

Programmation de pages WEB par l'exemple

Évolutions des modifications

Évolutions des modifications		
Création du document de base	Marc Emonet	09/11/2020
Reprise profonde du document (durant le confinement)	Marc Emonet	19/03/2021
Divers compléments	Marc Emonet	20/01/2022
Précisions sur tableaux arborescents	Marc Emonet	20/05/2023

TABLE DES MATIERES

1. LES BASES DE LA PROGRAMMATION SUR LE WEB	6
1.1. Architecture générale des processus	6
1.2. Les différents langages	7
1.3. Les protocoles	8
1.4. Éléments de base des pages Web	8
1.5. Développements informatiques	9
1.5.1. Les Systèmes de Gestion de Contenu	9
1.5.2. Les éditeurs de textes	9
1.5.3. Le développement local	10
1.5.4. Les plateformes de développement communautaire	10
1.5.5. Testeurs de codes	11
1.6. Origine des sources de ce présent recueil d'information	11
2. Le HTML	11
2.1. Le langage HTML c'est quoi ?	11
2.2. Les balises de bases en HTML	11
2.3. Principe général du HTML	12
2.4. Racine principale	13
2.5. Métadonnées du document	13
2.6. Sectionnement du contenu	15
2.7. La conception des textes	17
2.8. Sémantique du texte en ligne	20
2.9. Les images et les médias	25
2.10. Les contenus embarqués	26
2.11. Les scripts	29
2.12. Gestion de l'édition	29
2.13. Les cadres	29
2.14. Les tableaux	31
2.15. Les formulaires	36
2.16. Éléments interactifs	37
2.17. Web-Components	37
2.18. Autres HTML5	38
2.19. Quelques balises et attributs dépréciés à ne plus utiliser	38
2.20. Lectures complémentaires (et très utiles)	39
3. LE STYLE EN CASCADE AVEC CSS	39
3.1. Principe des CSS	39
3.1.1. Feuille de style externe	39
3.1.2. Feuille de style interne (inclus dans l'entête du fichier html)	40
3.1.3. Style CSS inclus dans le code html	40

3.1.4.	Style CSS appliqué sous condition	40
3.2.	Les commandes CSS	40
3.2.1.	Mise en forme d'un texte	40
3.2.2.	Gérer les marges en CSS	40
3.2.3.	Sélecteurs id et class	40
3.2.4.	Image de fond	41
3.2.5.	Effets rollover	41
3.2.6.	Aligner une image et du texte	41
3.2.7.	Puces en images	41
3.3.	Le CSS3	41
3.3.1.	Réaliser des coins arrondis	41
3.3.2.	Créer des ombres sur les boîtes et du texte	41
3.3.3.	Images multiples dans un même background	41
3.3.4.	Réaliser des fonds en dégradé sans image	42
3.3.5.	Déclarer une police de caractère non standard avec @font-face	42
3.3.6.	Animations et transitions d'images ou d'objets	42
3.3.7.	Les Media-Queries	42
3.3.8.	Boutons en CSS	42
3.3.9.	Zoom d'image	42
3.3.10.	Vignettes (Thumbnails)	42
3.3.11.	Menu déroulant	43
3.3.12.	Générer du contenu	43
3.3.13.	Formulaires	43
3.3.14.	Rasmol et objet	43
3.4.	DESCRIPTION DETAILLÉE DES COMMANDES CSS	43
3.4.1.	Police de caractère, tailles, structure	43
3.4.2.	Alignement	45
3.4.3.	Couleur et fond	45
3.4.4.	Dimension des boîtes	47
3.4.5.	Marges extérieures et intérieures (padding)	47
3.4.6.	Bordures	48
3.4.7.	Affichage et positionnement	49
3.4.8.	Valeurs utilisées	52
3.5.	Générateur de feuille de style (CSS)	53
4.	JAVASCRIPT	53
4.1.	La balise <script>	54
4.2.	L'attribut universel « id »	54
4.2.1.	L'élément « id »	54
4.2.2.	Le sélecteur « id »	54
4.3.	Document	55
4.4.	Element	55
4.5.	Affichage des données de sortie	56
4.6.	Les variables	56
4.7.	Blocs de code	57
4.8.	Liste des principaux mots-clés :	57
4.9.	Opérateurs arithmétiques	57
4.10.	Affectations	58
4.11.	Objets	59
4.12.	Evènements	59

4.13.	Les fonctions	59
4.14.	Les tableaux	60
4.15.	Les conditions	61
4.16.	Les boucles	61
4.17.	Résumé sur l'emploi des 3 langages	61
5.	JQUERY	62
5.1.	Syntaxe	62
5.2.	Sélecteurs jQuery	62
6.	XML	65
7.	JSON	65
8.	PHP & MySQL	66
8.1.	Instructions PHP	67
8.2.	Les variables	67
8.3.	Les opérations	67
8.4.	Les conditions	67
8.5.	Les boucles	68
8.6.	Les tableaux	68
8.7.	Les fonctions	69
8.8.	Inclusion d'un menu de page	69
8.9.	Manuel PHP	69
9.	Résumé des langages utilisés pour une page Web	69
9.1.	Récapitulatif de base	69
9.2.	Suivant les types de pages	70
9.3.	Les différentes méthodes utilisées :	70
10.	Autres langages de programmation	71
10.1.	Langage C	71
10.2.	Ruby	71
10.3.	Java	71
10.4.	Python	71
10.5.	Quelques autres langages	71
10.6.	Conversion entre langages	71
10.7.	Synthèse des principaux langages utilisés	71
10.7.1.	Évolution des langages dans le temps	71
10.7.2.	Paysage général de la situation	72
10.7.3.	Synthèse générale	72
10.7.4.	Comparaison	72
11.	EXEMPLES PRATIQUES	73
11.1.	Exemple d'utilisation de quelques balises	73

11.1.1.	Cellpadding	73
11.1.2.	Cellspacing	74
11.1.3.	Width	74
11.1.4.	Attributs pour les cellules (<td> et <th>)	75
11.1.5.	Align	75
11.1.6.	Border	76
11.2.	Autres exemples de programmation	76
11.2.1.	Exemple des commandes avec CSS	76
11.2.2.	Exemple max et min des caractères en CSS	77
11.2.3.	Intégration de photo avec CSS	77
11.2.4.	Intégration de photo avec zoom avec CSS	78
11.3.	Autres modules :	78
11.3.1.	Comment zoomer au survol	78
11.3.2.	Position d'une infobulle/Tooltip (en dessous)	79
11.3.3.	Incorporation d'une image « au survol »:	80
11.4.	Exemples de menus déroulants (sous l'onglet)	80
11.5.	Exemples sur les tableaux	83
11.5.1.	Intégration d'un tableau	83
11.5.2.	Commande pour la constitution d'un tableau :	83
11.5.3.	Les balises <th> (intitulés de colonne)	84
11.5.4.	Pour la gestion des blancs dans cellules	85
11.6.	Exemple comparatif de tableaux scrollables	85
11.7.	Exemple avec plusieurs colonnes collantes	88
11.8.	Exemples d'autres propriétés obtenues avec JavaScript	89
11.8.1.	Test bouton et Zindex de w3school	89
11.8.2.	Animation : direction et changement de couleur	89
11.8.3.	Passage alternatif d'un tableau à un autre	90
11.8.4.	Modifier un texte au clic	91
11.8.5.	Apparition/disparition du contenu d'une ligne dans un tableau	91
11.8.6.	Apparition/disparition d'une colonne dans un tableau	93
11.8.7.	Apparition/disparition de lignes et de colonnes avec JavaScript	94
11.8.8.	Création d'une arborescence (Tree view) de lignes	95
11.8.9.	Tableau avec arborescence de lignes et de colonnes	97
11.9.	Comparaison des tables sous jQuery	103
12.	SYNTHESE DES COMMANDES	104
12.1.	Résumé des commandes HTML	104
12.2.	Liste de commandes CSS	110
12.3.	Commandes par ordre alphabétique :	110
INDEX		111

1. LES BASES DE LA PROGRAMMATION SUR LE WEB

1.1. Architecture générale des processus

Les clients/utilisateurs se connectent par l'intermédiaire des navigateurs sur les réseaux pour accéder aux serveurs qui stockent des pages **Web**, des sites ou des applications... On dit que le Web est un protocole applicatif qui fonctionne sur un mode « client-serveur ».

Un navigateur (comme Google Chrome, Mozilla, Internet Explorer, Safari, Opéra...) :

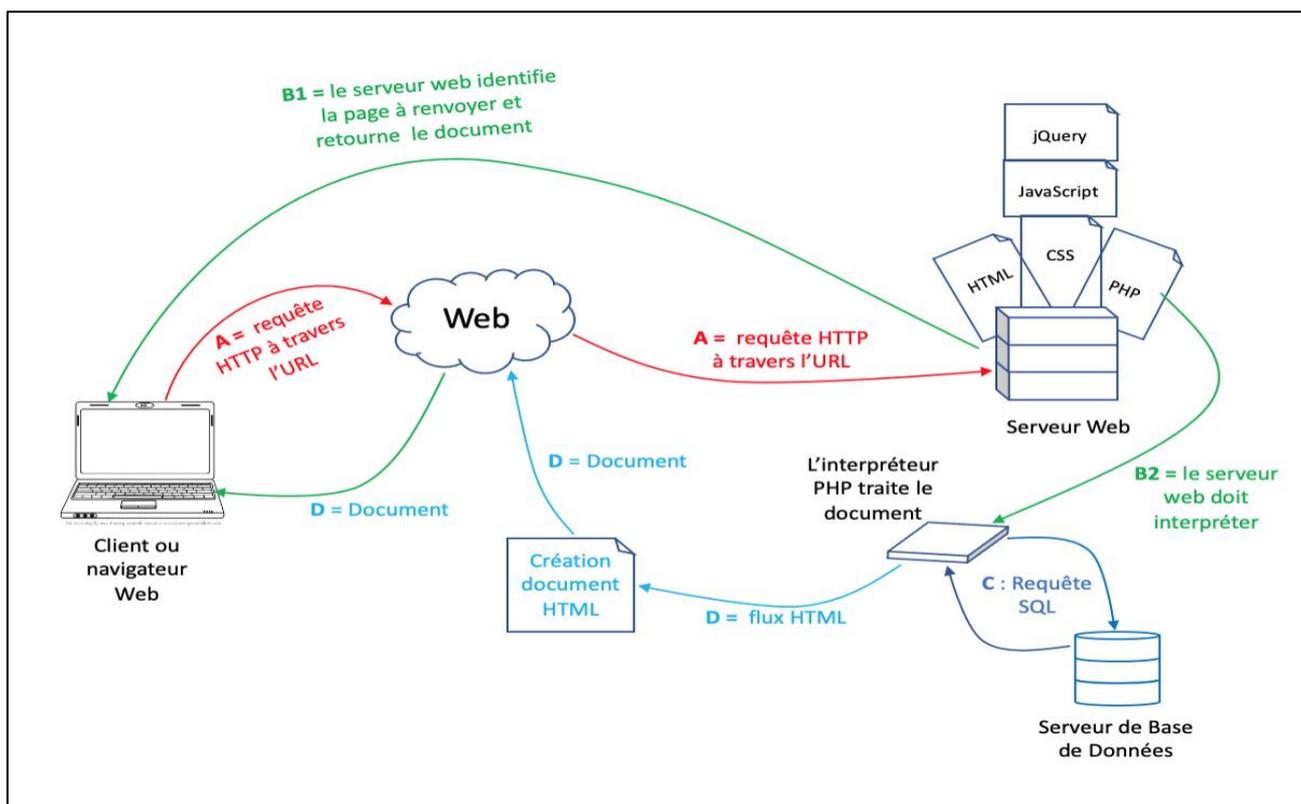
- utilise le protocole **HTTP** (Hyper Text Transfer Protocol) si l'**URL** (Uniform Resource Locator) commence par *http://* (ou **FTP** si c'est *ftp://*), puis il,
- consulte le **DNS** (Domain Name System) pour trouver l'adresse **IP** (de type :173.194.67.94) du site Web à partir de son nom de domaine connu (de type www.google.fr) et enfin,
- envoi à travers l'**URL** des requêtes HTTP (ou FTP) en langage compréhensible entre le client et le serveur **WWW** (World Wide Web).

La première requête vise à obtenir la page d'accueil ayant pour nom : *index.html*. Le serveur web va chercher sur son disque dur le fichier correspondant, puis il récupère son contenu et envoi ce flux de données au navigateur dont il connaît sa propre adresse IP.

A noter que si l'on connaît l'adresse IP du site web recherché, il est possible de la taper directement dans l'URL (<http://173.194.67.94>) pour accéder au site.

Le protocole HTTP permet d'envoyer ou de recevoir des informations entre les clients et les serveurs web ; les échanges se faisant sans aucun cryptage.

Le protocole **TCP/IP** (Transmission Control Protocol/Internet Protocol) définit la manière dont les données vont transiter sur le Web



Si l'adresse du site commence par HTTPS, alors les échanges se feront en langage sécurisé et crypté (le « S » désignant « Secure ») ; le protocole HTTPS est né de la combinaison du protocole HTTP et

d'un protocole de cryptage **SSL** (Secure Socket Layer) puis **TLS** (Transport Layer Security) pour la couche transport. Grâce à un échange de clés entre eux, le client et le serveur vont établir une connexion chiffrée dont le contenu ne pourra être lu que par eux.

Le bouton « verrou-vert » du navigateur permet de vérifier la validité du certificat SSL. (NB : l'outil GlobalSign à l'adresse : <https://globalsign.sslabs.com/analyze.html> permet de vérifier les protocoles de cryptage activés sur le serveur d'un site donné).

Le site Web est constitué de fichiers de codes (HTML, CSS, PHP ou Javascript) ou de ressources (images, musiques, vidéos, documents Word et Pdf).

Il existe deux types de site Web :

- Les **sites statiques** réalisent à partir des langages HTML et CSS correspondants à des « vitrines » que seul le propriétaire du site (le webmaster) peut modifier.
- Les **sites dynamiques** réalisent aussi à partir des langages HTML et CSS mais aussi en PHP, MySQL ou JavaScript qui permettent au visiteur de modifier les pages (sans intervention du webmaster).

1.2. Les différents langages

Chaque langage a ses spécificités propres :

- le **HTML** est un langage Hypertexte (permettant de créer des documents multimédias et de réaliser des liens entre eux) et à Balises (permettant de structurer et de formater des textes) ; il permet de créer la structure et le contenu des pages Web et permet de définir les différents éléments de la page tels que les titres, les paragraphes et les images ainsi que les formulaires,
- le **CSS** permet de spécifier les styles visuels et la mise en page générale des pages Web tels que les couleurs, les polices de caractère, les bordures,...
- le **JavaScript** est un langage interprété riche et parfaitement adapté à la programmation interactive des pages Web, il est utilisé pour créer des animations, des menus déroulants et des validateurs de formulaires.
- le **PHP** est un langage scripting de base de la programmation (coté serveur) qui est open source (est disponible et ouvert à tous) et qui est compilé. Il est utilisé pour créer des applications web dynamiques et interactifs de type : site de commerce électronique, blogs, forums...
- le **SQL** permet la gestion des Base de Données,
- le **XML** est le format universel d'échanges de données.

Les langages **compilés** s'exécutent rapidement ; en effet, le code source est compilé en binaire (grâce à un compilateur comme GCC ou JAVAC) et sera interprété par l'ordinateur mais sera incompatible entre les différentes machines et Systèmes d'Exploitation (SE) ; les principaux avec leurs spécificités sont :

- **C** (applications systèmes),
- **C++** (applications avec interfaces graphiques),
- **C#** (autres applications),
- **Java** (applications pour systèmes embarqués)...

Les langages **interprétés** sont plus lents car le code source reste tel quel mais ils sont « portables » sur tous type de machine et de SE, les principaux sont :

- **Perl** (administration système),
- **Python**,
- **Basic**,
- **JavaScript**,

- **Ruby...**

En résumé : en complément du langage de base HTML, on utilise le langage CSS pour la mise en page, le JavaScript pour les interactions et le PHP pour la communication avec les bases de données.

Remarque : une page Web peut s'ouvrir (être créée et modifiée) avec tout éditeur de texte (Notepad, Notepad++, Word...) mais on ne visualise alors que le code (source) de la page ; pour visualiser le « rendu », il faut utiliser un navigateur Web (Explorer, Chrome, Safari, Firefox, Opera...) qui interprète le code, met en forme la page et appelle les fichiers liés à celle-ci.

Le langage utilisé par le réseau mondial du Web est appelé « protocole ».

1.3. Les protocoles

Le principal est le **HTTP** mais il en existe bien d'autres comme :

- le **FTP** (File Transfert Protocol) pour le transfert des fichiers,
- l'**IRC** (Internet Relay Chat) pour le dialogue en direct,
- le **NNTP** (Network News Transfert Protocol) pour l'envoi et la lecture des News,
- le **POP** (Post Office Protocol) pour la récupération des courriels,
- le **SMTP** (Simple Mail Transfert Protocol) pour l'envoi de courriels...

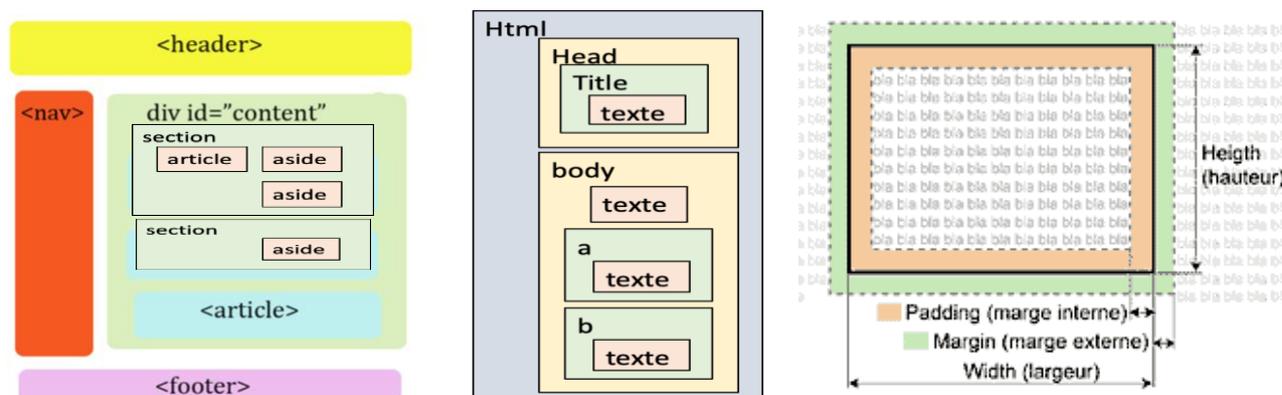
L'application **cliente** pour le protocole HTTP est un navigateur WEB (comme Chrome ou Firefox), pour le protocole FTP c'est le client FTP (comme Filezilla ou Cyberduk).

Mais plusieurs applicatifs ont accès à différents protocoles comme Outlook Express qui lit les courriels (POP) ou les news (NNTP) .

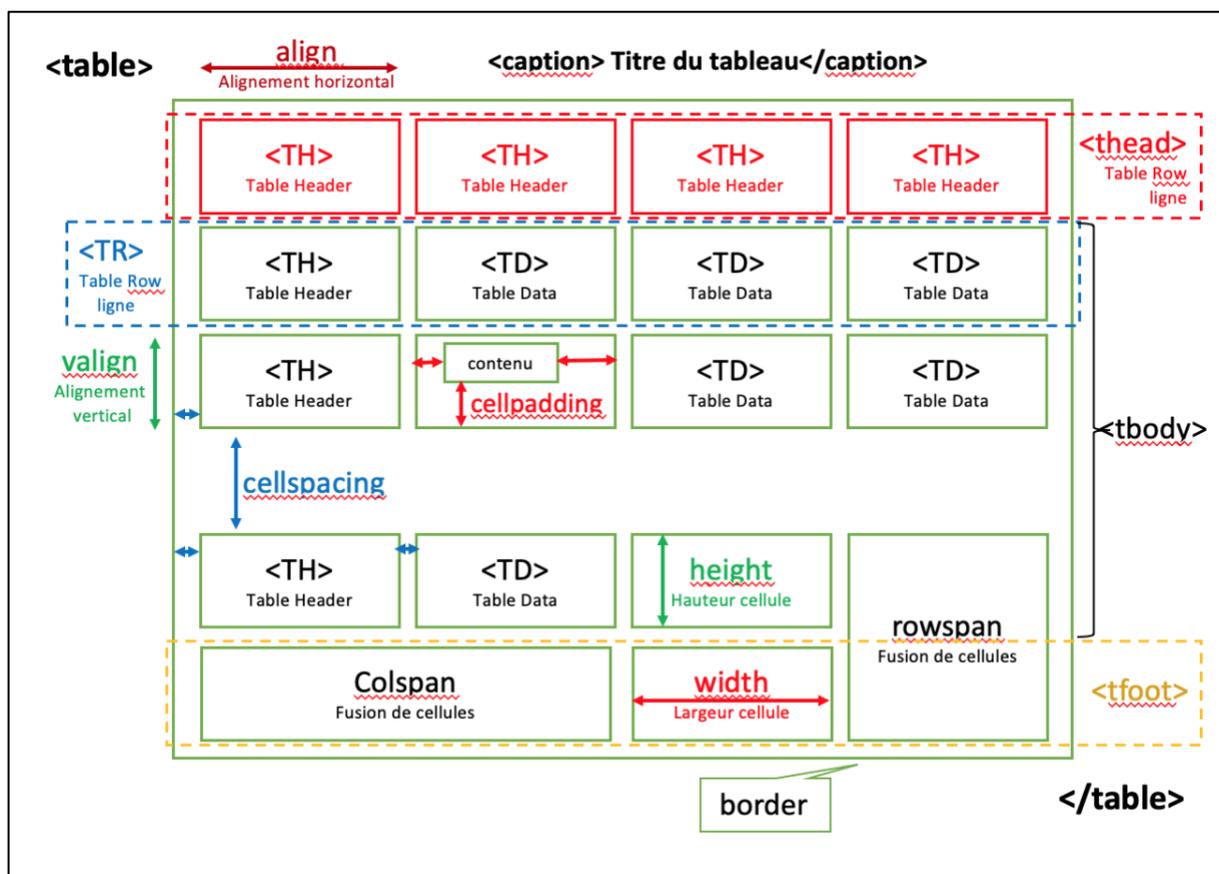
Si l'on souhaite accéder au serveur FTP à l'adresse <ftp.free.fr> il suffit de taper l'URL <ftp://ftp.free.fr>

1.4. Éléments de base des pages Web

Principales zones identifiées sur les pages écrans des sites :



Détails sur la conception des tableaux :



1.5. Développements informatiques

1.5.1. Les Systèmes de Gestion de Contenu

Le **Système de Gestion de Contenu** (SGC) ou Content Management System (CMS) est un logiciel qui permet aux utilisateurs de **créer, d’éditer et de publier** les contenus de sites Web (ou blog) comme par exemple des textes, des illustrations et des éléments multimédias... sans avoir à écrire de code.

Il en existe plus de 300, le plus répandu est **WordPress** (65%), ensuite il y a : Shopify, Wix, Squarespace, Joomla, Drupal, Blogger, Bitrix, Magento, OpenCart, Prestashop, Weebly, Typo3, Bigcommerce, **Adobe Dreamweaver**, Contao, Neos, Crafr, Grav, Kirby, Bolt...

Tous sont des « éditeurs de site Web » de type **WYSIWYG** (acronyme pour What You See Is What You Get).

1.5.2. Les éditeurs de textes

La création des pages Web se fait par un **éditeur de texte** (ou de code) comme : **Bloc Note**, **Note Pad++**, **TextEdit** (sur Mac), **Sublime Text**, **Visual Studio Code (VSCode)**, **Atom**, **IntelliJ IDEA**, **HelloWebFree** (qui est un éditeur de texte en ligne gratuit assez rudimentaire ne supportant pas toujours JavaScript) **ou PHP Storm** (ce dernier étant très complet mais payant).

Ces outils peuvent aller de l’édition de texte jusqu’à l’environnement de développement intégré (IDE).

Malheureusement les éditeurs de textes ne sont pas acceptés par tous les appareils et systèmes d'exploitation ! Ci-après le tableau de correspondance :

	Windows	Mac	Linux
SublimeText	X	X	X
NotePad ++	X	-	-
Brucket	X	X	X
jEdit	X	X	X
PSPad	X	-	-
ConTEXT	X	-	-
Smultron	-	X	-
TextWrangler	-	X	-

1.5.3. Le développement local

Pour créer un site Web, il peut être utile de le préparer sur un site fictif réalisé sur un serveur local de son ordinateur ; pour se faire, il existe plusieurs solutions de développement local de Web.

Le premier est constitué de 3 modules suivant les systèmes d'exploitation, à savoir :

- **WAMP** pour Windows (suivant la contraction de **Windows Apache, MySQL et PHP**),
- **MAMP** pour MAC et Windows et
- **LAMP** pour Linux.

Ce système utilise un serveur web **Apache**, une base de données **MySQL** et les langages de programmation : Python, PHP et Perl. Une fois MAMP installé sur son ordinateur, il est possible de créer les lignes de codes avec son éditeur de texte **Sublime Text** et de copier son projet sous forme projet .html dans le répertoire htdocs créé par MAMP afin de travailler en local avant de l'envoyer grâce à FileZilla.app sur l'hébergeur de son choix. MAMP permet d'associer les **gestionnaires de paquets** (qui automatisent les processus d'intégration des logiciels) tels que **Homebrew, Flatpak ou Ninite**.

Un autre outil de développement gratuit existe c'est **DevKinsta** qui permet aussi de concevoir, développer et déployer un site WordPress pour les machines MacOS ou Windows. <https://kinsta.com/fr/blog/outils-developpement-web/>

1.5.4. Les plateformes de développement communautaire

Les équipes de développements **DevOps** correspondent à un mouvement de mutualisation de développeurs d'applications qui suivent la méthodologie **Agile** (règles de planification et de développement par décomposition du travail en tâches et en étapes).

Le **Git** est un système de **contrôle de version** distribué (et non centralisé comme SVN ou VCS) Open Source (libre). Le code source d'un projet est géré dans Git.

Le contrôle de version permet aux équipes DevOps de modifier prudemment chaque évolution apportée du code source d'un logiciel en assurant son suivi. Le code d'un projet est organisé en arborescence de fichiers.

Le **GitHub** est un site de partage de code qui est géré avec le système de gestion de version Git ; il est utilisé par 82% des développeurs professionnels. Les autres outils de collaboration sont : Mou, Jira, Google Suite, Gitlab, Confluence, Trello, Équipes Microsoft, ...

Le **W3C** (World Wide Web Consortium) est une communauté internationale qui développe des standards pour améliorer le Web, ils mènent principalement leurs travaux dans le GitHub.

A noter qu'il existe aussi une plateforme dérivée du GitHub pour améliorer l'accessibilité des handicapés : le WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications)

Enfin de nombreuses autres plateformes existent proposant des missions à des candidats développeurs potentiels : LeHibou, WeLoveDevs, LesJeudis, Freelanceday, Codeur, Welcome to the Jungle, Freelance Republik... Avec éventuellement un portage salarial (comme Skalis) pour les rémunérer ; d'autres proposent des formations et enfin d'autres proposent des services de développement moyennant des financements.

1.5.5. Testeurs de codes

Différentes applications existent pour vérifier le résultat des codes HTML, CSS et JavaScript ; c'est notamment : <https://codepen.io/pen/> qui permet de voir directement le résultat des codes inscrits.

Une autre petite application très pratique « HelloWebFree » permet de voir directement l'action des codes HTML avant son transfert sur le Web.

1.6. Origine des sources de ce présent recueil d'information

Les principales origines des sources et des informations rassemblées dans ce document sont :

<http://fr.wikipedia.org>

<https://developer.mozilla.org/fr/docs/Web/HTML/Element>

<http://css.mammouthland.net/balises-html.php>

<http://stephane.chalamet.free.fr/html/tableaux.html>

<https://openclassrooms.com/fr/>

<https://wdi.centralesupelec.fr/appliouaibe/>

2. Le HTML

2.1. Le langage HTML c'est quoi ?

Le *HyperText Markup Langage*, est le langage de balisage conçu pour représenter les pages web. Il est parfois utilisé conjointement avec le langage de programmation : JavaScript et des feuilles de style en cascade (CSS). HTML est une des trois inventions à la base du *World Wide Web*, avec le *HyperText Transfer Protocol* (HTTP) et les adresses web (URL) créés en 1989.

2.2. Les balises de bases en HTML

La balise `<!Doctype...>` doit être déclarée au début des documents HTML. Elle permet de préciser la version du langage utilisé pour le codage, l'adresse de la définition de ce langage, la langue utilisée pour coder le document et le type de document HTML.

Même si le HTML 5 n'est pas implémenté par les navigateurs anciens, et même si on n'utilise pas les éléments HTML propres à HTML 5 on peut mettre ce doctype : `<!DOCTYPE html>`

Lorsque le document n'utilise pas que des feuilles de style, (la majorité des documents actuels), la balise Transitional doit être utilisée :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
```

Lorsque le document utilise uniquement des feuilles de style pour mettre en forme le document, c'est la balise Strict qui doit être utilisée (seuls les navigateurs les plus récents peuvent afficher ces pages sans problème).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/strict.dtd">
```

Enfin, lorsque le document utilise des cadres (cf. section cadres), c'est la balise Frameset qui doit être utilisée :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/frameset.dtd">
```

2.3.Principe général du HTML

En 1969 Goldfarb, Mosher et Lorie créent le premier métalangage : le SGML (Standard Generalized Markup Language).

En 1991 pour structurer les pages Web, Tim Berners-Lee crée la première version de HTML (Hyper Texte Markup Language) qui deviendra en 1994 HTML2 et en 1996 la version modifiée par le groupe W3C deviendra HTML3 offrant plusieurs possibilités dont les tableaux, applets, scripts...; Ensuite en 1998 la version HTML4 permettra l'utilisation de frame (découpage de page Web en plusieurs parties, tableaux complexes, formulaires...et d'exploiter les feuilles de styles CSS ; Enfin en 2000, HTML5 apportera des améliorations comme l'inclusion de vidéos, un meilleur agencement du contenu ...

En 1998 XML (eXtensible Markup Language) a été créé, puis en 2000 c'est XHTML (eXtensible Hyper Text Markup Language) qui est réécrit au format HTML à partir de XML.

Coder en HTML ou en XHTML revient donc au même, tant qu'on n'exploite pas de XML. Le principe général à respecter est que :

- Toutes les balises doivent être écrites en minuscules
- Soit, elles encadrent un texte, une image ou tout autre contenu et doivent être fermées (*exemple* : `<p> texte de paragraphe </p>`).
- Soit, elles permettent de définir des propriétés spéciales (ou attributs qui prennent certaines valeurs) (*exemple* : `<p align="center" > Texte du paragraphe centré </p>`)
- Les non-fermantes devront comporter un slash avant la fermeture de la balise (*exemple* : `
` pour un saut de ligne ou `<hr width="50%" size="5px" />` pour insertion d'une ligne horizontale dont la dimension représente 50% de la largeur de la fenêtre et une épaisseur de 5 pixels).
- Tous les attributs doivent être entre guillemets.
- Les balises de commentaires sont entre `<!--...-->`
- Les balises doivent être correctement imbriquées entre elles.

Par exemple : `<p></p> <p>Balises imbriquées</p>`

Autre exemple : `<p> ceci est un texte en gras et <i>en italique</i> </p>`

Attention avant de se lancer dans le HTML5 ! Quelques précautions sont à prendre, notamment vis-à-vis des versions d'Internet Explorer antérieures à 9 qui ne reconnaissent pas toujours le langage. A recommander la lecture de l'introduction au HTML5. (Soit : <https://www.cssdebutant.com>)

2.4. Racine principale

Élément	Description	Contient
<code><html></code> <code></html></code>	Cette balise représente la racine d'un document HTML (ou XHTML). Elle doit être placée en début de document, immédiatement après la balise DOCTYPE et se ferme à la fin du document.	<code><head></head></code> <code><body></body></code> <code><lang></lang></code>

2.5. Métadonnées du document

Les métadonnées contiennent des informations à propos de la page. On y retrouve les informations à propos des styles, des scripts et des données destinées à aider les moteurs de recherche et les navigateurs. Les métadonnées relatives à la mise en forme ou aux scripts peuvent être définies à même la page ou sous la forme d'un lien vers un autre fichier.

Élément	Description	Attribut
<code><base></code> <code></base></code>	<p>Cette balise permet de signaler au document que toutes les URL relatives du document ne partent pas du dossier en cours, mais, d'une position définie dans cette balise.</p> <p>Elle permet aussi de définir la zone d'affichage par défaut lorsque l'attribut target de la balise de lien n'est pas définie. Il ne peut y avoir qu'un seul élément <code><base></code> au sein d'un document.</p>	<p><i>href="URL"</i> : Définit la position relative par défaut d'un lien hypertexte.</p> <p><i>target="..."</i> : zone d'affichage par défaut lorsqu'un lien est utilisé, et que son attribut target n'est pas défini.</p> <ul style="list-style-type: none"> • <i>nom</i> d'un cadre : La page ciblée par le lien s'affichera dans le cadre dont vous aurez donné le nom (si toutefois celui-ci existe). • <i>_blank</i> : La page ciblée par le lien s'affichera dans une nouvelle fenêtre. • <i>_parent</i> : La page s'affiche en occupant la place qu'occupe le cadre qui définit le cadre ou se trouve de la page actuelle. • <i>_self</i> : La page s'affichera dans la fenêtre ou le cadre ou se situe le lien. • <i>_top</i> : La page s'affichera dans la fenêtre ouverte actuellement en occupant toute la fenêtre (ceci fera disparaître les éventuels cadres de la fenêtre).
<code><head></code> <code></head></code>	Cette balise entête du document est placée juste après la balise <code><html></code> fournit des informations générales (métadonnées) sur le document, incluant son titre et des liens ou des définitions vers des scripts et feuilles de style. Cette balise	<code><title></code> <code><meta></code> <code><link></code> <code><style></code> <code><script></code>

	<p>doit être fermée avant la balise d'ouverture du corps du document (balise <body>).</p>	
<p><body> </body></p>	<p>Cette balise placée juste après la fermeture de la balise <head> (</head>), elle représente le corps du document HTML, c'est à dire tout ce qui apparaîtra. Cette balise comporte certains attributs affectant l'ensemble du document.</p> <p>Il ne peut y avoir qu'un seul élément <body> par document.</p> <p>Attention, si on utilise une image d'arrière-plan, il faut penser aussi à définir une couleur de fond qui permet de lire le texte si l'image d'arrière-plan n'est pas lue par le butineur. Cela permet notamment à l'utilisateur de lire le contenu du document même si on utilise une couleur de police blanche, et une image de fond sombre, alors que le butineur de l'utilisateur ne lit pas l'image de fond.</p>	<p>onload="..." : permet de nommer un script définit qui s'exécute au chargement de la page.</p> <p>onunload="..." : permet de nommer un script définit qui s'exécute à la fermeture de la page.</p> <p>Certains attributs sont déclassés en XHTML en faveur des feuilles de style :</p> <p>bgcolor="couleur" : Permet de définir une couleur d'arrière-plan.</p> <p>background="couleur" : type de relation avec le document cité en ressource.</p> <p>text="couleur" : couleur du texte.</p> <p>link="couleur" : couleur des liens avant activation.</p> <p>vlink="couleur" : couleur des liens lorsqu'ils sont activés.</p> <p>alink="couleur" : couleur des liens déjà visités.</p>
<p><link> </link></p>	<p>Cette balise permet de définir les relations existantes entre ce document et d'autres documents.</p> <p>Exemple d'utilisation de la balise link pour expliquer la relation entre la page en cours, et la page située à l'adresse :</p> <pre>http://www.mon_site/monstyle : <link rel="stylesheet" type="text/css" href="http://www.mon_site.com/monstyle" title="monstyle"></pre> <p>Ici, la relation indique que la page http://www.mon_site.com/monstyle est une relation relative aux feuilles de style utilisées.</p>	<p>href="..." : URL de la ressource.</p> <p>rel="..." : type de relation avec le document cité en ressource.</p> <p>rev="..." : type de relation inverse avec le document cité en ressource.</p> <p>type="..." : définit le type de contenu Internet.</p> <p>media="..." : définit le support de destination</p> <ul style="list-style-type: none"> • <i>screen</i> : visuel (écran). • <i>print</i> : impression. • <i>projection</i> : projection. • <i>braille</i> : en braille. • <i>speech</i> : audio (parlé). • <i>all</i> : tous. <p>target="..." : zone d'affichage par défaut lorsque le lien est utilisé</p> <ul style="list-style-type: none"> • <i>nom d'un cadre</i> : La page ciblée par le lien s'affichera dans le cadre dont un nom aura été donné (si toutefois celui-ci existe). • <i>_blank</i> : La page ciblée par le lien s'affichera dans une nouvelle fenêtre. • <i>_parent</i> : La page s'affiche en occupant la place qu'occupe

		<p>le cadre qui définit le cadre ou se trouve de la page actuelle.</p> <ul style="list-style-type: none"> • <i>_self</i> : La page s'affichera dans la fenêtre ou le cadre ou se situe le lien. • <i>_top</i> : La page s'affichera dans la fenêtre ouverte actuellement en occupant toute la fenêtre (ceci fera disparaître les éventuels cadres de la fenêtre).
<p><code><meta></code> <code></meta></code></p>	<p>Cette balise représente toute information de métadonnées qui ne peut pas être représentée par un des éléments (<code><base></code>, <code><link></code>, <code><script></code>, <code><style></code> ou <code><title></code>), elle peut être utilisée plusieurs fois.</p> <p>Exemple de balises meta:</p> <ul style="list-style-type: none"> • <code><meta name="author" content="stephane chalamet" /></code> : Nom des auteurs d'un document. Ici, l'auteur est stephane chalamet. • <code><meta name="generator" content="gnu-emacs" /></code> : Nom de l'outil ayant permis de créer la page web. Ici GNU Emacs. • <code><meta name="keywords" content="HTML, XHTML, code" /></code> : Mots clés du document (utilisé par certains moteur de recherche pour référencer leur documents). Ici, HTML, XHTML et code. • <code><meta http-equiv="Content-Script-Type" content="text/javascript" /></code> : Langage utilisé ici, Javascript. 	<p>http-equiv="..." : nom de l'entête réponse HTTP.</p> <p>name="..." : nom de la méta-donnée définie dans l'attribut content.</p> <p>content="..." : valeur de la méta-donnée, nommée dans l'attribut name.</p> <p>scheme="..." : donne la clé d'interprétation de la méta-donnée.</p>
<p><code><style></code> <code></style></code></p>	<p>Cette balise sert à introduire une description de styles internes. Ces styles seront utilisés dans la page.</p> <p>Elle permet aussi de déclarer les codes CSS.</p> <p>Plusieurs styles peuvent être définis et ne seront utilisés que lorsqu'ils seront appelés avec l'attribut style d'une balise supportant cet attribut. Les balises supportant l'attribut style sont listées dans l'annexe sur les attributs communs</p>	<p>type="..." : définit le type de contenu Internet.</p> <p>media="..." : définit le support de destination</p> <ul style="list-style-type: none"> • <i>screen</i> : visuel (écran). • <i>print</i> : impression. • <i>projection</i> : projection. • <i>braille</i> : en braille. • <i>speech</i> : audio (parlé). • <i>all</i> : tous. <p>title=".." : donne un titre au style.</p>
<p><code><title></code> <code></title></code></p>	<p>Cette balise définit le titre du document (qui est affiché dans la barre de titre du navigateur ou dans l'onglet de la page). Cet élément ne peut contenir que du texte, les balises qu'il contiendrait seraient ignorées.</p>	

2.6. Sectionnement du contenu

Organiser le contenu d'une page en différentes sections permet d'avoir une structure logique au sein d'un document. Grâce à ces éléments, on peut créer un plan pour la page, ajouter des titres pour identifier les sections et également gérer un en-tête et un bas de page.

Élément	Description	Attribut
<code><address></code> <code></address></code>	Cet élément en général situé en bas d'une page web, indique le nom et les coordonnées des auteurs .	
<code><article></code> <code></article></code>	Cet élément représente du contenu autonome dans un document , une page, une application, ou un site. Ceci peut être un message sur un forum, un article de journal ou de magazine, une parution sur un blog, un commentaire d'utilisateur, un widget ou gadget interactif, ou tout autre élément de contenu indépendant. Ce contenu est prévu pour être distribué ou réutilisé indépendamment (par exemple dans un flux de syndication).	
<code><aside></code> <code></aside></code>	Cet élément (en français, " aparté ") représente une partie d'un document dont le contenu n'a qu'un rapport indirect avec le contenu principal du document.	
<code><header></code> <code></header></code>	Cet élément représente un groupe de contenu introductif ou de contenu aidant à la navigation. Il peut contenir des éléments de titre, mais aussi d'autres éléments tels qu'un logo, un formulaire de recherche, etc.	
<code><footer></code> <code></footer></code>	Cet élément représente le pied de page de la section ou de la racine de sectionnement la plus proche. Un pied de page ou de section contient habituellement des informations sur l'auteur de la section, les données relatives au droit d'auteur (<i>copyright</i>) ou les liens vers d'autres documents en relation.	
<code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , <code><h6></code>	Les éléments <code><h1></code> à <code><h6></code> représentent six niveaux de titres dans un document, <code><h1></code> est le plus important et <code><h6></code> est le moins important. Un élément de titre décrit brièvement le sujet de la section qu'il introduit.	<i>align="..."</i> : permet de choisir l'alignement du titre. <ul style="list-style-type: none"> ○ <i>left</i> : le titre est aligné à gauche. ○ <i>center</i> : le titre est centré. ○ <i>right</i> : le titre est aligné à droite. ○ <i>justify</i> : le titre est justifié.
<code><hgroup></code> <code></hgroup></code>	Cet élément représente un titre de plusieurs niveaux pour une section d'un document. Il regroupe un ensemble d'éléments <code><h1></code> - <code><h6></code> .	

Élément	Description	Attribut
<code><main></code> <code></main></code>	Cet élément représente le contenu majoritaire du <code><body></code> du document. Le contenu principal de la zone est constitué de contenu directement en relation, ou qui étend le sujet principal du document ou de la fonctionnalité principale d'une application.	
<code><nav></code> <code></nav></code>	Cet élément représente une section d'une page ayant des liens vers d'autres pages ou des fragments de cette page. Autrement dit, c'est une section destinée à la navigation dans un document (avec des menus, des tables des matières, des index, etc.).	
<code><section></code> <code></section></code>	Cet élément représente une section générique d'un document, par exemple un groupe de contenu thématique. Une section commence généralement avec un titre.	

2.7. La conception des textes

Le contenu HTML textuel permet d'organiser des blocs ou des sections de contenu entre la balise ouvrante `<body>` et la balise fermante `</body>`. Ces éléments sont cruciaux pour l'accessibilité et le référencement car ils permettent d'identifier le sens du contenu.

Élément	Description	Attribut
<code><blockquote></code> <code></blockquote></code>	Cette balise (qui signifie <i>bloc de citation</i>) sert à signaler une citation (longue) d'un ouvrage. Le texte est généralement affiché avec une indentation (voir les notes ci-après). Une URL indiquant la source de la citation peut être donnée grâce à l'attribut <code>cite</code> tandis qu'un texte représentant la source peut être donné via l'élément <code><cite></code> .	<code>cite="URL"</code> : URL d'où provient la citation.
<code><dd></code> <code></dd></code>	Cette balise (pour <i>définition description</i>) indique la définition d'un terme au sein d'une liste de définitions (élément <code><dl></code>). Cet élément ne peut apparaître qu'en tant qu'élément appartenant à une liste de définition et doit être précédé d'un élément <code><dt></code> .	
<code><dir></code> <code></dir></code>	Cette balise (pour <i>directory</i>) est utilisée comme un conteneur pour un répertoire (c'est-à-dire un ensemble de fichiers). Des styles et icônes peuvent être appliqués par l'agent utilisateur. Cet élément obsolète ne doit pas être utilisé, il peut être remplacé par	

Élément	Description	Attribut
	l'élément <code></code> qui permet de représenter des listes et, entre autres, des listes de fichiers.	
<code><div> </div></code>	<p>Cette balise (qui signifie <i>division du document</i>) est un conteneur générique qui permet d'organiser le contenu sans représenter rien de particulier.</p> <p>Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style (en utilisant les attributs <code>class</code> ou <code>id</code>) ou parce qu'ils partagent des attributs aux valeurs communes, tel que <code>lang</code>. Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique (par exemple <code><article></code> ou <code><nav></code>) n'est approprié.</p> <ul style="list-style-type: none"> ○ 	<p>L'unique attribut de cette balise est déclassé en HTML en faveur des feuilles de style: <code>align="..."</code> : permet de choisir l'alignement du texte.</p> <ul style="list-style-type: none"> ○ <code>left</code> : le texte est aligné à gauche. ○ <code>center</code> : le texte est centré. ○ <code>right</code> : le texte est aligné à droite. <code>justify</code> : le texte est justifié.
<code><center> </center></code>	Équivalent à la balise <code>div</code> , mais le texte est forcément centré .	
<code><dl></code>	Cette balise représente une liste de descriptions sous la forme d'une liste de paires associant des termes (fournis par des éléments <code><dt></code>) et leurs descriptions ou définitions (fournies par des éléments <code><dd></code>). On utilisera par exemple cet élément pour implémenter un glossaire.	
<code><dt></code>	Cette balise identifie un terme dans une liste de définitions ou de descriptions. Cet élément n'apparaît qu'en tant qu'élément enfant d'un élément <code><dl></code> et est généralement suivi d'un élément <code><dd></code> .	
<code><figcaption></code>	Cette balise (pour <i>figure caption</i> en anglais) permet d'indiquer un sous-titre , une légende, associé à une figure ou à une illustration (cette dernière étant représentée par l'élément <code><figure></code> qui est le parent direct de la légende). L'élément <code><figcaption></code> est optionnel ; s'il n'est pas présent, la figure n'aura pas de légende.	
<code><figure></code>	Cette balise représente une figure (un schéma), qui peut être accompagné d'une légende grâce à l'élément <code><figcaption></code> . Il est normalement référencé de manière unique. C'est souvent une image, une illustration, un diagramme, un fragment de code ou un schéma auquel il est fait référence dans le texte principal mais qui peut être utilisé sur une autre page ou dans une annexe sans que cela affecte le contenu principal.	

Élément	Description	Attribut
<hr>	Cette balise représente un changement thématique entre des éléments de paragraphe (par exemple, un changement de décor dans un récit, un changement de sujet au sein d'une section).	align ="..." : permet de choisir l'alignement de la barre. <ul style="list-style-type: none"> ○ <i>left</i> : la barre est aligné à gauche. ○ <i>center</i> : la barre est centré. ○ <i>right</i> : la barre est aligné à droite. ○ <i>justify</i> : la barre est justifié. width ="..." : largeur en pixels, ou pourcentage de la fenêtre. color ="..." : couleur size ="..." : taille en hauteur en pixels. noshade ="noshade" : Cet attribut n'admet que la valeur noshade qui signifie que la barre est pleine (grisée). Si l'attribut n'est pas utilisé la barre sera creuse, et l'intérieur prendra la couleur de fond de la page.
<hr/>	Barre de séparation horizontale.	
	Cette balise représente une liste à puce sans ordre particulier.	
	Cette balise représente une liste numérotée . Les éléments d'une telle liste sont généralement affichés avec un indicateur ordinal pouvant prendre la forme de nombres, de lettres, de chiffres romains ou de points. La mise en forme de la numérotation n'est pas utilisée dans la description HTML mais dans la feuille de style CSS associée grâce à la propriété list-style-type .	
	Cette balise est utilisée pour représenter un élément dans une liste . Il doit être contenu dans un élément parent : une liste ordonnée (), une liste non ordonnée () ou un menu (<menu>). Dans les menus et les listes non ordonnées, les éléments de liste sont habituellement affichés en utilisant des puces. Dans les listes ordonnées, ils sont habituellement affichés avec compteur croissant à gauche, tel qu'un nombre ou une lettre.	
<p>	Cette balise représente un paragraphe de texte . Les paragraphes sont généralement représentés comme des blocs et séparés par un espace vertical, leur première ligne est également	align ="..." : permet de choisir l'alignement du titre. <ul style="list-style-type: none"> ○ <i>left</i> : le titre est aligné à gauche. ○ <i>center</i> : le titre est centré.

Élément	Description	Attribut
	parfois indentée. Les paragraphes sont des éléments blocs .	<ul style="list-style-type: none"> o <i>right</i> : le titre est aligné à droite. o <i>justify</i> : le titre est justifié.
<code><pre></code>	Cette balise représente du texte préformaté , généralement écrit avec une police à chasse fixe. Le texte est affiché tel quel, les espaces utilisés dans le document HTML seront retranscrits.	<i>width</i> ="..." : indique la largeur du texte préformaté.
<code>< !... ></code> Ou <code>/*...*/</code>	Ces balises permettent de faire des commentaires sans qu'ils soient pris en compte dans le document Web.	

2.8. Sémantique du texte en ligne

Les éléments pour le texte en ligne peuvent être utilisés pour définir la signification, la structure ou la mise en forme d'un terme, d'une ligne ou d'un fragment de texte.

Élément	Description	Attribut
<code><a></code>	Cette balise (pour ancre ou <i>anchor</i> en anglais) définit un hyperlien vers un autre endroit de la même page ou vers une autre page sur le Web.	<code>Website</code>
<code><abbr></abbr></code>	Cette balise représente une abréviation ou un acronyme et permet, de façon optionnelle, d'en fournir une description complète. S'il est présent, l'attribut title doit contenir cette même description complète et rien d'autre.	
<code><bdi></bdi></code>	Cette balise (ou élément d'isolation de texte bidirectionnel) isole une portée (<i>span</i>) de texte pouvant être formatée dans une direction différente de celle du texte qui l'entoure. Cela permet, par exemple, de présenter correctement une citation en arabe (écrit de droite à gauche) au sein d'un texte écrit en français (écrit de gauche à droite).	
<code><bdo></bdo></code>	Cette balise change la direction d'affichage et de lecture du texte situé entre les balises d'ouverture et de fermeture.	<i>lang</i> ="..." : langue utilisée par le texte (en-US: anglais US, fr: français,...). D'autres valeurs sont référencées dans l'annexe sur les attributs communs %i18n <i>dir</i> ="..." : direction d'écriture / lecture du texte. <ul style="list-style-type: none"> o <i>ltr</i> : left to right : écriture / lecture de gauche à droite.

Élément	Description	Attribut
		<ul style="list-style-type: none"> o <i>rtl</i> : right to left : écriture / lecture de droite à gauche.
<code>
</code>	<p>Cette balise crée un saut de ligne (un retour chariot) forcé dans le texte sans pour autant mettre un espace entre les 2 paragraphes, comme il y en aurait en fermant le paragraphe et en ouvrant un nouveau.</p> <p>Il s'avère utile lorsque les sauts de ligne ont une importance (par exemple lorsqu'on écrit une adresse ou un poème).</p>	<p>clear="..." : détermine où se situe le début de la prochaine ligne lorsque le texte est situé à côté d'un objet tel qu'une image.</p> <ul style="list-style-type: none"> o <i>none</i> : le texte continue à côté de l'objet. o <i>left</i> : le texte continue en dessous de l'objet situé à gauche du texte. o <i>right</i> : le texte continue en dessous de l'objet situé à droite du texte.
<code><cite></cite></code>	<p>Cette balise contient le titre d'une œuvre telle que le nom d'un ouvrage, d'une chanson, d'un film, d'une sculpture... Cet élément doit inclure le titre de l'œuvre. Cette référence peut être abrégée en accord avec les conventions d'usages pour l'ajout des métadonnées de citations.</p>	
<code><code></code></code>	<p>Cette balise représente un fragment de code machine qui doit être situé de part et d'autre du listing du code d'un programme. Par défaut, l'agent utilisateur utilise une police à chasse fixe (<i>monospace</i>) afin d'afficher le texte contenu dans cet élément.</p>	
<code><data></data></code>	<p>Cette balise relie un contenu à une version de ce contenu interprétable par un ordinateur. Si le contenu possède une composante temporelle, l'élément <code>{{HTML_Element("time")}}</code> doit être utilisé.</p>	
<code><dfn></dfn></code>	<p>Cette balise est utilisée afin d'indiquer le terme qui est en train d'être défini dans une phrase ou un paragraphe (sa définition est écrite dans son parent <code>{{HTML_Element("p")}}</code> ou <code>{{HTML_Element("dt")}}</code>/<code>{{HTML_Element("dd")}}</code> ou <code>{{HTML_Element("section")}}</code> le plus proche).</p>	
<code><i></i></code>	<p>Cette balise représente un morceau de texte qui se différencie du texte principal. Cela peut par exemple être le cas pour des termes techniques, des phrases dans une langue étrangère ou encore l'expression des pensées d'un personnage. Le</p>	

Élément	Description	Attribut
	contenu de cet élément est généralement affiché en italique .	
<code><kbd></kbd></code>	Cette balise représente une plage de texte en ligne indiquant la saisie de texte par l'utilisateur à partir d'un clavier, d'une saisie vocale ou de tout autre dispositif de saisie de texte. Par convention, le rend par défaut le contenu d'un élément <code><kbd></code> en utilisant sa police monospace, bien que cela ne soit pas requis par le standard HTML.	
<code><mark></mark></code>	Cette balise représente un texte marqué ou surligné à cause de sa pertinence dans le contexte. Il peut par exemple être utilisé afin d'indiquer les correspondances d'un mot-clé recherché au sein d'un document.	
<code><q></q></code>	Cette balise indique que le texte qu'il contient est une citation en incise. La plupart des navigateurs modernes entoure le texte de cet élément avec des marques de citation. Cet élément est destiné aux citations courtes qui ne nécessitent pas de sauts de paragraphe. Pour les plus grandes citations, on utilisera l'élément <code><blockquote></code> .	
<code><ruby></ruby></code>	Cette balise représente une annotation Ruby. Les annotations Ruby servent à afficher la prononciation des caractères d'Asie Orientale .	
<code><rb></rb></code>	L'élément de base ruby est utilisé afin de délimiter le composant texte de base d'une annotation <code><ruby></code> . Autrement dit, le texte qui est annoté.	
<code><rt></rt></code>	Cette balise indique la composante texte d'une annotation Ruby, il est notamment utilisé pour la prononciation, la traduction ou la translittération des caractères d'Asie orientale. Cet élément est toujours contenu dans un élément <code><ruby></code> .	
<code><rp></rp></code>	Cette balise est utilisée pour fournir ce qui fera office de parenthèse aux navigateurs qui ne prennent pas en charge les annotations Ruby.	
<code><rtc></rtc></code>	Le code source de cet exemple interactif est disponible dans un dépôt GitHub. Ne pas hésitez pas à copier https://github.com/mdn/interactive-examples et à envoyer une <i>pull-request</i> !	

Élément	Description	Attribut
<code><s></s></code> <code><strike></code> <code></strike></code>	Cette balise permet d'afficher du texte qui est barré (soit il n'est plus pertinent ou il est obsolète). <code><s></code> ne doit pas être employé pour indiquer des éditions dans un document (on utilisera alors <code></code> et <code><ins></code>).	
<code><samp></code> <code></samp></code>	Cette balise est un élément qui permet de représenter (en relief) un résultat produit par un programme informatique. Il est généralement affiché avec la police à chasse fixe du navigateur (par exemple en Courier ou en Lucida Console).	
<code><small></code> <code></small></code>	Cette balise permet de représenter des commentaires ou des textes à écrire en plus petits caractères (des termes d'un contrat, des mentions relatives au droit d'auteur, etc.) quelle que soit la présentation.	
<code><big> </big></code>	Cette balise permet de représenter des commentaires ou des textes à écrire en plus gros caractères	
<code><tt> </tt></code>	Texte à largeur ou à chasse fixe.	
<code></code>	Cette balise est un conteneur générique en ligne (inline) pour les contenus phrasés. Il ne représente rien de particulier. Il peut être utilisé pour grouper des éléments afin de les mettre en forme (grâce aux attributs <code>class</code> ou <code>id</code> et aux règles <code>CSS</code>) ou parce qu'ils partagent certaines valeurs d'attribut comme <code>lang</code> . Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique n'est approprié. <code></code> est très proche de l'élément <code><div></code> , mais l'élément <code><div></code> est un élément de bloc , alors que <code></code> est un élément en ligne .	<code>#9660;</code>
<code></code>	Cette balise (pour emphase) est utilisée afin de marquer un texte sur lequel on veut insister en mettant le texte en exergue . Les éléments <code></code> peuvent être imbriqués, chaque degré d'imbrication indiquant un degré d'insistance plus élevé.	
<code></code>	Cette balise permet d'attirer l'attention du lecteur sur le contenu d'un élément sans que ce contenu revêt une importance particulière. Anciennement utilisé pour mettre le texte en gras . Cet élément ne doit pas être utilisé pour mettre en forme des éléments, c'est la propriété CSS <code>font-weight</code> qu'il	

Élément	Description	Attribut
	faut utiliser. Si l'élément est d'une importance particulière, on utilisera l'élément HTML <code></code> .	
<code></code> <code></code>	Cette balise indique que le texte a une importance particulière ou un certain sérieux voire un caractère urgent (plus important que <code></code>). Cela se traduit généralement par un affichage en gras .	
<code><sub></sub></code>	Cette balise est utilisée, pour des raisons typographiques, afin d'afficher du texte souscrit (ou en indice) (plus bas et généralement plus petit) par rapport au bloc de texte environnant.	
<code><sup></sup></code>	Cette balise est utilisée, pour des raisons typographiques, afin d'afficher du texte en exposant (plus haut et généralement plus petit) par rapport au bloc de texte environnant.	
<code><time></time></code>	Cette balise permet de représenter une période donnée. Cet élément permet d'utiliser l'attribut <code>datetime</code> afin de traduire la date ou l'instant dans un format informatique (permettant aux moteurs de recherche d'exploiter ces données ou de créer des rappels).	
<code><tt></tt></code>	Cette balise (pour <i>Teletype Text</i>) crée un élément en ligne, écrit dans la police à chasse fixe par défaut du navigateur. Cet élément a été conçu pour mettre en forme du texte comme s'il apparaissait sur un affichage à largeur fixe tel qu'un télécopieur.	
<code><u></u></code>	Cette balise permet d'afficher un fragment de texte qui est annoté avec des éléments non textuels. Par défaut, le contenu de l'élément est souligné . Cela pourra par exemple être le cas pour marquer un texte comme étant un nom propre chinois, ou pour marquer un texte qui a été mal orthographié.	
<code><var></var></code>	Cette balise représente une variable dans une expression mathématique ou un texte lié à la programmation. Son contenu est généralement représenté avec une version italique de la police environnante utilisée, toutefois, ce comportement peut dépendre du navigateur utilisé.	

Élément	Description	Attribut
<code><wbr></wbr></code>	Cette balise permet de représenter un emplacement où casser la ligne si nécessaire. Le navigateur pourra alors utiliser cet emplacement pour effectuer un saut de ligne si le texte est trop long et qu'en temps normal, une règle empêche le saut de ligne.	
<code><acronym></acronym></code>	Se place de part et d'autre d'un acronyme (abréviation dont les initiales sont prononcées comme un mot, comme SIDA, OTAn,...).	
<code><basefont /></code>	Tout texte situé après cette balise prendra pour valeur de base les valeurs des attributs de cette balise.	<i>size="..."</i> : taille de base des polices (entre 1 et 7). <i>color="..."</i> : couleur de base de la police d'écriture. <i>face="..."</i> : polices à utiliser de préférence. elles sont inscrites dans l'ordre de préférence, et séparées par des virgules. Le butineur affichera le texte avec la première police d'écriture disponible sur l'ordinateur de l'utilisateur.
<code> </code>	Tout texte situé entre ces balises prendra pour aspect, les valeurs des attributs de cette balise.	
<code><blink></code>	Caractère ancien inusité	

2.9. Les images et les médias

HTML prend en charge différents fichiers multimédias pour les images, les fichiers audios et vidéos.

Élément	Description	Attribut
<code><area></code>	Cet élément définit une zone particulière d'une image et peut lui associer un lien hypertexte . Il n'est utilisé qu'au sein d'un élément <code><map></code> .	
<code></code>	Cet élément permet de représenter une image dans un document. Le type d'image est normalement pour une photo en .jpg ou .jpeg (dégradant la photo mais étant moins lourd). En .png pour tout ce qui n'est pas une photo (graphique par exemple) ou en .gif avec une image pouvant être animée. Éviter .bmp qui est beaucoup trop lourd.	Se combine avec l'attribut : <ul style="list-style-type: none"> • src qui va chercher l'image où elle se trouve soit chemin absolu : <code>http://www.site.com/fliut.jpg</code> soit chemin relatif (dans dossier « image ») <code>src="image/fleur.jpg"</code>.

Élément	Description	Attribut
		<ul style="list-style-type: none"> • alt qui est un texte alternatif qui n'apparaît que s'il n'y a pas d'image. L'élément <code></code> est un élément remplacé
<code><map></code>	Cet élément est utilisé avec des éléments <code><area></code> afin de définir une image cliquable divisée en régions.	
<code><track></code>	Cet élément est utilisé comme élément fils d'un élément <code><audio></code> ou <code><video></code> et permet de fournir une piste texte pour le média (par exemple afin de gérer automatiquement les sous-titres). Les pistes texte utilisées avec cet élément sont formatées selon le format WebVTT (ce sont des fichiers .vtt) (WebVTT pour <i>Web Video Text Tracks</i>) ou selon le format Timed Text Markup Language (TTML) .	
<code><audio></code>	Cet élément est utilisé afin d'intégrer un contenu sonore dans un document. Il peut contenir une ou plusieurs sources audio représentées avec l'attribut <code>src</code> ou l'élément <code><source></code> . S'il y a plusieurs sources, l'agent utilisateur choisira celle qui convient le mieux.	
<code><video></code>	Cet élément intègre un contenu vidéo dans un document.	

2.10. Les contenus embarqués

En plus du contenu multimédia, un document HTML peut embarquer d'autres contenus (bien que les interactions soient plutôt limitées).

Élément	Description	Attribut
<code><applet></code>	Cet élément pour les applets, définit l'intégration d'une applet Java ; mais il est désormais déprécié en faveur de <code><object></code> .	
<code><embed></code>	Cet élément représente un point d'intégration pour une application externe ou pour du contenu interactif (autrement dit, pour un <i>plug-in</i>).	
<code><frame></code> <code></frame></code>	Cette balise permet de définir d'une part ce que contiendra un cadre , et d'autre part, une partie de l'apparence du cadre.	name="..." : Permet de donner un nom au cadre flottant. Cela permet d'afficher une page dans ce cadre

Élément	Description	Attribut
	<p>Exemple d'utilisation d'un jeu de cadre simple: Code: <pre><frameset cols="20%,*" > <frame src="menu.html" name="gauche" /> <frame src="site.html" name="principal" /> </frameset></pre> </p> <p>Au démarrage de cette page, la page menu.html est affiché dans le cadre de gauche nommé <i>menu</i>. Ce cadre a une largeur de 20% de la taille de la fenêtre. La page site.html est, elle, affiché dans le cadre de droite nommé <i>principal</i> occupant le reste de la fenêtre.</p> <p>Si la page menu.html contient un lien <code>accueil du site </code> .</p> <p>Lorsque ce lien sera activé, la page menu.html restera en place, et c'est la page du cadre nommé principal (celui de droite) qui changera pour afficher la page accueil.html</p>	<p>lorsque l'attribut target d'une balise <code></code> prend pour valeur le nom du cadre.</p> <p>src="URL" : Affiche la page html désignée par la valeur de cet attribut dans le cadre lorsqu'il est chargé pour la première fois.</p> <p>frameborder="..." : Affiche une bordure au cadre si l'attribut prend la valeur 1, ou aucune bordure s'il prend la valeur 0.</p> <p>marginwidth="..." : définit la distance en pixels entre les bordures gauche et droite du cadre, et le contenu de la page html qui est chargé dans celui-ci.</p> <p>marginheight="..." : définit la distance en pixels entre les bordures supérieur et inférieur du cadre, et le contenu de la page html qui est chargé dans celui-ci.</p> <p>scrolling:"..." : permet de définir si des ascenseurs sont disponible dans le cadre.</p> <ul style="list-style-type: none"> • <i>auto</i> : ascenseurs disponible lorsqu'ils sont nécessaires (lorsque le contenu d'un cadre est plus large, ou haut que le cadre lui-même. • <i>yes</i> : des ascenseurs verticaux, et horizontaux sont toujours visibles, même s'ils ne sont pas nécessaires • <i>no</i> : aucun ascenseur n'est disponible même s'ils sont nécessaires
<p><iframe></p>	<p>Cette balise permet d'inclure un cadre à l'intérieur d'un document html.</p> <p>Le contenu de ce cadre pourra être changé de la même manière qu'un cadre normal, mais disparaîtra lorsque la page html qui le contient changera.</p> <p>Notion de cadre flottant.</p> <p>Si votre butineur interprète les cadres flottants, vous devriez en avoir un exemple ci-dessous</p> <p>Le code qui permet d'intégrer ce cadre flotant est:</p>	<p>Même attribut que pour frame avec en plus</p> <p>height="..." : définit la hauteur du cadre en pixels.</p> <p>width="..." : définit la largeur du cadre en pixels, ou en pourcentage de la largeur de la fenêtre.</p> <p>Certains attributs sont déclassée en XHTML en faveur des feuilles de style:</p> <p>align="..." : permet d'aligner horizontalement le cadre flottant.</p> <ul style="list-style-type: none"> • <i>left</i> : alignement du cadre à gauche • <i>center</i> : cadre centré

Élément	Description	Attribut
	<code><iframe src="iframe.html" align="center" width="80%" height="50%" scrolling="auto"></iframe></code>	<ul style="list-style-type: none"> • <i>right</i> : alignement du cadre à droite • <i>justify</i> : cadre justifié
<code><noembed></code>	Cet élément dépassé et non standardisé de fournir une alternative de contenu pour les navigateurs ne supportant pas l'élément <code><embed></code> ou des catégories de contenu qu'un auteur aimerait utiliser. Cet élément a été rendu obsolète à partir de la version HTML 4.01 et a été remplacé par <code><object></code> . Le contenu alternatif doit être inséré entre la balise d'ouverture et celle de fermeture de <code><object></code>	
<code><object></code>	Cet élément représente une ressource externe qui peut être interprétée comme une image, un contexte de navigation imbriqué ou une ressource à traiter comme un <i>plugin</i> .	
<code><param></code>	Cet élément définit les paramètres qui peuvent être employés dans un élément <code><object></code> .	
<code><picture></code>	Cet élément est un conteneur utilisé afin de définir zéro ou plusieurs éléments <code><source></code> destinés à un élément <code></code> . Le navigateur choisira la source la plus pertinente selon la disposition de la page (les contraintes qui s'appliquent à la boîte dans laquelle l'image devra être affichée), selon l'appareil utilisé (la densité de pixels de l'affichage par exemple avec les appareils hiDPI) et selon les formats pris en charge (ex. WebP pour les navigateurs Chromium ou PNG pour les autres). Si aucune correspondance n'est trouvée parmi les éléments <code><source></code> , c'est le fichier défini par l'attribut <code>src</code> de l'élément <code></code> qui sera utilisé.	
<code><source></code>	Cet élément définit différentes ressources média pour un élément <code><picture></code> , <code><audio></code> ou <code><video></code> . C'est un élément vide : il ne possède pas de contenu et ne nécessite pas de balise fermante. Il est généralement utilisé pour	

Élément	Description	Attribut
	distribuer le même contenu en utilisant <u>les différents formats pris en charge par les différents navigateurs</u> .	

2.11. Les scripts

Afin de créer du contenu dynamique et des applications web, des langages de script peuvent être utilisés pour manipuler le document HTML. Certains éléments HTML permettent de gérer les liens entre les scripts et le document.

Élément	Description
<code><canvas></code>	Cet élément permet de modifier une zone graphique via un script (habituellement en JavaScript ou grâce à WebGL). Il peut par exemple être utilisé afin de dessiner des graphiques, manipuler des images ou jouer des animations.
<code><noscript></code>	Cet élément définit un fragment HTML qui doit être affiché si les fonctionnalités de script ne sont pas prises en charge ou si elles sont désactivées.
<code><script></code>	Cet élément est utilisé pour intégrer ou faire référence à un script exécutable. Cela fait généralement référence à du code JavaScript mais ce peut également être un autre type de script (par exemple WebGL).

2.12. Gestion de l'édition

Ces éléments permettent d'indiquer si des portions du texte ont été modifiées.

Élément	Description
<code></code>	Cet élément représente une portion de texte ayant été supprimée d'un document. Cet élément est souvent (mais pas nécessairement) affiché rayé.
<code><ins></code>	Cet élément représente un fragment de texte qui a été ajouté dans un document.

2.13. Les cadres

Élément	Description	Attribut
<code><frameset></code> <code></frameset></code>	Cette balise permet de signaler que la page actuelle est une description du découpage d'une fenêtre en cadre. Elle remplace la balise <code><body></code> .	cols="..." : définit le nombre de "colonnes" partageant la fenêtre, ou bien, la taille en pixels ou en pourcentage et séparés par des virgules de chacun des cadres.

Élément	Description	Attribut
	<p>La description du partage de la fenêtre doit se terminer par la balise <code></frameset></code> .</p> <p>Entre des balises <code><frameset></code> et <code></frameset></code>, il peut y avoir :</p> <ul style="list-style-type: none"> soit la description avec des balises <code><frame /></code> d'un ou plusieurs cadres déclaré dans la balise <code><frameset></code> soit une autre déclaration de découpage avec un nouveau couple de balise <code><frameset></code> <code></frameset></code>. <p>Attention, utiliser uniquement des balises <code><frameset></code> ne sert pas à grand-chose puisqu'elles permettent juste de définir des zones dans une page, mais, n'incluent aucun contenu.</p> <p>Pour inclure un contenu dans les cadres de la page, il faut utiliser les balises <code><frame /></code> .</p>	<p>Dans ce dernier cas, la valeur <code>"*"</code> correspond à "tout le reste de la place disponible".</p> <p>rows="..." : définit le nombre de "lignes" partageant la fenêtre, ou bien, la taille en pixels ou en pourcentage et séparés par des virgules de chacun des cadres. Dans ce dernier cas, la valeur <code>"*"</code> correspond à "tout le reste de la place disponible".</p> <p>frameborder="..." : indique si les bordures des cadres du frameset doivent être affichées.</p> <ul style="list-style-type: none"> <code>0</code> : aucune bordure <code>1</code> : les bordures apparaissent <p>Les attributs pour DHTML de la balise <code><frameset></code></p> <p>onload="..." : permet de déclarer un évènement au chargement de la page du frameset.</p> <p>onunload="..." : permet de déclarer un évènement à la fermeture de la page du frameset.</p>

Résumé des cadres :

Code	Effet
<code><frameset cols="2"</code>	Partage une fenêtre en 2 cadres séparés verticalement.
<code><frame src="page_1.html" /></code>	La page html_1.html sera chargée dans le premier cadre (celui de gauche).
<code><frame src="page_2.html" /></code>	La page html_2.html sera chargée dans le deuxième cadre (celui de droite).
<code></frameset></code>	Fermeture de la description de ce groupe de cadre.
<code><frameset rows="20%,*"</code>	Partage une fenêtre en 2 cadres séparés horizontalement.
<code><frame src="page_3.html" /></code>	La page html_3.html sera chargée dans le premier cadre (celui du haut) qui aura une largeur de 20% de la largeur de la fenêtre.
<code><frame src="page_4.html" /></code>	La page html_4.html sera chargée dans le deuxième cadre (celui du bas), et occupera le reste de la fenêtre.
<code></frameset></code>	Fermeture de la description de ce groupe de cadre.
<code><frameset rows="2"</code>	Partage une fenêtre en 2 cadres séparés horizontalement.
<code><frame src="page_3.html" /></code>	La page html_3.html sera chargée dans le premier cadre (celui du haut).
<code>frameset cols="2"</code>	Partage le deuxième cadre en 2 cadres séparés verticalement.
<code><frame src="page_1.html" /></code>	La page html_1.html sera chargée dans le premier cadre de ce groupe, et sera donc situé en bas à gauche.

Code	Effet						
<code><frame src="page_2.html" /></code>	La page html_2.html sera chargée dans le deuxième cadre de ce groupe, et sera donc situé en bas à droite.						
<code></frameset></code>	Fermeture de la description du groupe de cadre inférieur.						
<code></frameset></code>	Fermeture de la description du cadre général.						
<pre> <frameset cols="2"> <frameset rows="3"> <frame src="page_1.html"/> <frame src="page_2.html"/> <frame src="page_3.html"/> </frameset> <frameset rows="3"> <frame src="page_4.html"/> <frame src="page_5.html"/> <frame src="page_6.html"/> </frameset> </frameset> </pre>	<p>Ici, est présenté une manière plus correcte de découper une fenêtre pour obtenir un résultat similaire à l'exemple de la mauvaise utilisation de balise.</p> <p>La fenêtre sera donc découpée en deux colonnes ayant chacune trois lignes de cadre.</p> <p>Les pages seront chargées de la manière suivante :</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>page_1.html</td> <td>page_4.html</td> </tr> <tr> <td>page_2.html</td> <td>page_5.html</td> </tr> <tr> <td>page_3.html</td> <td>page_6.html</td> </tr> </table>	page_1.html	page_4.html	page_2.html	page_5.html	page_3.html	page_6.html
page_1.html	page_4.html						
page_2.html	page_5.html						
page_3.html	page_6.html						

2.14. Les tableaux

Il existe 4 balises principales pour définir un tableau :

1. La balise `<table>` qui permet d'annoncer l'ouverture d'un tableau et de contrôler son apparence.
2. La balise `<tr>` qui permet de créer une ligne.
3. La balise `<th>` qui permet de créer une cellule de titre.
4. La balise `<td>` qui permet de créer une cellule de donnée.

Un tableau doit être contenu entre les balises `<table>` et `</table>`.

Le contenu du tableau doit être rempli ligne par ligne.

Chaque ligne doit commencer par la balise `<tr>` et se terminer par la balise `</tr>`.

Une ligne est constituée de cellules qui doivent être délimitées par les balises :

- `<th>` et `</th>` pour les cellules correspondant à des titres, et par
- `<td>` et `</td>` pour les cellules correspondant à des données.

Ainsi un tableau avec 2 cellules de titre sur la première ligne, et 2 cellules de données sur la deuxième ligne doit être codé de la manière suivante :

Code d'un tableau avec 2 cellules de titre, et 2 cellules de données.	
Code	Effet
<code><table border="2" ></code>	ouverture du tableau ayant des bordures de 2 pixels
<code><tr></code>	début d'une ligne (ligne 1)
<code><th>titre 1</th></code>	cellule de titre (cellule 1 de la ligne 1)
<code><th>titre 2</th></code>	cellule de titre (cellule 2 de la ligne 1)
<code></tr></code>	fin de la ligne (fin de la ligne 1)
<code><tr></code>	début d'une ligne (ligne 2)
<code><td>donnée 1</td></code>	cellule de donnée (cellule 1 de la ligne 2)
<code><td>donnée 2</td></code>	cellule de donnée (cellule 2 de la ligne 2)
<code></tr></code>	fin de la ligne (fin de la ligne 2)
<code></table></code>	fin du tableau

Le résultat est le tableau suivant :

titre 1	titre 2
donnée 1	donnée 2

Élément	Description	Attribut
<p><code><table></code> <code></table></code></p>	<p>Cet élément permet de représenter un tableau de données, c'est-à-dire des informations exprimées sur un tableau en deux dimensions.</p>	<p><code>width="nb%"</code> : définit la largeur de la table. <code>cols="..."</code> : définit le nombre de colonnes de la table. <code>border="..."</code> : définit la largeur en pixel des bordures du tableau. <code>cellspacing="..."</code> : définit l'espacement entre les cellules d'un tableau. <code>cellpadding="..."</code> : définit l'espacement entre les bordures d'une cellule et ce qu'elle contient. <code>frame="..."</code> : définit les bords visibles d'une table (les autres sont donc invisibles), l'attribut peut prendre les valeurs suivantes:</p> <ul style="list-style-type: none"> <code>void</code>: aucune bordure <code>above</code>: bordure supérieure uniquement <code>below</code>: bordure inférieure uniquement <code>hsides</code>: bordure supérieur et inférieur uniquement <code>lhs</code>: bordure gauche uniquement <code>rhs</code>: bordure droite uniquement <code>vsides</code>: bordure gauche et droite uniquement <code>box</code>: toutes les bordures <code>border</code>: toutes les bordures <p><code>rules="..."</code> : définit les bordures intérieur visibles d'une table (les autres sont donc invisibles), l'attribut peut prendre les valeurs suivantes:</p> <ul style="list-style-type: none"> <code>none</code>: aucune bordure intérieur <code>groups</code>: uniquement les bordures qui entourent les groupes de ligne et les groupes de colonnes. <code>rows</code>: bordure entre les lignes uniquement <code>cols</code>: bordure entre les colonnes uniquement <code>all</code>: toutes les bordures <p>Attention, certains navigateurs comme IE 5.5 gèrent mal ces attributs si un seul des 2 est utilisé ; il est donc préférable de définir les 2 attributs même si vous n'en a besoin que d'un seul.</p>

Exemple d'utilisation de la balise table	
Code	Résultat
<pre><table width="80%" border="10" cellspacing="15" cellpadding="8"> <tr> <td>première cellule</td> <td>deuxième cellule</td> <td>troisième cellule</td> </tr> <tr> <td>quatrième cellule</td> <td>cinquième cellule</td> <td>sixième cellule</td> </tr> </table></pre>	

Élément	Description	Attribut
<p><code><caption></code> <code></caption></code></p>	<p>Cet élément représente la légende (ou le titre) d'un tableau. Il doit être le premier élément parmi les descendants de l'élément <code><table></code>. La mise en forme CSS peut placer cet élément à un autre endroit par rapport au tableau et on pourra notamment utiliser les propriétés <code>caption-side</code> et <code>text-align</code>.</p>	<p>align : permet d'aligner horizontalement la légende.</p> <ul style="list-style-type: none"> <code>top</code> : titre au-dessus du tableau. <code>bottom</code> : titre au-dessous du tableau.

Exemple d'utilisation de la balise caption	
Code	Résultat
<pre><table border="2"> <caption>Titre du tableau</caption> <tr> <th>titre 1</th> <th>titre 2</th> </tr> <tr> <td>donnée 1</td> <td>donnée 2</td> </tr> </table></pre>	

Élément	Description	Attribut
<p><code><th></code> <code></th></code></p>	<p>Cet élément définit une cellule d'entête d'un tableau pour un groupe de cellule. La nature de ce groupe est définie grâce aux attributs <code>scope</code> et <code>headers</code>.</p>	<p>axis="..." : nom abrégé axes="..." : noms des axes définissant des en-têtes de colonne et de ligne correspondant à la cellule. rowspan="..." : nombre de ligne sur lesquelles s'étend une cellule colspan="..." : nombre de colonnes sur lesquelles s'étend une cellule align="..." : permet d'aligner le contenu des cellules.</p> <ul style="list-style-type: none"> <code>left</code> : alignement à gauche <code>center</code> : contenu centré <code>right</code> : alignement à droite <code>justify</code> : contenu justifié <code>char</code> : alignement par rapport à un caractère. <p>char="..." : définit le caractère d'alignement des contenus des cellules lorsque la valeur de l'attribut align est "char" .</p>
<p><code><td></code> <code></td></code></p>	<p>Cet élément <code><td></code> définit une cellule d'un tableau qui contient des données. Cet élément fait partie du <i>modèle de tableau</i>.</p>	
<p><code><tr></code> <code></tr></code></p>	<p>Cet élément définit une ligne de cellules dans un tableau. Une ligne peut être constituée d'éléments <code><td></code> (les données des cellules) et <code><th></code> (les cellules d'en-têtes).</p>	

Élément	Description	Attribut
		<p>charoff="..." : permet de donner la position en pourcentage de la cellule, ou en pixels du premier caractère d'alignement (valeur de l'attribut char) sur une ligne.</p> <p>valign="..." : permet de définir l'alignement verticale du tableau</p> <ul style="list-style-type: none"> • <i>top</i> : contenu situé en haut des cellules. • <i>middle</i> : contenu situé au milieu des cellules. • <i>bottom</i> : contenu situé au bas des cellules.
<p><code><colgroup></code> <code></colgroup></code></p>	<p>Cet élément définit un groupe de colonnes au sein d'un tableau.</p>	<p>span="..." : nombre de colonne dans le groupe.</p> <p>width="..." : largeur de chaque colonne du groupe</p> <p>align : définir l'alignement horizontale du contenu des cellules du groupe.</p> <ul style="list-style-type: none"> • <i>left</i> : alignement à gauche • <i>center</i> : contenue centré • <i>right</i> : alignement à droite • <i>justify</i> : contenu justifié • <i>char</i> : alignement par rapport à un caractère. <p>char="..." : définit le caractère d'alignement des contenus des cellules lorsque la valeur de l'attribut align est "char" .</p> <p>charoff="..." : permet de donner la position en pourcentage de la cellule, ou en pixels du premier caractère d'alignement (valeur de l'attribut char) sur une ligne.</p> <p>valign="..." : définir l'alignement verticale du contenu des cellules du groupe.</p> <ul style="list-style-type: none"> • <i>top</i> : contenu situé en haut des cellules. • <i>middle</i> : contenu situé au milieu des cellules. • <i>bottom</i> : contenu situé au bas des cellules.
<p><code><col></code> <code></col></code></p>	<p>Cet élément définit une colonne appartenant à un tableau et est utilisé afin de définir la sémantique commune à toutes ses cellules. On trouve généralement cet élément au sein d'un élément <code>{{HTMLElement("colgroup")}}</code>.</p>	

Exemple d'utilisation des attributs colspan et rowspan des balises <th> et <td>

On veut créer un tableau ayant l'aspect suivant (les chiffres identiques appartiennent en fait à une seule cellule).

1	2	2	3
1	4	5	5
1	6	5	5

Code	Effet
<pre> <table align="center" width="80%" border="2"> <tr> <td rowspan="3" > 1 </td> <td colspan="2" > 2 </td> <td > 3 </td> </tr> <td > 4 </td> <td colspan="2" > 5 </td> </tr> <tr> <td > 6 </td> </tr> </table> </pre>	

Élément	Description	Attribut
<code><thead></code> <code></thead></code>	Cet élément définit un ensemble de lignes qui définit l' en-tête des colonnes d'un tableau.	align="..." : permet d'aligner le contenu des cellules. <ul style="list-style-type: none"> <i>left</i> : alignement à gauche <i>center</i> : contenu centré <i>right</i> : alignement à droite <i>justify</i> : contenu justifié <i>char</i> : alignement par rapport à un caractère. char="..." : définit le caractère d'alignement des contenus des cellules lorsque la valeur de l'attribut align est "char" charoff="..." : permet de donner la position en pourcentage de la cellule, ou en pixels du premier caractère d'alignement (valeur de l'attribut char) sur une ligne. valign="..." : permet de définir l'alignement verticale du contenu des cellules dans leur case: <ul style="list-style-type: none"> <i>top</i> : contenu situé en haut des cellules. <i>middle</i> : contenu situé au milieu des cellules. <i>bottom</i> : contenu situé au bas des cellules.
<code><tbody></code> <code></tbody></code>	Cet élément permet de regrouper un ou plusieurs éléments <tr> afin de former le corps d'un tableau HTML (<code><table></code>).	
<code><tfoot></code> <code></tfoot></code>	Cet élément permet de définir un ensemble de lignes qui résument les colonnes d'un tableau.	

exemple permettant de situer la position des différents groupes de lignes		
Les cellules suivantes sont comprises entre les balises <code><thead align="right" ></code> et <code></thead></code>		
head 1	head 2	head 3
head 4	head 5	head 6
Les cellules suivantes sont comprises entre les balises <code><tfoot align="left" ></code> et <code></tfoot></code>		
foot 1	foot 2	foot 3
foot 4	foot 5	foot 6

exemple permettant de situer la position des différents groupes de lignes		
Les cellules suivantes sont comprises entre les balises <code><thead align="right" ></code> et <code></thead></code>		
head 1	head 2	head 3
head 4	head 5	head 6
Les cellules suivantes ne sont comprises entre aucune balise de type <code>thead</code> , <code>tbody</code> , ou <code>tfoot</code> .		
basic 1.1	basic 2.1	basic 3.1
basic 4.1	basic 5.1	basic 6.1
Les cellules suivantes sont comprises entre les balises <code><tbody align="center" ></code> et <code></tbody></code>		
body 1.1	body 2.1	body 3.1
body 4.1	body 5.1	body 6.1
Les cellules suivantes ne sont comprises entre aucune balise de type <code>thead</code> , <code>tbody</code> , ou <code>tfoot</code> .		
basic 1.2	basic 2.2	basic 3.2
basic 4.2	basic 5.2	basic 6.2
Les cellules suivantes sont comprises entre les balises <code><tbody align="justify" ></code> et <code></tbody></code>		
body 1.2	body 2.2	body 3.2
body 4.2	body 5.2	body 6.2
Les cellules suivantes ne sont comprises entre aucune balise de type <code>thead</code> , <code>tbody</code> , ou <code>tfoot</code> .		
basic 1.3	basic 2.3	basic 3.3
basic 4.3	basic 5.3	basic 6.3

2.15. Les formulaires

HTML fournit différents éléments qui permettent de créer des formulaires pouvant être remplis par les utilisateurs du site ou de l'application web.

Élément	Description
<code><button></code> <code></button></code>	Cet élément est utilisé afin de créer un contrôle interactif ayant la forme d'un bouton et qui pourra être utilisé dans un formulaire ou dans le document.
<code><datalist></code> <code></datalist></code>	Cet élément contient un ensemble d'éléments <code><option></code> qui représentent les valeurs possibles pour d'autres contrôles.
<code><fieldset></code> <code></fieldset></code>	Cet élément est utilisé afin de regrouper plusieurs contrôles interactifs ainsi que des étiquettes (<code><label></code>) dans un formulaire web.
<code><form></code> <code></form></code>	Cet élément représente une section d'un document qui contient des contrôles interactifs permettant à un utilisateur d'envoyer des données à un serveur web.
<code><input></code> <code></input></code>	Cet élément <code><input></code> est utilisé pour créer un contrôle interactif dans un formulaire web qui permet à l'utilisateur de saisir des données. Les saisies possibles et le comportement de l'élément <code><input></code> dépend fortement de la valeur indiquée dans son attribut <code>type</code> .
<code><label></code> <code></label></code>	Cet élément représente une légende pour un objet d'une interface utilisateur. Il peut être associé à un contrôle en utilisant l'attribut <code>for</code> ou en plaçant l'élément du contrôle à l'intérieur de l'élément <code><label></code> . Un tel contrôle est appelé <i>contrôle étiqueté</i> par l'élément <code><label></code> .

Élément	Description
<code><legend></code> <code></legend></code>	Cet élément représente une légende pour le contenu de son élément parent <code><fieldset></code> .
<code><meter></code> <code></meter></code>	Cet élément représente une valeur scalaire dans un intervalle donné ou une valeur fractionnaire.
<code><optgroup></code> <code></optgroup></code>	Cet élément utilisé dans un formulaire, permet de créer un groupe d'options parmi lesquelles on peut choisir dans un élément <code><select></code> .
<code><option></code> <code></option></code>	Cet élément utilisé dans un formulaire, permet de représenter un contrôle au sein d'un élément <code><select></code> , <code><optgroup></code> ou <code><datalist></code> . Cet élément peut donc représenter des éléments d'un menu dans un document HTML.
<code><output></code> <code></output></code>	Cet élément représente un conteneur dans lequel un site ou une application peut injecter le résultat d'un calcul ou d'une action utilisateur.
<code><progress></code> <code></progress></code>	Cet élément indique l'état de complétion d'une tâche et est généralement représenté par une barre de progression.
<code><select></code> <code></select></code>	Cet élément représente un contrôle qui fournit une liste d'options parmi lesquelles l'utilisateur pourra choisir.
<code><textarea></code> <code></textarea></code>	Cet élément représente un contrôle qui permet d'éditer du texte sur plusieurs lignes.

2.16. Éléments interactifs

HTML fournit différents éléments qui permettent de créer des interfaces utilisateur interactives.

Élément	Description
<code><details></code> <code></details></code>	Cet élément est utilisé comme un outil permettant de révéler une information. Un résumé ou un intitulé peuvent être fournis grâce à un élément <code><summary></code> .
<code><dialog></code> <code></dialog></code>	Cet élément représente une boîte de dialogue ou un composant interactif (par exemple un inspecteur ou une fenêtre).
<code><menu></code> <code></menu></code>	Cet élément représente un groupe de commandes que l'utilisateur peut utiliser ou activer. Il peut être utilisé afin de créer des menus (affichés en haut d'un écran par exemple) et des menus contextuels (qui apparaissent au clic-droit ou après avoir cliqué sur un bouton).
<code><menuitem></code> <code></menuitem></code>	Cet élément représente une commande qu'un utilisateur peut utiliser via un menu contextuel ou un menu rattaché à un bouton.
<code><summary></code> <code></summary></code>	Cet élément représente une boîte permettant de révéler le contenu d'un résumé ou d'une légende pour le contenu d'un élément <code><details></code> . En cliquant sur l'élément <code><summary></code> , on passe de l'état affiché à l'état masqué (et vice versa) de l'élément <code><details></code> parent.

2.17. Web-Components

Les composants web (ou « *Web Components* ») permettent de créer et d'utiliser des éléments personnalisés comme s'ils étaient des éléments HTML classiques. Cela permet également de créer ses propres versions des éléments standards HTML.

Élément	Description
<code><content></code> <code></content></code>	Cet élément était utilisé au sein d'un shadow DOM comme point d'insertion. Cet élément est désormais obsolète et n'était pas conçu pour être utilisé dans un document HTML simple mais pour être employé dans des composants web . Il est désormais remplacé par l'élément <code><slot></code> qui permet de créer un point d'insertion pour un <i>shadow DOM</i> au sein du DOM.
<code><element></code> <code></element></code>	Cet élément était conçu pour être utilisé afin de définir des éléments DOM personnalisés, il a été retiré de la spécification. Il a été retiré en faveur d'outils JavaScript qui permettront de créer de nouveaux éléments personnalisés, par exemple avec les Web Components.
<code><shadow></code> <code></shadow></code>	Cet élément inclut uniquement les attributs universels .
<code><slot></code> <code></slot></code>	Cet élément représente un emplacement d'un composant web qu'on peut remplir avec son propre balisage. On peut ainsi obtenir un document construit avec différents arbres DOM. Cet élément fait partie des outils relatifs aux composants web (Web Components) .
<code><template></code> <code></template></code>	Cet élément (ou <i>Template Content</i> ou modèle de contenu) est un mécanisme utilisé pour stocker du contenu HTML (côté client) qui ne doit pas être affiché lors du chargement de la page mais qui peut être instancié et affiché par la suite grâce à un script JavaScript.

`` : lien hypertexte

2.18. Autres HTML5

`<header></header>` : en-tête

`<section></section>` : section

`<article></article>` : sous-section pour un article (de blog par exemple)

`<nav></nav>` : bloc de navigation

`<aside></aside>` : encart

`<figure></figure>` : images et/ou vidéos ←----

`<footer></footer>` : pied-de-page

2.19. Quelques balises et attributs dépréciés à ne plus utiliser

- `<center></center>` (pour centrer)
- `` (pour les polices de caractères)
- `<strike></strike>` (pour barrer du texte)
- `<s></s>` (pour souligner du texte)
- `<applet></applet>` (pour les applets java)
- align
- bgcolor
- noshade

- text
- vlink

2.20. Lectures complémentaires (et très utiles)

- [Le Doctype qu'il vous faut \(pompage.net\)](#)
- [Servir du XHTML 1.0 \(W3C\)](#)
- [Index des éléments HTML](#)
- [Index des attributs](#)
- [HTML5](#)
- [Nouveautés HTML5 \(Alsacreation\)](#)

3. LE STYLE EN CASCADE AVEC CSS

C'est un langage pour styliser les pages Web.

Sources du document :

<http://css.mammothland.net/premiers-pas-en-css.php>

<http://www.css-faciles.com/index.php>

3.1. Principe des CSS

Le design d'un site évolue toujours au fil du temps. Le problème, lorsqu'on n'utilise pas de feuilles de style, c'est qu'il faut reprendre toutes les pages une à une pour modifier une police de caractère ou une couleur de fond... Avec les "**Cascading Style Sheets**" (CSS), ce lourd handicap est résolu.

Le CSS 1 a été créé en 1996 définissant les marges, les couleurs et les polices de caractères des pages web ; en 1999, CSS 2 a permis d'utiliser un positionnement plus précis et le CSS 3 a apporté les bordures arrondies, les dégradés, les ombres...

C'est dans la feuille de style que l'on va déclarer toute la mise en forme des pages : la couleur de fond, les polices de caractère, leurs couleurs, etc.

Les éléments de style sont soit :

- **créés dans un fichier .css externe** (recommandé),
- **intégrés dans l'entête <head> du fichier .html** (valable pour tout le document suivant),
- **intégrés entre les balises <style> et </style> du fichier .html** (peu recommandé).

3.1.1. Feuille de style externe

Une feuille de style CSS externe peut se faire avec un éditeur de texte et il portera l'extension **.css**. On la liera ensuite à la page html à l'aide d'un **link** placé dans l'en-tête de la page (entre <head> et </head>).

Par exemple (conçu à l'aide d'un éditeur de texte) :

Coté HTML (nom du fichier *test.html*) : `<meta charset="utf-8" />` (indique l'encodage utilisé)
`<link rel="stylesheet" href="style.css" />`

Coté CSS (nom du fichier *style.css*) : `p {color :blue ; }`

3.1.2. Feuille de style interne (inclus dans l'entête du fichier html)

Le code CSS est inséré dans une balise <style> à l'intérieur de l'entête <head>

```
<head><meta charset="utf-8" /><style> p {color :blue ; } </style></head>
```

3.1.3. Style CSS inclus dans le code html

Pour attribuer un style à un endroit unique du document :

<code><body> <p style="color : blue ;" >texte</p></body></code>	texte
---	-------

3.1.4. Style CSS appliqué sous condition

Appliquer un style suivant les attributs <class> ou <id> sur les balises de titres, paragraphe, image...:

<code><h1 class="introduction">texte1</h1> <p class="introduction">texte 2</p> texte 3 <style>.introduction {color : red}</style></code>	texte1 texte 2 texte 3
--	------------------------------

A noter que l'attribut <id> est identique à <class> mais ne peut être utilisé qu'une seule fois et la propriété du fichier CSS commence par # (et non le « . »).

<code> <style> #logo{propriete} </style></code>	☒Logo du site
---	---------------

3.2. Les commandes CSS

3.2.1. Mise en forme d'un texte

Il est très important de bien choisir le formatage des polices de caractères utilisées sur une page web, car pour une majorité de sites, le principal contenu est sous forme textuelle.

La taille des caractères se déclare par l'attribut *font-size* :

- Taille absolue : *font-size* : *1in* (1 inch= 2,45 cm = 72 pt= 6 pc ou px)
- Taille relative : *font-size* : *1em* (=100%)
 large=1.4 em=140%ex ,
 medium=1em=ex (em =taille normale) ,
 small=0.8 em=80%ex
- Type police : *font-family* : *arial*, (« *comic sans MS* », *sans serif*...)
- Alignement : *texte-align* : *center*, (*justify*, *right*...)
- Couleur texte : *color* : # 000000 (=noir, #FFFFFF=blanc ou *rgb(0,0,0)*, *rgb(255,255,255)*)

3.2.2. Gérer les marges en CSS

Pour affecter des marges aux différents éléments d'une page web, les propriétés CSS à utiliser sont **margin**, pour les marges extérieures, et **padding** pour les marges intérieures.

3.2.3. Sélecteurs id et class

Les sélecteurs CSS : **id** identifie un élément unique (comme une balise ou un bouton) et **class** sert à regrouper des éléments qui ont des propriétés communes, ils servent tous les deux à définir des styles particuliers que l'on veut reproduire de façon ponctuelle ou récurrente dans les pages (X)HTML.

3.2.4. Image de fond

Mettre une image de fond dans une page web est courant. Avec tout éditeur de page html, l'option classique est de générer un code du type : `<body background="images/image_de_fond.png">`, auquel on peut même ajouter `bgproperties="fixed"` qui rend fixe l'image de fond ; mais cet attribut ne fonctionne qu'avec MSIE. Avec les feuilles de style on peut obtenir beaucoup plus !

3.2.5. Effets rollover

Pour tout effet de **rollover** (c'est à dire de changement d'aspect d'un élément au survol de la souris), la pseudo-classe CSS à utiliser est : **hover**. Cela permet d'exécuter plusieurs tâches : changements d'apparence de texte, de couleur de fond, d'apparition (ou de disparition) d'image, de zoom d'image, etc. Cependant Internet Explorer n'implémente cette propriété qu'à partir de sa version 7.

3.2.6. Aligner une image et du texte

Pour positionner une image à droite ou à gauche d'un texte, on utilise la propriété CSS **float** qui fait "flotter" les éléments :

Ex : ``

3.2.7. Puces en images

Des listes à puces avec des images sont plus attrayantes que les styles donnés par défaut (disc, circle, square). Pour cela, inutile d'incorporer une image à chaque item dans le code (x)html. Grâce aux CSS, on peut automatiser facilement la mise en image des listes à puces.

3.3. Le CSS3

<http://css.mammothland.net/css3/>

3.3.1. Réaliser des coins arrondis

Arrondir les coins est sûrement l'un des effets graphiques les plus recherchés par les webdesigners. Pour cela, on a longtemps utilisé des images avec plus ou moins de bonheur et de facilité pour obtenir l'effet recherché. Tous les navigateurs récents permettent à présent d'arrondir les coins de cadres, d'images, de tableaux etc.

3.3.2. Créer des ombres sur les boites et du texte

Ombre des boites ou du texte sans images c'est possible ! Exemple : `text-shadow:1px 1px 2px gray;`

3.3.3. Images multiples dans un même background

La propriété **background-image** n'acceptait, en CSS 2.1, qu'une seule image. Pour pouvoir en mettre plusieurs à différents endroits d'une page, il fallait déclarer plusieurs **div** ayant chacun sa propre image

de fond. Lourd... Les CSS3 permettent des **background** multiples, y compris Internet Explorer (version 9).

3.3.4. Réaliser des fonds en dégradé sans image

Avant les CSS3, pour réaliser de jolis boutons ou tout autre fond de couleur en dégradé, il convenait de réaliser une image que l'on déclarait en image de fond à l'aide de **background-image**.

On peut maintenant s'en passer grâce à de nouvelles valeurs de **background** : **linear-gradient** et **radial-gradient**.

3.3.5. Déclarer une police de caractère non standard avec @font-face

On ne pouvait jusqu'alors utiliser qu'un petit nombre de fontes pour ses sites web : Arial, Helvetica, Times New Roman...! Ou alors il fallait faire des images. Grâce à **@font-face** on peut à présent utiliser des polices de caractère exotiques et laisser cours à son inventivité.

3.3.6. Animations et transitions d'images ou d'objets

L'ère des gifs animés est morte et celle de flash est en déclin grâce aux CSS3 et à HTML5 qui apportent leur flopée de nouveaux outils pour animer nos sites ! **animation** et **transition** permettent d'apporter du mouvement aux pages web en quelques lignes de code.

3.3.7. Les Media-Queries

Avec l'avènement des smartphones ou autre tablette, la multiplication des tailles et des résolutions d'écran, il devient de plus en plus aléatoire de ne réaliser qu'une seule disposition graphique de sa page web... Comment en effet visualiser correctement sur un smartphone une page qui aurait été pensée pour une résolution minimale de 1900px ?

Les **Media-Queries**, permettent de cibler différents cas et ainsi d'adapter la restitution de sa page html à différentes caractéristiques des terminaux.

3.3.8. Boutons en CSS

Le bouton élégant qui change d'aspect au passage de la souris est très utilisé sur les pages web. Longtemps, la majorité de ces boutons étaient animés par un javascript ou pire (car plus lourd) par un applet java. Avec les CSS la légèreté et la simplicité est de mise pour créer de beaux boutons.

3.3.9. Zoom d'image

Zoomer une image, à l'aide des **CSS2**, lors de son survol de la souris peut se faire très facilement. Il suffit de combiner le pseudo-élément **:hover** à la balise **img** (image). Mais tant que MSIE6 existera, il faudra utiliser d'autres moyens pour le même résultat. En effet, il n'implémente **:hover** que pour les liens...

3.3.10. Vignettes (Thumbnails)

Zoomer une image avec le langage CSS est aisé. Faire des vignettes zoomable façon "thumbnail" demande un peu plus de dextérité.

3.3.11. Menu déroulant

Le menu déroulant sera fait sous forme d'une liste à puces que l'on va contraindre à devenir horizontale, mais ce sera, en plus, une liste à puces imbriquée qui va être provoquée par **rollover**.

3.3.12. Générer du contenu

Avec les pseudo-éléments **:before** et **:after**, on peut automatiser l'inclusion de caractères ou d'images grâce à l'attribut **content**. On peut aussi faire ressortir des informations de "noyées" dans le code source.

3.3.13. Formulaires

La tentation est grande de mettre les champs dans des tableaux pour les aligner, et de rajouter du javascript pour un effet focus ou se passer un bouton d'envoi. Et l'accessibilité dans tout ça ??? Mais avec les CSS, on peut améliorer l'aspect d'un formulaire et rester accessible !

3.3.14. Rasmol et object

La balise **<object>** a été introduite dans les [spécifications du HTML 4.0](#) [en] pour [présenter des objets](#) (applets java, videos, sons, flash etc.) dans les pages HTML.

Auparavant, la balise **<embed>** était couramment utilisée, mais celle-ci, initialement balise propriétaire de Netscape, n'a jamais été homologuée par le W3C.

Les molécules 3D "rasmol" font partie de ce type d'objets à utiliser avec la balise **<object>**.

3.4. DESCRIPTION DETAILLEE DES COMMANDES CSS

3.4.1. Police de caractère, tailles, structure

Commandes	Description	Choix des valeurs	Exemples
font	Cumule les propriétés de police.	Combine : font-family, font-face, font-size, font-weight, font-style, font-variant.	Voir ci-après
font-family	Liste des noms de police de caractères	Indique les noms de polices possibles par <u>ordre de préférence</u> : font-family:police1, police2, police3; Utilisez des guillemets si le nom de la police comporte des espaces. Essayez de toujours mettre comme dernière police possible : "serif" ou "sans-serif". NB : le nom de la police doit toujours être placé en dernier dans la liste.	<i>font-family</i> : "Arial Black", Arial, Verdana, serif; police1, police2, police3, serif, sans-serif, monospace
@font-face	Police personnalisée	Nom et source de la police	
font-size	Taille relative ou absolue de la police. Les modifications relatives se font par rapport à la taille de base de l'élément	Plusieurs unités sont possibles : • px (pixels) • % (pourcentage, 100% = normal) • em (taille relative, 1.0 = normal) • ex (taille relative à la hauteur de la lettre "x". 1.0 = normal) • Valeur absolue :	ex : <i>1.3em, 16px, 120%...</i>

	dans lequel se trouve la balise. Si une balise em avec une taille <i>larger</i> se trouve dans un paragraphe de taille <i>medium</i> , la balise em sera de taille <i>large</i>	<ul style="list-style-type: none"> o <i>xx-small</i> : très très petit o <i>x-small</i> : très petit o <i>small</i> : petit o <i>medium</i> : moyen o <i>large</i> : grand o <i>x-large</i> : très grand o <i>xx-large</i> : très très grand • Valeur relative <ul style="list-style-type: none"> o <i>Larger</i> : plus grand o <i>Smaller</i> : plus petit o +/- entier o +/-pourcentage 	
font-weight	Epaisseur de la police. L'effet dépend de la police utilisée.	<ul style="list-style-type: none"> • <i>bold</i> : gras • <i>bolder</i> : plus gras • <i>lighter</i> : plus fin • <i>normal</i> : pas gras (par défaut) 	
font-style	Style de la police	<ul style="list-style-type: none"> • <i>italic</i> : italique • <i>oblique</i> : autre façon de mettre en italique • <i>normal</i> : normal (par défaut) 	
font-variant	Petites capitales	<ul style="list-style-type: none"> • <i>small-caps</i> : petites majuscules • <i>normal</i> : normal (par défaut) 	
text-decoration	Soulignement, ligne au-dessus, barré ou clignotant	<ul style="list-style-type: none"> • <i>underline</i> : souligné • <i>overline</i> : ligne au-dessus • <i>line-through</i> : barré • <i>blink</i> : clignotant • <i>none</i> : normal (par défaut) 	
text-transform	Mise en majuscule	<ul style="list-style-type: none"> • <i>uppercase</i> : met en majuscules toutes les lettres du texte • <i>lowercase</i> : met en minuscules toutes les lettres du texte • <i>capitalize</i> : met en majuscule la première lettre de chaque mot • <i>none</i> : normal (par défaut) 	<i>Font : bold, 16px, Arial ;</i> /*Cela mettra le texte en gras, 16 pixels, Arial.*/*
text-shadow	permet de créer une ombre au texte.	<ul style="list-style-type: none"> • <i>normal</i> : ne change rien • <u>Valeur d'Epaisseur</u> 	<i>5px 5px 2px blue</i> (horizontale, verticale, fondu, couleur)
text-align	Alignement horizontal du texte	<ul style="list-style-type: none"> • <i>left</i> : à gauche (par défaut) • <i>center</i> : centré • <i>right</i> : à droite • <i>justify</i> : texte justifié (prend toute la largeur de la page) 	
text-indent	Indentation (retrait) de la première ligne de l'élément	<ul style="list-style-type: none"> • <i>pourcentage</i> : pourcentage • <u>Valeur d'épaisseur</u> 	<i>25px</i>
letter-spacing	Espacement des lettres entre elles.	<ul style="list-style-type: none"> • <i>normal</i> : ne change rien • <u>Valeur d'épaisseur</u> 	
word-spacing	Espacement des mots (en plus de la	<ul style="list-style-type: none"> • <i>normal</i> : ne change rien • <u>Valeur d'épaisseur</u> 	

	taille habituelle de l'espace).		
direction	permet de choisir le sens du texte.	<ul style="list-style-type: none"> • <i>ltr</i> : de gauche à droite, • <i>rtl</i> : de droite à gauche 	

3.4.2. Alignement

Commandes	Description	Choix des valeurs	Exemples
white-space	Césure, indique comment les espaces sont gérés	<ul style="list-style-type: none"> • <i>normal</i> : le passage à la ligne est automatique (par défaut) • <i>nowrap</i> : pas de passage à la ligne automatique, à moins qu'une balise xHTML comme <code>
</code> ne soit présente. • <i>pre</i> : le passage à la ligne se fait tel que le texte a été saisi dans le code source. 	
word-wrap	Césure forcée	break-word, normal	
vertical-align	Alignement vertical (cellules de tableau ou éléments inline-block uniquement) dans un bloc	<p>A utiliser dans des cellules de tableau.</p> <ul style="list-style-type: none"> • <i>baseline</i> : collé au bas de l'élément • <i>underline</i> : le milieu de l'élément est aligné avec le milieu du bloc dans lequel il se trouve. • <i>sub</i> : en indice • <i>super</i> : en exposant • <i>middle</i> : au milieu • <i>text-top</i> : le sommet de l'élément est aligné avec le sommet de l'élément dans lequel il se trouve • <i>text-bottom</i> : le bas de l'élément est aligné avec le bas de l'élément dans lequel il se trouve • <i>top</i> : le sommet de l'élément est aligné avec le sommet de l'élément le plus haut du bloc dans lequel il se trouve • <i>bottom</i> : le bas de l'élément est aligné avec le bas de l'élément le plus bas du bloc dans lequel il se trouve 	
line-height	Hauteur de la ligne	<ul style="list-style-type: none"> • <i>normal</i> : ne change rien • nombre réel • <i>pourcentage</i> : pourcentage relatif à la taille de font-size • <u>Valeur d'Epaisseur</u> 	18px, 120%, normal...

3.4.3. Couleur et fond

<https://htmlcolorcodes.com/fr/>

<https://www.rapidtables.com/web/color/html-color-codes.html>

Commandes	Description	Choix des valeurs	Exemples
color	Couleur du texte	<ul style="list-style-type: none"> • <i>transparent</i> : (aucune couleur) • <u>Valeur de couleur</u> 	Ex : blue, rgb (128,255,0), #CC48A1 (le plus universel)
background	Cumule les propriétés de l'arrière plan	Indique une ou plusieurs valeurs issues des propriétés background-image, background-repeat, background-attachment, background-position. L'ordre des valeurs n'a pas d'importance et il n'est pas nécessaire de mettre toutes les valeurs de ces propriétés (au moins une suffit)	<i>background:url ("images/fond.png") no-repeat fixed top right;</i> /* Le fond fond.png reste affiché en haut à droite de l'écran et n'est pas répété. */
background-color	Couleur de fond	Même fonctionnement que color. Cela définit cette fois la couleur de fond du texte	
background-image	Image de fond Indique l'url de l'image (notation absolue ou relative)	background-image: url("images/fond.png"); /* Notation relative */ background-image: url("http://www.monsite.com/images/fond.png"); /* Notation absolue */	
background-attachment	Fond fixe ou mobile quand on déplace la barre d'ascenseur	<ul style="list-style-type: none"> • <i>fixed</i> : le fond reste fixe quand on descend plus bas sur la page • <i>scroll</i> : le fond défile avec le texte (par défaut) 	<i>background-attach : fixed ;</i>
background-repeat	Répétition de l'image d'arrière-plan	<ul style="list-style-type: none"> • <i>repeat</i> : le fond se répète (par défaut) • <i>repeat-x</i> : le fond ne se répète que sur une ligne, horizontalement • <i>repeat-y</i> : le fond ne se répète que sur une colonne, verticalement • <i>no-repeat</i> : le fond ne se répète pas, il n'est affiché qu'une fois 	
background-position	Définit la position de la première image à partir de laquelle sera définie la répétition.	2 façons de faire : <ul style="list-style-type: none"> • En notant une distance en px ou %, par rapport au coin en haut à gauche. • En utilisant des valeurs prédéfinies, <ul style="list-style-type: none"> ○ une pour la verticale : <ul style="list-style-type: none"> <i>top</i> : en haut, verticalement <i>center</i> : au milieu, verticalement <i>bottom</i> : en bas, verticalement ○ une pour l'horizontale : <ul style="list-style-type: none"> <i>left</i> : à gauche, horizontalement <i>center</i> : au centre, horizontalement <i>right</i> : à droite, horizontalement 	Ex1 : background-position:50px 200px; /* 50 px à droite, 200px en bas */ Ex2 : background-position : bottom right; /* en bas à droite */ (x y), top, center, bottom, left, right
opacity	Transparence	0.5	

3.4.4. Dimension des boîtes

Commandes	Description	Choix des valeurs	Exemples
width	Largeur	<i>auto</i> : automatique. <u>Valeur d'épaisseur</u>	150px, 80%...
height	Hauteur	<i>auto</i> : automatique. <u>Valeur d'épaisseur</u>	
min-width	Largeur minimale	dito 150px, 80%...	
max-width	Largeur maximale	dito 150px, 80%...	
min-height	Hauteur minimale	dito 150px, 80%...	
max-height	Hauteur maximale	dito 150px, 80%...	

3.4.5. Marges extérieures et intérieures (padding)

Commandes	Description	Choix des valeurs	Exemples
margin	Cumule les propriétés de marge. Combine: <ul style="list-style-type: none"> margin-top, margin-left, margin-right, margin-bottom 	Indiquez de 1 à 4 valeurs à la suite. Selon le nombre de valeurs mises, la signification change : <ul style="list-style-type: none"> 1 valeur : ce sera la marge pour le haut, le bas, la gauche et la droite 2 valeurs : la première correspond à la marge pour le haut et le bas, la seconde pour la gauche et la droite 3 valeurs : la première correspond à la marge du haut, la seconde aux marges à gauche et à droite, la troisième à la marge du bas 4 valeurs : respectivement la marge du haut, de la droite, du bas, de la gauche. 	Exemple : <i>margin:20px 5px; /* 20px de marge en haut et en bas, 5px à gauche et à droite */</i> Ex : <i>23px 5px 23px 5px</i> (haut, droite, bas, gauche)
margin-top	Marge en haut	Indiquer une valeur comme 20px, 1.5em...	Ex : <i>23px</i>
margin-left	Marge à gauche		
margin-right	Marge à droite		
margin-bottom	Marge en bas		
margin-width_* margin-bottom-width margin-left-width margin-right-width margin-top-width	Dimension de la marge	<ul style="list-style-type: none"> <i>none</i> : aucune marge <i>Pourcentage</i> de la fenêtre ou du cadre <i>Valeur d'épaisseur</i> 	
padding	Cumule les propriétés des marges intérieures	Combine: <ul style="list-style-type: none"> padding-top, padding-right, 	Ex : <i>23px 5px 23px 5px</i>

		<ul style="list-style-type: none"> padding-bottom padding-left. 	(haut, droite, bas, gauche)
padding-left	Marge intérieure à gauche	Indiquer une valeur comme 20px, 1.5em...	Ex : 23px
padding-right	Marge intérieure à droite		
padding-bottom	Marge intérieure en bas		
padding-top	Marge intérieure en haut		
padding-width * padding-bottom-width padding-left-width padding-right-width padding-top-width	Dimension de la marge intérieure : c'est-à-dire l'espacement entre le contenu d'un élément et ses bordures. Il affecte soit l'ensemble des cotés d'un élément, soit un coté de celui-ci (top: supérieure, bottom: inférieure, left: gauche, right: droite).	<ul style="list-style-type: none"> none : aucune marge Pourcentage de la fenêtre ou du cadre Valeur d'épaisseur 	

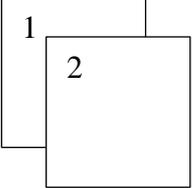
3.4.6. Bordures

Commandes	Description	Choix des valeurs	Exemples
border	Cumule les propriétés de bordure.	Combine : border-width, border-color, border-style. Existe aussi en version : <ul style="list-style-type: none"> border-top, border-right, border-bottom, border-left. 	Border : medium (fin en épaisseur) solid (uni) #000000 (noir)
border-width	Épaisseur de bordure	<ul style="list-style-type: none"> thin : fine medium : moyenne thick : épaisse Valeur d'épaisseur 	
border-color	Couleur de bordure	<ul style="list-style-type: none"> transparent : (aucune couleur) Valeur de couleur 	ex : #CF1A20...
border-style	Style de bordure	<ul style="list-style-type: none"> none : pas de bordure (par défaut) dotted : en pointillé dashed : ligne en tirets solid : unie 	ex : 3px solid black

		<ul style="list-style-type: none"> • double : ligne double (nécessite une taille de bordure de 3px minimum) • groove : en relief • inset : effet 3D « enfoncé » • outset : effet 3D « en relief » • ridge : autre effet 3D 	
border-radius	Bordure arrondie	5px pour les 4 coins, ou différent pour chaque coin.	
border-left	Bordure à gauche	Indiquer la couleur, l'épaisseur et le type de bordure pour la bordure gauche. L'ordre n'a pas d'importance.	<i>border-left : 2px inset blue ; /* Bordure bleue de 2px avec effet 3D « enfoncé » */</i>
border-collapse		Bordures collées	<i>Border-collapse = collapse</i>
box-shadow	Ombre de boîte	6px 6px 0px black (horizontale, verticale, fondu, couleur)	

3.4.7. Affichage et positionnement

Commandes	Description	Choix des valeurs	Exemples
display	Affichage : indique comment doit être affiché l'élément.	None : l'élément ne sera pas affiché (par défaut) block : l'élément devient de type « block » (bloc, comme <p>) inline : l'élément devient de type « inline » (en ligne, comme) list-item : l'élément devient de type « élément de liste à puce » (comme) table : les éléments se placent en colonne	<i>.display{ display:none;} <div class="display" : ceci est invisible </div></i>
visibility	Visibilité de l'élément	hidden : masqué visible : visible (par défaut) display : none, fait complètement disparaître l'élément, tandis que visibility:hidden, masque l'élément, qui continue quand même à prendre de la place sur l'écran.	<i>.visibility{ visibility:hidden;} <div class="visibility" : ceci est invisible </div></i>
clip	Affichage d'une partie de l'élément	rect (0px, 60px, 30px, 0px) rect (haut, droite, bas, gauche)	
overflow	(débordement) Comportement en cas de dépassement !	visible : tout l'élément sera affiché (par défaut). hidden : l'élément sera coupé s'il dépasse les limites définies par height et width. On ne pourra pas voir la partie du texte coupée. scroll : tout comme hidden, l'élément sera coupé s'il dépasse les limites. Mais le	

		navigateur ajoutera des barres de défilement pour qu'on puisse voir la suite du texte. auto : c'est le navigateur qui décide d'ajouter des barres de défilement ou pas en fonction des cas. Bien souvent, utiliser cette valeur revient à utiliser la valeur "scroll".	
float	Définit le positionnement de l'élément flottant	left : à gauche. right : à droite. none : non flottant (par défaut).	
clear	Arrêt de l'élément flottant	left : à gauche. right : à droite. both : gauche ou à droite	
position	Positionnement	static : position normale par défaut relative : position relative par rapport à la position "normale" de l'élément fixed : position fixe (fonctionne comme la position absolue). L'élément reste à sa position même quand on descend plus bas dans la page. absolute : position absolue par rapport au coin en haut à gauche sticky : position hybride entre relative et fixe : l'élément est en positionnement relatif jusqu'à un seuil puis devient fixe au delà.	Position: relative ; top :40px ; left :40ps ;
top	Position par rapport au haut	Valeur en px, %, em... A utiliser pour un positionnement absolu, fixe ou relatif.	Ex : 20px
bottom	Position par rapport au bas		
left	Position par rapport à la gauche		
right	Position par rapport à la droite		
z-index	Ordre d'affichage en cas de superposition. La plus grande valeur est affichée par-dessus les autres.	Par exemple en cas de positionnement absolu, si 2 éléments se chevauchent, z-index permet d'indiquer quel élément doit être affiché au-dessus de l'autre. Indiquez un nombre. Plus ce nombre est élevé, plus l'élément sera affiché en avant. Exemple, si 2 éléments sont positionnés en absolus avec un z-index : 1 pour l'un et z-index : 2 pour l'autre, c'est celui qui a un z-index de 2 qui sera affiché par-dessus.	
list-style-type	Précise le type de puce ou de numérotation à utiliser pour les listes.	<ul style="list-style-type: none"> • Pour les listes non ordonnées () : <ul style="list-style-type: none"> o disc : un disque plein (par défaut). o circle : un cercle. o square : un carré. 	

		<ul style="list-style-type: none"> o <i>none</i> : aucune puce ou numérotation ne sera utilisée. • Pour les listes ordonnées () : <ul style="list-style-type: none"> o <i>decimal</i> : des nombres 1, 2, 3, 4, 5... (par défaut) o <i>decimal-leading-zero</i> : des nombres commençant par zéro (01, 02, 03, 04, 05...). Ne fonctionne pas sur Internet Explorer o <i>upper-roman</i> : numérotation romaine, en majuscules (I, II, III, IV, V...) o <i>lower-roman</i> : numérotation romaine, en minuscules (i, ii, iii, iv, v...) o <i>upper-alpha</i> : numérotation alphabétique, en majuscules (A, B, C, D, E.) o <i>lower-alpha</i> : numérotation alphabétique, en minuscules (a, b, c, d, e..) o <i>lower-greek</i> : numérotation grecque. Ne fonctionne pas sur Internet Explorer 	
list-style-position	Précise où se trouve la puce/numérotation par rapport au texte des éléments de liste.	<ul style="list-style-type: none"> • <i>outside</i> : puce à l'extérieur, et texte en retrait. • <i>inside</i> : puce intégrée au texte. Le tout est en retrait. 	
list-style-image	Définit la puce personnalisée comme une image dont il faut préciser le chemin d'accès.	Indiquer l'url de l'image qui servira de puce. <i>URL</i> : url de l'image <i>none</i> : aucune image	<i>list-style-image: url("images/puce.png");</i>
list-style	Cumule les propriétés de liste.	Combine : list-style-type, list-style-position, list-style-image. Il n'est pas nécessaire de mettre toutes les valeurs et l'ordre n'a pas d'importance.	<i>list-style: inside square ;</i>
a		<ul style="list-style-type: none"> a:link : lien non-visité a:visited : lien visité a:hover : lien survolé a:active : lien activé 	
scrollbar	Barre de défilement	<ul style="list-style-type: none"> • <i>scrollbar-3dlight-color</i> : permet de modifier la couleur pour les effets de relief. Valeur possible : une couleur. • <i>scrollbar-arrow-color</i> : permet de modifier la couleur pour les pointeurs de défilement. Valeur possible : une couleur. • <i>scrollbar-base-color</i> : permet de modifier la couleur de la surface de la barre de défilement. Valeur possible : une couleur. 	

		<ul style="list-style-type: none"> • <i>scrollbar-darkshadow-color</i> : permet de modifier la couleur pour les ombres. Valeur possible : une couleur. • <i>scrollbar-face-color</i> : permet de modifier la couleur de la surface de la barre de défilement. Valeur possible : une couleur. • <i>scrollbar-highlight-color</i> : permet de modifier la couleur pour le bord haut et le bord gauche. Valeur possible : une couleur. • <i>scrollbar-shadow-color</i> : permet de modifier la couleur pour le bord droit et le bord du bas. Valeur possible : une couleur. • <i>scrollbar-track-color</i> : permet de modifier la couleur pour la barre de défilement non cachée par le pointeur de défilement. Valeur possible : une couleur. 	
--	--	---	--

3.4.8. Valeurs utilisées

Valeurs de couleurs

Les valeurs de couleur peuvent s'écrire de différentes manières :

Nom de couleur parmi les suivants : aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.

Les autres couleurs ne fonctionnent pas forcément...

Couleur hexidécimale à 3 chiffres : s'écrit #xyz où x est le rouge, y le vert, et z le bleu. x y et z vont de 0 à 9 et de a à f.

Couleur hexidécimale à 6 chiffres : s'écrit #xyyz où xx est le rouge, yy le vert, et zz le bleu. x, y et z vont de 0 à 9 et de a à f.

Entiers : s'écrit *rgb(x,y,z)*, ou x est le rouge, y le vert, et z le bleu. x, y et z sont des entiers qui compris entre 0 et 255.

Pourcentage : s'écrit *rgb(x%,y%,z%)*, ou x est le rouge, y le vert, et z le bleu. x, y et z sont des reels compris entre 0.0 et 100.0 .

	black = "#000000"		green = "#008000"
	silver = "#C0C0C0"		lime = "#00FF00"
	gray = "#808080"		olive = "#808000"
	white = "#FFFFFF"		yellow = "#FFFF00"
	maroon = "#800000"		navy = "#000080"
	red = "#FF0000"		blue = "#0000FF"
	purple = "#800080"		teal = "#008080"
	fuchsia = "#FF00FF"		aqua = "#00FFFF"

Valeurs des dimensions

Les valeurs d'épaisseur s'écrivent avec un nombre suivit d'un type de valeur parmi les suivants :

em : taille de la police.

px : pixels

pt : points (1/72e de pouce)

ex : valeur x-height de la police.

in : inch (pouces=2.52cm)

Sachant que les unités valent : 1em =12pt =16px = 100%

3.5. Générateur de feuille de style (CSS)

Formulaire automatique proposé par ce site : <http://css.mammothland.net/genecss.php>

4. JAVASCRIPT

Principales sources d'information :

- <https://www.w3schools.com/js/default.asp>
- https://www.w3schools.com/js/js_examples.asp
- <https://www.geeksforgeeks.org/introduction-to-javascript/>

Autres formations très complètes :

- <https://openclassrooms.com/fr/search?page=1&query=javascript&language=fr>
- <https://yard.onl/sitelycee/cours/js/ Js.html?Lirelesfichiers.html>

Les balises HTML créent des objets, JavaScript permet de les manipuler en tant que gestionnaire d'évènements.

JavaScript est :

- un **langage de programmation** (écriture de code source = ensemble d'instructions lues par l'ordinateur qu'il exécute)
- un ensemble de **scripts** (le code source est alors interprété voire compilé)
- **orienté objet** (les objets de base étant : images, dates, chaînes de caractères...)

Il est principalement employé dans les **pages web** interactives (interactivité des animations et utilisation des formulaires) ; c'est donc à ce titre une partie essentielle des **applications Web**.

Ce langage se caractérise par 2 concepts principaux : sa syntaxe et son DOM (Document Object Model). Ce modèle d'objet de document est une interface de programmation représentant le HTML d'une page Web et permettant d'accéder aux éléments de cette page et de pouvoir les modifier grâce au langage JavaScript.

Le point de départ du DOM est le **document** qui représente la page et contient les **fonctions**, pour ; accéder aux **éléments** que l'on souhaite modifier en faisant appel :

- à son **id** (identité unique) et on utilise pour cela la fonction : `getElementById (<id>)`
- à la **classe**, on utilise alors la fonction : `getElementByClassName(<class>)`
- par son **nom de balise** avec `getElementsByTagName(<name>)`
- par l'utilisation des **sélecteurs CSS** avec `querySelector(<selector>)`

Exemple des principales différences :

HTML	JavaScript	Résultat
<pre><p id="para">Some text here</p> <button onclick="changeColor('blue');">blue</button> <button onclick="changeColor('red');">red</button></pre>	<pre>function changeColor(newColor) { var elem = document.getElementById('para'); elem.style.color = newColor; }</pre>	<p>Some text here</p> <p>blue red</p>

Une expression JavaScript est en général composée d'opérateurs (addition, soustraction...) et d'opérandes (constantes et variables).

4.1. La balise <script>

En HTML le code JavaScript est inséré entre cette balise qui elle-même se situe entre <body> et </body> ou <head> et </head>.

```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

Mais les scripts peuvent aussi être placés dans des fichiers externes par exemple myScript.js :

```
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

avec :

```
<script src="myScript1.js"></script>
<script src="myScript2.js"></script>
```

ou

```
<script src="https://www.w3schools.com/js/myScript1.js"></script>
```

4.2. L'attribut universel « id »

Cet attribut définit un identifiant (chaîne de caractères) pour l'ensemble du document.

Code	Résultat
<pre>.output {font: 1rem;} #exciting{box: 2px;} <p>Paragraphe</p> <p id=exciting> le même dans une boîte </p></pre>	<p>Paragraphe</p> <div style="border: 1px solid black; padding: 2px;">le même dans la boîte</div>

4.2.1. L'élément « id »

La propriété de cet élément représente l'identifiant reflétant l'id global de l'attribut. Cet identifiant est utilisé pour extraire l'élément en utilisant *getElementById*.

Syntaxe :

```
var isStr = elt.id ;
Elt.id=idStr ;
```

4.2.2. Le sélecteur « id »

Le sélecteur d'identifiant id permet de cibler un élément grâce à la valeur de son attribut.

Code	Résultat
<pre><!DOCTYPE html> <html> <body> <h2>Que fait JavaScript?</h2> <p id="demo">JavaScript peut changer le contenu de HTML</p></pre>	<p>JavaScript peut changer le contenu de HTML</p> <p>Hello JavaScript!</p> <div style="border: 1px solid black; padding: 2px;">Clique moi!</div>

<pre><button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!'">Clique moi!</button> </body> </html></pre>	
<pre>#identifie {background-color: white;} <spam id="identifie">voici un spam</spam> <spam> et un autre (sans identifiant)</spam></pre>	<p>Voici un spam et un autre (sans identifiant)</p>

4.3. Document

L'interface « Document » correspond à n'importe quelle page Web comprenant notamment les éléments tels que <body> ou <table>.

La syntaxe est : `var document = new document ()`

De nombreuses propriétés existent :

- `Document.characterSet` : jeux de caractères utilisés dans le document,
- `Document.doctype` : renvoi le DTD (Document Type Definition) du document courant,
- `Document.body` : body envoie l'élément <body> du document en cours,
- `Document.image` : renvoie une liste des images du document cournyt,
- `Document.width` : renvoie la largeur du document, ...

Le terme « `getElementById()` » renvoi du « Document » un objet « Element » avec la propriété de « id ».

La syntaxe est : `var element = document.getElementById(id) ;`

4.4. Element

Element est la classe de base la plus générique dont bénéficient tous les objets qui représentent les éléments d'un Document. Element hérite des propriétés de son interface parente : [Node](#) (et donc indirectement des propriétés du parent de celle-ci : [EventTarget](#)).

De nombreuses propriétés existent :

- `Element.childElementCount` : renvoie le nombre d'éléments enfants de cet élément,
- `Element.classList` : renvoie un objet DOMTokenList contenant ma liste des attributs de classe,
- `Element.className` : une chaîne de caractère DOMString qui représente la classe de l'élément,
- `Element.clientWidth` : renvoie un nombre représentant la largeur intérieur de l'élément,
- `Element.id` : une chaîne de caractères DOMString qui représente l'identifiant de l'élément,
- `Element.scrollLeft` : qui représente le décalage à gauche de l'élément lié au défilement,
- `Element.ariaColCount` : indique le nombre de colonnes d'un tableau, d'une grille ou d'une grille arborescente...

La syntaxe est :

```
var ariaColCount = element.ariaColCount;
element.ariaColCount = ariaColCount
```

Code		Résultat
<pre><!DOCTYPE html> <html> <body> <h2>JavaScript HTML Events</h2></pre>	<pre><!DOCTYPE html> <html> <body> <h2>JavaScript HTML Events</h2></pre>	<p>JavaScript HTML Events</p>

<pre><button onclick="document.getElementById('demo'). innerHTML=Date()">The time is?</button> <p id="demo"></p> </body> </html></pre>	<pre><button onclick="displayDate()">The time is?</button> <script> function displayDate() { document.getElementById("demo").innerH TML = Date(); } </script> <p id="demo"></p> </body> </html></pre>	<p>The time is?</p> <p>Sat Feb 05 2022 19:06:31</p>
<pre><!DOCTYPE html> <html> <body> <h2>JavaScript HTML Events</h2> <button onclick="this.innerHTML=Date()">The time is?</button> </body> </html></pre>		

4.5. Affichage des données de sortie

JavaScript peut afficher les données soit en écrivant dans :

- un élément HTML avec *innerHTML*
- la sortie HTML à l'aide de *document.write ()*
- une boîte d'alerte en utilisant *window.alert()*
- la console du navigateur en utilisant *console.log()*.

Ci après les différentes possibilités :

Code			Résultat
<pre><html> <body></pre>			11
<pre><p id="demo"></p> <script> document.getElementById("demo") .innerHTML = 5+6 ; </script></pre>	<pre><script> document.write(5+6) ; </script></pre>	<pre><script> Window.alert(5+6) ; </script></pre>	
<pre></html> </body></pre>			

Pour obtenir le résultat `<button type="button" onclick="document.write(5+6)">total</button>`

Pour imprimer la page `<button on click="window.print()">imprimer cette page</button>`

4.6. Les variables

Il existe 4 façons de déclarer une variable en JavaScript : *var*, *let*, *const* et *rien*

Exemple d'instructions équivalentes à 2 variables sont :

Code		Résultat
<pre><p id="demo"></p> <script></pre>		Le total est 11.
<pre>var x = 5; var y = 6; document.getElementById("demo").innerHTML ="Le total est" +x+y ;</pre>	<pre>const valeur1 = 5; const valeur2 = 6; let total = valeur1+valeur 2; document.getElementById("demo").innerHTML ="Le total est" +total ;</pre>	
<pre></script></pre>		

Instructions équivalentes à 3 variables sont :

Code			Résultat
<pre><p id="demo"></p> <script></pre>			La valeur de z est 11.
<pre>var x = 5; var y = 6; var z = x + y;</pre>	<pre>let x = 5; let y = 6; let z = x + y</pre>	<pre>var x, y, z; x = 5; y = 6; z = x + y;</pre>	
<pre>document.getElementById("demo").innerHTML = "La valeur de z est" + z + "."; </script></pre>			

4.7. Blocs de code

Instructions regroupées en blocs de code entre accolade :

Code	Résultat
<pre><button type="button" onclick="myFunction()">Click Me!</button> <p id="demo1"></p> <p id="demo2"></p> <script> function myFunction() { document.getElementById("demo1").innerHTML = "Hello Dolly!"; document.getElementById("demo2").innerHTML = "How are you?"; } </script></pre>	<div style="border: 1px solid black; display: inline-block; padding: 2px;">Click Me!</div> Hello Dolly! How are you?

4.8. Liste des principaux mots-clés :

Mot-clé	La description
var	Déclare une variable
let	Déclare une variable de bloc
const	Déclare une constante de bloc
continue	Saute d'une boucle et commence par le haut
if ... else	Marque un bloc d'instructions à exécuter, en fonction d'une condition
do ... while	Exécute un bloc d'instructions et répète le bloc, tant qu'une condition est vraie
for	Marque un bloc d'instructions à exécuter, tant qu'une condition est vraie
function	Déclare une fonction
try ... catch	Implémente la gestion des erreurs dans un bloc d'instructions
debugger	Arrête l'exécution de JavaScript et appelle (si disponible) la fonction de débogage
break	Termine un commutateur ou une boucle
return	Quitte une fonction
switch	Marque un bloc d'instructions à exécuter, en fonction des différents cas

4.9. Opérateurs arithmétiques

Signes	Description
+	Addition
-	Soustraction
*	Multiplication
**	Exponentiation (ES2016)

/	Division
%	Modulo (Remainder) (ex5%2=1 correspondant au reste de la division de 5 par 2)
++	Incrément
--	Décrément

Code		Résultat
<pre><script> function myFunction(p1, p2) { return p1 * p2; } document.getElementById ("demo").innerHTML = myFunction(4, 3); </script></pre>	<pre><script> var x = myFunction(4, 3); document.getElementById("demo") .innerHTML = x; function myFunction(a, b) { return a * b; } </script></pre>	<pre><script> Var résultat = 4*3 ; Arlert(result) ; </script></pre> <p>12</p>
<pre><script> document.getElementById("demo").innerHTML = "The temperature is " + toCelsius(77) + " Celsius"; function toCelsius(fahrenheit) { return (5/9) * (fahrenheit-32); } </script></pre>		<p>The temperature is 25 Celsius</p>

4.10. Affectations

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
<<=	x <<= y	x = x << y
>>=	x >>= y	x = x >> y
>>>=	x >>>= y	x = x >>> y
&=	x &= y	x = x & y
^=	x ^= y	x = x ^ y
=	x = y	x = x y
**=	x **= y	x = x ** y

Code	Résultat
<pre><script> var x = 10; x+ = 5; document.getElementById("demo").innerHTML =x; </script></pre>	<p>15</p>
<pre><script> let x = 5 + 4; let y = "6" + 5 + 4; let z = 6 + 5 + "4"; document.getElementById("demo").innerHTML = x + "
" + y + "
" + z; </script></pre>	<p>9 654 114</p>

4.11. Objets

Un objet « Élément » correspond à l'Id spécifié

Code		Résultat
<pre><script> const person = { firstName: "John", lastName: "Doe", age: 50, eyeColor: "blue" };</pre>		John is 50 years old.
<pre>document.getElementById("demo").innerHTML = person.firstName + " is " + person.age + " years old."; </script></pre>		
<pre><script> const person = { firstName: "John", lastName : "Doe", id: 5566 }; document.getElementById("demo").innerHTML = person["firstName"] + " " + person["lastName"]; </script></pre>	<pre><script> const person = { firstName: "John", lastName: "Doe", id: 5566, fullName: function() { return this.firstName + " " + this.lastName; } }; document.getElementById("demo").innerHTML = person.fullName(); </script></pre>	John Doe

4.12. Evènements

L'interface **Event** représente un évènement qui se produit dans le DOM ; cet évènement peut être déclenché par une action extérieure ou à partir d'un programme en appelant par exemple la méthode [HTML`Element`.click\(\)](#) sur un élément ou en définissant l'évènement avant de l'envoyer sur une cible avec la méthode [EventTarget.dispatchEvent\(\)](#).

L'interface « **EventTarget** » est implémentée par des objets qui peuvent recevoir des évènements. Les 3 méthodes [Element](#), et ses enfants, ainsi que [Document](#) et [Window](#), sont les cibles d'évènement les plus courantes,

Plusieurs méthodes existent :

EventTarget.addEventListener() : Enregistre un gestionnaire d'évènements d'un type d'évènement spécifique sur le EventTarget

EventTarget.removeEventListener() : Supprime le récepteur d'évènement du EventTarget

EventTarget.dispatchEvent() : Envoie u évènement au EvetTarget

La propriété « **currentTarget** », accessible en lecture seule et associée à l'interface [Event](#), identifie la cible courante pour l'évènement lorsque celui-ci traverse le DOM. Elle fait référence à l'élément sur lequel le gestionnaire d'évènement a été attaché. *Event.target* peut être un descendant de *event.currentTarget*.

Syntaxe : `var currentEventTarget = event.currentTarget ;`

4.13. Les fonctions

Une fonction peut être appelée lorsqu'un évènement se produit, elle se compose de **son nom** suivi d'un **argument** (ou paramètre constitué de valeurs) mis entre parenthèses.

Code		Résultat
<pre><html> <body> <h2>Demo JavaScript in Body</h2> <p id="demo">A Paragraph.</p> <button type="button" onclick="myFunction()">Try it</button> <script> function myFunction() { document.getElementById("demo").innerHTML = "Paragraph changed."; } </script> </body> </html></pre>		<p>Demo JavaScript in Body ou in Head</p> <p>A Paragraphe ou Paragraph changed.</p> <p>Try</p>
<pre><script> function myFunction() { document.getElementById("demo").innerHTML = "Paragraph changed."; } </script> </head> <body> <h2>Demo JavaScript in Head</h2> <p id="demo">A Paragraph.</p> <button type="button" onclick="myFunction()">Try it</button> </body></pre>		

4.14. Les tableaux

Code			Résultat
<pre><p id="demo"></p> <script></pre>			<p>Saab, Volvo, BMW</p>
<pre>const cars = ["Saab", "Volvo", "BMW"];</pre>	<pre>const cars = new Array("Saab", "Volvo", "BMW");</pre>	<pre>const cars = []; cars[0] = "Saab"; cars[1] = "Volvo"; cars[2] = "BMW";</pre>	
<pre>document.getElementById("demo").innerHTML = cars ; </script></pre>			

Code		Résultat
<pre><script> const person = ["John", "Doe", 46]; document.getElementById("demo").innerHTML = person[0]; </script></pre>	<pre><script> const person = {firstName:"John", lastName:"Doe", age:46}; document.getElementById("demo").innerHTML = person.firstName; </script></pre>	<p>John</p>
<pre><script> const fruits = ["Banana", "Orange", "Apple", "Mango"]; document.getElementById("demo").innerHTML = fruits.length; </script></pre>	<pre><script> const fruits = ["Banana", "Orange", "Apple", "Mango"]; document.getElementById("demo").innerHTML = fruits[0]; </script></pre>	<p>Banana</p>
<pre><script> const fruits = ["Banana", "Orange", "Apple", "Mango"]; let text = ""; let fLen = fruits.length; for (let i = 0; i < fLen; i++) { text += "" + fruits[i] + ""; } text += ""; document.getElementById("demo").innerHTML = text;</pre>	<pre>let text = ""; fruits.forEach(myFunction); text += ""; function myFunction(value) { text += "" + value + ""; }</pre>	<ul style="list-style-type: none"> Banana Orange Apple Mango

<code></script></code>	
------------------------------	--

4.15. Les conditions

Une condition retourne une valeur booléenne : true ou false.

```
if (confirm('si c'est OK')){ alert('C'est bon') :} else ('c'est pas bon !')}
```

Switch imbrique plusieurs conditions entre elles.

4.16. Les boucles

Ce sont soit

- une incrémentation (on ajoute une unité à un nombre : `var number=0 ; number++ ; alert(number)`,
- une boucle while (tant que : `var number=1 ; while(number<10){ ; number++ : } alert(number) ;`
- une boucle go while (faire tant que : `do{ instruction 1 ; ... instruction3 ; } while (condition) ;`
- une boucle for (`for(initialeistaion, ; condition ; incrémentation) { instruction 1 ; ... instruction3 ; } lucidité`

Code		Résultat
<pre><script> const cars = ["BMW", "Volvo", "Saab", "Ford"]; let i, len, text; for (i = 0, len = cars.length, text = ""; i < len; i++) { text += cars[i] + "
"; } document.getElementById("demo").innerHTML = text; </script></pre>		<p>BMW Volvo Saab Ford</p>
<pre>let i = 0; let len = cars.length; let text = ""; for (; i < len;) { text += cars[i] + "
"; i++;}</pre>		

Les fichiers avec extension .js sont des documents texte qui contiennent des lignes de codes de JavaScript, ils peuvent être lus par Adobe Dreamweaver ou Text Ediitor, Adobe ColdFusion...

4.17. Résumé sur l'emploi des 3 langages

Dans un répertoire de travail on crée 3 fichiers :

- `index.html` dans lequel il y a :

```
<head><meta charset="utf-8"><title>Le trio</title><link rel="stylesheet" href="style.css"> </head>
<script src="script.js"></script>
```

- `style.css`

```
h1{ text-align : center ;}
```

- `script.js`

```
alert("Le JavaScript existe !)
```

5. JQUERY

jQuery est une **bibliothèque de JavaScript**, cette multiplateforme libre a été créée en 2006 pour simplifier l'écriture des scripts coté client dans le code HTML. Elle comprend et facilite les fonctionnalités suivantes :

- Manipulation des documents HTML / DOM
- Manipulation CSS
- Gestion d'événements
- Effets et animations
- AJAX (=JavaScript et XML asynchrone) c'est l'art d'échanger simplement des données avec le serveur suivant une API (Application Programming Interface = ensemble de protocoles facilitant l'intégration de logiciels d'applications°.
- Utilitaires

Pour utiliser jQuery sur son site Web, soit :

- on télécharge la bibliothèque jQuery à partir de jquery.com et on place le fichier téléchargé dans le même répertoire que les pages à utiliser.

La bibliothèque est alors référencée grâce au script :

```
<head>
<script src="jquery-3.5.1.min.js"></script>
</head>
```

- on inclue à partir d'un CDN (Content Delivery Network)

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
```

5.1. Syntaxe

La syntaxe de base est : **\$ (*sélecteur*). *action* ()**

- Un signe \$ pour définir / accéder à jQuery
- Un (*sélecteur*) pour "interroger (ou rechercher)" des éléments HTML
- Une *action* jQuery () à effectuer sur le ou les éléments

Exemples:

\$(this).hide() - masque l'élément courant.

\$("p").hide() - masque tous les éléments <p>.

\$(".test").hide() - masque tous les éléments avec class = "test".

\$("#test").hide() - masque l'élément avec id = "test".

5.2. Sélecteurs jQuery

Exemples de simplification des sélecteurs avec jQuery :

JavaScript	jQuery
myElement = document.getElementById("id01");	myElements = \$("#id01");

myElement = document.getElementsByTagName("p");	myElements = \$("p");
myElement = document.getElementsByClassName("intro");	myElements = \$(".intro");
myElement = document.querySelectorAll("p.intro");	myElements = \$("p.intro");
myElement.textContent = "Hello France" ;	myElements.text("Hello France");
myText = document.getElementById("02");	myText = \$("#02").text() ;
myElement.innerHTML = "<p>Salut France</p>" ;	myElement.html = "<p>Salut France</p>" ;
Content = myElement.innerHTML ;	Content = myElement.html() ;
myElement.style.display = "none" ;	myElement.hide() ;
myElement.style.display="" ;	myElement.show () ;
Document.getElementById("demo").style	\$("#demo").css("font-size","35px");

Selecteurs	Exemples	Signification
*	\$("*")	All elements
#id	\$("#lastname")	The element with id="lastname"
.class	\$(".intro")	All elements with class="intro"
.class,.class	\$(".intro,.demo")	All elements with the class "intro" or "demo"
element	\$("p")	All <p> elements
el1,el2,el3	\$("h1,div,p")	All <h1>, <div> and <p> elements
:first	\$("p:first")	The first <p> element
:last	\$("p:last")	The last <p> element
:even	\$("tr:even")	All even <tr> elements
:odd	\$("tr:odd")	All odd <tr> elements
:first-child	\$("p:first-child")	All <p> elements that are the first child of their parent
:first-of-type	\$("p:first-of-type")	All <p> elements that are the first <p> element of their parent
:last-child	\$("p:last-child")	All <p> elements that are the last child of their parent
:last-of-type	\$("p:last-of-type")	All <p> elements that are the last <p> element of their parent
:nth-child(n)	\$("p:nth-child(2)")	All <p> elements that are the 2nd child of their parent
:nth-last-child(n)	\$("p:nth-last-child(2)")	All <p> elements that are the 2nd child of their parent, counting from the last child
:nth-of-type(n)	\$("p:nth-of-type(2)")	All <p> elements that are the 2nd <p> element of their parent
:nth-last-of-type(n)	\$("p:nth-last-of-type(2)")	All <p> elements that are the 2nd <p> element of their parent, counting from the last child
:only-child	\$("p:only-child")	All <p> elements that are the only child of their parent
:only-of-type	\$("p:only-of-type")	All <p> elements that are the only child, of its type, of their parent
parent > child	\$("div > p")	All <p> elements that are a direct child of a <div> element
parent descendant	\$("div p")	All <p> elements that are descendants of a <div> element
element + next	\$("div + p")	The <p> element that are next to each <div> elements
element ~ siblings	\$("div ~ p")	All <p> elements that appear after the <div> element
:eq(index)	\$("ul li:eq(3)")	The fourth element in a list (index starts at 0)
:gt(no)	\$("ul li:gt(3)")	List elements with an index greater than 3
:lt(no)	\$("ul li:lt(3)")	List elements with an index less than 3
:not(selector)	\$("input:not(:empty)")	All input elements that are not empty
:header	\$(":header")	All header elements <h1>, <h2> ...
:animated	\$(":animated")	All animated elements
:focus	\$(":focus")	The element that currently has focus
:contains(text)	\$(":contains('Hello'))	All elements which contains the text "Hello"
:has(selector)	\$("div:has(p)")	All <div> elements that have a <p> element
:empty	\$(":empty")	All elements that are empty
:parent	\$(":parent")	All elements that are a parent of another element

<code>:hidden</code>	<code>\$("p:hidden")</code>	All hidden <code><p></code> elements
<code>:visible</code>	<code>\$("table:visible")</code>	All visible tables
<code>:root</code>	<code>\$(":root")</code>	The document's root element
<code>:lang(<i>language</i>)</code>	<code>\$("p:lang(de)")</code>	All <code><p></code> elements with a <code>lang</code> attribute value starting with "de"
<code>[attribute]</code>	<code>\$("[href]")</code>	All elements with a <code>href</code> attribute
<code>[attribute=value]</code>	<code>\$("[href='default.htm']")</code>	All elements with a <code>href</code> attribute value equal to "default.htm"
<code>[attribute!=value]</code>	<code>\$("[href!='default.htm']")</code>	All elements with a <code>href</code> attribute value not equal to "default.htm"
<code>[attribute\$=value]</code>	<code>\$("[href\$='.jpg']")</code>	All elements with a <code>href</code> attribute value ending with ".jpg"
<code>[attribute =value]</code>	<code>\$("[title='Tomorrow']")</code>	All elements with a <code>title</code> attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen
<code>[attribute^=value]</code>	<code>\$("[title^='Tom']")</code>	All elements with a <code>title</code> attribute value starting with "Tom"
<code>[attribute~=value]</code>	<code>\$("[title~='hello']")</code>	All elements with a <code>title</code> attribute value containing the specific word "hello"
<code>[attribute*=value]</code>	<code>\$("[title*='hello']")</code>	All elements with a <code>title</code> attribute value containing the word "hello"
<code>:input</code>	<code>\$(":input")</code>	All input elements
<code>:text</code>	<code>\$(":text")</code>	All input elements with <code>type="text"</code>
<code>:password</code>	<code>\$(":password")</code>	All input elements with <code>type="password"</code>
<code>:radio</code>	<code>\$(":radio")</code>	All input elements with <code>type="radio"</code>
<code>:checkbox</code>	<code>\$(":checkbox")</code>	All input elements with <code>type="checkbox"</code>
<code>:submit</code>	<code>\$(":submit")</code>	All input elements with <code>type="submit"</code>
<code>:reset</code>	<code>\$(":reset")</code>	All input elements with <code>type="reset"</code>
<code>:button</code>	<code>\$(":button")</code>	All input elements with <code>type="button"</code>
<code>:image</code>	<code>\$(":image")</code>	All input elements with <code>type="image"</code>
<code>:file</code>	<code>\$(":file")</code>	All input elements with <code>type="file"</code>
<code>:enabled</code>	<code>\$(":enabled")</code>	All enabled input elements
<code>:disabled</code>	<code>\$(":disabled")</code>	All disabled input elements
<code>:selected</code>	<code>\$(":selected")</code>	All selected input elements
<code>:checked</code>	<code>\$(":checked")</code>	All checked input elements
<code>[attribute]</code>	<code>\$("[href]")</code>	All elements with a <code>href</code> attribute
<code>[attribute=value]</code>	<code>\$("[href='default.htm']")</code>	All elements with a <code>href</code> attribute value equal to "default.htm"
<code>[attribute!=value]</code>	<code>\$("[href!='default.htm']")</code>	All elements with a <code>href</code> attribute value not equal to "default.htm"
<code>[attribute\$=value]</code>	<code>\$("[href\$='.jpg']")</code>	All elements with a <code>href</code> attribute value ending with ".jpg"
<code>[attribute =value]</code>	<code>\$("[title='Tomorrow']")</code>	All elements with a <code>title</code> attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen
<code>[attribute^=value]</code>	<code>\$("[title^='Tom']")</code>	All elements with a <code>title</code> attribute value starting with "Tom"
<code>[attribute~=value]</code>	<code>\$("[title~='hello']")</code>	All elements with a <code>title</code> attribute value containing the specific word "hello"
<code>[attribute*=value]</code>	<code>\$("[title*='hello']")</code>	All elements with a <code>title</code> attribute value containing the word "hello"
<code>:input</code>	<code>\$(":input")</code>	All input elements
<code>:text</code>	<code>\$(":text")</code>	All input elements with <code>type="text"</code>
<code>:password</code>	<code>\$(":password")</code>	All input elements with <code>type="password"</code>
<code>:radio</code>	<code>\$(":radio")</code>	All input elements with <code>type="radio"</code>
<code>:checkbox</code>	<code>\$(":checkbox")</code>	All input elements with <code>type="checkbox"</code>
<code>:submit</code>	<code>\$(":submit")</code>	All input elements with <code>type="submit"</code>
<code>:reset</code>	<code>\$(":reset")</code>	All input elements with <code>type="reset"</code>
<code>:button</code>	<code>\$(":button")</code>	All input elements with <code>type="button"</code>
<code>:image</code>	<code>\$(":image")</code>	All input elements with <code>type="image"</code>
<code>:file</code>	<code>\$(":file")</code>	All input elements with <code>type="file"</code>
<code>:enabled</code>	<code>\$(":enabled")</code>	All enabled input elements
<code>:disabled</code>	<code>\$(":disabled")</code>	All disabled input elements
<code>:selected</code>	<code>\$(":selected")</code>	All selected input elements
<code>:checked</code>	<code>\$(":checked")</code>	All checked input elements

6. XML

XML, (Extensible Markup Language), désigne un langage informatique (ou métalangage) utilisé, dans la conception des sites Web et pour faciliter les **échanges d'informations** sur Internet. Ce langage a pour mission de formaliser (décrire, structurer et stocker) des données textuelles sur internet.

XML permet d'afficher les données car c'est un langage de balisage ; la balise étant un mot clé qui se trouve entre « < » et « > » par exemple : <xml>.

Une fois le format XML défini, on peut utiliser le fichier XSD qui va contenir la définition de la structure puis on va créer un fichier XSLT pour transformer le fichier XML en fichier HTML et en CSV (pour Excel).

La première ligne du fichier XML commence toujours par :

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Le fichier XML peut être créé et lu par les éditeurs : Bloc-notes de Windows ou TextEdit par Mac mais peut aussi être incorporé dans une feuille de calcul Excel. Les données de XML sont conservées et HTML les affiche ; elles sont incluses entre des balises non standardisées comme <note></note> et peuvent être imbriquées les unes dans les autres pour créer des arborescences.

- <https://xml.developpez.com/cours/>
- <https://sebsauvage.net/comprendre/xml/>

7. JSON

JSON (JavaScript Object Notation) est **un format d'échanges de données** (entre un serveur et un client), c'est un format de texte indépendant de tout langage mais utilisant des conventions du langage C (incluant JavaScript et Python), il utilise l'extension : .json. Le fichier est une alternative plus simple que XML

Les développeurs utilisent JSON pour travailler avec AJAX (Asynchronous JavaScript and XML).

Il est construit suivant 2 structures de données :

- Une collection de paires nom/valeur, réalisé par la plupart des langages sous la forme d'un objet, d'un enregistrement, missionnaire, table de hachage, liste à clé ou tableau associatif.
- Une liste ordonnée de valeurs, réalisé par la plupart des langages sous la forme d'un tableau, d'un vecteur, d'une liste ou d'une séquence...

En JSON, ces 2 structures se retrouvent sous les formes suivantes :

- Un **objet** (entouré d'**accolades**) qui comporte les éléments suivants :
 - Les **noms, (ou clés)** constitués de chaînes de caractères entourées de **guillemets**.
 - Les **valeurs** qui sont un type de données JSON valide, pouvant se présenter sous la forme d'un tableau, d'un objet, d'un booléen, d'un nombre ou nul.

JavaScript	JSON
{ name : "John" }	{ "name": "John" }

- Un **tableau** (ARRAYS) par [valeur 1, valeur2]

JavaScript	JSON
[valeur 1, valeur2]	[valeur 1, valeur2]

<code>{prénom : "Tom", Nom : "Dupond", Loisir : ["football", "lecture", "natation"]}</code>	<code>{"prénom" : "Tom", "Nom" : "Dupond", "Loisir" : ["football", "lecture", "natation"]}</code>
---	---

Pour interpréter du JSON et convertir des valeurs en JSON à partir de l'objet JavaScript on utilise : les méthodes (correspondant aux spécifications ECMAScript) suivantes :

- La méthode `stringify()` qui convertit une valeur JavaScript en chaîne JSON.
- La méthode `parse()` qui analyse une chaîne de caractères JSON et construit la valeur JavaScript ou l'objet décrit par cette chaîne.

JSON étant du texte, on peut le stocker dans une base de données, dans stockage client (cookies ou localStorage) dans un fichier texte ou via son propre format de fichier qui utilise l'extension : `.json`.

Une fois le contenu JSON stocké on peut le récupérer et le parser. Pour cela on utilise : les méthodes (correspondant aux spécifications ECMAScript) suivantes :

- La méthode `JSON.stringify()`
- La méthode `JSON.parse()`

Différences entre XML et JSON :

XML	JSON
<pre><employees> <employee><prenom>John</prenom><nom>Doe</nom> </employee> <employee><prenom>Arlette</prenom><nom>Smith</nom> </employee> <employee><prenom>Jean</prenom><nom>Dupont</nom></e mployee></employees></pre>	<pre>{"employees": [{"prenom":"John","nom":"Doe"}, {"prenom":"Arlette","nom":"Smith"}, {"prenom":"Jean","nom":"Dupont"}]}</pre>
Nécessité d'avoir un analyseur XML	Plus rapide et moins verbeux
Orienté document et peut afficher les données car c'est un langage de balisage. Ne supporte pas les tableaux	Orienté données sans les afficher ; prend en charge les tableaux.

8. PHP & MySQL

PHP (Personal Home Page) est un langage (impératif et orienté objet) de programmation libre pour produire des pages Web dynamiques via le serveur HTTP.

Pour que le navigateur envoie une requête au serveur HTTP correspondant, il faut mettre :

```
< ?php echo 'Hello world' ; ?> ou bien :
< ? = 'Hello world' ?>
```

Au départ (en 1994) le langage PHP était une bibliothèque logicielle écrite en langage Perl puis en C ; il a notamment permis de créer des sites comme Facebook et Wikipédia...

Ce langage s'appuie sur le serveur Web : **Apache** qui répond aux requêtes. Les principaux « concurrents » au PHP sont **Java, Python, Ruby, ASP.Net, C++, Django...**

MySQL est une base de données utilisée par PHP. Ses Principaux « concurrents » sont Oracle, Microsoft SQLserver, SQLite, Post...

Pour la création des sites dynamiques, l'ordinateur doit se comporter comme un serveur et a donc besoin du **serveur Web Apache** qui délivre les pages Web (html) aux visiteurs, le PHP qui est un plug-in pour Apache, il le rend capable de traiter les pages web dynamiques en PHP et le logiciel de Base de données MySQL qui organise les données.

8.1. Instructions PHP

L'instruction de base echo permet d'afficher du texte ; la balise permet de mettre le texte en gras, l'ensemble devant être inséré dans des codes HTML

```
< ?php echo «ceci est du <strong>texte</strong> »; ?>
```

Ce code source peut être copié dans l'éditeur de texte, puis enregistré avec l'extension .php comme affichertexte.php dans le dossier : /applications/MAMP/htdocs/tests.

Les commentaires se font par //... pour 1 ligne et /*...*/ pour plusieurs lignes.

Pour connaître le type des éventuelles erreurs, la commande est : *phpinfo()* ;

8.2. Les variables

Les variables sont toujours précédées de \$ et se terminent par ";"

Elles peuvent prendre différentes valeurs :

- **string** = chaîne de caractères (texte).....\$nom_du_visiteur= "Dupond" ;
- **int** = nombre entier.....\$age_du_visiteur= 17 ;
- **float** = nombre décimaux avec virgule
- **bool** = booléen : vrai ou faux
- **null** = vide ou nul
-

Exemples de variables avec concaténation :

Code	Résultat
< ?php \$age_du_visiteur=17 ; echo "Le visiteur a" ; echo \$age_du_visiteur ; echo " ans" ; ?>	Le visiteur a 17 ans
< ?php \$age_du_visiteur=17 ; echo 'le visiteur a' . \$age_du_visiteur . ' ans' ; ?>	
< ?php \$txt ="W3Schools.com" ;echo "J'aime".\$txt."!" ;	J'aime W3Schools.com !

8.3. Les opérations

Toutes les opérations sont possibles y compris le modulo (%)

Code	Résultat
< ?php \$x =5 ;\$y=4 ; echo \$x+\$y ; ?>	9
< ?php \$nombre=10 ; \$resultat=(\$nombre+5)*\$nombre ; ?>	150
< ?php \$nombre=10% 3 ; ?>	1 (car la division ne tombe pas juste)

8.4. Les conditions

Toutes les conditions seront incluses entre 2 accolades {}.

La condition de base est « if...else », les mesures étant :

- == égal ;
- < inférieur ;
- > supérieur ;
- >= supérieur ou égal ;
- <= inférieur ou égal ;
- != différente de.

Autres conditions : **while** pour « tant que » ; **&&** pour Et ; **||** pour Ou et **switch** pour plusieurs conditions successives.

```
< ?php $note=10; switch($note) {
case 0 :echo "c'est nul" ;break ;
case 10 :echo "c'est juste" ;break ;
case 20 :echo "c'est super" ;break ;
default : echo "pas de note " ;} ?>
```

La structure ternaire

```
< ?php $age=24; $majeur = ($age>=18)? true : false ; ?>
```

Si l'âge est >= 18, la valeur true sera affectée à la variable \$majeur

8.5. Les boucles

Permet de faire des itérations **tant que**...le test n'est pas atteint.

Code	Résultat
<pre>< ?php \$nombre_de_ligne=1; while (\$nombre_de_ligne <=10) {echo 'ceci est la ligne n°'. \$nombre_de_ligne.'
' ; \$nombre_de_ligne++ ; } ?></pre>	ceci est la ligne n°1, 2, ..10
<pre>< ?php for(\$nombre_de_ligne=1;\$nombre_de_ligne <=10) \$nombre_de_ligne++ {echo 'ceci est la ligne n°'. \$nombre_de_ligne.'
' } ?></pre>	

8.6. Les tableaux

La variable **array** (tableau) peut prendre plusieurs valeurs :

Code	Résultat
<pre>< ?php \$prenoms= array ('Francois', 'Michel', 'Nicole') ; ?></pre>	Francois Michel Nicole
<pre>< ?php \$prenoms= array ('Francois', 'Michel', 'Nicole') ; for (\$numero<3 ;\$numero++) {echo \$prénom [\$numero]. '
' ;} ?></pre>	
<pre><?php \$coordonnees = array ('prenom' => 'François', 'nom' => 'Dupont', 'adresse' => '3 Rue du Paradis', 'ville' => 'Marseille');?></pre>	Francois Dupond 3 Rue du Paradis Marseille
<pre><?php \$coordonnees = array ('François', 'Dupont', '3 Rue du Paradis', 'Marseille'); foreach(\$prenoms as \$element) {echo \$element.'
' ;} ?></pre>	
<pre><?php \$cars = array (array("L1",22,18),array("L2",15,13),array("L3",5,2),);</pre>	Chapitre 0 <ul style="list-style-type: none"> • L1 • 22

<pre> for (\$row = 0; \$row < 3; \$row++) { echo "<p>Chapitre \$row</p>"; echo ""; for (\$col = 0; \$col < 3; \$col++) { echo "".\$cars[\$row][\$col].""; } echo ""; } ?> </pre>	<ul style="list-style-type: none"> • 18 <p>Chapitre 1</p> <ul style="list-style-type: none"> • L2 • 15 <p>Chapitre 2</p> <ul style="list-style-type: none"> • L3 • 5 • 2
--	--

8.7. Les fonctions

Il existe de très nombreuses fonctions (plus de 1000) qui recherchent et remplacent des mots dans une variable, qui envoient un fichier sur un serveur, qui créent une image...

Une fonction peut retourner une valeur :

```
< ?php $volume=calculCub(4) ; ?> le résultat donne pour valeur 64.
```

Certaines fonctions permettent de traiter des chaînes de caractères comme str_replace qui recherche et remplace un texte (ici b par p) :

```
< ?php $variable=str_replace('b', 'p', 'bim bam boum') ; echo $variable ; ?> le résultat est pim pam poum.
```

D'autres fonctions permettent d'écrire en minuscule (strtolower) ou de récupérer une date ou une heure : \$jour=date('d') ; \$mois=date('m') ; \$annee=date('Y') ; \$heure=date('h') ; \$minute=date('i') ;

Exemple du calcul du volume d'un cône : $\pi.R^2.h.1/3$ à l'aide de l'instruction return qui renvoie le volume :

```
< ?php function VolumeCone($rayon, $hauteur)
{$volume=3.14*$rayon*$rayon*$hauteur *(1/3) ; return $volume ; }
$volume =volumeCone(3,1) ; echo 'le volume d'un cône de rayon 3 et de hauteur 1 est de '$volume ; ?>
```

8.8. Inclusion d'un menu de page

```
< ?php include("menu.php") ?>
```

8.9. Manuel PHP

Il est accessible à l'adresse suivante : <https://www.php.net/manual/fr/funcref.php>

9. Résumé des langages utilisés pour une page Web

Issu de :

<https://www.imedias.pro/cours-en-ligne/programmation/specificites-des-langages-web/combinaisons-imbrications-langages-page-web/>

et de :

https://static.oc-static.com/prod/ebooks/apprenez-a-creer-votre-site-web-avec-html5-et-css3_2016.pdf

9.1. Récapitulatif de base



Code vu par un éditeur de texte

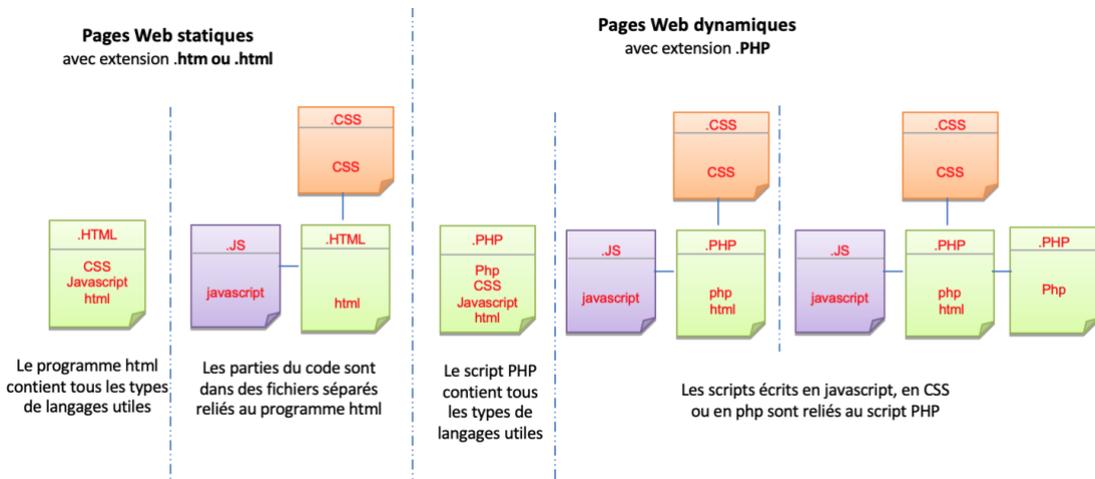


Résultat vu du navigateur avec code Html seul (sans CSS)

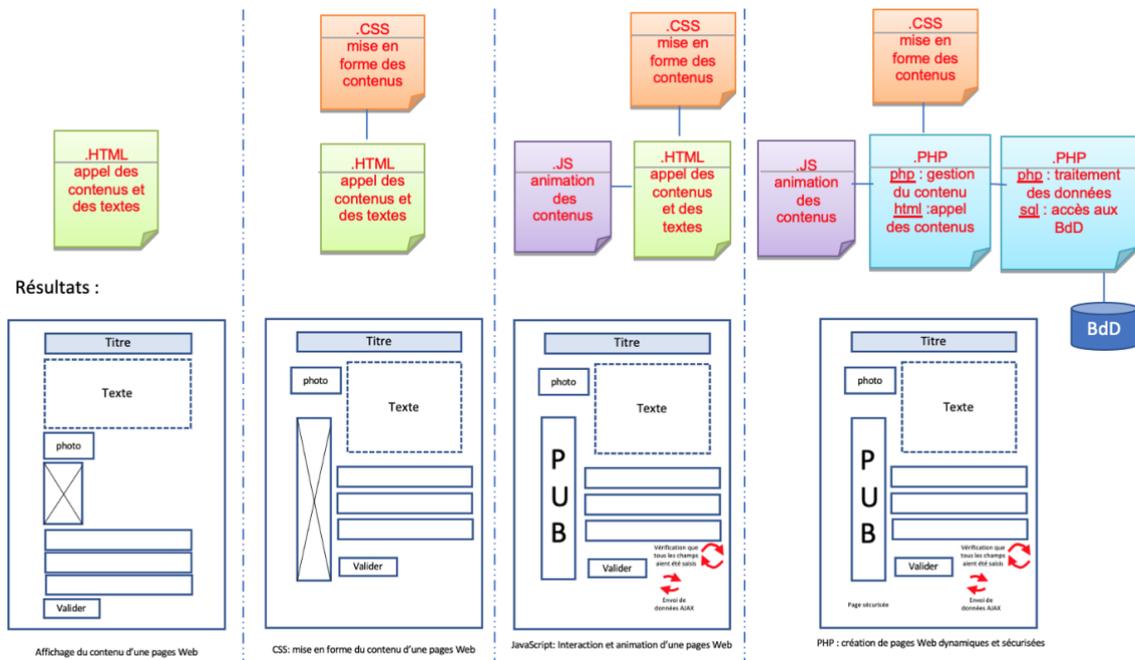


Résultat vu du navigateur avec code Html et CSS

9.2. Suivant les types de pages



9.3. Les différentes méthodes utilisées :



10. Autres langages de programmation

10.1. Langage C

Un des plus vieux langages de programmation, il est essentiellement utilisé pour les systèmes d'exploitation, le développement de logiciels et le hardware.

Il fonctionne sur différentes plateformes et peut être intégré sur n'importe quel microprocesseur.

Tous les autres langages en sont plus ou moins des dérivés.

Pour une programmation « orienté objet », le C++ a été créé. Un autre langage dérivé est le C# qui est puissant et flexible. « Objective C » est un langage pour un usage général et orienté objet qui ajoute une messagerie de style Smalltalk au langage C (utilisé notamment par Apple, OSX et iOS).

10.2. Ruby

Ruby est un langage de programmation d'usage général, orienté objet, dynamique et réflexif. Il est utilisé pour les interfaces utilisateur graphique, les applications web et le développement web.

Il est simple, facile à apprendre mais lent et avec peu de documentation.

10.3. Java

Java est le langage le plus connu utilisé par les développeurs pour créer et faire fonctionner les applications sur ordinateurs, smartphone, consoles, logiciels. Il évolue régulièrement.

10.4. Python

Python est un langage de programmation open source, populaire, interprété côté serveur, simple, puissant et non compilé ; principalement utilisé pour le développement de jeux vidéo et pour les interfaces utilisateur graphique. (Utilisé notamment par Google, Instagram, YouTube, Spotify...).

10.5. Quelques autres langages

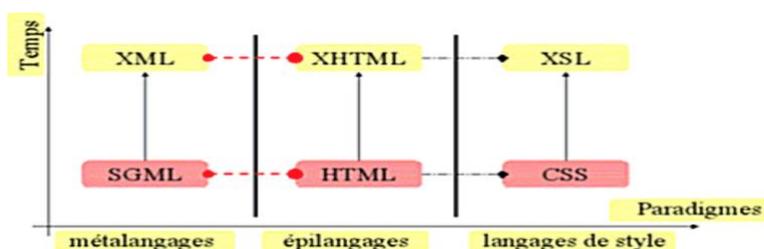
Rust (vitesse et sécurité), Kotlin (Android), Scala (combine programmation fonctionnelle et orienté objet), Swift (rapide et sûr utilisé par Apple), Golang (Open source, développé par Google).

10.6. Conversion entre langages

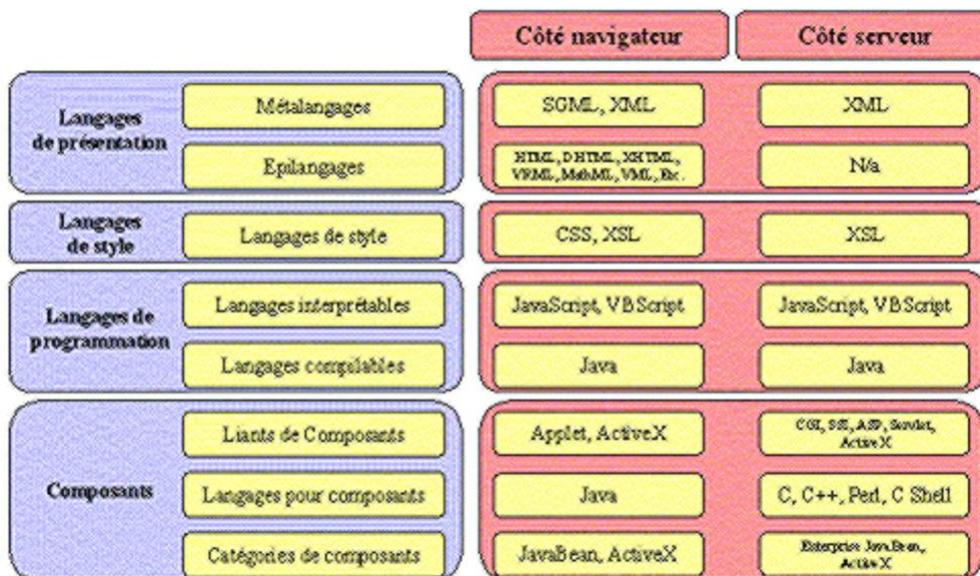
Il est possible de passer du langage XML ou JSON en HTML pour se faire, utiliser le convertisseur ASPOSE. <https://products.aspose.app/words/fr/conversion>

10.7. Synthèse des principaux langages utilisés

10.7.1. Évolution des langages dans le temps



10.7.2. Paysage général de la situation



10.7.3. Synthèse générale

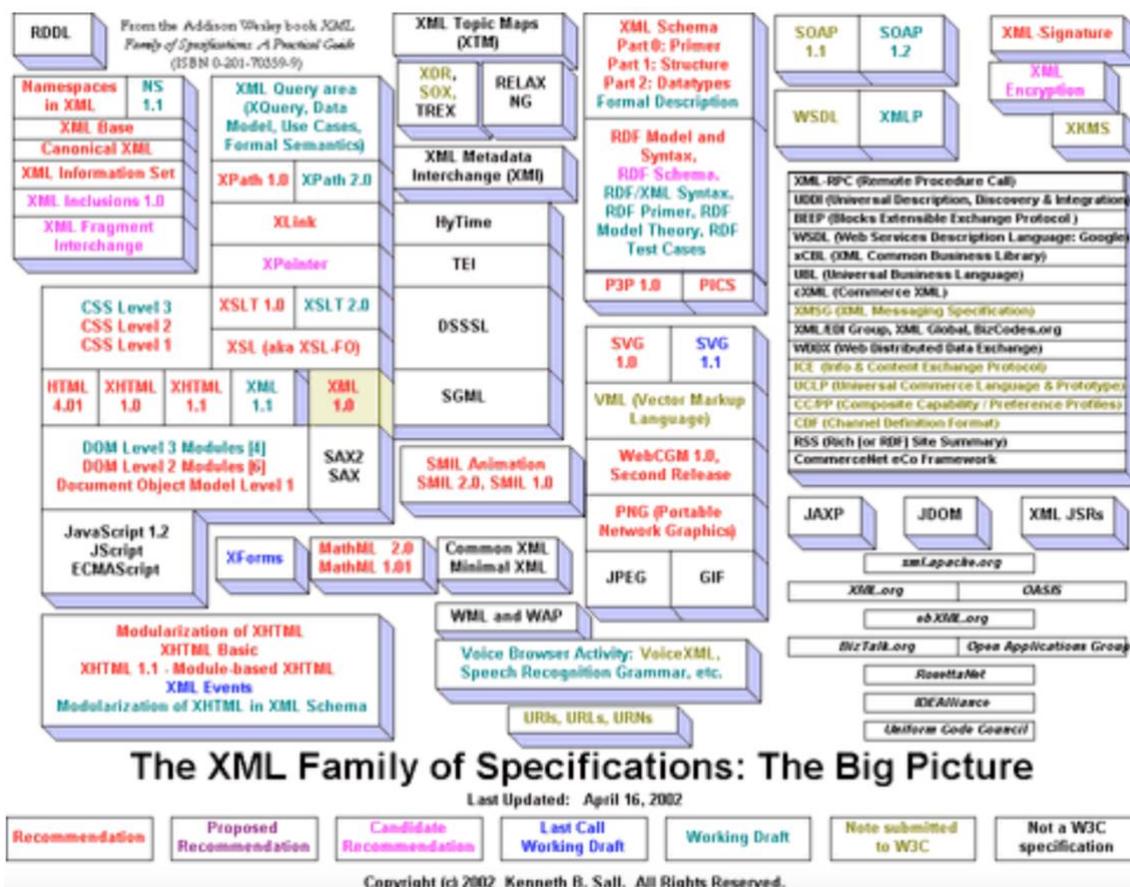


Tableau issu de <https://sebsauvage.net/comprendre/xml/>

10.7.4. Comparaison

PYPL	Tiobe	CodingDojo	IEEE Jobs	IEEE Open	IEEE Trending
Java	Java	SQL	C	C++	C
Python	C	Java	Java	Python	C++
PHP	C++	JavaScript	Python	C	Python
C#	C#	C#	C++	Java	Java
JavaScript	Python	Python	JavaScript	Swift	Swift
C++	JavaScript	C++	C#	JavaScript	R
C	PHP	PHP	PHP	C#	JavaScript
Objective-C	Assembly	iOS	Ruby	Ruby	Ruby
R	VB.NET	Ruby/Rails	HTML	PHP	Go
Swift	Perl		Swift	Ruby	C#
Matlab	Delphi		Assembly	HTML	PHP
Ruby	Ruby		Ruby	Go	Scala
VBA	Swift		Scala	Scala	Arduino
Visual Basic	Objective-C		Shell	Objective-C	Assembly
Scala	Matlab		Perl	Shell	Shell
Perl	Groovy		SQL	Arduino	Objective-C
lua	Visual Basic		Objective-C	Assembly	HTML
Delphi	Ruby		Matlab	Matlab	Rust
Go	Go		Visual Basic	Lua	Haskell
Haskell	PL/SQL		Go	Perl	Visual Basic

Base de comparaison	Python	JavaScript	C#
Environnement	Python se suffit à lui-même.	JavaScript fonctionne dans la plupart des navigateurs sans aucune exigence de plugin.	C# a besoin d'un plugin comme un Silver light.
Accès à l'hôte	Natif	A travers le module FS par exemple.	Natif
Typage	Typé dynamiquement	Typé dynamiquement	Typé statiquement
Ecosystème	Beaucoup de bibliothèques	Le plus de bibliothèques	1 bibliothèque principale
Surcharge d'opérateur	Possède la surcharge d'opérateur	Ne possède pas la surcharge d'opérateur	Possède la surcharge d'opérateur
Créateur	Créé par Guido van Rossum à Centrum Wiskunde & Informatica	Développé par Netscape (Mozilla)	Développé par Microsoft
Périmètre applicatif	Big Data et Machine Learning	WEB	Client Lourd et application
Maintenabilité	Correcte (nombreuse bibliothèque mais stables)	Difficile à maintenir (nombreuses bibliothèques)	Robuste mais langage complexe

11. EXEMPLES PRATIQUES

11.1. Exemple d'utilisation de quelques balises

11.1.1. Cellpadding

On peut modifier l'espacement entre le texte et les bords des cellules avec l'attribut cellpadding placé dans la balise table.

Voir : https://www.w3schools.com/html/tryit.asp?filename=tryhtml_table_cellpadding

Exemple :

```
<table border="1" cellpadding="15">
  <tr>
    <th>Prénom</th>
```

Prénom	Nom	Profession
Mike	Stuntman	Cascadeur
Mister	Pink	Ganster

<pre> <th>Nom</th> <th>Profession</th> </tr> <tr> <td>Mike</td> <td>Stuntman</td> <td>Cascadeur</td> </tr> <tr> <td>Mister</td> <td>Pink</td> <td>Gangster</td> </tr> </table> </pre>	
---	--

11.1.2. Cellspacing

On peut aussi modifier l'espacement entre deux cellules avec l'attribut **cellspacing** placé dans la balise **table**.

Exemple :

<pre> <table border="1" cellpadding="15" cellspacing="10"> <tr> <th>Prénom</th> <th>Nom</th> <th>Profession</th> </tr> <tr> <td>Mike</td> <td>Stuntman</td> <td>Cascadeur</td> </tr> <tr> <td>Mister</td> <td>Pink</td> <td>Gangster</td> </tr> </table> </pre>	
---	--

11.1.3. Width

On peut spécifier la largeur du tableau avec l'attribut **width** placé dans la balise **table**. Cette valeur sera indiquée en pixel (`width="400"`) ou en pourcentage (`width="100%"` de la fenêtre).

Exemple :

<pre> <table border="1" cellpadding="10" cellspacing="1" width="100%"> <tr> <th>Prénom</th> <th>Nom</th> <th>Profession</th> </tr> <tr> <td>Mike</td> <td>Stuntman</td> <td>Cascadeur</td> </tr> <tr> <td>Mister</td> <td>Pink</td> <td>Gangster</td> </tr> </table> </pre>	
---	--

```

</tr>
<tr>
  <td>Mister</td>
  <td>Pink</td>
  <td>Gangster</td>
</tr>
</table>

```

11.1.4. Attributs pour les cellules (<td> et <th>)

Pour chaque cellule, on peut spécifier une largeur grâce à l'attribut **width** placé dans la balise **td** ou **th**. A noter que :

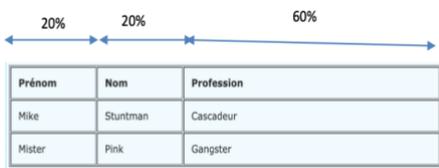
- La largeur peut être spécifiée en pixel (*width="100"*) ou en pourcentage (*width="33%"*)
- Attention à la cohérence des largeurs données à chaque cellule. Si on utilise des pourcentages, le total doit être égal à 100% ! Si on utilise des pixels, le total ne doit pas dépasser la largeur du tableau.
- Il n'est pas utile de préciser la largeur de chaque cellule. **On donne généralement une largeur à la première cellule d'une colonne** et celle-ci sera valable pour toute la colonne.

Exemple :

```

<table border="1" cellpadding="10" cellspacing="1" width="100%">
<tr>
  <th width="20%">Prénom</th>
  <th width="20%">Nom</th>
  <th width="60%">Profession</th>
</tr>
<tr>
  <td>Mike</td>
  <td>Stuntman</td>
  <td>Cascadeur</td>
</tr>
<tr>
  <td>Mister</td>
  <td>Pink</td>
  <td>Gangster</td>
</tr>
</table>

```



Prénom	Nom	Profession
Mike	Stuntman	Cascadeur
Mister	Pink	Gangster

11.1.5. Align

Pour chaque cellule, on peut aussi préciser l'**alignement horizontal** du texte grâce à l'attribut **align** qui peut prendre les valeurs suivantes :

- left (texte aligné à gauche)
- center (texte aligné au centre)
- right (texte aligné à droite)
- justify (texte justifié, peut utilisé sur le web car pas bien rendu)

Pour chaque cellule, on peut enfin préciser l'**alignement vertical** du texte grâce à l'attribut **valign** (vertical align) qui peut prendre les valeurs suivantes :

- top (texte aligné en haut)
- middle (texte aligné au milieu)
- bottom (texte aligné en bas)

Exemple :

<pre> <table border="1" cellpadding="10" cellspacing="1" width="100%"> <tr> <th width="20%" align="center">Prénom</th> <th width="20%" align="center">Nom</th> <th width="60%" align="left">Profession</th> </tr> <tr> <td align="center" valign="top">Mike</td> <td align="center" valign="top">Stuntman</td> <td align="left" valign="top"> Cascadeur
Acteur
Pilote. </td> </tr> <tr> <td align="center" valign="top">Mister</td> <td align="center" valign="top">Pink</td> <td align="left" valign="top">Gangster</td> </tr> </table> </pre>	<table border="1"> <thead> <tr> <th>Prénom</th> <th>Nom</th> <th>Profession</th> </tr> </thead> <tbody> <tr> <td>Mike</td> <td>Stuntman</td> <td>Cascadeur Acteur Pilote.</td> </tr> <tr> <td>Mister</td> <td>Pink</td> <td>Gangster</td> </tr> </tbody> </table>	Prénom	Nom	Profession	Mike	Stuntman	Cascadeur Acteur Pilote.	Mister	Pink	Gangster
Prénom	Nom	Profession								
Mike	Stuntman	Cascadeur Acteur Pilote.								
Mister	Pink	Gangster								

11.1.6. **Border**

On peut se servir de tableaux pour mettre en page des textes ou des images sur plusieurs colonnes. On précisera alors `border="0"` pour que le cadre du tableau ne soit pas visible.

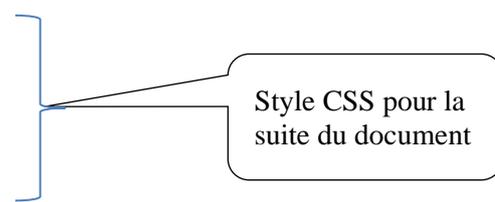
Exemple :

<pre> <table border="0" cellpadding="5" cellspacing="0" width="100%"> <tr> <td valign="top"> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p> <p>Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla.</p></td> <td align="right" valign="top"> </td> </tr> </table> </pre>	<p>Ce qui donne dans le navigateur :</p> <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. </p> <p> Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla </p> 
---	--

11.2. **Autres exemples de programmation**

11.2.1. **Exemple des commandes avec CSS**

Commandes CSS intégrées entre balises `<style>` elles même entrer balises `<head>`)

<pre> <!DOCTYPE html> <html> <head> <meta charset="utf-8" /> <style> p { color: blue; } </style> <title>Premiers tests du CSS</title> </head> <body> <h1>Mon super site</h1> <p>Bonjour et bienvenue sur mon site !</p> <p>Pour le moment, mon site est un peu vide. Patientez encore un peu !</p> </body> </html> </pre>	 <div data-bbox="941 537 1420 683" style="border: 1px solid black; padding: 5px;"> <p>Mon super site</p> <p>Bonjour et bienvenue sur mon site !</p> <p>Pour le moment, mon site est un peu <i>vide</i>. Patientez encore un peu !</p> </div>
---	---

11.2.2. Exemple max et min des caractères en CSS

<pre> <!DOCTYPE html> <html> <head> <style> span { background-color: yellow; } span.ex1 { max-width: 150px; display: inline-block; } </style> </head> <body> <h1>The min-width Property</h1> <h2>min-width: none (default):</h2> Lorem ipsum dolor sit amet... <h2>min-width: 500px:</h2> Lorem ipsum dolor sit amet... </body> </html> </pre>	<div data-bbox="1037 1030 1356 1254" style="border: 1px solid black; padding: 10px;"> <p>The min-width Property</p> <p>min-width: none (default):</p> <p>Lorem ipsum dolor sit amet...</p> <p>min-width: 500px:</p> <p>Lorem ipsum dolor sit amet...</p> </div>
---	--

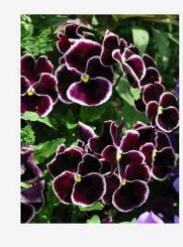
11.2.3. Intégration de photo avec CSS

<pre> <head> <style> .warning { border: 5px dotted #f00; background-color: #ff0; padding: .2rem; display: flex; } </pre>	
--	--

<pre> flex-direction: column; } .warning img { width: 100; } .warning p { font: small-caps bold 1.2rem sans-serif; text-align: center; } </style> </head> <div class="warning"> ← HTML <p>Beware of the leopard</p> </div> </pre>	
--	--

11.2.4. Intégration de photo avec zoom avec CSS

<http://css.mammothland.net/zoomer-une-image-avec-css.php>

<pre> <head> <style> ← CSS .zoom { height:400px; } .zoom p { text-align:center; } .zoom img { width:200px; height:267px; } .zoom img:hover { width:400px; height:600px; } </style> </head> <div class="zoom"> ← HTML <p> </p> </div> </pre>	<div style="text-align: center;"> <div data-bbox="1069 1108 1300 1164" style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">Image de base</div> <div data-bbox="1069 1243 1348 1310" style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">Image agrandie</div>  </div>
--	---

11.3. Autres modules :

11.3.1. Comment zoomer au survol

https://www.w3schools.com/howto/howto_css_zoom_hover.asp

```

<style> image{transition: transform .2s ; }
Image: hover {transform :scale(2) ; position :absolute ; left :5% ; }
<body><td><div class=image><img align= "right" scr= aa.gif" width="169" height="97"/></div></td>

```

<http://www.mytesting.fr/2016/11/comment-agrandir-une-image-au-passage-de-la-souris.html>

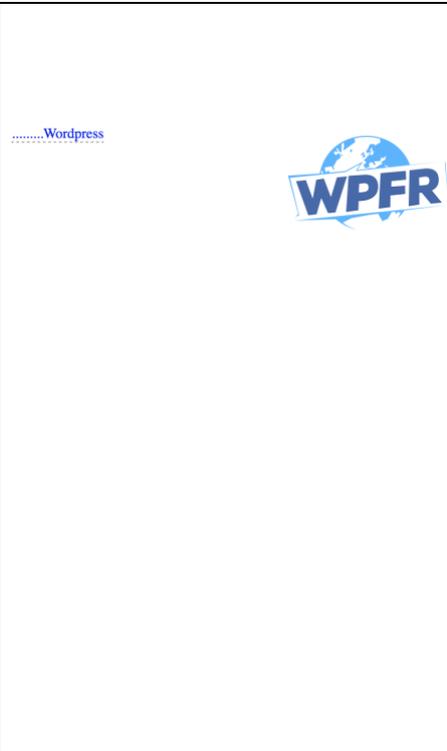
<pre> <div class=zoom> <style> .zoom { width: 320px; height: 240px; position: absolute; right:500px; bottom:300px; }https://www.w3schools.com/howto/howto_css_zoom_hover.asp .image { width: 100%; height: 100%; } .image img { /* La transition s'applique à la fois sur la largeur et la hauteur, avec une durée d'une seconde. */ -webkit-transition: all 1s ease; /* Safari et Chrome */ -moz-transition: all 1s ease; /* Firefox */ -ms-transition: all 1s ease; /* Internet Explorer 9 */ -o-transition: all 1s ease; /* Opera */ transition: all 1s ease; } .image: hover img { /* L'image est doublée */ -webkit-transform: scale(2); /* Safari et Chrome */ -moz-transform: scale(2); /* Firefox */ -ms-transform: scale(2); /* Internet Explorer 9 */ -o-transform: scale(2); /* Opera */ transform: scale(2); } </style> <div class=image> </pre>	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> Déplacement et zoom de l'image au passage de la souris </div> 
--	---

11.3.2. Position d'une infobulle/Tooltip (en dessous)

<pre> <!DOCTYPE html> <html> <style> .tooltip { position: relative; display: inline-block; border-bottom: 1px dotted black; } .tooltip .tooltiptext { visibility: hidden; width: 120px; background-color: black; color: #fff; text-align: center; border-radius: 6px; padding: 5px 0; /* Position the tooltip */ position: absolute; z-index: 1; top: 100%; </pre>	<p style="text-align: center;">Bottom Tooltip</p> <p style="text-align: center;">Move the mouse over the text below:</p> <div style="text-align: center; margin-top: 10px;"> Hover over me Tooltip text </div>
--	--

<pre> left: 50%; margin-left: -60px; } .tooltip:hover .tooltiptext { visibility: visible; } </style> <body style="text-align:center;"> <h2>Bottom Tooltip</h2> <p>Move the mouse over the text below:</p> <div class="tooltip">Hover over me Tooltip text </div> </body> </html> </pre>	
--	--

11.3.3. Incorporation d'une image « au survol »:

<pre> <style> a.survol { position:relative; border-bottom:1px dashed #808080; text-decoration: none; } a:hover.survol { text-decoration: none; background: none; } a.survol span {display: none;} a:hover.survol span { display: block; position: relative; top: -20px; left: 300px; z-index: 20; width:30%; padding:2px 4px; } </style>Wordpress </pre>	
---	---

11.4. Exemples de menus déroulants (sous l'onglet)

Code CCS

<pre> #menu-demo2, #menu-demo2 ul{ padding:0; margin:0; list-style:none; text-align:center; } #menu-demo2 li{ </pre>	
--	--

```
display:inline-block;
position:relative;
border-radius:8px 8px 0 0;
}
#menu-demo2 ul li{
display:inherit;
border-radius:0;
}
#menu-demo2 ul li:hover{
border-radius:0;
}
#menu-demo2 ul li:last-child{
border-radius:0 0 8px 8px;
}
#menu-demo2 ul{
position:absolute;
z-index: 1000;
max-height:0;
left: 0;
right: 0;
overflow:hidden;
-moz-transition: .8s all .3s;
-webkit-transition: .8s all .3s;
transition: .8s all .3s;
}
#menu-demo2 li:hover ul{
max-height:15em;
}
/* background des liens menus */
#menu-demo2 li:first-child{
background-color: #65537A;
background-image:-webkit-linear-gradient(top, #65537A 0%, #2A2333 100%);
background-image:linear-gradient(to bottom, #65537A 0%, #2A2333 100%);
}
#menu-demo2 li:nth-child(2){
background-color: #729EBF;
background-image:-webkit-linear-gradient(top, #729EBF 0%, #333A40 100%);
background-image:linear-gradient(to bottom, #729EBF 0%, #333A40 100%);
}
#menu-demo2 li:nth-child(3){
background-color: #F6AD1A;
background-image:-webkit-linear-gradient(top, #F6AD1A 0%, #9F391A 100%);
background-image:linear-gradient(to bottom, #F6AD1A 0%, #9F391A 100%);
}
#menu-demo2 li:last-child{
background-color: #CFFF6A;
background-image:-webkit-linear-gradient(top, #CFFF6A 0%, #677F35 100%);
background-image:linear-gradient(to bottom, #CFFF6A 0%, #677F35 100%);
}
/* background des liens sous menus */
#menu-demo2 li:first-child li{
background:#2A2333;
}
#menu-demo2 li:nth-child(2) li{
background:#333A40;
}
#menu-demo2 li:nth-child(3) li{
background:#9F391A;
}
#menu-demo2 li:last-child li{
```

```
background:#677F35;
}
/* background des liens menus et sous menus au survol */
#menu-demo2 li:first-child:hover, #menu-demo2 li:first-child li:hover{
background:#65537A;
}
#menu-demo2 li:nth-child(2):hover, #menu-demo2 li:nth-child(2) li:hover{
background:#729EBF;
}
#menu-demo2 li:nth-child(3):hover, #menu-demo2 li:nth-child(3) li:hover{
background:#F6AD1A;
}
#menu-demo2 li:last-child:hover, #menu-demo2 li:last-child li:hover{
background:#CFFF6A;
}
/* les a href */
#menu-demo2 a{
text-decoration:none;
display:block;
padding:8px 32px;
color:#fff;
font-family:arial;
}
#menu-demo2 ul a{
padding:8px 0;
}
#menu-demo2 li:hover li a{
color:#fff;
text-transform:inherit;
}
#menu-demo2 li:hover a, #menu-demo2 li li:hover a{
color:#000;
}
}
```

Code HTML

```
<ul id="menu-demo2">
  <li><a href="#">Lien menu 1</a>
</ul>
  <li><a href="#">lien sous menu 1</a></li>
  <li><a href="#">lien sous menu 1</a></li>
  <li><a href="#">lien sous menu 1</a></li>
  <li><a href="#">lien sous menu 1</a></li>
</ul>
</li>
  <li><a href="#">Lien menu 2</a>
</ul>
  <li><a href="#">Lien sous menu 2</a></li>
  <li><a href="#">Lien sous menu 2</a></li>
  <li><a href="#">Lien sous menu 2</a></li>
  <li><a href="#">Lien sous menu 2</a></li>
</ul>
</li>
  <li><a href="#">Lien menu 3</a>
</ul>
  <li><a href="#">lien sous menu 3</a></li>
  <li><a href="#">lien sous menu 3</a></li>
  <li><a href="#">lien sous menu 3</a></li>
  <li><a href="#">lien sous menu 3</a></li>
</ul>
</li>
</ul>
```



```

<li><a href="#">Lien menu 4</a>
<ul>
  <li><a href="#">Lien sous menu 4</a></li>
  <li><a href="#">Lien sous menu 4</a></li>
  <li><a href="#">Lien sous menu 4</a></li>
  <li><a href="#">Lien sous menu 4</a></li>
</ul>
</li>
</ul>
    
```

11.5. Exemples sur les tableaux

11.5.1. Intégration d'un tableau

```

<HTML>
  <HEAD>
<TITLE>le nom de page</TITLE>
  </HEAD>
<BODY BGCOLOR="#CCFFFF" TEXT="#000000"
MLINK="#ff0000">
  <H1>Le premier titre</H1>
  Le premier texte
  <BR>
  <FONT SIZE=6>est trop grand</FONT> et <BR>
  <p align=center><FONT SIZE=2><B>-----> est</B> trop
petit</FONT> </p>
  <HR size=1 width=50%>
  <BR>
  <table border-collapse= collapse border="1" solidblack;
cellpadding="10" cellspacing="1" width="100%">
  <tr>
    <th width="20%" align="center">Prénom</th>
    <th width="20%" align="center">Nom</th>
    <th width="60%" align="left">Profession</th>
  </tr>
  <tr>
    <td align="center" valign="top">Mike</td>
    <td align="left" valign="top">Stuntman de de de Stuntman
Stuntman</td>
    <td align="left" valign="top">
      Cascadeur Acteur Pilote.
    </td>
  </tr>
  <tr>
    <td align="center" valign="top">Mister</td>
    <td align="center" valign="top">Pink</td>
    <td align="left" valign="top">Gangster</td>
  </tr>
</table>
</BODY>
</HTML>
        
```

Le premier titre

Le premier texte
est trop grand et

-----> est trop petit

Prénom	Nom	Profession
Mike	Stuntman de de de Stuntman Stuntman	Cascadeur Acteur Pilote.
Mister	Pink	Gangster

ligne

Style du tableau

11.5.2. Commande pour la constitution d'un tableau :

<https://developer.mozilla.org/fr/docs/Web/HTML/Element/table>

<pre> } table { border-collapse: collapse; border: 2px solid rgb(200, 200, 200); letter-spacing: 1px; font-family: sans-serif; font-size: .8rem; } </style> <table> <caption>Alien football stars</caption> <tr> <th scope="col">Player</th> <th scope="col">Gloobles</th> <th scope="col">Za'taak</th> </tr> <tr> <th scope="row">TR-7</th> <td>7</td> <td>4,569</td> </tr> <tr> <th scope="row">Khires Odo</th> <td>7</td> <td>7,223</td> </tr> <tr> <th scope="row">Mia Oolong</th> <td>9</td> <td>6,219</td> </tr> </table> </pre>	<p style="color: red; font-weight: bold;">← HTML</p> <div style="text-align: center; margin-top: 20px;"> <p>Alien football stars</p> <table border="1" style="margin: auto; border-collapse: collapse; text-align: center;"> <thead> <tr style="background-color: #c6e0b4;"> <th>Player</th> <th>Gloobles</th> <th>Za'taak</th> </tr> </thead> <tbody> <tr style="background-color: #d9d9d9;"> <td>TR-7</td> <td>7</td> <td>4,569</td> </tr> <tr style="background-color: #d9d9d9;"> <td>Khires Odo</td> <td>7</td> <td>7,223</td> </tr> <tr style="background-color: #d9d9d9;"> <td>Mia Oolong</td> <td>9</td> <td>6,219</td> </tr> </tbody> </table> </div>	Player	Gloobles	Za'taak	TR-7	7	4,569	Khires Odo	7	7,223	Mia Oolong	9	6,219
Player	Gloobles	Za'taak											
TR-7	7	4,569											
Khires Odo	7	7,223											
Mia Oolong	9	6,219											

11.5.4. Pour la gestion des blancs dans cellules

Au sein d'un élément, (**white-space**) définit le comportement des textes avec gestion des espaces et des retours chariot ; voir le très bon détail dans le lien suivant :

<https://developer.mozilla.org/fr/docs/Web/CSS/white-space>

11.6. Exemple comparatif de tableaux scrollables

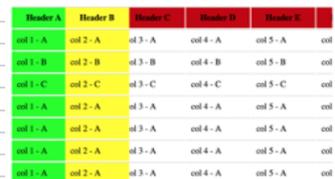
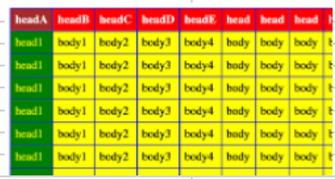
C'est-à-dire que le tableau se déplace en conservant le titre des lignes et des colonnes (th) qui restent fixes.

<https://nosmoking.developpez.com/demos/css/table-scrollable.php>

<https://jsfiddle.net/qwubvg9m/1/>

pour plusieurs colonnes qui restent fixes (trois la premières et la dernière) ainsi que les premières et dernières lignes on peut s'inspirer de ce site :

<https://glin.github.io/reactable/articles/sticky-columns.html#:~:text=To%20make%20multiple%20columns%20sticky,left%3A%20200px%3B%20%2C%20etc.>

https://isfiddle.net/qwubvg9m/1/	https://isfiddle.net/brianmerritt/8c5igs4b/		
1 colonne et une ligne d'entête fixes	2 colonnes fixe mais ligne horizontale d'entête mobile...	Synthèse	
programme	programme	programme	signification
			
NB: Dans certains cas (?) la 1ère ligne peut s'échapper!...			
<code><style></code>	<i>CSS inclus entre <style></style></i>	<code><Style></code>	<i>CSS inclus entre <style></style></i>
<code>div {</code>	<i>dimension de la boîte active</i>		<i>CSS inclus entre <style></style></i>
<code>max-width: 40em;</code>	<i>longueur de la boîte</i>		
<code>max-height: 15em;</code>	<i>hauteur de la boîte</i>		
<code>overflow: scroll;</code>	<i>comportement en cas de dépassement</i>		
<code>position: relative;</code>	<i>positionnement</i>		
<code>}</code>			
<code>table {</code>	<i>caractéristiques du tableau</i>	<code>table {</code>	<i>caractéristiques du tableau</i>
<code>position: relative;</code>	<i>position</i>	<code>table-layout: fixed;</code>	
<code>border-collapse: collapse;</code>	<i>bordures des cellules collées</i>	<code>width: 100%;</code>	
<code>}</code>		<code>*margin-left: -100px; /*ie7*/</code>	
<code>tbody th {</code>	<i>Entête verticale</i>	<code>td, th {</code>	
<code>position: -webkit-sticky; /* for Safari */</code>	<i>élément relatif puis fixe</i>	<code>vertical-align: top;</code>	
<code>position: sticky;</code>	<i>élément relatif puis fixe</i>	<code>border-top: 1px solid #ccc;</code>	
<code>left: 0;</code>	<i>couleur du fond = vert</i>	<code>padding: 10px;</code>	
<code>background: #0FF;</code>		<code>width: 100px;</code>	
<code>border-right: 1px solid #CCC;</code>		<code>}</code>	
<code>}</code>			
<code>thead th {</code>	<i>Entête horizontale</i>	<code>th {</code>	<i>entête horizontale</i>
<code>position: -webkit-sticky; /* for Safari */</code>	<i>élément relatif puis fixe</i>	<code>/* position: absolute;</code>	
<code>position: sticky;</code>		<code>*position: relative; /*ie7*/</code>	
<code>top: 0;</code>	<i>couleur du fond = rouge</i>	<code>/* left: 0; */</code>	
<code>background: #C00;</code>	<i>couleur du texte = vert</i>	<code>width: 100px;</code>	
<code>color: #0FF;</code>	<i>hauteur de l'entête</i>	<code>background-color: red;</code>	<i>Couleur de FondH = rouge</i>
<code>height: 50px;</code>		<code>color: green;</code>	<i>couleur du texte = vert</i>
<code>}</code>		<code>}</code>	
<code>thead th:first-child {</code>	<i>entête de la 1ère colonne</i>	<code>.hard_left {</code>	<i>1ère colonne de gauche</i>
<code>left: 0;</code>	<i>couleur du fond = bleu</i>	<code>position: absolute;</code>	
<code>background-color: blue;</code>		<code>*position: relative; /*ie7*/</code>	
<code>z-index: 1;</code>		<code>left: 0;</code>	
<code>}</code>		<code>width: 100px;</code>	
		<code>background: #00ff00</code>	<i>FondV 1ère entête= vert</i>
		<code>}</code>	
<code>td,</code>	<i>cellules du tableau et de l'entête</i>	<code>.next_left {</code>	<i>2ème colonne de gauche</i>
<code>th {</code>	<i>espace entre texte et cellule</i>	<code>position: absolute;</code>	
<code>padding: 0.25em;</code>		<code>*position: relative; /*ie7*/</code>	
<code>}</code>		<code>left: 100px;</code>	
		<code>width: 80px;</code>	<i>dimension de la 2ème colonne</i>
		<code>background: #ffff00</code>	<i>FondV 2ème entête= jaune</i>
		<code>}</code>	
		<code>.outer {position: relative}</code>	
		<code>.inner {</code>	
		<code>overflow-x: scroll;</code>	
		<code>overflow-y: visible;</code>	
		<code>width: 400px;</code>	
		<code>margin-left: 200px;</code>	
		<code>}</code>	
<code></style></code>			<code></style></code>
<code><div></code>		<code></style></code>	<code><div class="table-conteneur"></code>
<code><table></code>		<code><div class="outer"></code>	<code><table class="table-scrollable"></code>
<code><thead></code>		<code><div class="inner"></code>	
<code><tr></code>		<code><table></code>	<code><thead></code>
<code><th></th></code>		<code><tr></code>	<code><tr></code>
<code><th>headH1</th></code>		<code><th class="hard_left">Header A</th></code>	<code><th>headA</th></code>
<code><th>headH2</th></code>		<code><th class="next_left">Header B</th></code>	<code><th>headB</th></code>
<code><th>headH3</th></code>		<code><th>Header C</th></code>	<code><th>headC</th></code>
<code><th>head</th></code>		<code><th>Header D</th></code>	<code><th>headD</th></code>
<code><th>head</th></code>		<code><th>Header E</th></code>	<code><th>headE</th></code>
<code><th>head</th></code>		<code><th>Header A</th></code>	<code><th>head</th></code>
<code><th>head</th></code>		<code><th>Header B</th></code>	<code><th>head</th></code>
<code><th>head</th></code>		<code><th>Header C</th></code>	<code><th>head</th></code>
<code><th>head</th></code>		<code><th>Header D</th></code>	<code><th>head</th></code>
<code><th>head</th></code>		<code><th>Header E</th></code>	<code><th>head</th></code>
<code><th>head</th></code>		<code><th>Header E</th></code>	<code><th>head</th></code>
<code></tr></code>		<code></tr></code>	<code></tr></code>
<code></thead></code>			<code></thead></code>

<td>body</td>	<td>col 5 - B</td>	<td>body</td>
<td>body</td>	<td>col 5 - B</td>	<td>body</td>
</tr>	</tr>	</tr>
<tr>	<tr>	<tr>
<th>head</th>	<td class="hard_left">col 1 - A</td>	<td>head1</td>
<td>body</td>	<td class="next_left">col 2 - A</td>	<td>body1</td>
<td>body</td>	<td>col 3 - A</td>	<td>body2</td>
<td>body</td>	<td>col 4 - A</td>	<td>body3</td>
<td>body</td>	<td>col 5 - A</td>	<td>body4</td>
<td>body</td>	<td>col 1 - C</td>	<td>body</td>
<td>body</td>	<td>col 2 - C</td>	<td>body</td>
<td>body</td>	<td>col 3 - C</td>	<td>body</td>
<td>body</td>	<td>col 4 - C</td>	<td>body</td>
<td>body</td>	<td>col 5 - C</td>	<td>body</td>
<td>body</td>	<td>col 5 - C</td>	<td>body</td>
<td>body</td>	<td>col 5 - C</td>	<td>body</td>
</tr>	</tr>	</tr>
</tbody>	</tbody>	</tbody>
</table>	</table>	</table>
</div>	</div>	</div>

11.7. Exemple avec plusieurs colonnes collantes

<https://glin.github.io/reactable/articles/sticky-columns.html#:~:text=To%20make%20multiple%20columns%20sticky,left%3A%20200px%3B%20%2C%20etc.>

<pre> reactable(MASS::Cars93, columns = list(Manufacturer = colDef(class = "sticky left-col-1", headerClass = "sticky left-col-1"), Model = colDef(class = "sticky left-col-2", headerClass = "sticky left-col-2"), Type = colDef(class = "sticky left-col-3", headerClass = "sticky left-col-3"), Make = colDef(class = "sticky right-col-1", headerClass = "sticky right-col-1")), wrap = FALSE) .sticky { position: sticky !important; background: #fff; z-index: 1; } .left-col-1 { left: 0; } .left-col-2 { left: 100px; } .left-col-3 { left: 200px; </pre>	<p>Résultat : 3 colonnes fixe à gauche et 1 à droite</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Fabricant</th> <th>Modèle</th> <th>Taper</th> <th>Prix min.</th> <th>Prix</th> <th>Prix max.</th> <th>Faire</th> </tr> </thead> <tbody> <tr> <td>Acura</td> <td>Integra</td> <td>Petit</td> <td>12,9</td> <td>15,9</td> <td>18,8</td> <td>Acura Inte...</td> </tr> <tr> <td>Acura</td> <td>Légende</td> <td>Taille moye...</td> <td>29,2</td> <td>33,9</td> <td>38,7</td> <td>Légende A...</td> </tr> <tr> <td>Audi</td> <td>90</td> <td>Compact</td> <td>25,9</td> <td>29,1</td> <td>32,3</td> <td>Audi 90</td> </tr> <tr> <td>Audi</td> <td>100</td> <td>Taille moye...</td> <td>30,8</td> <td>37,7</td> <td>44,6</td> <td>Audi 100</td> </tr> <tr> <td>BMW</td> <td>535i</td> <td>Taille moye...</td> <td>23,7</td> <td>30</td> <td>36,2</td> <td>BMW 535i</td> </tr> </tbody> </table>	Fabricant	Modèle	Taper	Prix min.	Prix	Prix max.	Faire	Acura	Integra	Petit	12,9	15,9	18,8	Acura Inte...	Acura	Légende	Taille moye...	29,2	33,9	38,7	Légende A...	Audi	90	Compact	25,9	29,1	32,3	Audi 90	Audi	100	Taille moye...	30,8	37,7	44,6	Audi 100	BMW	535i	Taille moye...	23,7	30	36,2	BMW 535i
Fabricant	Modèle	Taper	Prix min.	Prix	Prix max.	Faire																																					
Acura	Integra	Petit	12,9	15,9	18,8	Acura Inte...																																					
Acura	Légende	Taille moye...	29,2	33,9	38,7	Légende A...																																					
Audi	90	Compact	25,9	29,1	32,3	Audi 90																																					
Audi	100	Taille moye...	30,8	37,7	44,6	Audi 100																																					
BMW	535i	Taille moye...	23,7	30	36,2	BMW 535i																																					

<pre>border-right: 1px solid #eee !important; } .right-col-1 { right: 0; border-left: 1px solid #eee !important; }</pre>	
--	--

11.8. Exemples d'autres propriétés obtenues avec JavaScript

11.8.1. Test bouton et Zindex de w3school

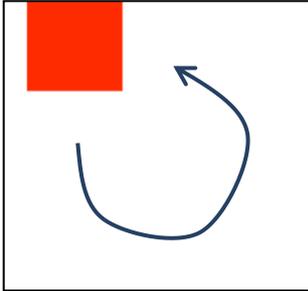
Changement de position des rectangles inspiré de https://www.w3schools.com/cssref/tryit.asp?filename=trycss_js_z-index

<pre><html> <head> <style> #myDIV { position: absolute; top: 100px; left: 0px; width: 100px; height: 300px; background-color: lightblue; border: 1px solid black; } #DIV2 { position: relative; top: 35px; left: 60px; width: 100px; height: 300px; background-color: coral; border: 1px solid black; } </style> </head> <body> <p>Cliquez sur le bouton "Try it" pour changer le Zindex de l'élément bleu</p> <button onclick="myFunction()">Try it</button> <div id="DIV2"> <h1>Voila!</h1> </div> <div id="myDIV"> </div> <script> function myFunction() { document.getElementById("myDIV").style.zIndex = "-1"; } </script> </body> </html></pre>	<p>Click the "Try it" button to change the z-index of the blue DIV element:</p> <p>Cliquez sur le bouton "Try it" pour changer le Zindex de l'élément bleu</p>
---	--

11.8.2. Animation : direction et changement de couleur

Circulation du carré avec changement de couleur...inspiré de https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_animation-direction

<pre><!DOCTYPE html> <html> <head> <style></pre>	
--	--

<pre> div { width: 100px; height: 100px; position: relative; background-color: red; animation-name: example; animation-duration: 4s; animation-iteration-count: infinite; } @keyframes example { 0% {background-color: red; left:0px; top:0px;} 25% {background-color: blue; left:0px; top:200px;} 50% {background-color: green; left:200px; top:200px;} 75% {background-color: yellow; left:200px; top:0px;} 100% {background-color: red; left:0px; top:0px;} } </style> </head> <body> <div></div> </body> </html> </pre>	
---	--

11.8.3. Passage alternatif d'un tableau à un autre

Changement de section successives inspiré de https://www.w3schools.com/cssref/tryit.asp?filename=trycss_scroll_behavior

<pre> <!DOCTYPE html> <html> <head> <style> html { scroll-behavior: smooth; } #section1 { height: 500px; width : 200px ; background-color: pink; } #section2 { height: 600px; width : 300px ; background-color: yellow; } </style> </head> <body> <h1>Smooth Scroll</h1> <div class="main" id="section1"> <h2>Section 1</h2> <p> Cliquez sur le lien pour voir l'effet scrolling.</p> Cliquez ici </div> <div class="main" id="section2"> <h2>Section 2</h2>

 Cliquez ici pour retourner à la Section 1 </div> <p>Note: The scroll-behavior property is not supported in Internet Explorer.</p> </body> </html> </pre>	
--	--

11.8.4. Modifier un texte au clic

Utilise la fonction « toggle » (c'est-à-dire une bascule ou un interrupteur) permet de basculer entre l'apparition et la disparition d'un nom de classe « mystyle ». Inspiré de https://www.w3schools.com/howto/howto_js_toggle_class.asp

<pre> <!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width, initial-scale=1"> <style> .mystyle { width: 100%; padding: 25px; background-color: coral; color: white; font-size: 25px; box-sizing: border-box; } </style> </head> <body> <p>Click the "Try it" button to toggle between adding and removing the "mystyle" class name of the DIV element:</p> <button onclick="myFunction()">Try it</button> <div id="myDIV"> This is a DIV element. </div> <script> function myFunction() { var element = document.getElementById("myDIV"); element.classList.toggle("mystyle"); } </script> </body> </html> </pre>	<div style="border: 1px solid blue; padding: 10px; margin-bottom: 20px;"> <p>Click the "Try it" button to toggle between adding and removing the "mystyle" class name of the DIV element:</p> <div style="border: 1px solid gray; display: inline-block; padding: 2px 5px; margin: 5px;">Try it</div> This is a DIV element. </div> <p>Ce qui donne au clic :</p> <div style="border: 1px solid blue; padding: 10px;"> <p>Click the "Try it" button to toggle between adding and removing the "mystyle" class name of the DIV element:</p> <div style="border: 1px solid gray; display: inline-block; padding: 2px 5px; margin: 5px;">Try it</div> <div style="background-color: coral; color: white; padding: 10px; text-align: center; font-size: 25px;"> This is a DIV element. </div> </div>
--	--

11.8.5. Apparition/disparition du contenu d'une ligne dans un tableau

(Opérationnel)

<https://codepen.io/Zonecss/pen/YJyMEL>

```

<style>
#tbl, #tbl td, #tbl th {
  border: 1px solid #000;
  border-collapse : collapse;
}
#tbl th {cursor:pointer}
#tbl td{visibility:collapse }
#tbl td.show{visibility:visible}
</style>
<head>
<script>
function showLigne(oEvent){
  var sClass="show",
      oBt = oEvent.currentTarget,
      oTr = oBt.parentNode,
      aCells = oTr.cells ;
  for(var i = 0; i<aCells.length ;i++){
    var oCel = aCells[i];
    if(oCel == oBt){
      continue;
    }
    if(oCel.classList.contains(sClass)){
      oCel.classList.remove(sClass)
    }else{
      oCel.classList.add(sClass)
    }
  }
}
//for
//fct
document.addEventListener('DOMContentLoaded',function(){
  var aBts = document.getElementsByClassName("bt");
  for(var i = 0; i < aBts.length ;i++){
    aBts[i].addEventListener("click", showLigne);
  }
});
</script>
<h1>Cacher le contenu d'une ligne dans un tableau</h1>
<p>Par défaut les données sont cachées.<br>
Au clique sur l'entête de la ligne, on affiche ou cache le contenu de la ligne.
</p>
<table id="tbl">
  <tr>
    <th class="bt">Ligne 1</th>
    <td>111</td>
    <td>222</td>
    <td>333</td>
  </tr>
  <tr>
    <th class="bt">Ligne 2</th>
    <td>444</td>
    <td>555</td>
    <td>666</td>
  </tr>
  <tr>
    <th class="bt">Ligne 3</th>
    <td>777</td>
    <td>888</td>
    <td>999</td>
  </tr>
</table>
<hr>
<div class="mise-en-page-lien">
  <div class="bloc-mise-en-page bloc-lien">
  </div>
</div>
</head>

```

Cacher le contenu d'une ligne dans un tableau

Par défaut les données sont cachées.
 Au clique sur l'entête de la ligne, on affiche ou cache le contenu de la ligne.

Ligne 1	111	222	333
Ligne 2	444	555	666
Ligne 3			

11.8.6. Apparition/disparition d'une colonne dans un tableau

<pre> <style> #tbl, #tbl td, #tbl th { border: 1px solid #000; border-collapse : collapse; } #tbl th {cursor:pointer} #tbl td{visibility:visible } #tbl td.show{visibility:collapse} </style> <head> <script> function showColonne(oEvent){ var sClass="show", oBt = oEvent.currentTarget, iPos = oBt.cellIndex, oTr = document.getElementById("tbl").rows; for(var i = 1; i<oTr.length ;i++){ var oCel = oTr[i].cells[iPos]; if(oCel.classList.contains(sClass)){ oCel.classList.remove(sClass) }else{ oCel.classList.add(sClass) } } } //for //fct document.addEventListener('DOMContentLoaded',function(){ var aBts = document.getElementsByClassName("bt"); for(var i = 0; i<aBts.length ;i++){ aBts[i].addEventListener("click", showColonne); } }); </script> <h1>Cacher le contenu d'une ligne dans un tableau</h1> <p>Par défaut les données sont visibles.
 Au clique sur l'entête de la colonne, on affiche ou cache le contenu de la ligne. </p> <table id="tbl"> <tr> <th class="bt">COLONNE 1</th> <th class="bt">COLONNE 2</th> <th class="bt">COLONNE 3</th> </tr> <tr> <td>111</td> <td>222</td> <td>333</td> </tr> <tr> <td>444</td> <td>555</td> <td>666</td> </tr> <tr> <td>777</td> <td>888</td> <td>999</td> </tr> </table> <hr> <div class="mise-en-page-lien"> <div class="bloc-mise-en-page bloc-lien"> <h2>Liens utiles CSS</h2> </pre>	<p style="text-align: center;">Cacher le contenu d'une colonne dans un tableau</p> <p>Par défaut les données sont visibles. Au clique sur l'entête de la colonne, on affiche ou cache le contenu de la ligne.</p> <table border="1" style="margin-left: auto; margin-right: auto; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>COLONNE 1</th> <th>COLONNE 2</th> <th>COLONNE 3</th> </tr> </thead> <tbody> <tr> <td>111</td> <td>222</td> <td>333</td> </tr> <tr> <td>444</td> <td>555</td> <td>666</td> </tr> <tr> <td>777</td> <td>888</td> <td>999</td> </tr> </tbody> </table>	COLONNE 1	COLONNE 2	COLONNE 3	111	222	333	444	555	666	777	888	999
COLONNE 1	COLONNE 2	COLONNE 3											
111	222	333											
444	555	666											
777	888	999											

11.8.7. Apparition/disparition de lignes et de colonnes avec JavaScript

Inspiré de <https://codepen.io/bazooka07/pen/pKJaQa>

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Cacher ou apparaitre ligne et colonne</title>
  <!-- See at https://forum.alsacreations.com/topic-5-83353-1.html -->
  <!-- Play at https://codepen.io/bazooka07/pen/pKJaQa -->
  <meta http-equiv="content-type" content="text/html;charset=utf-8" />
  <style>
    body { font-family: 'Noto Sans', Ubuntu, Arial; background: #666; }
    table { border-collapse: collapse; border: 1px solid #444; background: #fff;
  }

  tr:nth-of-type(2n) { background: yellow; }
  th:hover { background: red; }
  tr:hover { background: maroon; color: orange; }
  th { cursor: pointer; }
  td, th { text-align: center; padding: 0.3rem 0.5rem; }
  td { border-left: 1px solid red; }
  .hide { display: none; }
  </style>
</head>
<body>
<div id="toolbar">
  <button data-css="tr.hide">All rows</button>
  <button data-css="td.hide, th.hide">All cols</button>
</div>
<table id="id1">
  <tr>
    <th>&nbsp;</th>
    <th class="col">X1</th>
    <th class="col">X2</th>
    <th class="col">X3</th>
    <th class="col">X4</th>
  </tr>
  <tr>
    <th class="row">Y1</th>
    <td>L1C1</td>
    <td>L1C2</td>
    <td>L1C3</td>
    <td>L1C4</td>
  </tr>
  <tr>
    <th class="row">Y2</th>
    <td>L2C1</td>
    <td>L2C2</td>
    <td>L2C3</td>
    <td>L2C4</td>
  </tr>
  <tr>
    <th class="row">Y3</th>
    <td>L3C1</td>
    <td>L3C2</td>
    <td>L3C3</td>
    <td>L3C4</td>
  </tr>
  <tr>
    <th class="row">Y4</th>
    <td>L4C1</td>
    <td>L4C2</td>
    <td>L4C3</td>
    <td>L4C4</td>
  </tr>
  <tr>
    <th class="row">Y5</th>
    <td>L5C1</td>
    <td>L5C2</td>
    <td>L5C3</td>
    <td>L5C4</td>
  </tr>
  </table>

```

Ligne paire=jaune

Changement de couleur au passage

Bord à gauche cellule=rouge

Commande de réapparition

All rows		All cols			
	X1	X2	X3	X4	
Y1	L1C1	L1C2	L1C3	L1C4	
Y2	L2C1	L2C2	L2C3	L2C4	
Y3	L3C1	L3C2	L3C3	L3C4	
Y4	L4C1	L4C2	L4C3	L4C4	
Y5	L5C1	L5C2	L5C3	L5C4	

<pre> <tr> <th class="row">Y5</th> <td>L5C1</td> <td>L5C2</td> <td>L5C3</td> <td>L5C4</td> </tr> </table> <script> 'use strict'; // Rows and cells for hiding const table1 = document.getElementById('id1'); table1.addEventListener('click', function(event) { if(event.target.classList.contains('col')) { const cellIndex = event.target.cellIndex; /* alert('the column #' + cellIndex + ' is hiding');*/ const rows = table1.rows; for(var i=0, iMax=rows.length; i<iMax; i++) { rows[i].cells[cellIndex].classList.add('hide'); } } else if(event.target.classList.contains('row')) { const row = event.target.parentElement; /* alert('the row #' + row.rowIndex + ' is hiding');*/ row.classList.add('hide'); } else { /* alert('Pas là, malheureux!');*/ return; } event.preventDefault(); }); // Buttons document.getElementById('toolbar').addEventListener('click', function(event) { if(event.target.hasAttribute('data-css')) { const els = table1.querySelectorAll(event.target.getAttribute('data- css')); if(els != null) { for(var i=0, iMax=els.length; i<iMax; i++) { els[i].classList.remove('hide'); } } } }); </script> </body> </html> </pre>	<div style="border: 1px solid black; border-radius: 10px; padding: 5px; display: inline-block;">JavaScript</div>
---	--

11.8.8. Création d'une arborescence (Tree view) de lignes

Hiérarchie d'éléments et de sous-éléments inspiré de :
https://www.w3schools.com/howto/howto_js_treeview.asp

<pre> <!DOCTYPE html> <html> <head> <meta name="viewport" content="width=device-width, initial-scale=1"> <style> ul, #myUL { list-style-type: none; } #myUL { margin: 0; padding: 0; } </pre>	
---	--

```

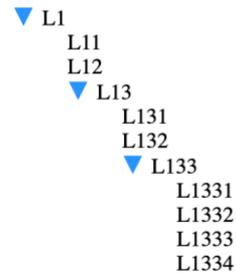
.box {
  cursor: pointer;
  -webkit-user-select: none; /* Safari 3.1+ */
  -moz-user-select: none; /* Firefox 2+ */
  -ms-user-select: none; /* IE 10+ */
  user-select: none;
}
.box::before {
  content: "\0025BA"; /*puce-triangle vers la droite*/
  color: black;
  display: inline-block;
  margin-right: 6px;
}
.check-box::before {
  content: "\0025BC"; /*puce-triangle vers le bas (ou transform : rotate(90deg)*/
  color: dodgerblue;
}
.nested {
  display: none;
}
.active {
  display: block;
}
</style>
</head>
<body>
<h2>Arborecence de lignes</h2>
<p>Certains éléments peuvent avoir des sous éléments</p>
<p>En cliquant sur la boite on ouvre ou ferme la branche</p>
<ul id="myUL">
  <li><span class="box">L1</span>
    <ul class="nested">
      <li>L11</li>
      <li>L12</li>
      <li><span class="box">L13</span>
        <ul class="nested">
          <li>L131</li>
          <li>L132</li>
          <li><span class="box">L133</span>
            <ul class="nested">
              <li>L1331</li>
              <li>L1332</li>
              <li>L1333</li>
              <li>L1334</li>
            </ul>
          </li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
</li>
</ul>
</li>
</ul>
</script>
var toggler = document.getElementsByClassName("box");
var i;
for (i = 0; i < toggler.length; i++) {
  toggler[i].addEventListener("click", function() {
    this.parentElement.querySelector(".nested").classList.toggle("active");
    this.classList.toggle("check-box");
  });
}
</script>
</body>
</html>

```

Arborecence de lignes

Certains éléments peuvent avoir des sous éléments

En cliquant sur la boite on ouvre ou ferme la branche



11.8.9. Tableau avec arborescence de lignes et de colonnes

Les lignes et les colonnes s'emboitent et se déplient, code inspiré de ChatGPT...

```

<!DOCTYPE html>
<html>
<head>
<title>Tableau interactif</title>
<style>
table {
border-collapse: collapse;
}
th, td {
border: 1px solid black;
padding: 5px;
text-align: center;
}
.hidden {
display: none;
}
</style>
</head>
<body>
<table>
<thead>
<tr>
<th>C0</th>
<th>C1<span id="toggle-c1" class="toggle">&#9658;</span></th>
<th class="toggle-c1">C2<span id="toggle-c2"
class="toggle">&#9658;</span></th>
<th class="toggle-c2">C3<span id="toggle-c3"
class="toggle">&#9658;</span></th>
<th class="toggle-c3">C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1<span id="toggle-l1" class="toggle">&#9660;</span></td>
<td>L1C1</td>
<td class="toggle-c1">L1C2</td>
<td class="toggle-c2">L1C3</td>
<td class="toggle-c3">L1C4</td>
</tr>
<tr class="toggle-l1">
<td class="toggle-l1">L11</td>
<td class="toggle-l1">L11C1</td>
<td class="toggle-l1 toggle-c1">L11C2</td>
<td class="toggle-l1 toggle-c2">L11C3</td>
<td class="toggle-l1 toggle-c3">L11C4</td>
</tr>
<tr>
<td>L2</td>
<td>L2C1</td>
<td class="toggle-c1">L2C2</td>
<td class="toggle-c2">L2C3</td>
<td class="toggle-c3">L2C4</td>
</tr>
</tbody>
</table>
<script>
const toggleC1 = document.getElementById('toggle-c1');
const toggleC2 = document.getElementById('toggle-c2');
const toggleC3 = document.getElementById('toggle-c3');
const toggleL1 = document.getElementById('toggle-l1');
const toggleableColumnsC1 = document.querySelectorAll('.toggle-c1');
const toggleableColumnsC2 = document.querySelectorAll('.toggle-c2');

```

C0	C1▶	C2▶	C3▶	C4
L1▼	L1C1	L1C2	L1C3	L1C4
L11	L11C1	L11C2	L11C3	L11C4
L2	L2C1	L2C2	L2C3	L2C4

C0	C1▼	C3▼
L1▼	L1C1	L1C3
L11	L11C1	L11C3
L2	L2C1	L2C3

Clic sur la puce de L1 : toute la ligne L11 se cache

clic sur la puce de la colonne C1 : supprime L11C2

clic sur la puce de la colonne C2 : supprime L11C3

clic sur la puce de la colonne C3 : supprime L11C4

la constante toggle-c1 déclarée stocke la référence à l'élément ayant pour identifiant toggle-c1 afin de modifier son apparence

Toggeable : l'élément colonne C1 peut être activé ou désactivé en cliquant dessus pour changer son apparence

```

const toggleableColumnsC3 = document.querySelectorAll('.toggle-c3');
const toggleableRowsL1 = document.querySelectorAll('.toggle-l1');
toggleC1.addEventListener('click', () => {
  toggleableColumnsC1.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC1.textContent = toggleC1.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});
toggleC2.addEventListener('click', () => {
  toggleableColumnsC2.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC2.textContent = toggleC2.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});
toggleC3.addEventListener('click', () => {
  toggleableColumnsC3.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC3.textContent = toggleC3.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});
toggleL1.addEventListener('click', () => {
  toggleableRowsL1.forEach(row => {
    row.classList.toggle('hidden');
  });
  toggleL1.textContent = toggleL1.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});
</script>
</body>
</html>

```

Exemple de tableau plus complet

```

<!DOCTYPE html>
<html>
<head>
  <title>Tableau interactif</title>
  <style>
    table, th, td {
      border-collapse: collapse;
      padding: 5px;
      text-align: center;
    }

    body { font-family: 'Noto Sans', Ubuntu, Arial; background: #666; }
    table { border-collapse: collapse; border: 1px solid #444; background:
#fff; }

    tr:nth-of-type(2n) { background: #edefa1; }
    th:hover { background: red; }
    tr:hover { background: yellow; color: maroon; }
    th { cursor: pointer; }
    td, th { text-align: center; padding: 0.3rem 0.5rem; }
    td { border: 1px solid blue; }

    .hidden {
      display: none;
    }
  </style>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>C0<span id="toggle-c1" class="toggle">&#9658;</span></th>
        <th class="toggle-c1">C1<span id="toggle-c2"
class="toggle">&#9658;</span></th>

```

C0▶	C1▶	C11	C12	C2▶	C21	C22	C3▶	C4
L1▼	L1C1	L1C11	L1C12	L1C2	L1C21	L1C22	L1C3	L1C4
L11	L11C1	L11C11	L11C12	L11C2	L11C21	L11C22	L11C3	L11C4
L12▼	L12C1	L12C11	L12C12	L12C2	L12C21	L12C22	L12C3	L12C4
L121	L121C1	L121C11	L121C12	L121C2	L121C21	L121C22	L121C3	L121C4
L122	L122C1	L122C11	L122C12	L122C2	L122C21	L122C22	L122C3	L122C4
L2▼	L2C1	L2C11	L2C12	L2C2	L2C21	L2C22	L2C3	L2C4
L21	L21C1	L21C11	L21C12	L21C2	L21C21	L21C22	L21C3	L21C4
L22	L22C1	L22C11	L22C12	L22C2	L22C21	L22C22	L22C3	L22C4

C0▶	C1▶	C11	C12	C2▼	C3▼
L1▶	L1C1	L1C11	L1C12	L1C2	L1C3
L2▶	L2C1	L2C11	L2C12	L2C2	L2C3

C0▶	C1▼	C3▼
L1▶	L1C1	L1C3
L2▶	L2C1	L2C3

```

    <th class="toggle-c1 toggle-c2">C11<span id="toggle-c2"
class="toggle"></span></th>
    <th class="toggle-c1 toggle-c2">C12<span id="toggle-c2"
class="toggle"></span></th>
    <th class="toggle-c2">C2<span id="toggle-c3"
class="toggle">&#9658;</span></th>
    <th class="toggle-c2 toggle-c3">C21<span id="toggle-c4"
class="toggle"></span></th>
    <th class="toggle-c2 toggle-c3">C22<span id="toggle-c4"
class="toggle"></span></th>
    <th class="toggle-c4">C3<span id="toggle-c5"
class="toggle">&#9658;</span></th>
    <th class="toggle-c5">C4<span id="toggle-c6"
class="toggle"></span></th>
</tr>
</thead>
<tbody>
<tr>
<td>L1<span id="toggle-l1" class="toggle">&#9660;</span></td>
<td class="toggle-c1">L1C1</td>
<td class="toggle-c1 toggle-c2">L1C11</td>
<td class="toggle-c1 toggle-c2">L1C12</td>
<td class="toggle-c2">L1C2</td>
<td class="toggle-c2 toggle-c3">L1C21</td>
<td class="toggle-c2 toggle-c3">L1C22</td>
<td class="toggle-c4">L1C3<span id="toggle-c7"
class="toggle"></span></td>
<td class="toggle-c5">L1C4</td>
</tr>
<tr class="toggle-l1">
<td class="toggle-l1">L11</td>
<td class="toggle-l1 toggle-c1">L11C1</td>
<td class="toggle-l1 toggle-c1 toggle-c2">L11C11</td>
<td class="toggle-l1 toggle-c1 toggle-c2">L11C12</td>
<td class="toggle-l1 toggle-c2">L11C2</td>
<td class="toggle-l1 toggle-c2 toggle-c3">L11C21<span id="toggle-c21"
class="toggle"></span></td>
<td class="toggle-l1 toggle-c2 toggle-c3">L11C22<span id="toggle-c22"
class="toggle"></span></td>
<td class="toggle-l1 toggle-c4">L11C3<span id="toggle-c8"
class="toggle"></span></td>
<td class="toggle-l1 toggle-c5">L11C4</td>
</tr>
<tr class="toggle-l1">
<td class="toggle-l1">L12<span id="toggle-l12"
class="toggle">&#9660;</span></td>
<td class="toggle-l1 toggle-c1">L12C1</td>
<td class="toggle-l1 toggle-c1 toggle-c2">L12C11</td>
<td class="toggle-l1 toggle-c1 toggle-c2">L12C12</td>
<td class="toggle-l1 toggle-c2">L12C2</td>
<td class="toggle-l1 toggle-c2 toggle-c3">L12C21<span id="toggle-c21"
class="toggle"></span></td>
<td class="toggle-l1 toggle-c2 toggle-c3">L12C22<span id="toggle-c22"
class="toggle"></span></td>
<td class="toggle-l1 toggle-c4">L12C3<span id="toggle-c8"
class="toggle"></span></td>
<td class="toggle-l1 toggle-c5">L12C4</td>
</tr>
<tr class="toggle-l12">
<td class="toggle-l12">L121</td>
<td class="toggle-l12 toggle-c1">L121C1</td>
<td class="toggle-l12 toggle-c1 toggle-c2">L121C11</td>
<td class="toggle-l12 toggle-c1 toggle-c2">L121C12</td>
<td class="toggle-l12 toggle-c2">L121C2</td>
<td class="toggle-l12 toggle-c2 toggle-c3">L121C21<span id="toggle-
c21" class="toggle"></span></td>

```

```

<td class="toggle-l12 toggle-c2 toggle-c3">L121C22<span id="toggle-
c22" class="toggle"></span></td>
<td class="toggle-l12 toggle-c4">L121C3<span id="toggle-c8"
class="toggle"></span></td>
<td class="toggle-l12 toggle-c5">L121C4</td>
</tr>
<tr class="toggle-l12">
<td class="toggle-l12">L122</td>
<td class="toggle-l12 toggle-c1">L122C1</td>
<td class="toggle-l12 toggle-c1 toggle-c2">L122C11</td>
<td class="toggle-l12 toggle-c1 toggle-c2">L122C12</td>
<td class="toggle-l12 toggle-c2">L122C2</td>
<td class="toggle-l12 toggle-c2 toggle-c3">L122C21<span id="toggle-
c21" class="toggle"></span></td>
<td class="toggle-l12 toggle-c2 toggle-c3">L122C22<span id="toggle-
c22" class="toggle"></span></td>
<td class="toggle-l12 toggle-c4">L122C3<span id="toggle-c8"
class="toggle"></span></td>
<td class="toggle-l12 toggle-c5">L122C4</td>
</tr>
<tr>
<td>L2<span id="toggle-l2" class="toggle">&#9660;</span></td>
<td class="toggle-c1">L2C1</td>
<td class="toggle-c1 toggle-c2">L2C11</td>
<td class="toggle-c1 toggle-c2">L2C12</td>
<td class="toggle-c2">L2C2</td>
<td class="toggle-c2 toggle-c3">L2C21<span id="toggle-c21"
class="toggle"></span></td>
<td class="toggle-c2 toggle-c3">L2C22<span id="toggle-c22"
class="toggle"></span></td>
<td class="toggle-c4">L2C3<span id="toggle-c9"
class="toggle"></span></td>
<td class="toggle-c5">L2C4</td>
</tr>
<tr class="toggle-l2">
<td class="toggle-l2">L21</td>
<td class="toggle-l2 toggle-c1">L21C1</td>
<td class="toggle-l2 toggle-c1 toggle-c2">L21C11</td>
<td class="toggle-l2 toggle-c1 toggle-c2">L21C12</td>
<td class="toggle-l2 toggle-c2">L21C2</td>
<td class="toggle-l2 toggle-c2 toggle-c3">L21C21<span id="toggle-c21"
class="toggle"></span></td>
<td class="toggle-l2 toggle-c2 toggle-c3">L21C22<span id="toggle-c22"
class="toggle"></span></td>
<td class="toggle-l2 toggle-c4">L21C3<span id="toggle-c9"
class="toggle"></span></td>
<td class="toggle-l2 toggle-c5">L21C4</td>
</tr>
<tr class="toggle-l2">
<td class="toggle-l2">L22</td>
<td class="toggle-l2 toggle-c1">L22C1</td>
<td class="toggle-l2 toggle-c1 toggle-c2">L22C11</td>
<td class="toggle-l2 toggle-c1 toggle-c2">L22C12</td>
<td class="toggle-l2 toggle-c2">L22C2</td>
<td class="toggle-l2 toggle-c2 toggle-c3">L22C21<span id="toggle-c21"
class="toggle"></span></td>
<td class="toggle-l2 toggle-c2 toggle-c3">L22C22<span id="toggle-c22"
class="toggle"></span></td>
<td class="toggle-l2 toggle-c4">L22C3<span id="toggle-c9"
class="toggle"></span></td>
<td class="toggle-l2 toggle-c5">L22C4</td>
</tr>
</tbody>
</table>

<script>
const toggleC1 = document.getElementById('toggle-c1');
```

```

const toggleC2 = document.getElementById('toggle-c2');
const toggleC3 = document.getElementById('toggle-c3');
const toggleC4 = document.getElementById('toggle-c4');
const toggleC21 = document.getElementById('toggle-c21');
const toggleC22 = document.getElementById('toggle-c22');
const toggleC5 = document.getElementById('toggle-c5');
const toggleC6 = document.getElementById('toggle-c6');
const toggleC7 = document.getElementById('toggle-c7');
const toggleC8 = document.getElementById('toggle-c8');
const toggleC9 = document.getElementById('toggle-c9');
const toggleL1 = document.getElementById('toggle-l1');
const toggleL12 = document.getElementById('toggle-l12');
const toggleL2 = document.getElementById('toggle-l2');

const toggleableColumnsC1 = document.querySelectorAll('.toggle-c1');
const toggleableColumnsC2 = document.querySelectorAll('.toggle-c2');
const toggleableColumnsC3 = document.querySelectorAll('.toggle-c3');
const toggleableColumnsC4 = document.querySelectorAll('.toggle-c4');
const toggleableColumnsC5 = document.querySelectorAll('.toggle-c5');
const toggleableColumnsC6 = document.querySelectorAll('.toggle-c6');
const toggleableColumnsC7 = document.querySelectorAll('.toggle-c7');
const toggleableColumnsC8 = document.querySelectorAll('.toggle-c8');
const toggleableColumnsC9 = document.querySelectorAll('.toggle-c9');

const toggleableRowsL1 = document.querySelectorAll('.toggle-l1');
const toggleableRowsL12 = document.querySelectorAll('.toggle-l12');
const toggleableRowsL2 = document.querySelectorAll('.toggle-l2');

toggleC1.addEventListener('click', () => {
  toggleableColumnsC1.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC1.textContent = toggleC1.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleC2.addEventListener('click', () => {
  toggleableColumnsC2.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC2.textContent = toggleC2.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleC3.addEventListener('click', () => {
  toggleableColumnsC3.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC3.textContent = toggleC3.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleC4.addEventListener('click', () => {
  toggleableColumnsC4.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC4.textContent = toggleC4.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleC5.addEventListener('click', () => {
  toggleableColumnsC5.forEach(column => {
    column.classList.toggle('hidden');
  });
  toggleC5.textContent = toggleC5.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

```

```
toggleC6.addEventListener('click', () => {
  toggleC6.parentNode.classList.toggle('hidden');
  toggleC6.textContent = toggleC6.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleC7.addEventListener('click', () => {
  toggleC7.parentNode.classList.toggle('hidden');
  toggleC7.textContent = toggleC7.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleC8.addEventListener('click', () => {
  toggleC8.parentNode.classList.toggle('hidden');
  toggleC8.textContent = toggleC8.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

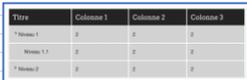
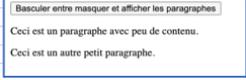
toggleC9.addEventListener('click', () => {
  toggleC9.parentNode.classList.toggle('hidden');
  toggleC9.textContent = toggleC9.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleL1.addEventListener('click', () => {
  toggleableRowsL1.forEach(row => {
    row.classList.toggle('hidden');
  });
  toggleL1.textContent = toggleL1.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleL12.addEventListener('click', () => {
  toggleableRowsL12.forEach(row => {
    row.classList.toggle('hidden');
  });
  toggleL12.textContent = toggleL12.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});

toggleL2.addEventListener('click', () => {
  toggleableRowsL2.forEach(row => {
    row.classList.toggle('hidden');
  });
  toggleL2.textContent = toggleL2.textContent === '\u25BC' ? '\u25B6' :
'\u25BC';
});
</script>
</body>
</html>
```

11.9. Comparaison des tables sous jQuery

Table arborescente à partir d'une table HTML	Masquer ou afficher éléments	Bascule masquer/afficher	Élément glissant alternativement
<p>http://gajiks.github.io/jquery-tbltree/index.html</p> <p>Programme</p> <p>souhaité</p> 	<p>https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_show</p> <p>Programme</p> <p>signalisation</p> <p>obtenue</p> <p>Fonctionne très bien...</p> 	<p>https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_toggle</p> <p>Programme</p> <p>signalisation</p> <p>Basculer entre masquer et afficher les paragraphes</p> <p>Ceci est un paragraphe avec peu de contenu.</p> <p>Ceci est un autre petit paragraphe.</p> 	<p>https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_sl</p> <p>Programme</p> <p>signalisation</p> <p>Élément glissant alternativement</p> <p>Cliquez pour faire glisser le panneau vers le bas ou vers le haut</p> <p>Bonjour le monde!</p> 
<pre><!DOCTYPE html> <html> <Head> <script src = "http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.mi n.js" > </script> <script type = "text / JavaScript" src = "http: //ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery- ui.min.js " > </script> <script type = "text / javascript" src = ". /js/jquery.tbltree.js" > </script> <link type = "text / css" href = "css / jquery.tbltree.css" rel = " feuille de style " > <script type = "text / javascript" src = ". /js/jquery.cookie.js" > </script> <script> \$(function () { // initialiser avec les options par défaut \$("# table1").tbltree (); }); </script> <table id = "table1" > <tr></pre> <p><i>inclusion de jQuery dans la page</i></p> <p><i>inclusion de jQuery-ui dans la page</i></p> <p><i>inclusion du plugin dans la page</i></p> <p><i>inclusion du plugin dans la page</i></p> <p><i>code JavaScript</i></p> <p><i>constitution de l'arbre à partir de la création du tableau HTML n ajoutant les attributs 'row-id' et 'parent-id' aux balises 'tr'.</i></p>	<pre><!DOCTYPE html> <html> <Head> <script src="https://ajax.googleapis.com/ajax/libs/jquery /3.5.1/jquery.min.js"></script> <script> \$(document).ready(function){ \$("#hide").click(function){ \$("#p").hide(); }); \$("#show").click(function){ \$("#p").show(); }); }; </script> </head> <body></pre> <p><i>masque l'élément html</i></p> <p><i>affiche l'élément html</i></p>	<pre><!DOCTYPE html> <html> <Head> <script src="https://ajax.googleapis.com/ajax/libs /jquery/3.5.1/jquery.min.js"></script> <script> \$(document).ready(function){ \$("#button").click(function){ au click du bouton bascule de l'élément \$("#p").toggle(); }); \$("#panel").slideToggle("slow"); }); }; </script> <style> #panel, #flip { padding: 5px;</pre> <p><i>au click du bouton</i></p> <p><i>bascule de l'élément</i></p>	<pre><!DOCTYPE html> <html> <Head> <script src="https://ajax.googleapis.com/ajax/lib s/jquery/3.5.1/jquery.min.js"></script> <script> \$(document).ready(function){ \$("#flip").click(function){ \$("#panel").slideToggle("slow"); }); }; </script> <style> #panel, #flip { padding: 5px;</pre>

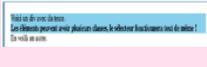
<pre><th> Titre </th> <th> Colonne 1 </th> <th> Colonne 2 </th> <th> Colonne 3 </th> </tr></pre>		<pre><p>If you click on the "Hide" button, I will disappear.</p></pre>	<pre></head> <body></pre>			<pre>text-align: center; background-color: #e5eccc;</pre>	
<pre><tr row-id = "1" > <td> Niveau 1 </td> <td class = "data" > 2 </td> </tr></pre>	<p><i>Nom d'attribut qui est défini pour les <tr> balises et identifie l'ID de la ligne.</i></p>	<pre><button id="hide">Hide</button> <button id="show">Show</button></pre>		<pre><button>Toggle between hiding and showing the paragraphs</button></pre>	<p>inscription sur le bouton</p>	<pre>} border: solid 1px #c3c3c3;</pre>	
<pre><tr row-id = "1.1" parent-id = "1" > <td> Niveau 1.1 </td> <td class = "data" > 2 </td> </tr> <tr row-id = "2" > <td> Niveau 2 </td> <td class = "data" > 2 </td> </tr> <tr row-id = "2.1" parent-id = "2" > <td> Niveau 2.1 </td> <td class = "data" > 2 </td> </tr></pre>	<p><i>Nom d'attribut qui est défini pour les <tr> balises et identifie l'ID de la ligne parent. Remarque: cet attribut doit être ignoré pour les nœuds racine.</i></p>	<pre></body> </html></pre>		<pre><p>This is a paragraph with little content.</p> <p>This is another small paragraph.</p></pre>	<p>inscription de la ligne 1 inscription de la ligne 2</p>	<pre>#panel { padding: 50px; display: none; }</pre>	
<pre></tr></pre>	<p><i>a faire sélection des lignes sur 2ème colonne</i></p>						
<pre></table> </head> </html></pre>						<pre><div id="flip">Click to slide the panel down or up</div> <div id="panel">Hello world!</div></pre>	
	<pre>\$("# table1").bitree ({ treeColumn: 1, saveState: vrai });</pre>					<pre></body> </html></pre>	

Autre tableau arborescent :

<https://homfen.github.io/dataTables.treeGrid.js/>

12. SYNTHÈSE DES COMMANDES

12.1. Résumé des commandes HTML

code	Baïse/commande	Attribut	valeurs	Description	Exemples	résultat
HTML						
Base du document						
	<!DOCTYPE>			permet de définir la version HTML utilisée		
	<html></html>			début du document HTML contient les éléments <head> et <body>		
	<head></head>			entête du document qui contient les méta-informations : <title><meta><link><style><script>		
	<body></body>			corps du document contient toutes les autres balises...		
	<title></title>			titre affiché dans la "barre de titre" du navigateur (se place entre les balises <head>)		
	<meta>			inclu dans les sections <head> donne des informations (méta-données) traitées par le serveur.		
		name		informations sur l'auteur, la date de publication	<meta name="author">	
		content		Donne une valeur à une propriété repérée par son nom	<meta name="author" content="Ed Tittel">	
		http-equiv		crée un lien entre l'élément et un entête de réponse HTML	<meta http-equiv="Copyright" content="Lanwright, inc. --1999">	
		scheme		Identifie le modèle employé pour interpréter les valeurs de propriété.	<meta http-equiv="Copyright" content="Lanwright, inc. --1999">	
		charset="valeur"	UTF-8	indique l'encodage utilisé		
	<style></style>			Permet de définir l'affichage d'une partie du document HTML		
		type		Spécifie le type de feuille de style utilisé	<type="text/css2">	
		média			h1 color:red;	
	<script></script>			contient le script de JavaScript ou fait référence à un fichier externe	<script src="script.js"></script>	
		type				
		src				
		defer		exécution différée à la fin du document		
		async		exécution asynchrone		
		integrity				
	</script></script>			lu lorsque la lecture des scripts est désactivée sur le navigateur		
	<base></base>			sert de base dans toute la section <body>		
		href="url"		définit l'url	<base href="http://lanw.com">	
		target="menu"		spécifie la fenêtre où les pages dsont appelées par <a>		
	<!-- abcd -->			commentaire entre balise non lu par le navigateur		
	<applet></applet>			pour les applets java		
structure du document						
	<main></main>			Contenu principal. Structure la section majeure qui en regroupe d'autres		
	<article></article>			Contenu autonome dans un document		
	<aside></aside>			aparté		
	<section></section>			élément générique		
	<nav></nav>			regroupe la sélection des principaux liens pour naviguer		
	<header></header>			en-tête: contenu introductif		
	<footer></footer>			pied de page		
	<address></address>			Coordonnées des auteurs	<address>Marc Emonet </address>	marc.emonet@wanadoo.fr
	<word-break></break>			définit la façon dont la césure s'applique pour les endroits où le texte dépasserait de sa boîte de contenu		
		normal		passage à la ligne classique		
		break-all		La césure peut être insérée après n'importe quel caractère		
		keep-all		dito normal		
		break-word		dito normal et overflow-wrap: break-word		
hiérarchisation						
	<h1></h1>			titre 1er niveau de grosseur		
	<h2></h2>			titre 2ème niveau de grosseur		
	...			et ainsi de suite		
	<h5></h5>			titre 5ème niveau	Titre de niveau 5	Titre de niveau 5
	<h6></h6>			titre 6ème niveau (de base)	<h6> Titre de niveau 6</h6>	Titre de niveau 6
		align	"left"; "center"; "right"; justify"	positionnement de titre		
	<p></p>			paragraphe avec 2 retours à la ligne	<p>Ceci est un simple paragraphe</p>	Ceci est un simple paragraphe
	 			mise à la ligne du texte (nb: pas de fin de balise)		
	<wbr>			Césure (coupure) de mot lorsque la ligne est trop longue		
	<hr></hr>			Trace un trait horizontal		
		align	left; "center"; "right"	positionne le trait dans la page	<hr align="left">	
		noshade		supprime l'ombrage du trait	<hr noshade>	
		size		définit l'épaisseur du trait	<hr size="10">	
		width="valeur"	pixel ou %	définit la largeur du trait en pixel ou % de la fenêtre	<hr width="50%">	
conteneurs génériques						
	<div></div>			division logique de secteurs pouvant avoir des styles différents		
		align="x"	x:center; right; left; justify	alignement du paragraphe	<div align="left">	
		style="valeur"			<div style="color:navy">	
		class="x"	valeurs déclarées dans CSS		CSS: classy {background-color: skyblue}; toto {font-weight: bold;} <div class="classy">Voici un div avec du texte</div> <div class="toto classy truc">Les éléments peuvent avoir plusieurs classes, le sélecteur fonctionnera tout de même !</div></div>En voilà un autre.</div>	
				conteneur de type en ligne		
		style		applique des informations de style à n'importe quel fragment de code HTML contenu dans le document	<h2>Sans ce titre le mot couleur est en rouge</h2>	Sans ce titre le mot couleur est en rouge
Statut du contenu						
	<ins></ins>			pour surligner du texte (stably)		
				suppression d'un texte ancien qui est barré		
	<strike></strike> ou 			pour barrer du texte obsolète	<p>un texte peu avoir un élément 	un texte peu avoir un élément
Blocs sémantiques						
	<blockquote></blockquote>		cite="texte"	bloc de citation	<blockquote> Tout corps plongé dans l'eau subit une poussée de bas en haut...</blockquote>	Tout corps plongé dans l'eau subit une poussée de bas en haut...
	<pre></pre>			texte préformaté : ce qui est mis à l'intérieur de ces balises respecte la présentation	<h2>annuaire</h2> <pre>Nom telephone Jean 010204</pre>	annuaire Nom telephone Jean 010204
	<code></code>			Texte provenant d'instruction d'un programme informatique (caractère à pas fixe)	<p>la commande<code>SORTIE</code>met en exergue</p>	la commande SORTIE met en exergue
	<kbd></kbd>			Texte devant être saisi au clavier (caractère à pas fixe)	<kbd>IPCONFIG</kbd>	IPCONFIG
	<var></var>			Représente une variable destinée à l'ordinateur	<var>adresse IP</var>	adresse IP
	<samp></samp>			Définit la sortie à pas fixe des chaînes de caractères d'un programme ou d'un script	<samp>Afrique Asie</samp>	Afrique Asie
		src="url"		source de l'image		

	<code>name="texte"</code>		étiquette de la balise		
	<code>alt="texte"</code>		texte de remplacement si l'image ne s'affiche pas		
	<code>align="x"</code>	x=bottom, middle, top; left, right	position de l'image par rapport au cadre		
	<code>border="pixels"</code>	de 0 à 5	largeur de la bordure de l'image		
	<code>hspace="pixels"</code>		largeur du vide autour de l'image		
	<code>vspace="pixels"</code>		hauteur du vide autour de l'image		
	<code>width="pixels"</code>		longueur de l'image		
	<code>height="pixels"</code>		hauteur de l'image		
balises en ligne sémantiques					
	<code><time></time></code>		définit une date et une heure		
		<code>datetime</code>		<code><ns datetime="2001-11-03T19:22:00+1:00">texte inséré le 03 novembre 2001 à 19h22 GMT+1</ns></code>	texte inséré le 03 novembre 2001 à 19h22 GMT+1
	<code><acronym></acronym></code>		Acronyme	<code><acronym>SIDA</acronym></code> (Syndrome d'ImmunoDéficiency Acquis)	SIDA (Syndrome d'ImmunoDéficiency Acquis)
	<code><abbr></abbr></code>		abréviation	<code><abbr>CA</abbr></code> (California)	CA (California)
	<code><cite></cite></code>		mention d'une œuvre	<code><cite>Coder en HTML 4</cite></code> par <code><cite>Stéphane Chalamet</cite></code>	Coder en HTML 4 par Stéphane Chalamet
	<code><q></q></code>		Marquage d'une citation, ils seront placés entre guillemets		
	<code><dfn></dfn></code>		repère les termes d'un document HTML et le met en italique	<code><p><dfn>chat</dfn></code> est félin</p>	un <i>chat</i> est un félin
mise en forme du texte					
	<code><mark></mark></code>		insertion dans un texte ancien sera surligné		
	<code><u></u></code>		pour souligner du texte		
	<code></code>		mise en gras du texte	<code><p>salut les copains</p></code>	salut les copains!
	<code><i></i></code>		mise en italique du texte	<code><p>Texte mis en italique</p></code>	Texte mis en <i>italique</i>
	<code></code>		forte insistance avec mise en gras	<code><p>utilisez le BP q'en cas de danger</code>	utilisez le BP q'en cas de danger
	<code></code>		met en <i>exergue</i> et en gras	<code><p>pour ouvrir cette porte poussez le BP</p></code>	pour ouvrir cette porte <i>poussez</i> le BP
	<code><small></small></code>		mise en petit caractères	<code><p>Texte <small>plus petit</small></p></code>	Texte plus petit
	<code><big></big></code>		affichage plus gros du texte	<code><p>Texte <big>plus gros</big></p></code>	Texte plus gros
	<code><sup></sup></code>		met en ^{x²}	Le mètre carré s'écrit M²	Le mètre carré s'écrit M ²
	<code><sub></sub></code>		met en _{x₀}	La formule chimique de l'eau est H₂_O	La formule chimique de l'eau est H ₂ O
	<code><bdo></bdo></code>		Langue principale utilisée dans le texte et sens de l'affichage		
		<code>dir</code>	LTR(left to right; RTL(right to left)	<code><bdo dir="RTL">4321</bdo></code> (->4321)	4321
		<code>lang</code>	Langue principale utilisée dans le texte		
	<code><bdi></bdi></code>		Isolément du sens de l'écriture		
	<code></code>		définit les polices de caractères compris entre les balises		
		<code>color="x"</code>	x=red, green,...	<code></code>	
		<code>face="x"</code>	x=Time Roman; Arial,...	<code></code>	
		<code>size="x"</code>	x= 1 à 7 (à 3 par défaut)	<code></code> (soit 3-2=1)	
	<code><basefont></basefont></code>		Police de caractères utilisé par défaut dans tout le texte d'un document		
		<code>color="x"</code>	x=red, green, ou #FF0000	<code><basefont color="red"></code>	
		<code>face="x"</code>	x=Time Roman; Arial,...	<code><basefont face="Arial"></code>	
		<code>size="x"</code>	x= 1 à 7 (à 3 par défaut)	<code><basefont size="2"></code> (soit 3-2=1)	
	<code><center></center></code>		Provoque un retour à la ligne et un centrage du texte compris entre les balises	<code><p>ce texte est mis <center>au milieu</center>mais pas le reste</p></code>	
	<code></code>		lien hypertexte avec une autre page HTML		
	<code><tt></tt></code>		Affiche le texte avec une police à pas fixe de type courrier	<code><p>Ce texte a des <tt>lettres à largeur fixe</tt></p></code>	Ce texte a des lettres à largeur fixe
	<code><blink></blink></code>		caractères clignotants		
listes					
	<code><dl></dl></code>		Liste des définitions avec indentation		
	<code><dt></dt></code>		Titre de définition		
	<code><dd></dd></code>		contenu de la définition de description	<code><h2>titres</h2><dt><dt>chat<dt>mammifères,<dt>poule<dt>oiseaux</dt></code>	titre chat mammifère...
	<code></code>		liste à puces	<code> 1er élément 2ème élément 3ème élément</code>	- 1er élément - 2ème élément - 3ème élément
	<code></code>		liste numérotée	<code><ol start="4"><li type="A">chat<li type="1">souris</code>	0 chat 5 souris
	<code></code>		Valeur initiale de la sérialisation pour 		
			contient les éléments d'une liste		
		<code>type="valeur"</code>	DISC, SQUARE, CIRCLE	respectivement: gros point noir; petit carré plein; petit cercle creux, pour 	
		<code>type="valeur"</code>	1; a; A; i; l	respectivement : chiffres Arabes; Lettres minuscules; majuscules; Romains minuscules; majuscules pour 	0 chat 5 souris
		<code>value="nombre"</code>	Valeur initiale de la sérialisation pour 		
Multimédia					
	<code><figure></figure></code>		correspond au paragraphe <p> dans lequel on place des images.		
	<code><figcaption></figcaption></code>		légende de figure		
	<code><details></details></code>	open	Informations ou contrôles additionnels		
	<code><summary></summary></code>		Résumé, légende ou intitulé		
	<code></code>		remplacé par 		
			insertion d'une image située dans un dossier	<code></code>	
		<code>src="adresse"</code>	indique le chemin d'accès à la photo	<code><p></code> Ce texte est situé à droite de l'image. Ce texte est situé à la ligne, à droite de l'image. <code>clear="left" /></code>	Ce texte est situé à droite de l'image. est situé à la ligne, à droite de l'image
		<code>alt="texte"</code>	le texte s'affiche si la photo n'apparaît pas		
		<code>style="float"</code>	positionne l'image dans un texte	Ex : <code></code>	
		<code>title="texte"</code>	ajoute une info bulle		
	<code><map></map></code>		carte cliquable pour image		
	<code><picture></picture></code>		image à sources multiples		
	<code><area></area></code>		zone cliquable		
	<code><audio></audio></code>		Son ou élément audio		
	<code><video></video></code>		vidéo		
	<code><source></source></code>		Source alternative audio, image ou vidéo		
	<code><track></track></code>		Piste de texte		
Autres zones					
	<code><canvas></canvas></code>		Zone de dessin 2D au 3D		
	<code><object></object></code>		Permet de faire appel à un objet externe		
	<code><param></param></code>		Définit les paramètres d'exécution		
	<code><embed></embed></code>		Intégration de contenus externe		
	<code><template></template></code>		Modèle de partition de documents		

<dialog></dialog>			Bloc de dialogue utilisateur		
<frameset></frameset>			structure d'un document comportant des cadres (remplace <body>)		
	rows="valeur"	n, %, *	n= hauteur en pixels, %=hauteur dépendant de la fenêtre, *=correspond à ce qui reste		
	cols		définit le nombre et les largeurs des cadres verticaux de l'espace		
	bordercolor		définit la couleur des bordures		
	frameborder="x"	x= yes ou no	existence ou non d'une bordure autour du cadre		
<frame>			les cadres sont utilisés pour la division d'une page web en plusieurs sous-cadres	<frame src="mapage.htm" name="moncadre">	
	src="url"		image		
	name="nom"		définit le nom du cadre		
	id="nom"		définit l'identificateur unique au sein du document		
	marginwidth="valeur"		Largeur des marges latérales entre deux cadres		
	marginheight="valeur"		Largeur des marges verticales entre deux cadres		
	noresize		empêche la modification des cadres		
	scrolling="x"	x= yes; no; auto	affichage ou non d'une barre de défilement		
</frame></iframe>			Cadre de navigation imbriquée		
<noframe></noframe>			affichage d'un message pour les navigateurs ne reconnaissant pas les cadres	<noframe>body>votre navigateur ne reconnait pas les cadres</body></noframe>	votre navigateur ne reconnait pas les cadres
Tableau					
<table></table>			tableau		
	align="x"	x=left, center, right	position		
	border="nb"	de 0 à 10 pixels	épaisseur de bordure (0 par défaut)		
	cellpadding="nb"	de 0 à 20	espace entre contenu et la bordure (1 par défaut)	<table border cellpadding="14">	
	cellspacing="nb"	de 0 à 20 pixels	espace entre 2 cellules proches	<table border cellspacing="10">	
	width="valeur"	pixel ou %	définit la largeur du trait en valeur pixel ou relative en % de la fenêtre	<table border width="90%">	
	valign="x"	x=top, middle, bottom	position verticale haut, milieu, bas du contenu dans la cellule		
	nowrap		empêche le retour à la ligne du contenu d'une cellule		
	colspan="nb"		précise le nombre de colonnes la cellule doit utiliser		
	rowspan="nb"		précise le nombre de lignes la cellule doit utiliser		
	bgcolor="x"	x=red, green, ... ou #FF0000	suivant la balise <table>, <tr> ou <td> attribuée à tout le tableau, une ligne ou une colonne la couleur		
	frame	above; hsid; lhs; border; void; below; vsides; rhs; box	définit les côtés du tableau sur lesquels dessiner une bordure : au dessus, frontière, néant, dessous, boîte		
	rules	none; groups; rows; cols; all	définit la manière dont les traits doivent apparaître au sein du tableau : sans traits, les traits n'apparaissent que sur les groupes, affichés entre les lignes, entre les colonnes, partout		
	summary		définit un texte alternatif à utiliser afin de décrire le tableau		
<caption></caption>			titre du tableau (centré sur la largeur du tableau)	<caption align="top">c'est le titre du tableau</caption>	c'est le titre du tableau
<tr></tr>			ligne du tableau		
<th></th>			cellule de légende de ligne ou de colonne en gras	https://developer.mozilla.org/fr/docs/Web/HTML/Element/th	
	scope	row; col; rowgroup; colgroup; auto	l'entête s'applique respectivement à la ligne, la colonne, au groupe de ligne, de colonne...		
	headers		liste de chaîne de caractères séparés par des espaces		
<td></td>			cellule du tableau (colonne)		
<thead></thead>			ensemble de lignes définissant l'entête des colonnes voir : https://developer.mozilla.org/fr/docs/Web/HTML/Element/thead	<table><thead><tr><th>Col 1</th><th>Col 2</th></tr></thead><tbody><tr><td>Contenu 1</td><td>Contenu 2</td></tr></tbody></table>	
<tbody></tbody>			regroupe les lignes d'un tableau permettant un défilement horizontal en conservant l'entête et le pied		
<tfoot></tfoot>			ensemble de lignes résumant le pied des colonnes		
<colgroup></colgroup>			crée un groupe de colonnes à l'aide de span (uniformité de la largeur) ou de col		
<col>			précise la structure des colonnes		
Liens					
<a>			lien hypertexte	découvrir HTML5	
<link></link>			Relations externes, utilisé pour lier une feuille de style CSS au document HTML au moyen de la relation stylesheet		
formulaire					
<form></form>			formulaire		
	action="adresse"				
	method="x"	x=get;			
<fieldset></fieldset>			Regroupe les informations de la même thématique dans un formulaire		
<legend></legend>			Légende d'un regroupement de champs		
<label></label>			étiquette représentant une légende pour un objet d'une interface utilisateur; associé avec <input>	<label>Cliquez ici<input type="text" id="Utilisateur" name="Nom" /></label>	<input type="text" value=""/>
	input type="x"	x= color; date; month; file; url; email;			
	for			<label for="Utilisateur">Cliquez ici</label><input type="text" id="Utilisateur" name="Nom" />	<input type="text" value=""/>
	form				<input type="button" value="Envoyer"/>
<input>			élément de saisie	<input type="submit" value="Envoyer">	
	type="x"	x=checkbox; text; submit	respectivement pour case à cocher ou pour du texte (par défaut) ou pour un bouton poussoir qui acquiesce.	<label for="ptipois">Aimez-vous les petits-pois ?</label><input type="checkbox" name="petitspois" id="ptipois">	Aimez-vous les petits-pois ? <input type="checkbox"/>
	value="texte"		positionne du texte dans le formulaire	<input type="text" value="Saisir un texte ici">	<input type="text" value="Saisir un texte ici"/>
	name="nom"				
	id="..."				
<button></button>			Crée un bouton étiquetés dans un formulaire	<button>cliquez moi</button>	<input type="button" value="cliquez moi"/>
<textarea></textarea>			Zone de texte de plusieurs lignes et plusieurs colonnes		
<select></select>			Renferme une série d'options issues d'une liste déroulante		
<option></option>			Option de liste de choix		
<output></output>			Résultats d'une opération dans un formulaire		
<progress></progress>			Barre de progression correspondant au temps d'attente, manipulable avec JavaScript		
<meter></meter>			représente une jauge reflétant un état		
<datalist></datalist>			liste de suggestions		
CSS (valeur du style indiqué dans un fichier à part ou entre les balises <head> <style>div{...})					
border			bordures d'un tableau	https://developer.mozilla.org/fr/docs/Web/CSS/border	
border-collaps :	collaps	inset; outset	collées	border-collaps : collaps	
	separate		séparées (par défaut)		
boder-spacing :		0; 5px; 5px 1rem;	distance des bordures séparées: nulle, 5 pixels, + 1 rem vertical		

border-style :		none; dotted; dashed; solid; double; groove; ridge	sans, gros pointillés ronds, succession de tirets, ligne continue, lignes doubles, effet 3D en creux, en avant	CSS: b1{border-style:none} b2{border-style:dotted;} HTML: <table><tr><td class="b1">none</td><td class="b2">dotted</td></tr></table>	
border-width :		thick; 1em; 4px 1.25em; 4px 1.25em; 4px 1.25em	Fin, épais, ...		
border-color :		x:red; green;... ou #FF0000	couleur du trait		
background-color :		x:red; green;... ou #FF0000	couleur de l'arrière plan	background-color : red	
background-size :			définit la taille des images d'arrière-plan pour l'élément		
	contain		redimensionne l'image afin qu'elle soit la plus grande possible et que l'image conserve ses proportions		
	cover		L'image couvre toute la largeur ou la hauteur du conteneur et les parties qui dépassent sont rognées		
	auto		redimensionne l'image d'arrière-plan afin que ses proportions soient conservées		
	<length>		redimensionne l'image afin que celle-ci occupe la longueur indiquée dans la dimension concernée		
	<percentage>		redimensionne l'image d'arrière-plan en proportion de la taille de la zone dédiée à l'arrière-plan		
background-position :			permet de définir la position initiale, relative à l'origine		
	center		centre l'image		
	top; left; bottom; right		indique le bord par rapport auquel placer l'élément		
	length; percentage				
padding	-top; right; bottom; left		écart entre la partie utile et le cadre		
	padding: 1em		même espace aux 4 cotés	padding: 10px 50px 30px 0;	
	padding: 5% 10%		% sur vertical et horizontal		
	padding: 1em 2em 2em		valeur sur haut, horizontal et bas		
	padding: 5px 1em 0 2em		valeur sur haut, droit, bas, gauche		
display					
<display-outside>			définit le type d'affichage extérieur de l'élément. Autrement dit, ils définissent comment l'élément participe au flux		
	display: block;				
	display: inline;				
	display: run-in;				
<display-inside>			définit le type d'affichage intérieur de l'élément qui définit le contexte de formatage qui organisera le contenu de l'élément		
	display: flow;				
	display: table;				
	display: flex;				
	display: grid;				
<display-listitem>			génère une boîte de bloc pour le contenu et une boîte en ligne séparée pour l'élément de liste.		
<display-internal>					
<display-box>			peut générer une boîte		
.box(...)					
box-sizing :			La somme des différentes valeurs = dimension de l'ensemble		
	content;		texte ou image		
	padding:	valeur em ou px; %; inherit; initial; unset	espace autour du contenu : 1 valeur = même écart aux 4 cotés; 2 valeurs = vertical et horizontal; 3 = haut, droite et gauche, en bas; 4 = haut, droite, bas, gauche.	padding: 1em; padding: 5% 10%; padding: 5px 1em 0 2em;	
	border:		bordure		
	margin:		marge		
	content-box		largeur et hauteur du contenu utile (valeur par défaut)		
	border-box		dimension de l'élément dans son ensemble		
	inherit		dépend de la valeur précédente héritée		
<display-legacy>					
flex			est une propriété raccourcie qui définit la capacité d'un élément flexible à modifier ses dimensions afin de remplir l'espace disponible de son conteneur		
flex-direction			direction dans les éléments flexibles seront affichés		
	row		axe aligné en ligne de gauche à droite		
	row-reverse		axe aligné en ligne de droite à gauche		
	column		axe de haut en bas		
	column-reverse		axe de bas en haut		
flex-wrap	flex-wrap: wrap;		les éléments trop grands passent à la ligne suivante		
flex-flow	flex-flow: row wrap;		regroupe les flex-direction et flex-wrap	box (display: flex; flex-flow: row wrap;) <div class="box"><div>One</div><div>Two</div><div>Three</div></div>	
flex-grow			permet aux éléments flexibles de s'étendre		
flex-shrink			permet si l'espace est insuffisant de réduire l'élément	box (display: flex; flex-direction: row; flex-wrap: wrap;) <div class="box"><div class="one">One</div><div class="two">Two</div><div class="three">Three</div></div>	
flex-basis			définit la taille de l'élément en termes d'espace occupé		
align-items			permet d'aligner les éléments sur l'axe secondaire		
	stretch		avec les éléments flexibles étirés sur l'axe perpendiculaire	box (display: flex; align-items: flex-start;) <div class="box"><div>One</div><div>Two</div><div>Three</div></div>	
	flex-start		éléments alignés sur la ligne de début de l'axe secondaire		
	flex-end		éléments alignés sur la ligne de fin de l'axe secondaire		
	center		pour les aligner au centre		
justify-content			utilisée afin d'aligner les éléments le long de l'axe principal dans la direction définie par flex-direction		
	flex-start		place les éléments à partir de la ligne de début du conteneur sur l'axe principal		
	flex-end		les placer vers la fin et la valeur		
	center		permet de les centrer le long de l'axe principal.		
	space-around		espace est également réparti autour des éléments, y compris au début et à la fin		
	space-between		éparti l'espace disponible de façon égale entre chaque élément		
	space-evenly		l'espace est également réparti avec un espace entier au début et à la fin		
@font-face					

	font-family;				
	font-style:				
font-weight:		normal; bold; lighter; bolder; number;	épaisseur des caractères, respectivement : normal(400), gras(700), moins gras de l'élément parent, plus gras, valeur de 1 à 1000		
label{...}					
	white-space:		permet spécifier la manière dont votre navigateur va interpréter les espaces blancs et les types de caractères de retour à la ligne et la justification automatique		
		normal	fusionne les espaces blancs multiples et ajoute si nécessaire des sauts de ligne pour la justification du texte		
		nowrap	Les blancs sont regroupés comme avec normal mais les passages à la ligne automatiques sont supprimés		
		pre	ne fusionne pas les espaces blancs multiples et n'ajoute pas des sauts de ligne pour la justification du texte; saut de ligne avec 		
		pre-wrap	ne fusionne pas les espaces blancs multiples et ajoute automatiquement des sauts de ligne pour la justification du texte		
		pre-line	fusionne les espaces blancs multiples et ajoute des sauts de ligne pour la justification du texte		
		break-spaces	identique à celui de pre-wrap mais chaque séquence de blancs continue d'occuper un espace		
.container{...}					
	max-height:	150px			
	overflow-y:	auto			
	overflow-wrap;		force la césure		
	break-word				
sample{...}	text-transform:	uppercase			
	font:	1.5rem			
span.ex1{...}					
	min-width		largeur minimale d'un élément 		
	max-width		largeur maximale d'un élément 		
div{...}					
overflow:			gestion des dépassements du contenu d'un élément dans son bloc		
	visible;		en cas de dépassement, le contenu est affiché en dehors de la boîte de remplissage (padding). (par défaut)		
	hidden;		Le contenu rogné		
	clip;		Le contenu est aussi rogné		
	scroll;		apparition de barres de défilement pour voir le complément qui dépasse		
	auto		dito scroll		
	hidden visible;				
	inherit;				
	initial;				
	inset; outset				
	overlay;		dito auto mais barres de défilement à l'extérieur		
white-space			gestion des blancs au sein de l'élément		
	normal;		Les blancs sont regroupés et les passages à la ligne sont faits naturellement pour remplir les boîtes		
	nowrap;		Les blancs sont regroupés et les passages à la ligne sont supprimés		
	pre;		Les séries de blancs sont conservées et les sauts de ligne ne se font qu'avec saut de ligne et 		
	pre-wrap;		Les séries de blancs sont conservées et les sauts de ligne se font avec saut de ligne, et automatiquement.		
	pre-line;		Les séries de blancs sont regroupées et les sauts de ligne se font avec saut de ligne, et automatiquement		
	break-spaces;		dito pre-wrap mais les blancs occupent un espace		
width:			dimensions		
	<length>	(voir ci après les valeurs utilisées.)	unités absolue		
	<percentage>		largeur en pourcentage par rapport à la largeur du bloc		
	border-box		les dimensions concernent la boîte de bordure		
	content-box		les dimensions concernent la boîte de contenu de l'élément		
	auto		calcul automatique des dimensions		
	fill		Utilise la taille fill-available dans l'axe du sens de lecture.		
	max-content		La largeur intrinsèque préférée		
	min-content		La largeur intrinsèque minimum		
	available		La largeur du bloc contenant le bloc courant, moins la marge horizontale, la bordure et le remplissage (padding)		
	fill-content		la quantité la plus grande entre la largeur minimum et celle disponible		
position:			définit la façon dont un élément est positionné dans un document	https://developer.mozilla.org/fr/docs/Web/CSS/position	
	static	top; right; bottom; left; z-index	Cet élément (utilisé par défaut) est positionné dans le flux suivant la valeur indiquée	.static (position:static;)	
	relative	top; right; bottom; left;	L'élément est positionné dans le flux normal du document puis décalé, par rapport à lui-même	.relative (position: relative;top: -20px;left: 20px;background-color: white;width: 500px;)	
	fixed	top; right; bottom; left;	L'élément est positionné par rapport à la fenêtre du navigateur et reste à la même place quand la page défile		
	absolute	top; right; bottom; left;	dito fixed (est retiré du flux normal et aucun espace n'est créé pour l'élément sur la page) mais l'élément se positionne relativement à l'élément le plus proche.		
	sticky	top; right; bottom; left;	La position de la boîte est calculée en fonction du flux normal du document		
		z-index	Gère les superpositions des éléments		
img (float: right; margin(0 1em 1em))			positionne une image dans un texte	Ex : 	

12.2. Liste de commandes CSS

<https://developer.mozilla.org/fr/docs/Web/CSS/Reference> ou

12.3. Commandes par ordre alphabétique :

<http://www.css-faciles.com/proprietes-css-liste-alphabetique.php>

Autres références :

<http://openclassrooms.com/fr/courses>

Autres astuces :

<http://www.mytesting.fr/tag/astuces%20de%20blogueur/>

INDEX

Codes **HTML** en bleu, **CSS** en marron, **jQuery** en vert.

<p>@</p> <p>@font-face 43</p> <p><</p> <p><!Doctype...> 11</p> <p><a> 20</p> <p><abbr> 20</p> <p><acronym> 25</p> <p><address> 16</p> <p><applet> 26</p> <p><area> 25</p> <p><article> 16</p> <p><aside> 16</p> <p><audio> 26</p> <p> 23</p> <p><base> 13</p> <p><basefont /> 25</p> <p><bdi> 20</p> <p><bdo> 20</p> <p><big> 23</p> <p><blink> 25</p> <p><blockquote> 17</p> <p><body> 14</p> <p>
 21</p> <p><button> 36</p> <p><canvas> 29</p> <p><caption> 33</p> <p><center> 18</p> <p><cite> 21</p> <p><code> 21</p> <p><col> 34</p> <p><colgroup> 34</p> <p><content> 38</p> <p><data> 21</p> <p><datalist> 36</p> <p><dd> 17</p> <p> 29</p> <p><details> 37</p> <p><dfn> 21</p> <p><dialog> 37</p> <p><dir> 17</p> <p><div> 18</p> <p><dl> 18</p> <p><dt> 18</p> <p><element> 38</p> <p> 23</p> <p><embed> 26</p> <p><fieldset> 36</p> <p><figcaption> 18</p> <p><figure> 18</p> <p> 25</p> <p><footer> 16</p> <p><form> 36</p> <p><frame> 26</p>	<p><frameset> 29</p> <p><h1> 16</p> <p><head> 13</p> <p><header> 16</p> <p><hgroup> 16</p> <p><hr> 19</p> <p><html> 13</p> <p><i> 21</p> <p><iframe> 27</p> <p> 25</p> <p><input> 36</p> <p><ins> 29</p> <p><kbd> 22</p> <p><label> 36</p> <p><legend> 37</p> <p> 19</p> <p><link> 14</p> <p><main> 17</p> <p><map> 26</p> <p><mark> 22</p> <p><menu> 37</p> <p><menuitem> 37</p> <p><meta> 15</p> <p><meter> 37</p> <p><nav> 17</p> <p><noembed> 28</p> <p><noscript> 29</p> <p><object> 28</p> <p> 19</p> <p><optgroup> 37</p> <p><option> 37</p> <p><output> 37</p> <p><p> 19</p> <p><param> 28</p> <p><picture> 28</p> <p><pre> 20</p> <p><progress> 37</p> <p><q> 22</p> <p><rb> 22</p> <p><rp> 22</p> <p><rt> 22</p> <p><rtc> 22</p> <p><ruby> 22</p> <p><samp> 23</p> <p><script> 29</p> <p><section> 17</p> <p><select> 37</p> <p><shadow> 38</p> <p><slot> 38</p> <p><source> 28</p> <p> 23</p> <p><strike> 23</p> <p> 24</p> <p><style> 15</p> <p><sub> 24</p>
---	--

<summary>.....	37	I	
<sup>.....	24	id	41
<table>.....	32	J	
<tbody>.....	35	JavaScript.....	53
<td>.....	33	jQuery.....	62
<template>.....	38	L	
<textarea>.....	37	left	50
<tfoot>.....	35	letter-spacing	45
<th>.....	33	line-height	45
<thead>.....	35	liste	19
<time>.....	24	liste à puce	19
<title>.....	15	liste numérotée	19
<tr>.....	33	list-style	51
<track>.....	26	list-style-image	51
<tt>.....	23, 24	list-style-position	51
<u>.....	24	list-style-type	51
.....	19	M	
<var>.....	24	margin	47
<video>.....	26	métadonnées	15
<wbr>.....	25	N	
\$		niveaux de titres	16
\$ (sélecteur). action ()	62	O	
A		overflow	49
a	51	P	
AJAX.....	62	padding	48
Apache.....	67	paragraphe de texte	19
ava.....	71	position	50
B		Python.....	71
background	46	R	
border	48	right	50
bottom	50	Ruby.....	71
C		S	
class	41	saut de ligne	21
clear	50	scrollbar	51
clip	49	T	
color	46	text-align	44
commentaires.....	20	text-decoration	44
D		text-indent	44
direction	45	text-shadow	44
display	49	text-transform	44
F		titre du document	15
float	50	toggle	92
font	43	top	50
font-family	43	V	
font-size	43	vertical-align	45
font-style	44	visibility	49
font-variant	44	W	
font-weight	44	white-space	45
H			
height	47		

width	47	Z	
word-spacing	45	z-index	50
word-wrap	45		
X			
XML.....	65		