

# Introduction aux technologies du Web

*Premier aperçu des langages HTML et CSS.*

## 1 L'HyperText Markup Language (ou HTML)

### 1.1 Différences entre HTML et XHTML

**L'HTML.** L'HTML est un langage de balisage permettant de structurer le contenu de documents (pages web) et d'y inclure différentes ressources (liens, images ...). La plupart des pages web sont aujourd'hui écrites à l'aide d'HTML 5, la dernière version du langage définie par le *World Wide Web Consortium* (ou W3C), l'organisme chargé de standardiser les technologies du web (dont notamment l'HTML et le CSS).

**L'XHTML.** L'XHTML est une extension de l'HTML basée sur le langage XML. Il possède une syntaxe plus stricte et a pour but de produire des documents plus clairs car mieux organisés. Son développement a été abandonné à la fin des années 2000. Il est cependant encore intéressant de garder certaines de ces contraintes désormais révolues comme bonnes pratiques de développement.

**Des informations bien balisées.** Par exemple, en XHTML, toute information est contenue entre deux balises, l'une entrante (`<balise>`) et l'autre fermante (`</balise>`). Les informations sont ensuite organisées en s'imbriquant les unes dans les autres : on écrit ainsi `<a>Je suis un <b>Homme</b></a>` et non `<a>Je suis un <b>Homme</a></b>`.

En plus des doubles balises, certaines balises. Ces dernières constituent en fait l'information en elles-mêmes. Dans ce cas, la balise sera ouverte et fermée en même temps (`<balise />`).

Certaines balises peuvent (et parfois doivent) posséder des attributs. Ces attributs s'écrivent dans la balise ouvrante, sous la forme `attribut="valeur"`. On écrit ainsi `<livre auteur="Nietzsche">Le Gai Savoir</livre>` ou encore `<image titre="Renard de feu" />`.

*Il est à noter que si la valeur d'un attribut peut prendre n'importe quelle forme, les noms de balise et d'attribut doivent être écrits en minuscule.*

**Les contraintes du HTML.** Même si plus permissif que l'XHTML, l'HTML n'est pas pour autant laxiste. Le W3C impose ainsi des contraintes à respecter. Ces dernières aident au bon affichage d'une page web sur tous les navigateurs et facilitent la création de programmes parcourant automatiquement le web (robots d'indexation ou *webcrawlers*). En outre, de nombreuses personnes non/mal-voyantes accèdent au web via des logiciels spécialisés qui supposent que les pages web respectent la syntaxe HTML. Ne pas respecter la syntaxe revient donc à les **exclure** du web. Pour vérifier qu'une page web satisfait les standards W3C, il est possible de se rendre sur <http://validator.w3.org/>.

## 1.2 Structure d'une page web

Jusqu'à l'arrivée d'HTML 5, il était nécessaire de préciser la version d'HTML utilisée par la page web avant même d'écrire la page. Cette information, ou *doctype* (pour *document type*) est de nos jours réduite à son strict minimum (`<!doctype html>`) et n'est encore présente que par simple souci de compatibilité avec les anciens navigateurs.

Une fois la *doctype* écrite, il est ensuite possible d'écrire la page web. Puisque toutes les informations d'une page sont délimitées par des balises, la page web est elle-même délimitée par les balises `<html></html>`. Il est ensuite nécessaire de décrire au minimum l'en-tête et le corps de la page, à l'aide des balises `<head></head>` et `<body></body>`.

Le code HTML minimal d'une page web est donc :

<code>&lt;!doctype html&gt;</code>	Déclaration du type de document
<code>&lt;html&gt;</code>	Début de la page web
<code>&lt;head&gt;</code>	Début de l'en-tête
<code>...</code>	Contenu de l'en-tête
<code>&lt;/head&gt;</code>	Fin de l'en-tête
<code>&lt;body&gt;</code>	Début du corps
<code>...</code>	Contenu du corps
<code>&lt;/body&gt;</code>	Fin du corps
<code>&lt;/html&gt;</code>	Fin de la page web

### 1.2.1 L'en-tête d'une page web (`<head></head>`)

L'en-tête d'une page web contient une série d'informations sur la page web en elle-même, appelées **métadonnées**. Ces informations comprennent notamment le titre de la page web, son type de contenu, l'encodage dans lequel elle est écrite ou encore les ressources qui y sont liées sur le serveur (cf. section 2).

Certains types de métadonnées possèdent leur propre balises, comme le titre (`<title></title>`) ou le style (`<style></style>`, cf. section 2) de la page web. Les autres métadonnées sont incluses grâce à la balise générique `<meta name="" content="" />` où l'attribut `name` permet de définir le nom de l'information (encodage, type de contenu ...) et `content` permet de stocker l'information à proprement dite.

*Il est à noter que la balise `<title>` est capitale dans l'indexation d'une page web par les moteurs de recherche. Elle tend donc à être incluse dans les exemples de pages web minimales.*

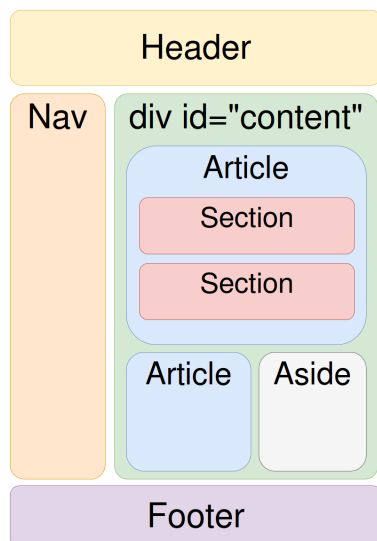


FIGURE 1 – Structure standard d'une page web en HTML 5.

### 1.2.2 Le corps d'une page web (`<body></body>`)

Outre les métadonnées, une page web possède également un contenu (son utilité serait sinon très limitée ...) organisé à l'aide d'un grand nombre de balises. C'est ce contenu qui sera affiché à l'utilisateur par le navigateur.

**Structure standard d'une page web** Le HTML 5 a permis de standardiser la structure d'une page web, en reprenant une organisation déjà largement répandue à son apparition (cf. figure 1).

Selon cette structure, une page web possède quatre éléments principaux :

**Header** A ne pas confondre avec l'en-tête de la page web, signalé par la balise `<head></head>` ! Le *header* accueille les éléments d'introduction à la page web (titre, logo ...).

**Nav** Cet élément comprend le(s) menu(s) qui permet(tent) la navigation dans le site web. Selon la page web, il peut être indépendant ou inclus dans un autre élément (le plus souvent le *header*).

**Content** Cet élément inclut le contenu de la page web en lui-même.

**Footer** Le pied de page, il inclut en général diverses informations sur le Copyright, les auteurs de la page et des liens vers d'autres pages web en relation.

Avant l'arrivée d'HTML 5, chacun de ces éléments était écrit dans la page web grâce à la balise `<div></div>` à laquelle on donnait un attribut identifiant `id` correspondant à sa fonction (*header*, *nav* ...). La balise `div` est une balise générique utilisée pour regrouper des morceaux de code HTML.

La standardisation d'HTML 5 a permis d'introduire des balises `div` spécialisées, portant le même nom que les éléments dans le figure 1. On écrit désormais `<footer>Pied de page</footer>` et non plus `<div id="footer">Pied de page</div>`. Outre la simplification à l'écriture, cette spécialisation permet également de donner un sens aux balises. Il est ainsi désormais possible aux moteurs de recherche de savoir que les éléments contenus dans la balise `header` sont des éléments très importants, c'est ce que l'on appelle le **web sémantique** (et ça compte beaucoup dans le référencement des pages web!).

*Notez que la balise `<content></content>` n'existe pas en HTML 5. A la place, les balises *article*, *section* et *aside* ont été introduites.*

**Article** Permet de regrouper du contenu qui se suffit à lui-même, qui peut être compris indépendamment du reste de la page web. Si tout le contenu d'une page web peut se regrouper en un seul *article*, on utilisera directement la balise `article` à la place de `<div id="content">`.

**Section** Comme son nom l'indique, permet de regrouper le contenu correspond à une section de document.

**Aside** Cette balise est destinée au contenu additionnel, un contenu "facultatif" venant en bonus du contenu principal de la page web.

**Formater son texte** Une fois le squelette de sa page déterminé grâce aux balises précédentes, d'autres balises permettent de "remplir" la page.

Les balises les plus souvent utilisées sont ainsi `<p></p>` et `<h1></h1>`, permettant d'écrire respectivement le texte de la page web et les titres (de section, paragraphe ...). Il existe plusieurs versions pour les balises de titre, en fonction de l'importance du titre. Ainsi, `h1` désigne un titre capital alors que `h6` désigne un titre très marginal (bien évidemment, les balises `h2`, `h3`, `h4` et `h5` existent également).

Il est possible de mettre en exergue certains mots contenus dans la page web, grâce aux balises `<em></em>` et `<strong></strong>`. Les navigateurs ont pour habitude de représenter les mots mis en emphase (*em*) en italique et les mots mis en exergue (*strong*) en gras. Cependant, ces balises ne doivent pas être utilisées à des buts esthétiques (cf. section 2) mais seulement pour signifier l'importance de certains mots dans un texte. Là encore, c'est de la sémantique et cela va influencer le référencement de la page web!

Note : l'**exergue** est généralement considérée plus forte que l'*emphase*, **strong** a donc plus de force qu'*em*.

**Intégrer des listes à vos pages web** HTML 5 considère plusieurs types de listes, dont les deux suivantes :

- les listes à puces, encadrées par la balise `<ul></ul>` pour *unordered list*.
- les listes ordonnées, encadrées par la balise `<ol></ol>` pour *ordered list*.

Quelque soit le type de liste, les éléments d'une liste seront eux-même encadrés par les balises `<li></li>`. On aura ainsi, suivant la nature de la liste, les exemples suivants :

```
<ul>
  <li>Raclette</li>
  <li>Fondue savoyarde</li>
  <li>Tartiflette</li>
</ul>

<ol>
  <li>Ski</li>
  <li>Snowboard</li>
  <li>Luge</li>
</ol>
```

- Raclette  
- Fondue savoyarde  
- Tartiflette

1. Ski  
2. Snowboard  
3. Luge

**Intégrer d'autres éléments** Le HTML permet également d'intégrer beaucoup d'autres éléments, et sa cinquième version a encore amélioré ses capacités.

Les liens (dits hypertextes) sont ainsi intégrés via la balise `<a></a>` (pour *ancree*). Pour qu'un lien puisse fonctionner, il est nécessaire d'indiquer l'adresse de la ressource cible (qui peut-être une autre page web, une vidéo, un morceau de musique ...) via l'attribut `href` (pour *hyperref*). Cette adresse peut aussi bien concerner une ressource présente sur la même page web, le même site ou un site complètement différent. Le code suivant renvoie ainsi vers le site *Alsacrations* : `<a href="http://www.alsacreations.com/">Un site très utile !</a>`.

Il est également possible d'insérer des images dans une page web via la balise `<img />`. Là encore, un attribut (`src`) doit indiquer la source de l'image (soit une URL, soit l'adresse de l'image sur le serveur de la page web). En plus de `src`, il est également important que la balise `img` possède l'attribut `alt`, qui donne une description *alternative* de l'image si cette dernière ne se charge pas (ou qui pourra être lue par un logiciel spécialisé pour les personnes mal/non-voyantes). Par exemple, la balise `` affichera le logo de Firefox.

Il est également possible d'intégrer d'autres éléments à votre page web, comme de la musique ou des vidéos. Ces éléments dépassant le cadre de ce cours, il ne seront pas abordés ici. Pour avoir plus de détails sur ces éléments et sur les autres "nouveauautés" d'HTML 5, rendez-vous sur <http://www.alsacreations.com/article/lire/750-HTML5-nouveautes.html>.

### 1.2.3 Les autres éléments présents dans une page web

En plus des éléments d'en-tête, donnant des informations sur la page web elle-même, et le corps d'une page web, contenant les informations qui seront affichées à l'utilisateur, il est également possible de placer d'autres éléments dans une page web.

**Commenter son code** Les commentaires sont des éléments qui ne seront pas affichés à l'utilisateur. Ils peuvent être placés n'importe où, entre les balises `<!--` et `-->`.

**Eléments non-HTML** Il est possible d'introduire du code CSS ou JavaScript dans une page web à l'aide des balises `<style></style>` et `<script></script>`. Si le CSS est traité dans la partie 2, le JavaScript sera lui traité au second semestre dans le cours d'*Applications du Web*.

## 1.3 Liste non exhaustive des balises HTML

Au-delà de leur rôle, on peut séparer les balises en deux types selon leur comportement dans une page web :

- les balises de type **bloc** se comportent comme des conteneurs et vont servir à structurer une page web. Elles peuvent contenir d'autres balises blocs ou des balises en-ligne. *C'est le cas des balises `div`, `p` ou encore `li`.*
- les balises de type **en-ligne** doivent se trouver à l'intérieur de balises blocs et ne peuvent pas contenir de balises bloc. *C'est le cas des balises `strong` ou `em`.*

Cette distinction tend à être abandonnée avec l'arrivée d'HTML 5. Cependant, elle reste utile pour comprendre la façon dont s'affiche certaines balises par défaut (cf. section 2).

La liste des balises les plus utilisées est disponible en annexe de ce document. Elle est conçue comme un aide-mémoire où les balises sont triées par domaine puis par ordre alphabétique.

Pour une liste exhaustive de toutes les balises HTML, le *Mozilla Developer Network* (<https://developer.mozilla.org/fr/docs/Web/HTML>), le site *Alsacrations* (<http://www.alsacreations.com/>) ainsi que dans une moindre mesure celui de *W3Schools* (<http://www.w3schools.com/tags/default.asp>) sont conseillés.

## 2 Les feuilles de style CSS

### 2.1 Séparation du fond et de la forme

Les feuilles de style en cascade (ou *Cascading Style Sheets*, CSS) permettent de définir la présentation des documents HTML. On peut donc y modifier le style de sa page web, notamment les polices, les couleurs, les marges, la hauteur ou la largeur de la page, des images et le positionnement des différents éléments.

On peut ainsi facilement séparer le *fond* de la page web (le code HTML) de sa *forme* (les feuilles CSS). Pour rappel, les balises HTML `<strong>` et `<em>` ne doivent donc pas être utilisées pour mettre en forme le texte mais simplement pour marquer son importance. C'est dans la feuille de style que l'on modifiera l'apparence de ces éléments importants (gras, italique, mais aussi couleur, soulignement, clignotement ...).

### 2.2 Fonctionnement d'une feuille de style

#### 2.2.1 Indiquer l'emplacement de la feuille de style au fichier HTML

Lorsque le navigateur charge une page web, il va lire les informations contenues dans les balises `<head>` pour afficher correctement la page. C'est donc entre ces balises que se place le lien de la feuille de style attachée au document HTML : `<link href="style.css" rel="stylesheet" type="text/css" />`.

*Il est également possible d'intégrer du code CSS dans une page web via la balise `<style></style>` ou l'attribut `style` plaçable dans n'importe quelle balise. Ces méthodes sont cependant déconseillées car elle ne permettent pas de facilement séparer le fond de la forme dans une page web !*

#### 2.2.2 Organisation de la feuille de style

Une feuille de style contient une ou plusieurs règles CSS décrivant comment les éléments HTML d'une page web doivent s'afficher. Chaque règle possède :

- un **sélecteur** permettant de déterminer à quel élément HTML s'applique la règle.
- une déclaration qui contient la **propriété** définie par la règle (couleur, police, taille du texte ...) et sa **valeur** (rouge, arial, 12px ...).

Une règle CSS suit donc la syntaxe suivante :

```
selecteur_1, ..., selecteur_n {  
    propriete_1: valeur_1;  
    ...  
    propriete_n: valeur_n;  
}
```

Il est donc possible de combiner plusieurs sélecteurs et plusieurs propriétés dans une même règle.

**Note :** en CSS, les commentaires sont encadrés par les symboles `/*` et `*/`.

Comme l'indique leur nom, les feuilles de styles sont dites *en cascade*. Si plusieurs règles affectent la même propriété d'un même élément, c'est la règle la plus précise qui l'emporte (celle qui possède le plus de sélecteurs). Si les deux règles possèdent la même précision, c'est la règle écrite en dernier qui *écrase* la règle antérieure.

#### Exemples :

Cette règle permet de sélectionner tous les titres de grande importance (*h1*) et de modifier la taille de leur police (*font-size*) à 12 pixels de hauteur (*12px*).

```
h1 {  
    font-size: 12px;  
}
```

Cette règle permet de sélectionner tous les paragraphes et les titres de grand importance (*p*, *h1*) et de modifier leur couleur (*color*) en rouge (*red*) et de mettre leur style (*font-style*) en italique (*italic*).

```
p, h1 {  
    color: red;  
    font-style: italic;  
}
```

**Remarque :** dans l'exemple précédent, les titres de grande importance (*h1*) seront à 12 pixels **ET** en rouge.

## 2.3 Les sélecteurs

### 2.3.1 Les sélecteurs simples

Les sélecteurs basiques correspondent aux noms des éléments HTML à sélectionner. Ainsi, pour sélectionner tous les textes compris entre les balises `<p>` d'une page, il suffit juste de mettre `p` en sélecteur.

On veut parfois pouvoir sélectionner seulement certains éléments d'un même type. Il est alors possible de définir, dans le code HTML, des **classes** ou des **identifiants** pour chaque balise via les attributs `class` ou `id` (comme avec la balise `<div id="content">`). Plusieurs balises peuvent appartenir à une même classe mais un identifiant n'est valable que pour une seule balise.

Le sélecteur d'une classe correspond au nom de cette classe précédé d'un point. De même pour un identifiant, le nom est alors précédé d'un dièse (penser aux mots-dièse de *Twitter* peut aider à s'en rappeler, `#ViveLaFrancophonie`).

**Exemple :**

`#titre_important` sélectionne l'élément HTML possédant l'attribut `id="titre_important"`.

`.conseil` sélectionne tous les éléments marqués de l'attribut `class="conseil"`.

### 2.3.2 Les sélecteurs avancés

Il est possible de sélectionner **un type d'élément contenu dans un autre** type d'élément : `"h1 em"` sélectionnera tous les textes mis en emphase dans les titres `h1`. Si l'on ne veut sélectionner que les éléments **directement** contenus dans un autre, on utilisera alors `"h1 > em"`.

On peut également sélectionner tous les éléments **situés à côté** d'un autre type d'élément : `"h1 + p"` sélectionnera tous les paragraphes placés juste après les titres `h1`. Il est aussi possible de ne sélectionner que les éléments d'un **certain type appartenant à une classe donnée**. Par exemple, `"p.conseil"` sélectionnera tous les éléments paragraphes marqués de l'attribut `class="conseil"`.

Il existe également en CSS des **pseudo-classes** pour les ancres HTML (les `<a>`). Ces pseudo-classes permettent de définir l'apparence des liens de façon plus précise :

**a :link** Sélectionne les liens non-visités.

**a :visited** Sélectionne les liens visités.

**a :hover** Sélectionne les liens survolés (apparence du lien lorsqu'il est survolé).

**a :active** Sélectionne les liens activés (apparence du lien lorsqu'il est cliqué).

Il existe bien entendu bon nombre d'autres sélecteurs qu'il est possible d'utiliser pour plus d'efficacité :

- `::first-line` (`::first-letter`) permet de ne sélectionner que la première ligne (lettre) d'une balise. *p::first-line sélectionnera donc la première ligne de tous les paragraphes de la page.*
- `:first-child` (`:last-child`) permet de sélectionner un élément s'il est le premier (dernier) enfant d'un élément. `p:first-child` sélectionnera tous les paragraphes qui sont les premiers enfants d'une autre balise.

Une liste plus conséquente de sélecteurs est disponible sur le site de *W3Schools* ([http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)).

## 2.4 Les propriétés CSS

Tout comme pour les éléments HTML, une liste non exhaustive des propriétés les plus utilisées est disponible en annexe de ce document. Elle est conçue comme un aide-mémoire où les balises sont triées par domaine puis par ordre alphabétique. Là encore, les sites précédemment conseillés (*MDN*, *W3Schools* ...) peuvent être utiles pour consulter la liste complète des propriétés CSS.

## 2.5 Le modèle des boîtes en CSS

Les boîtes constituent la structure de base de l’affichage d’une page HTML. La plupart des balises HTML est en effet représentée par une boîte dans le navigateur. Une balise imbriquée dans une autre dans le code HTML sera ainsi représentée par une boîte imbriquée dans une autre. On appelle cette organisation le *modèle de boîtes*.

### 2.5.1 La structure d’une boîte

Une boîte est caractérisée par quatre éléments :

- la marge extérieure (*margin*), l’espace entre cette boîte et les boîtes qui l’entourent.
- la bordure (*border*), par défaut, cette dernière a une épaisseur de 0 pixel et n’est donc pas visible.
- la marge intérieure (*padding*), l’espace entre la bordure et le contenu de la boîte.
- le contenu (*content*), ne dépendant pas du CSS mais du code HTML.

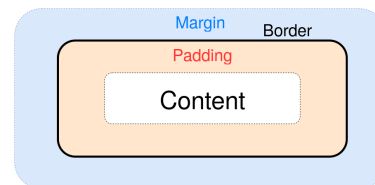


FIGURE 2 – Modèle de boîtes.

Si la balise correspondant à la boîte possède un arrière-plan, celui-ci ne sera affiché qu’à l’intérieur de la bordure, c’est à dire derrière le contenu et la marge intérieure. De plus, les marges intérieures et extérieures sont toujours transparentes.

Les propriétés CSS `margin`, `border` et `padding` permettent de modifier la taille des marges et de la bordure (0 correspondant à l’absence de marge/bordure). La taille peut-être exprimée en différentes unités (pixels `px`, centimètres `cm`, pouces `in` ...) ou en pourcentage de la taille de la boîte (%).

Il est également possible de modifier les marges et la bordure uniquement sur un côté du bloc en suffixant la propriété CSS correspondante par `-top`, `-left`, `-bottom`, `-right`. Par exemple, `margin-top` ne modifie que la marge extérieure située au-dessus de l’élément et `padding-right` ne modifie que la marge intérieure du côté droit de l’élément.

**Astuce :** les propriétés `padding` et `margin` agissent également comme des raccourcis selon le nombre de valeurs qui leur est affecté :

- avec **une seule** valeur, la propriété affecte tous les côtés de l’élément.
- avec **deux** valeurs, la *première* valeur affecte le haut et le bas et la *seconde* les côtés de l’élément.
- avec **trois** valeurs, la *première* valeur affecte le haut, la *seconde* les côtés et la *troisième* le bas de l’élément.
- avec **quatre** valeurs, les valeurs sont affectées dans l’ordre suivant : *haut, droite, bas, gauche* (comme le nommage des points d’un carré en géométrie).

#### Exemples :

Haut et bas à 1 pouce (`in` pour inches) et les côtés à 2 pouces.

```
p {  
    padding: 1in 2in;  
}
```

Haut à 2 pixels, côtés à 20% de la longueur de la boîte et bas à 1 centimètre.

```
p {  
    margin: 2px 20% 1cm;  
}
```

Les deux codes suivant sont équivalents :

```
p {  
    margin: 1em, 2em, 3em, 4em;  
}
```

```
p {  
    margin-top: 1em;  
    margin-right: 2em;  
    margin-bottom: 3em;  
    margin-right: 4em;  
}
```

Pour plus de précisions sur les unités utilisées ici, il est possible de se rendre sur le site de *W3Schools* à l’adresse suivante : <https://www.w3.org/Style/Examples/007/units.fr.html>.

### 3 Annexe 1 - Quelques balises HTML

Les balises soulignées correspondent à des balises de type *inline*, les autres sont des balises de type *block*.

Texte		
Balise	Nom	Description
<u>&lt;br /&gt;</u>	<i>line break</i>	! Cette balise est un élément vide ! Elle force le <b>retour à la ligne</b> .
<u>&lt;em&gt;</u>	<i>emphasis</i>	Met son contenu en <b>emphase</b> , mais de façon plus modérée que <strong>.
<h1>	<i>headings</i>	Délimite un <b>titre</b> . L'importance du titre varie avec la balise utilisée, de <h1> (titre important) à <h6> (titre dérisoire).
<p>	<i>paragraph</i>	Délimite un <b>paragraphe</b> de texte.
<u>&lt;strong&gt;</u>	<i>strong</i>	Permet de <b>mettre en exergue</b> son contenu.

Structuration du code		
<div>	<i>division</i>	Balise <b>neutre</b> utilisée pour regrouper d'autres balises dans un même bloc. L'apparition du XHTML 5 a vu l'introduction de balises plus spécifiques comme <section>, <header> ou <footer>. Ces balises se comportent comme <div> mais renseignent en plus le navigateur sur la <b>signification</b> de leur contenu. Ainsi, <footer> informe le navigateur que son contenu constitue le pied de page. On préférera donc, lorsque cela est possible, utiliser les nouvelles balises à la place de <div>.
<u>&lt;span&gt;</u>	<i>span</i>	Balise <b>neutre</b> utilisée pour regrouper plusieurs mots au sein d'un même texte.

Liens hypertextes		
<u>&lt;a&gt;</u>	<i>anchor</i>	Délimite un <b>lien hypertexte</b> . L'adresse du lien est indiquée via l'attribut <b>href</b> . On peut également afficher une infobulle au survol de la souris via l'attribut <b>title</b> .

Images		
<u>&lt;img /&gt;</u>	<i>image</i>	! Cette balise est un élément vide ! Marque la présence d'une <b>image</b> . Cette balise doit obligatoirement contenir les attributs <b>src</b> (source) et <b>alt</b> (texte alternatif). Le texte alternatif est une courte description de l'image, utile si l'image ne peut pas s'afficher ou si l'utilisateur est malvoyant.

Listes		
<li>	<i>list element</i>	Délimite un <b>élément</b> d'une liste. Cette balise est employée avec les balises <ol> ou <ul>.
<ol>	<i>ordered list</i>	Délimite une <b>énumération</b> , chaque élément étant déclaré via <li>.
<ul>	<i>unordered list</i>	Délimite une <b>liste</b> , chaque élément étant déclaré via <li>.

Tableaux		
<caption>	<i>caption</i>	Délimite la <b>légende</b> d'un tableau.
<table>	<i>table</i>	Délimite un <b>tableau</b> . Chaque ligne du tableau est ensuite déclarée individuellement.
<td>	<i>table datacell</i>	Délimite une <b>cellule de données</b> dans une ligne de tableau.
<th>	<i>table header</i>	Délimite une <b>cellule d'en-tête</b> dans une ligne de tableau.
<tr>	<i>table row</i>	Délimite une <b>ligne</b> de tableau. On déclare ensuite les éléments qui la composent.



## 4 Annexe 2 - Quelques propriétés CSS

Couleur et arrière-plan	
Propriété	Description
<code>background-color</code>	Spécifie la couleur de l'arrière plan. En CSS, il existe une liste de noms de couleur prédéfinis (disponibles ici : <a href="http://www.w3schools.com/cssref/css_colornames.asp">http://www.w3schools.com/cssref/css_colornames.asp</a> ) mais il est également possible d'exprimer n'importe quelle couleur avec son code hexadécimal précédé d'un dièse. (Il est possible d'utiliser des outils en ligne comme <i>paletton</i> pour accéder rapidement aux codes hexadécimaux).
<code>background-image</code>	Place une image en arrière-plan en indiquant la position de l'image via un lien http ( <code>background-image: url("http://www.monsite/photosympa.jpg");</code> ) ou via une adresse locale.
<code>background-repeat</code>	Permet de répéter l'arrière-plan via une mosaïque horizontale, verticale ou les deux.
<code>background-position</code>	Permet de positionner l'arrière-plan par rapport au point en haut à gauche de la fenêtre. Les coordonnées peuvent s'exprimer en plusieurs unités (pixels <code>px</code> , centimètres <code>cm</code> ...), en pourcentage de largeur/hauteur de fenêtre ou via des mots-clés ( <code>top</code> , <code>center</code> , <code>left</code> ...).
<code>background</code>	Raccourci pour régler toutes les propriétés précédentes, si une propriété manque, elle prendra sa valeur par défaut.
<code>color</code>	Définit la couleur du texte.

Police d'un élément	
<code>font-family</code>	Indique une liste de polices à utiliser prioritairement pour l'affichage d'un texte. Si la première police n'est pas disponible sur l'ordinateur du visiteur, le navigateur tentera de charger la seconde <i>etc.</i> Il est donc essentiel de toujours indiquer au moins une police standard en fin de liste ( <code>font-family: arial, verdana, sans-serif;</code> ).
<code>font-style</code>	Spécifie le style de la police ( <code>normal</code> , <code>italic</code> , <code>oblique</code> ).
<code>font-variant</code>	Choisit entre les variantes <code>normal</code> ou <code>small-caps</code> d'une police.
<code>font-weight</code>	Spécifie l'épaisseur de la police ( <code>normal</code> , <code>bold</code> ).
<code>font-size</code>	Règle la taille d'une police en pixels ( <code>px</code> ), pourcentage (%) ...
<code>font</code>	Raccourci pour les précédentes propriétés.

Texte d'un élément	
<code>text-indent</code>	Ajoute une indentation en début de paragraphe.
<code>text-align</code>	Règle l'alignement horizontal d'un texte ( <code>left</code> , <code>right</code> , <code>center</code> ou <code>justify</code> ).
<code>text-transform</code>	Règle la casse d'un texte ( <code>lowercase</code> , <code>uppercase</code> , <code>capitalize</code> ).
<code>text-decoration</code>	Détermine le soulignement d'un texte ( <code>underline</code> , <code>overline</code> , <code>line-through</code> , <code>none</code> ).
<code>letter-spacing</code>	Définit l'espacement entre les caractères du texte.

## Introduction aux technologies du Web

### *Exercices de rappel sur les langages HTML et CSS.*

### Exercice 1 - Un peu d'HTML

Editez le fichier `1_exo.html` pour reproduire le fichier `1_modele.png`. Vous utiliserez exclusivement du HTML.

Précisions :

- le titre de l'exercice est en `h1`.
- le lien 1 doit rediriger vers un site extérieur.
- le lien 2 doit rediriger vers la page `4_exo.html`.
- l'image correspond au fichier `img_tux.png` et doit posséder une description alternative.

### Exercice 2 - Utiliser une feuille de style

Ajoutez une feuille de style CSS au fichier `2_exo.html` afin de reproduire le fichier `2_modele.png`. Vous ne pouvez pas modifier le code HTML.

Précisions :

- toutes les propriétés CSS utilisées ici sont dans les fiches de rappel.
- le lien doit devenir rouge et ne plus être souligné lorsque la souris le survole.
- les couleurs utilisées dans ce TD sont : `black`, `red`, `blue`, `orange`, `magenta` et `purple`.

Rappel :

- il est également possible d'utiliser un code hexadécimal à la place du nom de la couleur (par exemple le code du rouge : `#FF0000`).

### Exercice 3 - Le modèle des boîtes

Ouvrez la page `3_exo.html` dans votre navigateur et la feuille de style `3_feuille_de_style.css` dans un éditeur.

Certaines règles ont déjà été ajoutées, vous les connaissez toutes sauf `border`. Cette propriété permet de régler l'apparence de la bordure d'un élément. La valeur affectée dans cette règle (`solid 2px black`) désigne successivement le style de la bordure (simple trait, double trait, pointillés ...), son épaisseur et sa couleur. Il est possible de préciser `border-xxxx` où `xxxx` désigne la partie de la bordure à modifier (`top`, `bottom`, `left`, `right`).

1. Testez les propriétés suivantes sur le bloc 0 et prenez le temps de bien identifier les différentes zones de la boîte (contenu, padding, bordure, margin) :

- `margin` à 1em, 100px, 10%, 2cm.
- `padding` à 1em, 100px, 10%, 2cm.
- faites varier l'épaisseur de la bordure.
- essayez de ne changer que le padding ou margin du dessus, dessous ou des côtés (`margin-top ...`).
- essayez de donner quatre valeurs différentes au margin (ou le padding) du dessus, dessous ou des côtés.

Par défaut, la balise `p` s'affiche comme une balise bloc, il est cependant possible de changer son mode d'affichage à l'aide de la propriété `display`.

2. Appliquez les propriétés suivantes aux blocs 1, 2 et 3 :
  - `margin` à 2em
  - `padding` à 2em
  - faites varier la propriété `display` entre les valeurs `block`, `inline`, `inline-block`, que remarquez-vous à chacun de ces changements ?

## Exercice 4 - Une liste horizontale ?

Ouvrez la page `4_exo.html` dans votre navigateur et la feuille de style `4_feuille_de_style.css` dans un éditeur.

Le but de cet exercice est d'améliorer l'apparence de cette lsite en changeant la façon dont ses éléments vont s'afficher pas à pas. A chaque étape, prenez le temps d'observer en quoi les changements de style influencent l'apparence de la page web :

1. Changez la façon dont s'affiche les éléments de la liste pour du `inline-block`.
2. Supprimez le soulignement des liens, changez leur couleur en violet (`purple`) et affichez les en PETITES MAJUSCULES.
3. Ajoutez un fond bleu clair (`lightblue`) aux liens.
4. Ajoutez un `padding` de 20px et un `margin` de 10px aux liens.
5. Changez la façon dont s'affiche les liens pour du `inline-block`.
6. Ajoutez une bordure à simple trait violette de 2px d'épaisseur aux liens.
7. Changez la couleur de fond des liens en blanc lorsqu'ils sont survolés.

Question bonus :

- Pourquoi a-t-il été nécessaire d'utiliser deux fois la propriété `display : inline-block` ?