



# Sommaire

|  |    |                                       |    |
|--|----|---------------------------------------|----|
| ☒ Sommaire .....                                     | 2  | ☒ Balises .....                       | 14 |
| ☒ Présentation .....                                 | 3  | Balises du document .....             | 14 |
| Introduction .....                                   | 3  | Balises d'en-tête .....               | 14 |
| Fonctionnement.....                                  | 3  | Balises de listing .....              | 14 |
| Historique.....                                      | 4  | Balises de tableau.....               | 15 |
| ☒ Navigateurs .....                                  | 6  | Balises de formulaire.....            | 15 |
| Définition .....                                     | 6  | Balises de texte .....                | 16 |
| Historiquement.....                                  | 6  | Balises de contenu .....              | 17 |
| Fonctionnement.....                                  | 6  | Balises générique de conteneur.....   | 17 |
| Statistiques.....                                    | 7  | Balises Objet.....                    | 15 |
| Interprétation .....                                 | 7  | Balise autofermante.....              | 18 |
| ☒ Editeurs .....                                     | 8  | Type BLOC et Type INLINE.....         | 18 |
| Definition .....                                     | 8  | ☒ Conception.....                     | 19 |
| Les éditeurs WYSIWYG (tel-tel).....                  | 8  | Mise en forme .....                   | 19 |
| Les éditeurs de texte.....                           | 9  | Structure.....                        | 20 |
| ☒ Syntaxe et Normes .....                            | 10 | Meta .....                            | 21 |
| World wide web (w3c).....                            | 10 | Texte, titre et couleurs .....        | 21 |
| Règle d'écriture.....                                | 10 | Fond de page .....                    | 22 |
| Balises .....  | 10 | Commentaires .....                    | 23 |
| Attributs .....                                      | 11 | Listes .....                          | 24 |
| Elements .....                                       | 11 | Tables HTML .....                     | 25 |
| Sémantique .....                                     | 12 | Liens .....                           | 27 |
| Differences et usage : HTML, XHTML, DHTML, CSS ..... | 12 | Images .....                          | 29 |
| Extension.....                                       | 13 | Formulaires .....                     | 30 |
|  |    | MultiMedia .....                      | 32 |
|  |    | Disposition des elements .....        | 33 |
|  |    | Feuille de style .....                | 34 |
|  |    | Script .....                          | 35 |
|  |    | ☒ Divers .....                        | 36 |
|  |    | Normes d'accessibilité.....           | 36 |
|  |    | Etape de création d'un Site web ..... | 36 |
|  |    | Hébergeur Web .....                   | 36 |

## INTRODUCTION

---

L'HTML (Hyper Text Markup Language) est un langage de conception, inventé pour représenter des pages web.

C'est un langage de balises qui permet de mettre en forme le contenu et d'inclure des éléments dans les pages web.

HTML est initialement dérivé du SGML (Standard Generalized Markup Language).

XHTML (eXtensible HyperText Markup Language) est un langage de conception amélioré afin de représenter des pages web pour le w3c (World Wide Web Consortium)

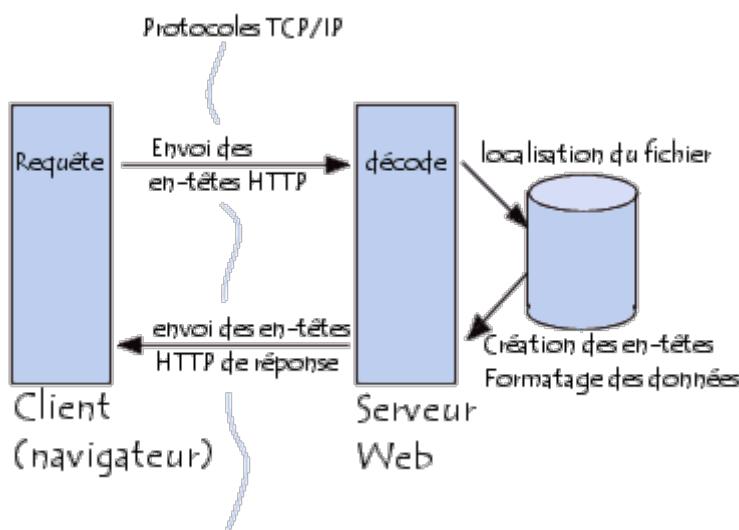
Le langage HTML n'est pas un langage compliqué mais est essentiel à comprendre car c'est le début de beaucoup de choses. Chaque site internet du monde entier comprend du HTML dans son code source.

Pour l'observer, rendez vous sur un site quelconque et faites un clic droit -> code source de la page (afficher la source pour Internet Explorer).

Le code HTML que vous visualisez est interprété par le serveur et vous renvoie ce que vous voyez à l'écran.

## FONCTIONNEMENT

---



Le navigateur effectue une requête HTTP.

Le serveur traite la requête puis envoie une réponse http.

Le protocole HTTP (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990.

Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML)

localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web.

Le HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur, développé pour le World Wide Web. HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

## HISTORIQUE

---

### 1969 : SGML

Charles Goldfarb, qui est alors chef de projet chez IBM, fait lancer par cette compagnie un langage descriptif.

Ce langage rend vite célèbre, Charles Goldfarb quitte alors cette compagnie pour développer un successeur de GML appelé SGML (Standard Generalized Markup Language), publié en 1986 comme norme ISO (ISO 8879:1986).

Les Communautés européennes s'associent à ce développement novateur dès 1984 et adoptent alors le SGML comme standard pour leurs très nombreuses publications officielles.

### 1989 : HTML

Tim Berners-Lee décrit la décision de baser HTML sur SGML comme étant aussi « diplomatique » que technique : techniquement, il trouvait SGML trop complexe, mais il voulait attirer la communauté hypertexte qui considérait que SGML était le langage le plus prometteur pour standardiser le format des documents hypertexte.

L'apparition du HTML se fait en 1989 par Tim Berners Lee.

Il est décrit comme étant un langage de balises permettant de communiquer sur Internet.

### 1993 : Mosaic

Avec le navigateur NCSA Mosaic, HTML connaît deux inventions majeures. D'abord l'invention de l'élément IMG permet d'intégrer des images (dans un premier temps, uniquement des GIF et des XBM) aux pages web (Mosaic 0.10). Ensuite l'invention des formulaires (Mosaic 2.0) rend le web interactif en permettant aux visiteurs de saisir des données dans les pages et de les envoyer au serveur web. Cette invention permet notamment de passer des commandes, donc d'utiliser le web pour faire du commerce électronique.

### 1994 : Netscape

Avec l'apparition de Netscape Navigator 0.9 le 13 octobre 1994, le support de nombreux éléments de présentation est ajouté : attributs de texte, clignotement, centrage, etc.

Les développeurs de navigateurs s'attachent à maximiser l'impact visuel des pages web en réponse aux demandes des utilisateurs.

Dans le même temps le W3C a été fondé par Tim Berners-Lee lorsqu'il a quitté l'Organisation européenne pour la recherche nucléaire (Cern) en octobre 1994.

## 1997 : HTML 4

Le W3C publie la spécification HTML 4.0 qui standardise de nombreuses extensions.

La variante stricte (strict) exclut des éléments et attributs dits « de présentation », destinés à être remplacés par les styles CSS.

## 2000 : XHTML

XHTML (Extensible HyperText Markup Language) est un langage de balisage servant à écrire des pages pour le World Wide Web et se fonde sur la syntaxe définie par XML.

## 2008 : HTML 5

Le W3C relance le développement du HTML et le fait évoluer sur de nombreux aspects, ce qui donne naissance à l'HTML5 et à l'abandon du XHTML.

## DEFINITION

---

Un navigateur Web est un logiciel conçu pour consulter et explorer des pages web.

C'est au minimum un client HTTP.

Les navigateurs Web sont des logiciels complexes et en constante évolution, principalement au niveau de la sécurité, car de nouveaux virus, spywares et autres scripts malveillants ne cessent de voir le jour, mais aussi au niveau des fonctionnalités car des nouveaux standards ou des révisions des standards voient régulièrement le jour.

## HISTORIQUEMENT

---

Dès 1995, Netscape Navigator devint le navigateur dominant, développé par Marc Andreessen, l'ancien responsable du développement de Mosaic. En 1995 sortit également Internet Explorer 1.

## FONCTIONNEMENT

---

L'utilisateur saisie dans le navigateur Web l'adresse URL de la ressource à consulter (en saisissant l'adresse ou en suivant un lien par exemple).

Le navigateur se connecte au serveur Web hébergeant la ressource visée grâce au protocole de communication (http).

Le moteur de rendu du navigateur traite cette ressource, télécharge les éventuelles ressources associées et affiche le résultat sur l'écran de l'utilisateur.

# Statistiques

Statistiques officielles provenant du W3C montrant les parts d'utilisation des différents navigateurs sur 10 années.

| Navigateur | I.Explorer | Firefox | Chrome     | Safari     | Opera |
|------------|------------|---------|------------|------------|-------|
| 2014       | 10.2 %     | 26.9 %  | 55.7 %     | 3.9 %      | 1.8 % |
| 2013       | 14.3 %     | 30.2 %  | 48.4 %     | 4.2 %      | 1.9 % |
| 2012       | 20.1 %     | 37.2 %  | 35.3 %     | 4.3 %      | 2.4 % |
| 2011       | 26.6 %     | 42.8 %  | 23.8 %     | 4.0 %      | 2.5 % |
| 2010       | 36.2 %     | 46.3 %  | 10.8 %     | 3.7 %      | 2.2 % |
| 2009       | 44.8 %     | 45.5 %  | 3.9 %      | 3.0 %      | 2.3 % |
| 2008       | 54.7 %     | 36.4 %  | inexistant | 1.9 %      | 1.4 % |
| 2007       | 58.6 %     | 31.0 %  | inexistant | 1.7 %      | 1.5 % |
| 2006       | 66.0 %     | 25.0 %  | inexistant | inexistant | 1.6 % |
| 2005       | 74.5 %     | 16.6 %  | inexistant | inexistant | 1.9 % |
| 2004       | 84.7 %     | 5.5 %   | inexistant | inexistant | 1.5 % |



Un navigateur permet de se connecter à un nom de domaine (ou adresse IP) afin de consulter des pages web hébergées sur un serveur quelque part dans le monde.

## INTERPRETATION

---

Les navigateurs web n'ont pas tous la même interprétation du code source qui constitue la page, c'est pourquoi il est essentiel de « tester » son site dans la phase de conception sur plusieurs navigateurs afin de traquer les éventuels bugs, décalage ou toutes autres surprises non désirées.

Un site doit être compatible avec tout navigateur et tout support (pc, tablette, smartphone, etc.).

Le w3c a fait le ménage dans le nombre considérable de balises existantes en HTML afin de donner la bonne marche à suivre à l'interprétation de ses balises par les navigateurs.

A ce jour, la plupart des navigateurs suivent les recommandations du w3c.

## DEFINITION

---

Un éditeur HTML (ou éditeur Web) est un logiciel conçu pour faciliter la préparation et la modification de documents écrits en HTML.

A l'origine, le code source d'une page web pouvait être écrit avec le simple bloc-notes fourni par Windows, ceci est toujours possible mais d'autres outils ont fait leur apparition depuis et sont beaucoup plus adaptés pour concevoir des pages web.

Dans ces éditeurs, il en existe deux catégories : les éditeurs wysiwyg et les éditeurs de texte.

## LES EDITEURS WYSIWYG (TEL-TEL)

---

Une interface utilisateur WYSIWYG permet de composer visuellement le résultat voulu, typiquement pour un logiciel de mise en page, un traitement de texte ou d'image.

C'est une interface « intuitive » : l'utilisateur voit directement à l'écran à quoi ressemblera le résultat final. « WYSIWYG » est l'acronyme de la locution anglaise « What you see is what you get », signifiant littéralement en français « tel affichage, tel résultat ».

### Avantages

- o facilité d'utilisation, prise en main rapide
- o observation immédiate du rendu graphique

### Inconvénients

- o manque de maîtrise sur la qualité du document HTML produit
- o risques d'incompatibilité avec des navigateurs non prévus par l'éditeur (plus récents, plus anciens, ou moins courants).

Parfois l'utilisateur peut contrôler le code source d'un éditeur WYSIWYG en le créant lui-même, mais cela revient à utiliser directement un éditeur de texte (dans ce cas, cela serait le choix le plus judicieux car les éditeurs de texte sont moins lourds).

Voici la liste des plus connus Adobe GoLive, Adobe Dreamweaver (anciennement Macromedia Dreamweaver), BlueGriffon, Komodo, KompoZer, Netlor, Microsoft Expression (anciennement Microsoft FrontPage), Mozilla Composer, Webacappela, OPEN BEXI HTML Builder.

## LES EDITEURS DE TEXTE

---

### Avantages

- o Assurance de la qualité du code source produit (c'est l'utilisateur qui le crée)

### Inconvénients

- o Ils requièrent donc une compétence dans le langage HTML

Voici la liste des plus connus Bluefish, Ecoder, HTMLedit, HTML Kit, Isofting eWriter Wisto, Notepad++, PSPad, Quanta+, SciTE .

## WORLD WIDE WEB (W3C)

---

### Définition

Le World Wide Web Consortium, abrégé par le sigle W3C, est un organisme de standardisation, fondé en octobre 1994 comme un consortium chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, RDF, CSS, PNG, SVG et SOAP.

On l'abordait dans le chapitre précédent, le w3c sert entre autres, à donner une « ligne directrice » aux navigateurs afin de se mettre d'accord sur l'interprétation des balises.

Afin de « tester » si sa page HTML est valide w3c, il suffit de se rendre au service de validation sur le site officiel du w3c <http://validator.w3.org/>.

## REGLE D'ECRITURE

---

L'avantage de concevoir un site en HTML est de structurer son code afin de séparer le contenu (tel que les textes, titres, paragraphes ainsi que tous les autres éléments qui composent la page) et de s'occuper de la mise en forme de ces éléments via le langage CSS dans un fichier séparé.

En HTML, vous serez obligés de respecter certaines règles, en voici quelques-unes :

## BALISES

---

### Définition

Forme concrète d'un élément. Nous parlerons aussi de balise ouvrante et fermante. Par exemple <HTML> est la balise ouvrante de l'élément HTML.

### Règles

- o toutes balises ouvertes doivent être fermées (nous optimisons l'utilisation des balises existantes et nous enlevons les balises qui ne sont plus utiles).
- o les noms des balises doivent être écrits en minuscule ;

### Exemple

```
Vrai : <span>texte</span>
Faux : <span>texte
Faux : <SPAN>texte</SPAN>
```

Certaines balises sont auto-fermantes.

## ATTRIBUTS

---

### Définition

Un attribut est une propriété d'une balise. Il permet de préciser le rôle ou certaines propriétés d'une balise dans le document.

### Règles

- o les noms des attributs doivent être écrits en minuscule ;
- o les valeurs des attributs doivent être entre guillemets doubles.

### Exemple

Vrai : <span class="maclasse">texte</span>

Faux : <span class=maclasse>texte</span>

## ELEMENTS

---

### Définition

Les éléments sont une notion abstraite. Il s'agit d'une dénomination pour cibler plusieurs choses dans un document.

### Règles

- o Chaque élément doit être imbriqué correctement.
- o Nous utilisons au mieux les feuilles de style

### Exemple

Vrai : <p>j'aime le <strong>sport</strong></p>

Faux : <p>j'aime le <strong>sport </p></strong>

## SEMANTIQUE

---

### Définition

La sémantique donne un sens aux éléments qui composent la page web.

### Exemple (aucun sens)

```
<p>J'ai un titre,      Voici mon texte suivi de mes citations      </p>
```

### Exemple (avec du sens)

```
<h1>J'ai un titre</h1>
<p>Voici mon texte<      /p>
<blockquote>  suivi de mes citations préférées      </blockquote>
```

### Conclusion

La sémantique permet d'améliorer l'accessibilité et le référencement de vos pages web sur les différents moteurs de recherche.

## DIFFERENCES ET USAGE : HTML, XHTML, DHTML, CSS

---

HTML dynamique, souvent abrégé DHTML, est un nom générique donné à l'ensemble des techniques utilisées par l'auteur d'une page web pour que celle-ci soit capable de se modifier elle-même en cours de consultation dans le navigateur web.

Le DHTML permet de coupler du javascript dans du HTML en utilisant certains attributs spécifiques (au DHTML) dans les tags comme par exemple onChange, onFocus, onMouseOver ou onClick.

### Exemple

```
<p onmouseover="javascript:alert('coucou');">survol      e le texte pour me voir</p>
```

Le XHTML est une version du HTML plus strict (dont l'objectif était de permettre une transition vers le XML) utilisé pendant plusieurs années mais progressivement abandonné (à partir de ~2011) au profit du HTML 5

Il ne faut pas confondre HTML et CSS, CSS sont des styles qui définissent la présentation de la page alors que HTML définit la structure de la page. Structure et présentation sont différentes mais complémentaires.

Exemple : Le HTML déclare les composants de la page, les feuilles de style CSS le mettent en forme.

- o nous écrirons du texte dans le HTML à l'intérieur de la balise « <p> »,
- o nous donnerons de la couleur au texte via la balise « p » en CSS.

## EXTENSION

---

Certaines personnes non averties parlent parfois de HTM au lieu de HTML.

A l'origine, les premières versions de windows ne prenaient en compte que les 3 premiers caractères de l'extension d'un fichier (ex : .txt, .exe, etc.). Donc pour une page HTML cela donnait un .htm.

Avec les systèmes d'exploitation actuels htm et HTML n'ont aucune différence.

HTM est l'extension tronquée à trois lettres d'un nom de fichier « .HTML », une limitation non actuelle que l'on trouve surtout sur d'anciens systèmes d'exploitation de Microsoft.

# Balises marqueur markup

Voici une liste (non exhaustive) des balises importantes en HTML :

## BALISES DU DOCUMENT

---

| Balises | Description                          |
|---------|--------------------------------------|
| <html>  | Balise d'ouverture de toute page web |
| <head>  | En-tête de la page                   |
| <body>  | Corps de la page                     |

## BALISES D'EN-TETE

---

| Balises    | Description  |
|------------|--|
| <link />   | Permet d'insérer une feuille de style ou d'afficher une icône de favicon   |
| <meta />   | Fixe les propriétés du document (auteur, description de la page)   |
| <script>   | Permet d'insérer un script (le plus souvent en JavaScript)   |
| <noscript> | Cette balise permet d'afficher un texte alternatif aux scripts de la page web (si le javascript a été désactivé par exemple)   |
| <style>    | Permet de définir une zone de mise en forme via du code CSS  |
| <title>    | Titre du document (ne s'affichant pas dans la page web mais sur la zone de titre du navigateur). Le texte contenu dans cette balise a une influence sur le référencement |

## BALISES DE LISTING

---

| Balises | Description                            |
|---------|--|
| <ul>    | Liste à puces non numérotée            |
| <ol>    | Liste à puces numérotée                |
| <li>    | Permet de créer un élément de la liste |
| <dl>    | Liste de définitions                   |
| <dt>    | Terme à définir                        |
| <dd>    | Définition du terme                    |

## BALISES DE TABLEAU

| Balises   | Description  |
|-----------|--|
| <table>   | Délimite un tableau                                      |
| <caption> | Permet de donner un titre au tableau                     |
| <tr>      | Ligne de tableau   |
| <th>      | Cellule d'en-tête du tableau (généralement mise en gras) |
| <td>      | Cellule du tableau                                       |
| <thead>   | permet d'insérer l'en-tête du tableau                    |
| <tbody>   | permet d'insérer le corps du tableau                     |
| <tfoot>   | permet d'insérer le pied du tableau                      |

## BALISES DE FORMULAIRE

| Balises    | Description  |
|------------|--|
| <form>     | La balise form délimite un formulaire. Elle s'accompagne de deux attributs :<br>method : comment vont circuler les données<br>action : Où va-t-on arriver après soumission du formulaire |
| <fieldset> | Permet de regrouper plusieurs éléments d'un formulaire   |
| <legend>   | Titre d'un groupe dans un formulaire<br>A utiliser à l'intérieur d'un <fieldset>   |
| <label>    | Titre d'un élément de formulaire. Cette balise s'accompagne de l'attribut for pour indiquer l'ID de l'élément auquel correspond le label.  |
| <input />  | Champ de formulaire.<br>Il existe de nombreux types de champs différents.  |
| <textarea> | Zone de saisie multi ligne.<br>La taille peut être définie grâce aux attributs rows et cols.   |
| <select>   | Liste déroulante.  |
| <option>   | Elément d'une liste déroulante.  |
| <optgroup> | Permet de regrouper plusieurs options de même type.  |

## BALISES OBJET

| Balises   | Description  |
|-----------|--|
| <object>  | Un objet peut être une animation flash, un applet java ou autre            |
| <param /> | La balise <param /> définit les paramètres d'objet dans une page web html. |

## BALISES DE TEXTE

| Balises      | Description   |
|--------------|---|
| <h1>         | Titre de niveau 1   |
| <h2>         | Titre de niveau 2   |
| <h3>         | Titre de niveau 3   |
| <h4>         | Titre de niveau 4   |
| <h5>         | Titre de niveau 5   |
| <h6>         | Titre de niveau 6   |
| <acronym>    | Cette balise sert à définir des acronymes (exemple : TGV = Train à grande vitesse).<br>Elle est accompagnée de l'attribut title.  |
| <abbr>       | Cette balise sert à déclarer une abréviation dans une page web.<br>L'attribut title permet de donner la signification de l'abréviation et l'attribut lang la langue de l'abréviation. |
| <blockquote> | Citation taille longue  |
| <cite>       | Citation taille moyenne   |
| <q>          | Citation taille courte  |
| <sup>        | Mise en exposant (exemple : m <sup>2</sup> )  |
| <sub>        | Mise en indice  |
| <strong>     | Strong permet de faire ressortir une partie de texte (texte en gras)  |
| <em>         | Em permet de faire ressortir une partie du texte (texte en italique)  |
| <br />       | Cette balise permet d'effectuer un retour à la ligne  |
| <p>          | La balise p pour paragraphe nous permettra d'insérer du texte   |
| <hr />       | La balise hr marque d'un trait la séparation entre deux « parties »   |
| <address>    | Cette balise permet de montrer une adresse (le texte est mis en italique)   |
| <del>        | Cette balise barre une partie d'un texte  |
| <ins>        | Cela permet d'indiquer qu'une partie du texte a été insérée (texte souligné)  |
| <dfn>        | Cela permet d'indiquer que le texte est une définition  |
| <kbd>        | La balise "keyboard" ( clavier ) permet d'indiquer un code que doit taper le visiteur   |
| <pre>        | Insère un saut de ligne à chaque fois que l'on passe à la ligne dans le document<br>HTML  |

## BALISES DE CONTENU

| Balises  | Description  |
|----------|--|
| <a>      | Lien hypertexte.<br>L'indication de l'url de destination s'effectue grâce à l'attribut href  |
| <img>    | Insère une image.<br>Les attributs src (pour indiquer l'adresse de l'image) et alt (pour indiquer un texte de remplacement) viennent compléter cette balise.   |
| <map>    | La balise map regroupe les différentes zones réactives d'une même image.   |
| <area>   | La balise area possède obligatoirement un attribut alt. L'attribut shape permet de définir quel type de zone nous souhaitons dessiner : "rect" (rectangle), "circle" ( cercle) ou "poly" (polygone). L'attribut coords permet de définir les coordonnées associées à la figure géométrique choisie |
| <audio>  | Audio  |
| <video>  | Vidéo  |
| <source> | Format source pour les balises <audio> et <video>  |
|          |  |

## BALISES GENERIQUE DE CONTENEUR

| Balises   | Description  |
|-----------|--|
| <span>    | Balise générique permettant de former un conteneur de type inline  |
| <div>     | Cette balise est en fait souvent utilisée pour regrouper d'autres balises de manière à former un conteneur. Il est ensuite intéressant d'utiliser le style CSS pour le positionner n'importe où. |
| <header>  | En-tête  |
| <nav>     | Balise menu permettant de regrouper les différents liens de navigation   |
| <footer>  | Pied de page   |
| <section> | Section de page  |
| <article> | Article (contenu)  |
| <aside>   | Partie destinée aux informations complémentaires   |

## BALISE AUTO-FERMANTE

Les balises de type AUTOFERMANTE HTML sont des balises qui sont ouvrantes et fermantes en même temps. Cela signifie qu'elles ne contiennent pas de contenu. Nous fermons donc la balise en ajoutant un espace et un slash (/) à la fin de la balise.

Quelques exemples de balises auto-fermantes :

<area />, <br />, <hr />, <img />, <input />, <link />, <meta />, <param />

## TYPE BLOC ET TYPE INLINE

### Type Bloc

C'est un élément (ou une zone) qui prend par défaut toute la largeur de la page.

Les balises de type BLOC HTML sont des boites d'éléments qui peuvent contenir d'autres éléments tels que du texte, des images, des liens ou autres... Elles peuvent contenir d'autres balises de type bloc et aussi contenir d'autres balises de type inline.

Par défaut, les blocs se placent toujours l'un en dessous de l'autre.

Quelques balises de type BLOC « block » : <p>, <ul>, <li>, <div>, <form>, <input>, <blockquote>, <table>, <h1>...<h6>,...

### Type Inline

C'est un élément (ou une zone) qui prend uniquement la place dont il a besoin

Une balise inline ne peut contenir uniquement une (ou plusieurs) balise inline, et n'a pas de dimensions à proprement parler (aucune largeur et aucune hauteur. Elle n'occupe que la place minimum nécessaire à son contenu).

Par défaut, les inline se placent toujours l'un à côté de l'autre.

Quelques exemples de balises de type Inline « en ligne » : <a>, <strong>, <em>, <span>, <img>, Description block, inline, etc.

## MISE EN FORME

---

La conception d'une page HTML se fait grâce à des composants tels que des balises.

La mise en forme de ces composants se fait grâce au langage CSS.

Ces balises peuvent recevoir uniquement un style via CSS si vous leur donnez un attribut class, id ou style :

- o class : indique le nom de la classe CSS à utiliser.

Cette classe sera ensuite appelée dans une feuille de style CSS et contiendra des propriétés permettant la mise en forme.

- o id : donne un nom à la balise.

Ce nom doit être unique sur toute la page car il permet d'identifier la balise.

L'id sera également appelé dans une feuille de style CSS.

Nous pouvons également nous servir de l'ID pour d'autres choses, comme par exemple pour un lien vers une ancre, pour un style CSS de type ID, pour des manipulations en Javascript etc.

- o style : cet attribut nous permet d'indiquer directement le code CSS à appliquer.

Nous ne sommes donc pas obligés d'avoir une feuille de style à part, nous pouvons y inscrire directement les attributs CSS.

D'une manière générale, il est préférable de ne pas utiliser cet attribut et de gérer la mise en forme par une feuille de style externe car cela rend notre site plus facile à mettre à jour par la suite.

## STRUCTURE

### Déclaration DOCTYPE

Dans un document HTML vous devrez spécifier le DOCTYPE.

```
<!DOCTYPE html>
<head>
    <title>Titre du site<           /title>
    <link rel="stylesheet" href="structure.css" />
</head>
<body>
</body>
</html>
```

#### Dans la partie HEAD (en-tête) :

Nous pouvons voir des informations sur le document, telles que le titre général de la page, le lien vers la feuille de style CSS.

Nous pourrions également y inscrire certaines balises meta donnant des informations pour le référencement du site ainsi qu'appeler des fichiers JavaScript.

#### Dans la partie BODY (corps) :

Nous pouvons observer que le contenu de notre document se trouve plus particulièrement à l'intérieur des balises `<body></body>`.

Le corps (balise body) contient ce qui sera affiché sur la page web.

Il est possible d'ajouter des commentaires, ce sont des annotations dans le code-source qui ne seront pas interprétées par le navigateur

## META

Les balises <meta /> contiennent les métas informations de la page web HTML.

Ainsi, la balise <meta /> vous donne des infos sur la description de la page web, ses mots-clés, son auteur, les règles spécifiques destinées aux robots des moteurs de recherche, la langue de la page web... L'attribut content est obligatoire.

```
<meta name="keywords" lang="fr" content="motcle1,mocle2" />
<meta name="description" content="Description de ma page web." />
<meta http-equiv="Content-Type" content="text/html; charset=iso - 8859 - 1" />
<meta http-equiv="Content-Language" content="fr" />
<meta http-equiv="Content-Script-Type" content="text/javascript" />
```

Les balises Meta description et Meta keywords sont importantes pour donner une description de son site web et indiquer des mots-clés. Il en existe d'autres.

## TEXTE, TITRE ET COULEURS

Nous nous situons à l'intérieur des balises <body> </body> et nous souhaitons afficher un titre et un texte.

Exemple



Certaines balises héritent de propriétés, c'est le cas de h1 ou de p qui héritent de marges, cela est bien sûr modifiable via les feuilles de style CSS.

Code

```
<h1>Titre de niveau 1</h1>
<p>Paragraphe...</p>

<h2>Titre de niveau 2</h2>
<p>Second paragraphe...</p>
```

La même chose en couleur :



The screenshot shows a white rectangular box with a thin gray border. Inside, there is a red title "Titre de niveau 1" and a red paragraph "Paragraphe...". Below it is a blue title "Titre de niveau 2" and a blue paragraph "Second paragraphe...".

Code

```
<h1 style="color:red">Titre de niveau 1</h1>
<p style="color:red">Paragraphe...</p>

<h2 style="color:blue">Titre de niveau 2</h2>
<p style="color:blue">Second paragraphe...</p>
```

Note

Vous l'aurez compris, le fait de mettre l'attribut « style » n'est pas le plus judicieux car nous ne séparons pas la mise en forme du contenu. Cependant étant donné que nous nous intéressons uniquement au HTML ici, nous l'utilisons quand même pour démontrer que nous pouvons mettre en forme le HTML.

## FOND DE PAGE

Le fond d'une page se fixe sur la balise ouvrante : <body>.

Nous pouvons à la fois définir une image de fond ou une couleur de fond.

Exemple



The screenshot shows a solid green rectangular background. Overlaid on it are the same red and blue text elements as the previous example: a red "Titre de niveau 1" and red "Paragraphe...", followed by a blue "Titre de niveau 2" and blue "Second paragraphe...".

## Code

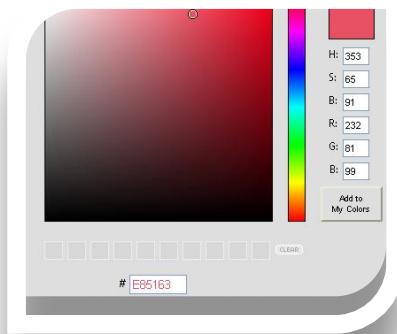
```
<body style="background: #06c626">
    <h1 style="color:red">Titre de niveau 1</h1>
    <p style="color:red">Paragraphe...</p>

    <h2 style="color:blue">Titre de niveau 2</h2>
    <p style="color:blue">Second paragraphe...</p>
</body>
```

Rouge vert bleu, abrégé RVB (ou RGB de l'anglais red green blue), est un format de codage des couleurs.

De nombreux sites Internet vous indiqueront les codes couleurs les plus utilisés.

L'outil le plus populaire permettant de vous générer le code couleur en fonction de votre choix reste sans aucun doute la palette de couleurs sur le logiciel PhotoShop.



## COMMENTAIRES

---

En programmation (ou en langage de conception comme ici), les commentaires sont des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas nécessaires à l'exécution du programme.

Les commentaires sont le plus souvent utilisés pour expliquer le code ; certains outils permettent d'extraire l'information disponible dans les commentaires pour générer automatiquement la documentation du programme.

Les commentaires en HTML se déclarent avec `<!-- texte -->`

## Code

```
<! -- Voici un commentaire -- >
```

## LISTES

Les listes sont des outils bien utiles pour présenter des informations. HTML en offre 3 types différents. Le premier type de liste est la liste non ordonnée, autrement appelée liste à puces.

### Exemple

Code

- 1er élément
- 2ème élément
- 3ème élément

```
<ul>
  <li>1er élément</li>
  <li>2ème élément</li>
  <li>3ème élément</li>
</ul>
```

Le second type de liste est la liste ordonnée. Ce type de liste sera présentée avec des petites puces avant le texte de chaque élément.

### Exemple

1. 1er élément
2. 2ème élément
3. 3ème élément

Code

```
<ol>
  <li>1er élément</li>
  <li>2ème élément</li>
  <li>3ème élément</li>
</ol>
```

Le dernier type de liste est la liste de définitions, permettant de créer des listes possédant un terme et sa définition.

### Exemple

- |                             |
|-----------------------------|
| Premier terme               |
| Définition du premier terme |
| Second terme                |
| Définition du second terme  |

## Code

```
<dl>
  <dt>Premier terme</dt>
  <dd>Définition du premier terme.</dd>
  <dt>Second terme</dt>
  <dd>Définition du second terme.</dd>
</dl>
```

## TABLES HTML

---

Souvent, les intégrateurs web vous mettront en garde contre l'utilisation des tables HTML (appelées aussi tableau HTML).

Pourquoi ? A l'origine les « tables » ou « tableaux » ont été créés pour présenter des données tabulaires. Ils ont été détournés par les designers afin de structurer des pages web complexes. L'inconvénient était qu'il n'était pas simple de mettre en place un design complexe à l'aide de ce système.

Aujourd'hui, et depuis déjà quelques années, la conception d'un design en structure de page web est beaucoup plus simple et se fait par des balises prévues à cet effet telles que les balises div.

Les avantages d'utiliser des blocs div à la place des « tables » ou « tableaux » sont multiples :

- o Avoir des sources sémantiquement correctes
- o Le code est beaucoup plus maintenable et moins complexe
- o Les pages sont plus accessibles

Néanmoins, si nous revenons à la fonction d'origine des « tables » ou « tableaux » (de représenter des données tabulaires), il est tout à fait normal et autorisé par les normes W3C/HTML de les utiliser.

Tant que nous ne cherchons pas à construire son design de site via des tables et que nous ne les emboîtons pas les unes dans les autres, le code sera valide w3c.

Informations :

Pour construire une newsletter, l'utilisation des « tables » ou « tableaux » sont encore préconisés.

Un tableau HTML est découpé en lignes contenant des cellules.

### Exemple

| Ma premiere table  |                    |                    |
|--------------------|--------------------|--------------------|
| ligne 1, colonne 1 | ligne 1, colonne 2 | ligne 1, colonne 3 |
| ligne 2, colonne 1 | ligne 2, colonne 2 | ligne 2, colonne 3 |
| ligne 3, colonne 1 | ligne 3, colonne 2 | ligne 3, colonne 3 |

### Code

```
<table border="1" cellpadding="5" cellspacing="10">
  <caption>Ma premiere table</caption>
  <tr>
    <th>ligne 1, colonne 1</th>
    <th>ligne 1, colonne 2</th>
    <th>ligne 1, colonne 3</th>
  </tr>
  <tr>
    <td>ligne 2, colonne 1</td>
    <td>ligne 2, colonne 2</td>
    <td>ligne 2, colonne 3</td>
  </tr>
  <tr>
    <td>ligne 3, colonne 1</td>
    <td>ligne 3, colonne 2</td>
    <td>ligne 3, colonne 3</td>
  </tr>
</table>
```

Dans la première ligne, la balise `<th>` désigne des cellules d'un type particulier : elles ne contiennent pas de données, mais l'en-tête (ou titre) des colonnes correspondantes.

Les balises `<tr>` et `</tr>` délimitent les lignes du tableau.

Dans les lignes suivantes, `<td>` spécifie les cellules de données.

L'élément `caption` va placer le titre, généralement au-dessus du tableau

Les cellules peuvent en recouvrir d'autres, que ce soit en largeur ou en hauteur. Pour qu'une cellule occupe x cellules vers la fin de la ligne, nous utilisons l'attribut colspan="x".

x".

Pour qu'une cellule occupe x cellules vers la fin de la colonne, nous utilisons l'attribut rowspan="

Exemple

|                         |                         |                    |
|-------------------------|-------------------------|--------------------|
| ligne 1 et 2, colonne 1 | ligne 1, colonne 2      | ligne 1, colonne 3 |
|                         | ligne 2, colonne 2      | ligne 2, colonne 3 |
| ligne 3, colonne 1      | ligne 3, colonne 2 et 3 |                    |

Code

```
<table border="1" cellpadding="5" cellspacing="10">
<tr>
  <td rowspan="2">ligne 1 et 2, colonne 1</td>
  <td>ligne 1, colonne 2</td>
  <td>ligne 1, colonne 3</td>
</tr>
<tr>
  <td>ligne 2, colonne 2</td>
  <td>ligne 2, colonne 3</td>
</tr>
<tr>
  <td>ligne 3, colonne 1</td>
  <td colspan="2">ligne 3, colonne 2 et 3</td>
</tr>
</table>
```

Voir aussi les balises : tbody, thead, tfoot.

## LIENS

---

L'aspect le plus intéressant du Web est cette formidable capacité à créer des liens de page en page, de site en site. Pour créer un lien, nous utilisons la balise a.

### Lien externe

Ce type de lien pointe vers un autre site.

## Code

```
<a href="http://www.tf1.fr/news">TF1 actualités</a>
```

### Lien local

Ce type de lien permet de lier deux pages de votre site entre elles.

Supposons que nous nous trouvons dans une page nommée « page.html »

## Code

```
<a href="page2.html"> Accédez à ma page 2</a>
```

Sur la page nommée « page2.html » il faudrait faire le code inverse :

```
<a href="page.html"> Retour vers ma page </a>
```

### Lien ouvrant une nouvelle fenêtre

L'attribut target permettra d'ouvrir la page dans une nouvelle fenêtre et ainsi ne pas quitter la page affichée à l'écran.

## Code

```
<a href="http://www.google.com/" target="_blank">Google</a>
```

### Lien vers une adresse email

Ce type de lien permet d'ouvrir le client de messagerie de l'utilisateur.

A noter qu'ils ne sont plus très utilisés car ils ne sont plus dans l'air du temps et sont susceptibles de récolter des spams.

## Code

```
<a href="mailto:nom@site.com"> contactez - moi</a>
```

### Lien vers un fichier

Ce type de lien permet de pointer vers un fichier à ouvrir ou à télécharger sur son ordinateur.

## Code

```
<a href="un_fichier.xls"> consulter le fichier excel </a>
```

## Lien « ancre »

Ce type de lien permet d'accéder rapidement à l'information qui nous intéresse en ciblant l'ancre (id) auquel un texte dépend.

### Code

```
<a href="#ici"> lire le texte </a>
<p id="ici"> texte </p>
```

Les liens servent également à la construction d'une zone de navigation couramment appelée « menu ».

## IMAGES

---

### Images locales

Afin de rendre vos documents plus attrayants, vous pouvez y insérer des images à l'aide de l'élément img. Nous considérons pour l'exemple que votre image soit en local.

### Code

```

```

Cet élément vous montre comment insérer une image dans un document et la syntaxe correcte en HTML pour un élément ne possédant pas de balise fermante.

La balise img ne contient, en effet aucun texte, il est donc inutile de la fermer.

La syntaxe des balises auto-fermantes consistant à ajouter un / à la fin du marqueur.

Nous ajoutons également un espace avant ce dernier pour permettre aux anciens navigateurs de comprendre la syntaxe du marqueur.

Si votre visiteur ne peut pas lire les images car il ne possède qu'un navigateur textuel ou vocal, il est indispensable de lui fournir un texte décrivant votre image avec l'attribut alt.

### Lien sur image

Vous pouvez aussi faire d'une image un lien hypertexte.

Par exemple le logo de votre organisation servant de lien pour revenir à la page d'accueil.

### Code

```
<a href="/"></a>
```

## FORMULAIRES

---

Les formulaires interactifs permettent aux auteurs de pages Web de doter leur page web d'éléments interactifs permettant par exemple un dialogue avec les internautes.

Le lecteur saisit des informations en remplissant des champs ou en cliquant sur des boutons, puis appuie sur un bouton de soumission (submit) pour l'envoyer soit à un URL, c'est-à-dire de façon générale à une adresse e-mail ou à un script de page web dynamique tel que PHP.

Un formulaire peut servir à recueillir les avis des visiteurs, leur e-mail en vue d'une newsletter. Cela peut également servir pour une inscription, un recrutement, un bon de commande, un sondage, un jeu... La balise `<form>` qui permet de construire un formulaire possède au moins deux attributs « method » et « action ».

### METHOD

METHOD indique sous quelle forme seront envoyées les réponses

« POST » est la valeur qui correspond à l'envoi de données côté « serveur », tandis que « GET » correspond à un envoi des données codées dans l'URL.

### ACTION

Indique l'adresse d'envoi. Où arrivons-nous une fois le formulaire soumis ?



|                          |                       |
|--------------------------|-----------------------|
| <b>Informations</b>      |                       |
| <b>Nom *</b>             |                       |
| <input type="text"/>     |                       |
| <b>Prénom *</b>          |                       |
| <input type="text"/>     |                       |
| <b>Adresse</b>           |                       |
| <input type="text"/>     |                       |
| <b>Date de naissance</b> |                       |
| Jour                     | 1                     |
| Mois                     | 1                     |
| Année                    | 1900                  |
| <b>Sex(e)</b>            |                       |
| homme:                   | <input type="radio"/> |
| femme:                   | <input type="radio"/> |
| <b>inscription</b>       |                       |

## Exemple de code

```

<form method="post" action="formulaire.php">
    <fieldset>
        <legend>Informations</legend>
        <label for="nom">Nom *</label>
        <input type="text" id="nom" name="nom"
placeholder="nom" /><br />
        <label for="prenom">Prénom *</label>
        <input type="text" id="prenom" name="prenom" /><br />
        <label for="adresse">Adresse</label>
        <textarea id="adresse" name="adresse" cols="16">
    </textarea>
    </fieldset>
    <fieldset>
        <legend>Date de Naissance</legend>
        <label for="jour">Jour</label>
        <select id="jour" name="jour">
            <option name="jour" value="1">1 </option>
        </select><br /><br />
        <label for="mois">Mois</label>
        <select id="mois" name="mois">
            <option name="mois" value="1">1</option>
        </select><br /><br />
        <label for="annee">Année</label>
        <select id="annee" name="annee">
            <option name="annee" value="1900">1900</option>
        </select>
    </fieldset>
    <fieldset>
        <legend>Sexe</legend>
        <p>homme:</p><input type="radio" name="sexe" value="M" /><br />
        <p>femme:</p> <input type="radio" name="sexe" value="F" />
    </fieldset>
    <input type="submit" class="submit" name="inscription"
value="inscription"/>
</form>

```

## Les différents composants d'un formulaire

Attention pour une bonne récupération des données saisies, il faut impérativement nommer chaque champ qui compose le formulaire via l'attribut « name ».

La balise label avec l'attribut for pointe sur l'id d'un input, cela permet de cliquer sur « l'étiquette » du champ et de retrouver son curseur à l'intérieur.

Fieldset et legend délimite des « zones » du formulaire.

Le type des « input » peut être : text, password, radio, checkbox, file, hidden, submit, reset, etc.

Nous pouvons également ajouter d'autres attributs tels que maxlength qui permet de limiter le nombre de caractères à saisir coté client, disabled qui « désactive » le champ à l'écriture ou encore checked qui coche une case par défaut.

Select option offre une liste déroulante. L'attribut multiple permet à l'utilisateur de faire plusieurs choix dans cette liste.

Textarea offre une zone de texte un peu plus grande (paramétrable avec les attributs rows et cols). L'attribut Placeholder permet de pré saisir un texte à l'intérieur d'une case de saisie.

## MULTIMEDIA

Certains éléments peuvent être intégrés aux pages web tels qu'une carte interactive (code googlemap fournit par google).

Les vidéos et les sons audio possèdent leur propre balise.

```
<video preload controls poster="cic.gif">
    <source src="cic.ogv" type="video/ogg" />
</video>
```

```
<audio src="jazz.mp3" controls></audio>
```

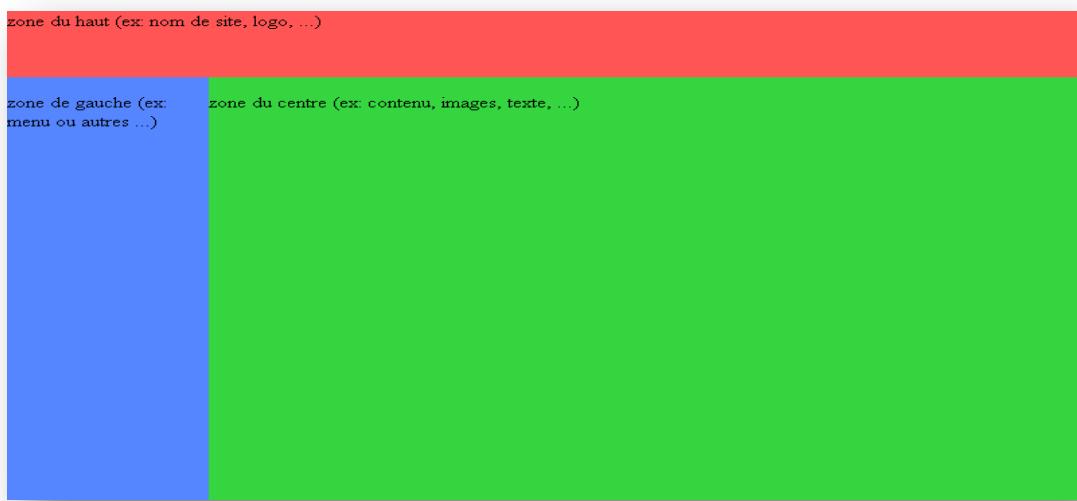
## DISPOSITION DES ELEMENTS

La balise DIV (qui est de type bloc) est un conteneur qui peut inclure toutes les autres balises HTML (paragraphes, table, liens, images ... etc. et également d'autres div).

Elle sert à structurer le document en plusieurs zones.

Nous employons la balise « div » pour « division ». Ces balises permettent de « former » un site web.

### Exemple



### Code

```
<div style="background: #336699; width: 800px; height: 600px;">
    <div style="background: #ff5555; width: 800px; height: 70px;">
        <p> zone du haut (ex: nom de site, logo, ...) </p>
    </div>
    <div style="background: #5585ff; width: 150px; height: 530px; float: left;">
        <p> zone de gauche (ex: menu ou autres ...) </p>
    </div>
    <div style="background: #35d540; width: 650px; height: 530px; float: left;">
        <p> zone du centre (ex: contenu, images, texte, ...) </p>
    </div>
</div>
```

En rouge, La zone du haut correspond à la zone de « header » où nous pouvons retrouver le nom du site, le logo, etc.

En bleu, la zone de gauche correspond à la zone de menu ou autres.

En vert, la zone du centre correspond à la zone du contenu, images, texte, etc.

La balise span permet sensiblement de faire la même chose, à la différence près que la balise span est une balise inline (en ligne) et que, syntaxiquement, nous ne pouvons pas imbriquer une div dans une balise span.

- o La balise span est utilisée pour créer des colonnes (blocs côte à côté)
- o La balise div est utilisée pour créer des blocs les uns en dessous des autres.

Lorsque nous commençons un site web, il faut toujours prévoir les zones de celui-ci et les faire ressortir avec des couleurs de fond, cela correspond à identifier les zones actuellement en chantier.

## FEUILLE DE STYLE

Nous l'avons vu, les feuilles de style CSS permettent de mettre en forme les éléments déclarés dans le HTML. Cela permet de séparer le fond (le contenu) de la forme (la mise en forme).

Pour insérer un style dans un document, il existe plusieurs méthodes :

### Exemple 1

```
<style>  
...  
</style>
```

1. La déclaration des propriétés CSS se fait dans le HTML.

Elle a le désavantage d'agrandir le code et nous devons répéter cette déclaration sur chaque page.

### Exemple 2

```
<link rel="stylesheet" href="style.css" type="text/css" media="screen" />
```

2. Par l'intermédiaire d'une feuille de style externe (située sur votre serveur).

Solution à recommander car elle permet une économie de code et une maintenance simplifiée.

### Exemple 3

```
<style type="text/css">  
@import url('http://www.site.com/style.css');  
</style>
```

3. Par l'intermédiaire d'une feuille de style externe (située sur un autre serveur par exemple).

## SCRIPT

---

Un script peut être inséré dans le HTML (généralement il s'agit de code JavaScript)

Là aussi, plusieurs méthodes s'offrent à nous :

### Exemple 1

```
<script>  
...  
</script>
```

1. Script inséré directement dans le contenu de la page (à répéter sur toutes les pages HTML).

### Exemple 2

```
<script src="url/fichier.js"></SCRIPT>
```

2. Script inséré via un fichier externe (solution recommandée).

## NORMES D'ACCESSIBILITE

---

L'accessibilité du Web est la problématique de l'accès aux services et contenus en ligne pour les handicapés et les seniors.

En respectant les standards HTML et CSS édictés par le W3C et les normes Accessiweb, vous assurerez à votre site une visibilité maximale pour tous les types de population : seniors, personnes handicapées, personnes utilisant des outils de navigations spéciales telles que des lecteurs d'écran, des claviers en braille, etc.

## ETAPE DE CREATION D'UN SITE WEB

---

Un site Web est un ensemble de pages Web hyper liées entre elles et mises en ligne à une adresse Web.

Pour créer un site Web en HTML/CSS, il faut :

- o Définir les zones du site, élaborer la structure (arborescence), le contenu et la présentation ;
- o Mettre en ligne en hébergeant le site sur un serveur Web continuellement connecté à Internet (accessible via un nom de domaine)

Il est également possible d'utiliser un système de gestion de contenu (CMS).

## HEBERGEUR WEB

---

Un hébergeur Internet (ou hébergeur Web) a pour vocation de mettre à disposition les internautes des sites web conçus et gérés par des tiers.

Il donne ainsi accès à tous les internautes au contenu déposé dans leurs comptes par les webmestres souvent via un logiciel FTP ou un gestionnaire de fichiers. Pour cela, il maintient des ordinateurs connectés 24 heures sur 24 à Internet (des serveurs web par exemple) par une connexion à très haut débit, sur lesquels sont installés des logiciels : serveur HTTP (souvent Apache), serveur de messagerie, de base de données...

Une fois que les pages HTML sont conçues et que le site est terminé, nous utilisons généralement un logiciel FTP afin d'envoyer les pages à son hébergement web (créées au préalable) afin qu'elles soient consultables par tout le monde.

Votre propre pc peut également faire office d'hébergement à condition qu'il reste toujours allumé.