

# Principes des langages de description (X)HTML(5)

## Sommaire

- I. Principes des langages de description (X)HTML(5)
- II. Présentation d'HTML 5
- III. Balises et attributs
- IV. DOM et navigateurs
- V. Introduction au CSS3
- VI. Les sélecteurs
- VII. Tour d'horizon des fonctionnalités
- VIII. Concepts avancés  
(mediaQueries, Responsive design, transitions)
- IX. Construction et bonnes pratiques HTML5 / CSS3  
(HTML5 Boilerplate, Bootstrap Twitter, Foundation,...)
- X. Introduction au JavaScript
- XI. UI (User Interface) - UX (User eXperience)

# I. Principes des langages de description (X)HTML(5)

## Présentation des 2 langages :

**HTML** - HyperText Markup Language est le format de données structurant une page web.

Ce langage est composé de balises, elles mêmes paramétrable grâce aux attributs. Nous verrons plus loin la structure d'une page HTML.

Un document HTML, peu importe la version utilisée, doit être déclaré par le biais d'une entête. Sans quoi votre page HTML n'est pas **valide** au regard des navigateurs web. Ils interpréteront alors votre page à leur guise ou plutôt en fonction des spécifications que leur auront introduit leur développeur. Ainsi Internet Explorer n'interprétera pas votre page de la même manière que Firefox ou Safari... ou un autre.

Trois éléments sont nécessaires pour déclarer une page :

### a) L'entête

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

Le **DTD** (Document Type Declaration) est la référence du format HTML mis en place par le W3C.

### b) La langue

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
```

La langue peut être changer dans le corps de la page pour une ligne ou un paragraphe. Par exemple, vous souhaitez afficher un texte français suivit de sa traduction en anglais. Pour ce faire, vous aurez dans un premier temps déclaré la langue française comme principale de votre document, puis pour les traductions, utilisé l'attribut `xml:lang` au coup par coup :

```
<p>La force est grande dans ma famille !</p>
<p xml:lang="en">The force is strong in my family !</p>
```

### c) Le type de contenu ou charset que l'on intégrera dans la balise <head> sous forme de balise META

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
```

La première partie de cette ligne est le type MIME du fichier.

Ici de l'HTML : `content="text/html;`

La deuxième partie est le jeu de caractères.

Ici de l'unicode : `charset=UTF-8"`

On pourra utiliser aussi iso-8859-1 ou iso-8859-15.

L'encodage le plus répandu est UTF-8.

## Historique des langages :

- 1991 HTML
- 1994 HTML 2
- 1996 CSS 1 + JavaScript
- 1997 HTML 4
- 1998 CSS 2
- 2000 XHTML 1
- 2002 Tableless Web Design (web Design sans tables)
- 2005 AJAX
- 2009 HTML 5



Le **World Wide Web Consortium**, est un [organisme de normalisation](#) à but non lucratif, fondé en [octobre 1994](#) chargé de promouvoir la compatibilité des technologies du [World Wide Web](#).

Le leitmotiv du W3C est

**« Un seul web partout et pour tous »**

### 1.1 Présentation de XHTML 1.0

Le XHTML est une transposition de la syntaxe HTML 4.

Voici les principales différences entre les 2 langages :

- Toute balise ouvrante doit être fermée, et les balises dites «vides» sont écrites avec un slash à la fin (par exemple: `<br />` au lieu de `<br>`).
- Les noms des balises et des attributs sont écrits en minuscules.
- Les valeurs des attributs sont encadrés par des 'quotes' (apostrophes droites) ou des "double quotes" (guillemets droits).
- Chaque attribut doit avoir une valeur (pas d'attribut vide comme `checked`, qui doit être écrit `checked="checked"`).
- Les éléments HTML doivent être correctement imbriqués :
- (`<strong><span>contenu</span></strong>` et pas `<strong><span>contenu</strong></span>`).

Voici la bonne manière de déclarer votre page en XHTML 1.0 de manière stricte

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
```

Cette entête ne vous autorisera aucune autre sémantique que celle de XHTML 1.0.

Si en revanche vous avez appris le langage HTML en version 4 et que vos réflexes ont la vie dure, vous pouvez opter pour une entête transitionnelle :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
```

Ainsi vous pourrez *mixer* les deux versions de langage XHTML 1.0 et HTML 4.1 .

## II. Présentation d'HTML 5



HTML 5 est la dernière version parue. La sémantique est un mixe d'HTML 4 et de XHTML 1, mais y introduit de nouvelles balises, de nouveaux attributs et de nouvelles API.

Dans cette nouvelle version, certaines balises se sont vues améliorées quand d'autres, nouvelles, ouvrent des perspectives multimédia étendues et jusque là complexes à mettre en œuvre (audio, vidéo et SVG).

En définitif, l'HTML 5 n'est plus un «simple» langage d'affichage de contenu, mais étend ses possibilités d'interaction avec le «Drag and Drop» ou la possibilité de mettre en place des notifications «Push».

**Ressources :** <http://www.w3schools.com/tags/default.asp>

### 2.1 L'entête

La déclaration a été revue et véritablement simplifiée :

```
<!doctype html> <!-- plus besoin de se référer au DTD -->
<html lang="fr">
<head>
  <meta charset="utf-8">
</head>
```

### 2.2 Les nouvelles balises

Voici les nouvelles balises parues dans la sémantique HTML 5 :

### 2.3 Description de la relation des liens

L'attribut a évolué. `rel` sera placé sur les balises `<a ...>`, `<link ...>` et `<area ...>`. Il permet de définir la nature de l'élément lié et sa relation.

```
<link rel="alternate" type="application/rss+xml" href="http://myblog.com/feed" />
<link rel="icon" href="/favicon.ico" />
<link rel="pingback" href="http://myblog.com/xmlrpc.php" />
<link rel="prefetch" href="http://myblog.com/main.php" />
...
<a rel="archives" href="http://myblog.com/archives">old posts</a>
<a rel="external" href="http://notmysite.com">tutorial</a>
<a rel="license" href="http://www.apache.org/licenses/LICENSE-2.0">license</a>
<a rel="nofollow" href="http://notmysite.com/sample">wannabe</a>
<a rel="tag" href="http://myblog.com/category/games">games posts</a>
...
```

## 2.4 Les attributs ARIA

(**A**ccessible **R**ich **I**nternet **A**pplications ou Applications Internet riches accessibles).

ARIA est une spécification du W3C qui permet d'ajouter de la sémantique ou des métas-données à un élément HTML, sous forme de rôles, d'états ou de propriété.

<code>&lt;article&gt;</code>	Détermine une section de la page propre à un contenu spécifique, comme un article
<code>&lt;aside&gt;</code>	Utilisée pour définir des informations complémentaires au sein d'une balise affichant un contenu spécifique
<code>&lt;audio&gt;</code>	Permet de faire référence à un média sonore et de le lire (musique, flux audio en streaming)
<code>&lt;bdi&gt;</code>	Permet d'isoler du texte dont la direction est différente du reste du texte qui l'entoure
<code>&lt;canvas&gt;</code>	Permet d'afficher des éléments graphiques générés en javascript
<code>&lt;command&gt;</code>	Définit une commande à exécuter (fonctionnement équivalent à la balise button)
<code>&lt;datalist&gt;</code>	Détermine une liste de données sélectionnables
<code>&lt;details&gt;</code>	Précise les détails supplémentaires qui peuvent être masqués ou affichés sur demande
<code>&lt;embed&gt;</code>	Définit un contenu embarqué, comme un plug-in
<code>&lt;figcaption&gt;</code>	Légende pour la balise figure
<code>&lt;figure&gt;</code>	définit un bloc d'informations pouvant contenir des images, des diagrammes, des photos, du code .....
<code>&lt;footer&gt;</code>	Définit le pied de page d'une section ou du document.
<code>&lt;header&gt;</code>	Représente l'en tête d'une section ou du document.
<code>&lt;hgroup&gt;</code>	Définit une section regroupant un ensemble de titres lorsque plusieurs niveaux de titres (de type <code>&lt;h1&gt;</code> ... <code>&lt;h6&gt;</code> ) sont présents
<code>&lt;keygen&gt;</code>	Permet de générer une clé sécurisée
<code>&lt;mark&gt;</code>	Met en valeur un texte marqué en le surlignant
<code>&lt;menu&gt;</code>	Définit un menu pouvant être constitué de commandes et d'autres menus enfants
<code>&lt;meter&gt;</code>	Permet de situer une mesure entre un minimum et maximum déterminé, pour afficher une jauge
<code>&lt;nav&gt;</code>	Définit une section dédiée à la navigation dans le site, comme un menu
<code>&lt;output&gt;</code>	Représente le résultat d'un calcul
<code>&lt;progress&gt;</code>	Définit une barre de progression permettant de donner l'état d'avancement d'une tâche
<code>&lt;ruby&gt;</code> , <code>&lt;rt&gt;</code> et <code>&lt;rp&gt;</code>	Annotations ruby
<code>&lt;section&gt;</code>	Définit une section regroupant des éléments d'un même thème dans la page

<code>&lt;source&gt;</code>	Permet de définir des contenus alternatifs pour le fichier vidéo (en cas de non support du format)
<code>&lt;summary&gt;</code>	Permet de définir un titre visible pour la balise details
<code>&lt;svg&gt;</code>	Permet de réaliser des dessins vectoriels
<code>&lt;time&gt;</code>	Définit de façon complète la sémantique d'une date ou d'une heure.
<code>&lt;track&gt;</code>	Permet d'insérer un sous-titre (au format WebVTT) à une vidéo affichée avec la balise video.
<code>&lt;video&gt;</code>	Permet d'insérer un contenu video en streaming
<code>&lt;wbr /&gt;</code>	Permet de définir des indications en cas de retour à la ligne au sein d'un texte

Entre autre, elles facilitent l'accessibilité des pages web aux personnes handicapées (mal-voyantes et aveugles). De la même manière, l'attribut **alt** (description des images) permet aux logiciels Lecteurs d'écran, une lecture à haute voix de la navigation en court.

L'attribut **role** définit le type de l'objet auquel il est attribué :

```
<ul id="tree1"
  role="tree"
  tabindex="0"
  aria-labelledby="label_1">
  <li role="treeitem" tabindex="-1" aria-expanded="true">Fruits</li>
  <li role="group">
    <ul>
      <li role="treeitem" tabindex="-1">Oranges</li>
      <li role="treeitem" tabindex="-1">Pineapples</li>
      ...
    </ul>
  </li>
</ul>
```

Lire cet article très complet sur ARIA : <http://letrainde13h37.fr/8/aria-il-serait-temps-de-sy-mettre/>

## 2.5 Nouveau types de formulaires

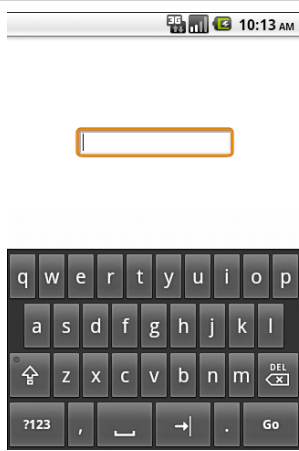
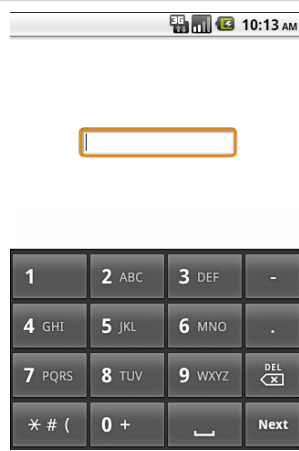
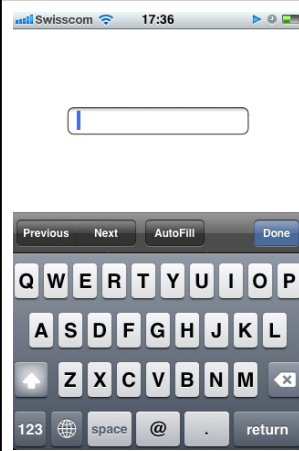
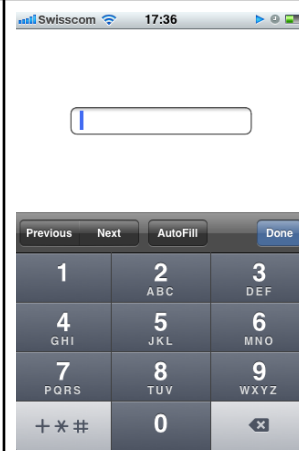
### a) Sur navigateur Web

<code>&lt;input type="text" required /&gt;</code>	<input type="text"/>
<code>&lt;input type="email" value="votre@email.com" /&gt;</code>	<input type="email" value="votre@email.com"/>
<code>&lt;input type="date" min="2010-08-14" max="2011-08-14" value="2010-08-14" /&gt;</code>	<input type="date" value="2010-08-14"/>
<code>&lt;input type="range" min="0" max="50" value="10" /&gt;</code>	<input type="range" value="10"/>
<code>&lt;input type="search" results="10" placeholder="Recherche..." /&gt;</code>	<input type="search" value="Recherche..."/>

<code>&lt;input type="tel" placeholder="(555) 555-5555" pattern="^\(?\d{3}\)?[-\s]\d{3}[-\s]\d{4}.*\$" /&gt;</code>	<input type="text" value="(555) 555-5555"/>
<code>&lt;input type="color" placeholder="e.g. #bbbbbb" /&gt;</code>	<input type="text" value="e.g. #bbbbbb"/>
<code>&lt;input type="number" step="1" min="-5" max="10" value="0" /&gt;</code>	<input type="text" value="0"/>

## b) Sur mobile

Les types de champs `<input>` feront se déployer, sur le mobile, un clavier contextuel.

<code>type="text"</code>	<code>type="number"</code>	<code>type="email"</code>	<code>type="tel"</code>
			

## 2.6 Créer une page HTML

### a) Les outils pour créer une page HTML

Pour créer une page HTML il vous faut un logiciel basique tel que Notepad (Windows) ou TextEdit (Mac). Un navigateur web pour visualiser le rendu de votre code. Vous n'avez pas besoin d'être connecté à internet pour visualiser votre travail.

### b) Les éléments obligatoires dans une page HTML

```
<!doctype html>
<html lang="fr">
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

**L'entête :** `<!doctype html>`

**L'élément html :** `<html lang="fr">`

C'est la délimitation du document HML. On y déclare la langue primaire du document avec l'attribut `lang`.

**L'élément head : <head>**

C'est la section dans laquelle on va déclarer les balises meta, appeler les feuilles de style, donner un titre à la page...

**L'élément body : <body>**

C'est le corps du document. On va y coder tous les éléments qui le constitue : Menus, textes, images etc.. Une fois cette structure posée, on va l'agrémenter un peu.

Ajoutons un titre à notre page web et insérons un paragraphe :

```
<!doctype html>
<html lang="fr">
  <head>
    <title>Site internet pour s'amuser</title>
  </head>
  <body>
    <p>Hourra ! Mon premier site Web.</p>
  </body>
</html>
```

La balise <title> est importante car, avec le concours des balises <meta>, servira aux moteurs de recherche.

Il ne nous reste plus qu'à enregistrer notre document. Le premier fichier qu'un serveur web va essayer de trouver se nomme index.html .

**c) Les couleurs et leur utilisation**

En HTML, les couleurs peuvent avoir différentes syntaxes.

Il en existe des couleurs que l'on peut appeler par leur nom (anglais). Mais elles sont basiques.

black (exemple)	blue (exemple)	fuchsia (exemple)
gray (exemple)	green (exemple)	lime (exemple)
maroon (exemple)	navy (exemple)	olive (exemple)
purple (exemple)	red (exemple)	silver (exemple)
teal (exemple)	white ( )	yellow (exemple)

Pour aller plus loin dans la personnalisation des couleurs, il existe le système Hexadécimal.

Ce système est issu du modèle RVB (Rouge Vert Bleu) - RGB en anglais.

Où la syntaxe est :

#000000 pour le noir - peut s'écrire aussi #000 ou black

#ffffff pour le blanc - peut s'écrire aussi #fff ou white

#ff0000 pour le rouge - peut s'écrire aussi red ou encore #f00

#333333 pour le gris sombre - peut s'écrire aussi #333

#f1f1f1 pour un gris très clair - ne peut s'écrire que comme ça.

Soit : #RRVBB - Liste des couleurs : <http://html-color-codes.info/Codes-couleur-HTML/>

Pour attribuer une couleur à un élément, on peut le faire via un attribut ou via CSS.

Avec un attribut style, voici ce que cela donne dans votre code :

```
<p style="background-color:#ff000; color:#fff;">Bonjour et bienvenue sur mon site.</p>
```

Bonjour et bienvenue sur mon site.



## III. Balises et attributs

### 3.1 Les balises

Les balises sont des étiquettes qui servent à marquer **le début et la fin d'un élément**.

Une balise commence toujours dans par le signe inférieur `<` et finissent toujours par le signe supérieur `>`.

La règle en HTML est que toute balise ouverte doit être fermée et ce en respectant l'ordre d'apparition des balises dans le code.

Une balise ouvrante pour un paragraphe s'écrit comme suit : `<p>`

Et la balise fermante pour ce même paragraphe s'écrit comme suit : `</p>`

La différence entre les deux balises est l'ajout du `/` dans la balise de fermeture. Le contenu viendra s'insérer entre ces balises ouvrante et fermante.

Liste exhaustive des balises HTML : <http://www.codeshttp.com/baliseh.htm>

Par exemple, voici comment on écrit un élément paragraphe en HTML :

```
<p>Bonjour et bienvenue sur mon site.</p>
```

soit :

Bonjour et bienvenue sur mon site.

Pour passer le mot «Bonjour» en gras on rajoute la balise `<strong>` ou `<b>` :

```
<p><strong>Bonjour</strong> et bienvenue sur mon site.</p>
```

soit :

**Bonjour** et bienvenue sur mon site.

Pour passer les mots «mon site» en italique on rajoute la balise `<em>` :

```
<p style="text-align:center;"><strong>Bonjour</strong> et bienvenue sur  
<em>mon site</em>.</p>
```

soit :

**Bonjour** et bienvenue sur *mon site*.

Ces assemblages de balises et de textes constituent des **noeuds**.

Il existe trois types de noeud :

- les noeuds-élément : `<p></p>`
- les noeuds-attribut `align="center"`
- et les noeuds-texte : *Bonjour et bienvenue sur mon site.*

On peut aussi imbriquer deux balises de mise en forme comme `<strong>` et `<em>` :

```
<p><strong>Bonjour</strong> et bienvenue sur  
<strong><em>mon site</em></strong>.</p>
```

soit :

**Bonjour** et bienvenue sur *mon site*.

Les éléments titres sont des balises un peut à part. Il existe 6 types de titre en HTML.

`<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>`. (où **h** est la première lettre de **heading**)

La taille de typographie de la balise h1 est conventionnellement plus grande que celle de la balise h2... etc. jusqu'à h6.

```
<h1>Un titre</h1>  
<h2>Un sous titre</h2>
```

Soit :

Un titre  
Un sous titre

### 3.2 Les balises vides

Comme vu précédemment, une balise s'ouvre et se ferme.

Il existe tout de même des balises dites «vides» qui se suffisent à elles même.

`<br/>` retour à la ligne dans un paragraphe

`<hr/>` séparation de deux éléments, sous forme d'un trait

`` balise image

`<link rel="..." type="..." href="..." />` appel d'un fichier externe

`<meta name="description" content="ma description" />` une balise méta

`<base href="..." target="_blank">` référence pour toutes les url de la page etc...

### 3.3 Les attributs de balise

Liste exhaustive des attributs de balise HTML : <https://developer.mozilla.org/fr/docs/Web/HTML/Attributes>

On peut appliquer aux balises des attributs afin de modifier leur apparence ou leur comportement au sein de la page HTML.

Voici comment se présente un attribut de contenu :

```
<table width="150">
```

Cela signifie que la table (tableau) mesurera 150px.

**Certains attributs, facultatifs, s'appliquent à toutes les balises, comme :**

- Les attributs génériques : `id`, `class`, `style`, `title`.
- Les attributs de langue : `dir`, `lang`.
- Les gestionnaires d'évènements utilisés avec le javascript : `onclick`, `ondblclick`, `onkeydown`, `onkeypress`, `onkeyup`, `onmousedown`, `onmousemove`, `onmouseout`, `onmouseover`, `onmouseup`

### D'autres sont obligatoire comme :

- `src` pour définir la source (URL = Uniform Resource Identifier, soit littéralement identifiant uniforme de ressource) de la balise `img`. Dans un formulaire `form`, on utilisera `action` pour cibler le fichier de traitement du dit formulaire.
- `alt` pour l'accessibilité des images aux lecteurs d'écrans.
- `width` et `height` pour définir la hauteur et la largeur des éléments tels que `img`, `object` et `iframe`.

### Certains attributs peuvent être raccourcis :

- `selected="selected"` pour les cases à cocher, les boutons radios, ... peut être écrit `selected`

```
<select>
  <option value="volvo">Volvo</option>
  <option value="audi" selected>Audi</option>
</select>
```

## 3.4 Les balises élémentaires

### Les titres

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

### Les listes

#### Non ordonnées

```
<ul>
  <li>Voiture</li>
  <li>Caravanes</li>
  <li>Avions</li>
</ul>
```

- Voiture
- Caravanes
- Avions

On peut imbriquer des listes dans des listes

```
<ul>
  <li>Voiture
    <ul>
      <li>Berline</li>
      <li>Citadine</li>
      <li>Grand tourisme</li>
    </ul>
  </li>
  <li>Caravanes</li>
  <li>Avions</li>
</ul>
```

- Voiture
  - Berline
  - Citadine
  - Grand tourisme
- Caravanes
- Avions

### Ordonnées

```
<ol>
  <li>Voiture</li>
  <li>Caravanes</li>
  <li>Avions</li>
</ol>
```

1. Voiture
2. Caravanes
3. Avions

### Les tables

```
<table width="500" border="1">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Johana</td>
    <td>barton</td>
    <td rowspan="2">78</td>
  </tr>
  <tr>
    <td>Robert</td>
    <td>Deniro</td>
  </tr>
  <tr>
    <td>John</td>
    <td colspan="2">Snow</td>
  </tr>
</table>
```

Jill	Smith	50
Johana	Barton	78
Robert	Deniro	
John	Snow	

### Les liens

```
<a href="http://www.google.fr" target="_blank">un lien</a>
```

[un lien](http://www.google.fr)

Pour les liens il existe quatre attributs `target` (cible) qu'il est primordial de connaître :

`_self` : C'est la valeur par défaut. Si aucun attribut n'est déclaré, c'est celui-là qui sera utilisé. il permet d'ouvrir le lien dans la page courante

`_blank` : Cet attribut ouvre une nouvel fenêtre ou un nouvel onglet pour afficher la cible

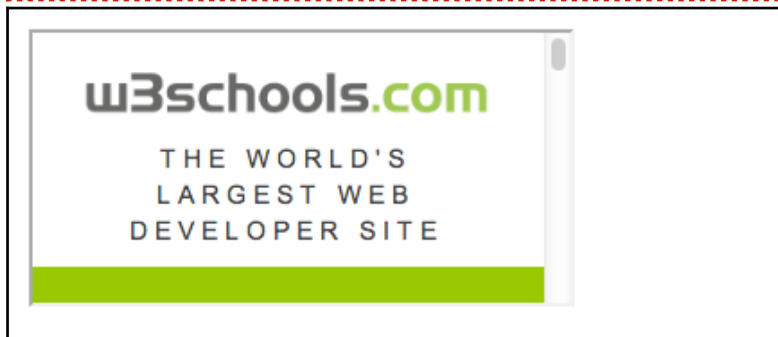
`_parent` : Cet attribut est utilisé pour cibler le cadre parent. On l'utilise quand on développe des pages avec des frame.

`_top` : Ouverture de la page cible dans la fenêtre hôte (par-dessus le `frameset`). Là encore on l'utilise quand on développe des pages avec des frame.

L'utilisation des frame est aujourd'hui rare car dépréciée en HTML. Mais cela peut arriver dans des sites développés en XHTML.

### Les iframe

```
<iframe src="http://www.w3schools.com">
<p>Votre navigateur ne supporte pas les iframe.</p>
</iframe>
```



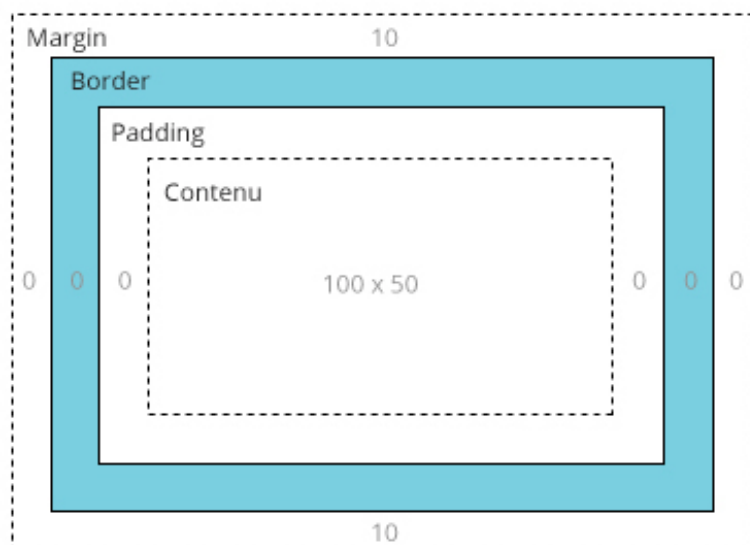
Les iframe peuvent être paramétrés grâce à des attributs :

Attributs	Valeurs	Description
<code>align</code>	left right top middle bottom	<b>Non supporté en HTML5.</b> Spécifie l'alignement d'un <code>&lt;iframe&gt;</code> par rapport à l'élément conteneur
<code>frameborder</code>	1 0	<b>Non supporté en HTML5.</b> Spécifie s'il y a ou non une bordure sur <code>&lt;iframe&gt;</code>
<code>height</code>	pixels	Spécifie la hauteur d'un <code>&lt;iframe&gt;</code>
<code>longdesc</code>	URL	<b>Non supporté en HTML5.</b> Spécifie une page contenant une longue description de l' <code>&lt;iframe&gt;</code>
<code>marginheight</code>	pixels	<b>Non supporté en HTML5.</b> Spécifie le margin top et bottom du contenu d'un <code>&lt;iframe&gt;</code>

marginwidth		pixels	<b>Non supporté en HTML5.</b> Spécifie le margin droite et gauche du contenu d'un <iframe>
name		text	Specifies the name of an <iframe>
sandbox		allow-forms allow-pointer-lock allow-popups allow-same-origin allow-scripts allow-top-navigation	Autorise des restrictions pour le contenu d'un <iframe>
scrolling		yes no auto	<b>Non supporté en HTML5.</b> Spécifie si on affiche ou non une scrollbar <iframe>
seamless		seamless	Spécifie que l'<iframe > devrait ressembler à la partie du document contenant. Non supporté par les navigateurs pour le moment.
src		URL	Spécifie l'adresse du document à afficher dans l'<iframe>
srcdoc		Code HTML	Spécifie le code HTML à écrire dans l'<iframe> Non supporté dans IE
width		pixels	Spécifie la largeur d'un <iframe>

### 3.4 Notion avancée de la structure d'un élément HTML

Un élément HTML se définit comme suit :



## IV. DOM et navigateurs

### 4.1 Le DOM selon le W3C

#### DOM (Document Object Model)

"DOM permet aux programmes et scripts d'accéder et de modifier dynamiquement le contenu, la structure et le style de documents XML ou HTML".

Avant la normalisation par le W3C (World Wide Web Consortium) chaque navigateur web possédait son propre DOM. Ce qui obligeait les intégrateurs/développeurs à écrire leurs documents web en deux versions (au moins) pour que tout le monde puisse visualiser la même chose.

Le programmeur (DOM) dispose d'objets standardisés autour du Javascript, qui ont des propriétés, des méthodes et des événements qui "interfacent" le document XML ou HTML.

#### En résumé :

- Un ensemble d'objets,
- Un modèle pour la façon dont ces objets peuvent être combinés (édité par le W3C),
- Et une interface pour y accéder et les manipuler (le navigateur web).

Autrement dit, c'est une technologie qui lit votre document HTML (ou XML), en extrait les noeuds (balises, attributs...) et vous en restitue une visualisation cohérente, pour tout le monde :  
Un site internet.

#### Cela nous sert à quoi ?

Cela sert à afficher :

Le renard **roux saute** parmi les poules...

Quand vous avez écrit :

```
<p>Le renard <span style="font-weight:bold; color:#cc3300;">roux</span>
<b><em>saute</em></b> parmi les <span style="background-color:#000033;
color:#000;">poules</span>...</p>
```

Au lieu de :

```
<p>Le renard style="font-weight:bold;
color:#cc3300;">roux</span> <b><em>saute</em></b> parmi
les <span style="background-color:#000033;
color:#000;">poules</span>...</p>
```

#### Peut-on voir ce DOM ?

Oui... ou du moins le rendu de son action / travail.

Pour cela, servez-vous des outils tels que Firebug (à installer dans Firefox) ou des panneaux de développement présents nativement dans vos navigateurs web.

Vous aurez l'impression de relire le code HTML que vous venez d'écrire...

#### Mais pas seulement.

Dans la majorité des cas, la représentation visuelle du DOM sera votre code puisque elle en est issue.

Mais il peut arriver que le DOM modifie votre code car il n'était pas conforme au standard.

Admettons que votre code HTML soit celui d'une table :

```
<table style="width:50%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    ...
  </tr>
</table>
```

Jill	Smith	50
Johana	Barton	78

Vous n'avez pas inscrit la balise `<tbody>` qui pourtant est constituante des tables. Et bien, elle aura été ajoutée par le DOM dans la représentation visuelle de votre code.

```
<table style="width:50%">
  <tbody>
    <tr>
      <td>Jill</td>
      <td>Smith</td>
      <td>50</td>
    </tr>
    <tr>
      ...
    </tr>
  </tbody>
</table>
```

Il vous sera possible de la cibler avec du javascript ou de la styliser avec du CSS sans l'avoir même écrit dans votre code. Mais là n'est pas vraiment l'intérêt du DOM, d'autant que si vous codez pas correctement vos pages, Le DOM n'agira pas comme un super héros qui remplace les balises manquantes...

### Manipulation du DOM par le Javascript

On peut manipuler le DOM avec du Javascript. Imaginons un élément présent dans votre code HTML :

```
<div id="maDiv"></div>
```

Une div vide possédant un id. (identifieur)

Et que plus tard dans le document on trouve :

```
<script>
  var container = document.getElementById("maDiv");
  container.innerHTML = "Un nouveau contenu !";
</script>
```

Vous verrez alors à l'écran *Un nouveau contenu !* en lieu et place de ... rien.

Et si vous utilisez votre inspecteur d'éléments, vous pourrez lire :

```
<div id="maDiv">Un nouveau contenu !</div>
```

Le code HTML sera différent de votre code source.

Le Javascript est un langage que le navigateur lit et utilise pour modifier votre code HTML.

mais c'est dans le DOM que cela se passe. De fait, beaucoup de choses que vous pourriez nommer des «*trucs javascript*» sont en fait des «*actions du DOM*»



## 4.2 Les navigateurs web

### a) Les différents navigateurs

Ce sont des logiciels que l'on utilise pour naviguer sur internet et visualiser les sites.

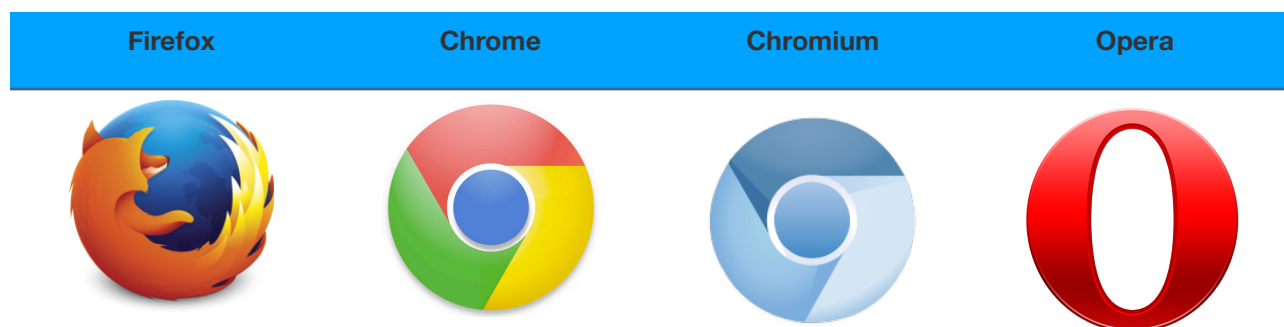
Ils ont tous leurs spécificités, mais ont le même rôle. Chaque OS possède le sien de manière native, mais il est possible de les télécharger et installer séparément.

- Ainsi sur Windows, on peut utiliser par défaut Internet Explorer. Celui-ci est en passe de disparaître et d'être remplacé par un navigateur web plus proche des spécifications W3C. Pour le moment le projet se nomme Spartan.
- Chez Apple le navigateur par défaut est Safari.
- Et enfin pour l'OS Linux Kubuntu, c'est Konqueror que l'on utilisera.



Cependant il est possible de remplacer le navigateur par défaut de votre machine par celui de votre choix. Ainsi vous pourrez opter pour Firefox, Chrome ou Chromium pour Linux etc...

Il existe une multitude de navigateurs pour les OS de bureau ou pour vos smart phones.



Pour connaître la répartition des navigateurs web dans le monde, vous pouvez vous rendre à l'adresse suivante :

<http://gs.statcounter.com/#browser-ww-monthly-201205-201205-map>

Vous trouverez sur ce site toutes les statistiques concernant les navigateurs web les plus répandus.

Vous pouvez aussi tester la compatibilité des navigateurs sur des sites dédiés tels que :

<http://html5test.com/index.html>

## b) Les moteurs de rendu

L'une des phrases que vous risquez d'entendre le plus souvent pendant votre carrière est :  
«Je comprends pas... Sur l'ordinateur de ma collègue, tout est nickel.. mais sur le mien ça va pas !»

**En effet, chaque navigateur web possède son propre moteur de rendu...** et... Oh surprise, votre site internet, fraîchement développé, présentera des différences d'affichage d'un navigateur à l'autre.

Il vous faudra alors tester votre site internet sur différents navigateurs...

Mais là encore, le moteur de rendu peut différer entre la version desktop et la version mobile.

Rassurez-vous, vous n'êtes pas obligé de posséder toutes les versions de navigateur, sur ordinateur et sur smart phone... Il existe des sites qui se chargent d'afficher votre production sur différents moteurs de rendu.

<http://netrenderer.com> pour les rendus sur Internet explorer depuis la version 5.5.

<http://spoon.net/browsers/> pour les utilisateurs de Windows.

<http://quirktools.com/screenfly/> Site proposant une visualisation assez globale.

Voici la liste des moteurs de rendu des navigateurs web (les plus répandus) :

## Webkit

*Développé par Apple et Nokia*

- Safari (Mac)
- Safari (iOS)
- Chrome (iOS)
- Dolphin Browser (Android, iOS)
- Navigateur natif (Android)

## Blink

*Un clone de Webkit développé par Google.*

- Chromium / Chrome (Linux, Windows, Mac)
- Chrome (Android)
- Opera (Linux, Windows, Mac)
- Opera Mini (Android, Windows Mobile)

## Gecko

*Moteur développé par Mozilla*

- Firefox (Linux, Windows, Mac)
- Firefox (Android)

## Trident

*Développé par Microsoft*

- Internet Explorer (Windows)
- Internet Explorer (Windows Mobile)

# Exercice

**Reproduire :**

**Bonjour et bienvenue dans ma librairie virtuelle.**

## Chapitre I

Aujourd'hui le jour se lève, le début d'une nouvelle aventure, celle de la vie durant un jour.

## Chapitre II

Maintenant le soir arrive, la fin de cette aventure extraordinaire qui se déroule aujourd'hui.

## Chapitre III

Au cœur de la nuit les esprits se reposent, préparant ainsi une nouvelle expérience de la vie, celle qui se déroulera demain.  
**Ne vivons pas sans le projet du lendemain !**

Holla	Que tal !
Muy bien !	Y tu ?
Ok	Ok

## Table des matières

### de notre livre électronique

- Chapitre I
- Chapitre II
  1. Introduction
  2. Développement
  3. Synthèse
  4. Antithèse
  5. Conclusion
- Chapitre III
- Chapitre IV

**Correction :**

```

<!doctype html>
<html lang="fr">
  <head>
    <title>Librairie virtuelle</title>
  </head>
  <body style="padding: 0; margin: 0;">
    <section style="background-color: #ccc; padding: 20px;">
      <h1>Bonjour et bienvenue dans ma librairie virtuelle.</h1>
    </section>
    <section style="padding: 20px;">
      <h2>Chapitre I</h2>
      <p>Aujourd'hui le jour se lève, le début d'une nouvelle aventure,
celle de la vie durant un jour.</p>
      <h2>Chapitre II</h2>
      <p>Maintenant le soir arrive, la fin de cette aventure
extraordinaire qui se déroula aujourd'hui.</p>
      <h2>Chapitre III</h2>
      <p>Au cœur de la nuit les esprits se reposent, préparant ainsi une
nouvelle expérience de la vie,
celle qui se déroulera demain.<br/>Ne <strong>vivons pas sans le projet
du landemain !</strong></p>
      <table border="1">
        <tr bgcolor="#ccc">
          <td width="250">Holla</td>
          <td width="250">Que tal !</td>
        </tr>
        <tr>
          <td>Muy bien !</td>
          <td>Y tu ?</td>
        </tr>
        <tr bgcolor="#ccc">
          <td>Ok</td>
          <td>Ok</td>
        </tr>
      </table>
    </section>
    <section style="background-color: #3acf80; padding: 20px; width: 75%;">
      <hgroup>
        <h2>Table des matières</h2>
        <h3>de notre livre élecronique</h3>
      </hgroup>
      <ul>
        <li>Chapitre I</li>
        <li>Chapitre II
          <ol>
            <li>Introduction</li>
            <li>Developpement</li>
            <li>Synthèse</li>
            <li>Antithèse</li>
            <li>Conclusion</li>
          </ol>
        </li>
        <li>Chapitre III</li>
        <li>Chapitre IV</li>
      </ul>
    </section>
  </body>
</html>

```

## V. Introduction au CSS3



### 5.1) Le CSS et les feuilles de style

CSS - Cascading Style Sheets. Le CSS est édité par le W3C.

[http://www.w3schools.com/css/css3\\_intro.asp](http://www.w3schools.com/css/css3_intro.asp)

On peut écrire le CSS de 2 manières. Soit dans une feuille de style ( un fichier \*.css) ou inline. La syntaxe sera toujours la même, à ceci prêt que l'appel ne sera pas géré de la même manière.

Inscrire du code inline :

```
<head>
  <style type="text/css">
    body {
      font-family: Verdana, Arial, Helvetica, sans-serif;
      font-size: 11px;
      color: #ffffff;
    }
  </style>
</head>
...
<!-- Ou au fil du code HTML -->
<section style="background-color: #ccc; padding: 20px;">...</section>
...
```

Pour appeler une feuille de style, la syntaxe et l'endroit d'appel dans la page HTML sont :

```
<head>
  <link rel="stylesheet" href="style.css" media="all">
</head>
```

En HTML5, il n'est plus strictement obligatoire de spécifier l'attribut `type`.

L'attribut `media` permet de cibler les médias de diffusion :

- `screen` Écrans
- `handheld` Périphériques mobiles ou de petite taille
- `print` Impression
- `aural` (CSS 2.0) / `speech` (CSS 2.1) Synthèses vocales
- `braille` Plages braille
- `embossed` Imprimantes braille
- `projection` Projecteurs (ou présentations avec slides)
- `tty` Terminal/police à pas fixe
- `tv` Téléviseur
- `all` Tous les précédents

## 5.2) Le CSS 3

*Quelles sont les nouvelles propriétés dans CSS3 ?*

- @font-face · Utiliser des polices personnalisées pour vos sites
- Animations CSS
  - animation-name
  - animation-duration
  - animation-timing-function
  - animation-iteration-count
  - animation-direction
  - animation-delay
  - animation-fill-mode
- background-clip · Rogner l'arrière-plan des boîtes HTML
- border-radius · Créer des coins arrondis sur les boîtes
- box-shadow · Ajouter une ombre sur les boîtes
- counters
- Dégradés CSS
  - linear-gradient
  - radial-gradient
- element()
- Filtres CSS · Des effets d'images pour le web
- rgba()
- text-shadow · Ajouter une ombre sur les textes
- Transformations 2D
- Transformations 3D
- Transitions CSS · Animer facilement en CSS toutes les propriétés
  - transition-property
  - transition-duration
  - transition-timing-function
  - transition-delay

*Les nouvelles propriété CSS3 ont de nouveaux sélecteurs :*

<http://debray-jerome.developpez.com/articles/les-selecteurs-en-css3/>

### Sélecteurs d'attributs

- [attr^="stringValue"]  
Ce sélecteur permet de sélectionner un élément DOM dont l'attribut "attr" commence exactement par la valeur "stringValue".
- [attr\$="stringValue"]  
Ce sélecteur permet de sélectionner un élément DOM dont l'attribut "attr" finit exactement par la valeur "stringValue".
- [attr\*="stringValue"]  
Ce sélecteur permet de sélectionner un élément DOM dont l'attribut "attr" comporte au moins une fois la valeur "stringValue".

## Combinateur d'adjacence directe

- `div~p`  
Permet d'ajouter un style à tous les éléments qui suivent un élément particulier.

## Pseudo-classes

- `:root`  
Ce sélecteur représente un élément qui est la racine d'un document. Par exemple, en HTML 4, l'élément est `html`.
- `:nth-child(expression)`  
Ce sélecteur permet de cibler tous les éléments en se basant sur leur position dans la liste des enfants de leur parent.

```
.exampleTable{
  width:100%;
  border:1px solid #444;
}
.exampleTable tr:nth-child(even){ /*tous les enfants aux numéros pairs*/
  background:#999999;
  text-shadow: 2px 2px 5px #111;
  color:#fff;
}
.exampleTable tr:nth-child(odd){ /*tous les enfants aux numéros impairs*/
  background:#990000;
  color:#fff;
}
.exampleTable tr:nth-child(3n){ /*tous les 3 enfants*/
  background:#045FB4;
  color:#fff;
}
.exampleTable tr:nth-child(7){ /*1'enfant numéro 7*/
  background:#006400;
  text-shadow: 2px 2px 2px #fff;
  color:#000;
}
```

- `:nth-last-child(expression)`  
Ce sélecteur accepte les mêmes arguments que `:nth-child()` et correspond au dernier enfant d'un élément parent.
- `:last-child`  
Correspond à `:nth-last-child(1)`.
- `:first-child`  
Correspond à `:nth-child(1)`.
- `:nth-of-type(expression)`  
Ce sélecteur représente un élément qui a expression frères du même type devant lui dans l'arbre DOM.
- ...

## Pseudo-éléments

- `::first-line`  
Applique la règle de style à la première ligne du texte de l'élément.
- `::first-letter`  
Applique la règle de style à la première lettre du texte de l'élément.
- `::selection`  
Applique la règle de style à la sélection du texte de l'élément faite par l'utilisateur.
- `::before` et `::after`  
Génère un contenu avant ou après un contenu d'un élément.

### 5.3) Les classes et ID

Les **classes** permettent de cibler un élément, ou un groupe d'éléments HTML. On peut alors les "customiser" à notre guise, tant dans l'apparence que dans son comportement.

Les **classes** peuvent être appelés plusieurs fois dans un document. En revanche, un **ID** ne peut être appelé qu'une seule fois. On se servira des **ID** pour la "customisation" visuelle, mais servira surtout à identifier un élément cible avec le javascript.

Une classe ou un ID ne peuvent JAMAIS commencer par un chiffre ex. : .1548

## VI. Les sélecteurs

### Structure de base :

```
sélecteur{  
  propriété: valeur;  
}
```

### Les sélecteurs simples se divisent en 5 catégories :

- Le sélecteur de type : `h1` | `p` | `div` | `section` | `ul` | `li` | `ol` | ...
- Le sélecteur universel : `*`
- Le sélecteur d'attribut : `[attr]` | `[lang]` | ...
- Le sélecteur d'ID : `#monId` | ...
- La pseudo classe : `:hover` | `:first-child` | ...

### Notions de cascade :

Le DOM va lire votre feuille de style de haut en bas et de gauche à droite comme nous le faisons avec une feuille de papier. Si vous déclarez plusieurs fois la même classe dans votre feuille de style, c'est la dernière occurrence qui pendra le dessus.



## VII. Tour d'horizon des fonctionnalités

Dans la feuille de style :

```
.paragrapheRouge{
  background-color:#ff000;
  color:#fff;
  border-radius:2px;
    -o-border-radius:2px;
    -webkit-border-radius:2px;
  -moz-border-radius:2px;
  -ms-border-radius:2px;
  padding:3px 5px 3px 5px;
}

/* ou */

.paragrapheRouge{ background-color:#ff000; color:#fff; border-radius:2px; -o-
border-radius:2px; -webkit-border-radius:2px; -moz-border-radius:2px; -
ms-border-radius:2px; padding:3px 50px 3px 5px; }

/* Syntaxe d'un ID */

#paragrapheRouge{ background-color:#ff000; color:#fff; border-radius:2px; -o-
border-radius:2px; -webkit-border-radius:2px; -moz-border-radius:2px; -
ms-border-radius:2px; padding:3px 50px 3px 5px; }
```

Dans la page HTML:

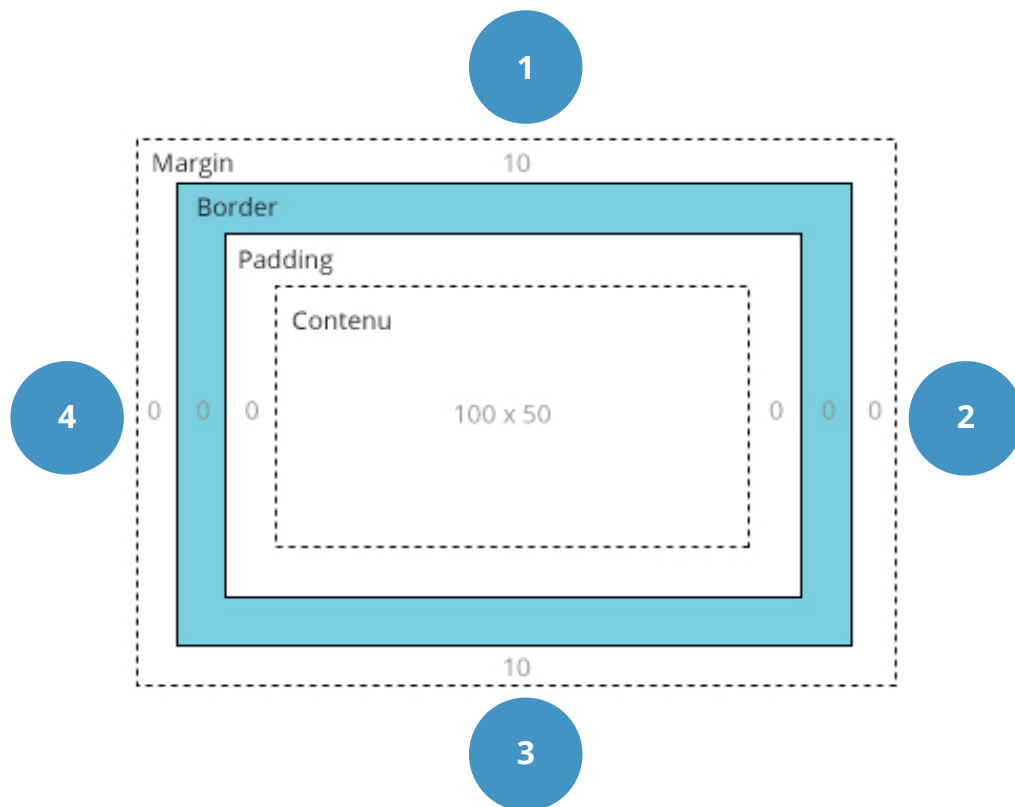
```
<p class="paragrapheRouge">Aujourd'hui le jour se lève, le début d'une
nouvelle aventure, celle de la vie durant un jour.</p>
```

ou

```
<p id="paragrapheRouge">Aujourd'hui le jour se lève, le début d'une nouvelle
aventure, celle de la vie durant un jour.</p>
```

Rendu :

Aujourd'hui le jour se lève, le début d'une nouvelle aventure, celle de la vie durant un jour.



On peut raccourcir les valeurs présentes dans une propriété comme pour les couleurs.  
Pourquoi et comment?

```
padding:3px 5px 3px 5px; = padding:3px 5px;
```

=>

```
padding-top:3px;
padding-right:5px;
padding-bottom:3px;
padding-left:5px;
```

et

```
padding:5px;
```

=>

```
padding-top:5px;
padding-right:5px;
padding-bottom:5px;
padding-left:5px;
```

### Les propriétés de base :

- background : #ff000 url(images/monimage.jpg) no-repeat; | repeat-x | repeat-y | 8px 15px ...
- background-color : #ff000; | transparent
- color : #ff000;
- font-family : "Verdana", arial, sans-serif;
- font-size : 14px; | % | em
- margin : 18px 12px 5px 8px;
- padding : 18px 12px 5px 8px;
- text-align : center; | left | right | justify ...
- text-decoration : none; | underline | overline ...

### Déclaration des tailles en générale :

Les tailles peuvent être déclarées en "px", "%" et/ou en "em"

Tout dépend si on veut qu'elle soit proportionnelles. (<http://pxtoem.com>)

### Les typos sans serif :

- Verdana (Mac/PC)
- Arial (Mac/PC)
- Trebuchet (PC)
- Helvetica (Mac)
- Tahoma (PC)
- Geneva (Mac)

### Les typos serif :

- Times new roman (PC)
- Times (Mac)
- Georgia (Mac/PC)
- Palatino Linotype (PC)
- Palatino (Mac)

### Les web-font :

Les web-fonts sont un super moyen de s'affranchir des typos de base des ordinateurs.

Google web font = <https://www.google.com/fonts>

### Les liens et les effets rollover :

L'élément lien <a> est particulier. Lors de sa déclaration dans les feuilles de style, on doit lui attribuer deux lignes... voir quatre.

```
a:link{ color:#3466e4; text-decoration:underline; }
a:hover{ color:#000; text-decoration:none; }
a:active{ color:#3466e4; text-decoration:underline; }
a:visited{ color:#ff000; text-decoration:underline; }
```

On peut appliquer la pseudo classe à à peu près tous les éléments d'une page HTML.

### Mise en page et positionnement :

En 2002 nous sommes passé au "**tableless design**".

Fini l'utilisation des tables pour construire une page HTML.

D'une part nous avons gagné en légèreté de code et ensuite nous avons gagné en flexibilité. Grâce à cette évolution, nous sommes aujourd'hui à l'ère du "**Responsive design**".

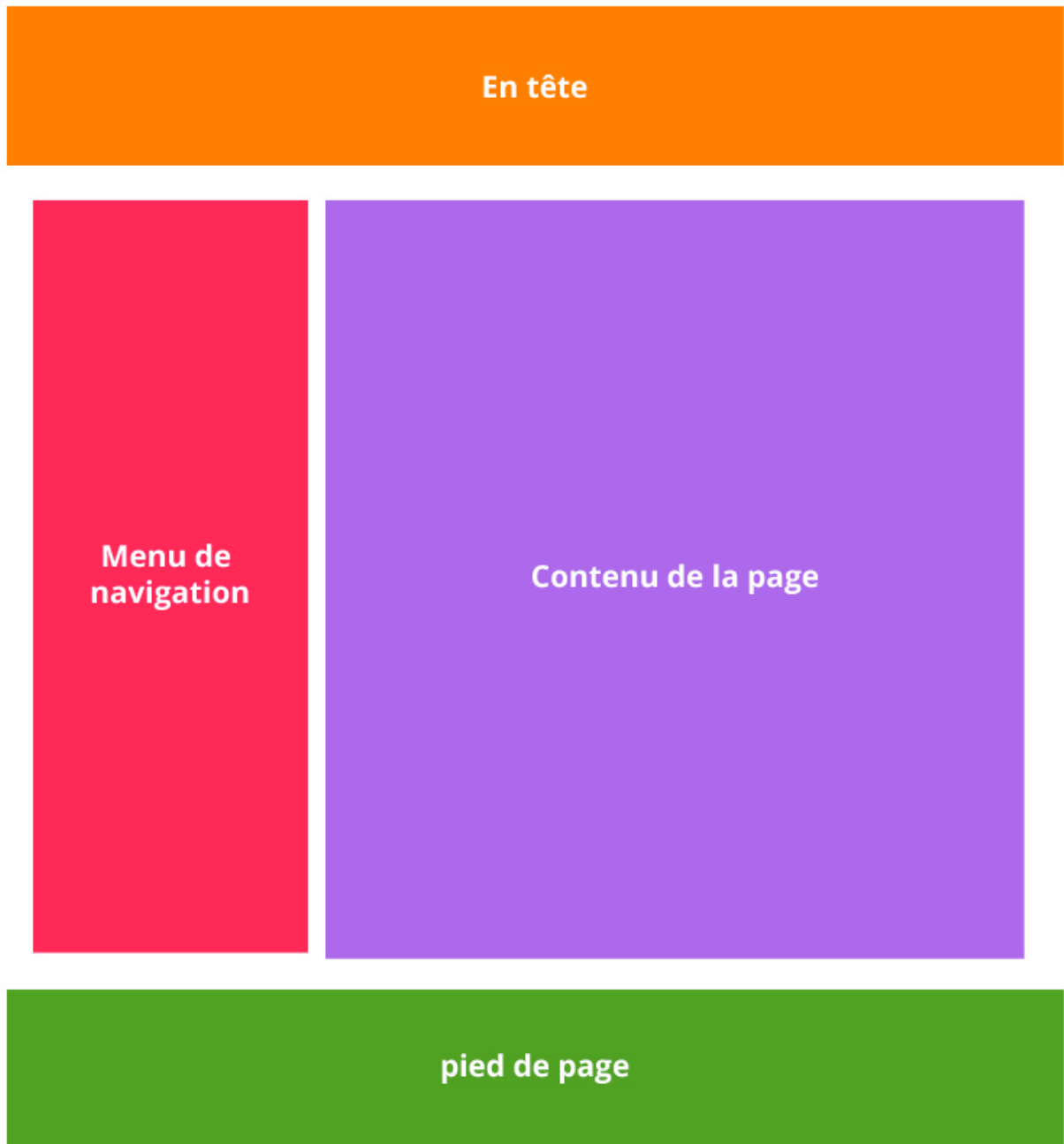
L'heure de l'intégration est arrivée. Nous venons de récupérer les fichiers de la part du designer.

Il faut réfléchir à la meilleure manière de découper tout ça de manière propre, générique et évolutive.

### Les notions importantes :

```
float : left | right | none
clear : left | right | both
overflow : hidden
```

Exemple :



## HTML

```

<!doctype html>
<html lang="fr">
<head>
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css" media="screen">
</head>
<body >
  <div class="entete">
    En tête
  </div>

  <main class="main">
    <nav class="menu">
      Menus
    </nav>

    <article class="contenu">
      Contenu de la page
    </article>
  </main>

  <footer>
    Pied de page
  </footer>
</body>
</html>

```

## CSS

```

header, main, nav, article, footer {
  padding: 1px 0px;
}
header{
  background-color: #FF9900;
  text-align: center;
}
main {
  max-width: 960px;
  margin: auto;
}
nav {
  float: left;
  width: 240px;
  background-color: #FF3366;
}
article {
  margin-left: 245px;
  background-color: #9966FF;
}
footer {
  background-color: #669933;
  text-align: center;
  clear: both;
}

```

Pour que tous les éléments fassent la même hauteur et donner un aspect de table HTML, il faut utiliser la propriété `display` avec le sélecteur `table`. Le contenu de ces cellules s'aligne par défaut en haut, mais on peut changer ça en déclarant un `vertical-align:middle` pour centrer le contenu verticalement, ou un `vertical-align:bottom` pour l'aligner en bas. Soit :

```
header, main, nav, article, footer {
  padding: 1px 0px;
}
header {
  background-color: #FF9900;
  text-align: center;
}
main {
  max-width: 960px;
  margin: auto;
  display: table;
}
nav {
  float: left;
  width: 240px;
  background-color: #FF3366;
  display: table-cell;
}
article {
  margin-left: 245px;
  background-color: #9966FF;
  display: table-cell;
}
footer {
  background-color: #669933;
  text-align: center;
  clear: both;
}
```

## VIII. Concepts avancés

(mediaQueries, Responsive design, transitions)

### Animations en CSS3 :

Avant CSS3, le changement de couleur se faisait immédiatement au rollover d'un élément. Grâce à `transition`, on peut faire en sorte que la transition d'un état se fasse "en douceur". Avant la propriété `transition` il fallait passer par le javascript pour obtenir cela.

**Non compatible IE < 10**

HTML

```
<p class="doigt">
<a href="#doigt">On ne montre pas du doigt !</a></p>
```

CSS

```
.doigt a {
  padding: 10px 10px 10px 77px;
  /* pour les vieux navigateurs */
  background: #FAC27E url(/images/main.png) no-repeat 5px 10px;
  /* pour les navigateurs récents */
  background: rgba(255,140,0,0.4) url(/images/main.png) no-repeat 5px 10px;
  border: 1px solid #FF8C00;
  border-radius: 5px;
}
p.doigt a:hover, p#doigt a:focus {
  color: black;
  background-color: #ff8c00; /* pour les vieux navigateurs */
  background-color: rgba(255,140,0,1);
  transition: background-color 2s linear;
  /* équivaut à :
  transition-property: background-color;
  transition-duration: 2s;
  transition-timing-fonction: linear; */
}
```

Certains navigateurs ont besoin de préfixes :

```
-webkit-transition: background-color 2s linear;
-moz-transition: background-color 2s linear;
-ms-transition: background-color 2s linear;
-o-transition: background-color 2s linear;
transition: background-color 2s linear;
```

### Media-queries CSS3 :

La spécification media-queries permet d'appliquer les mêmes styles différemment en fonction des périphériques se connectant à votre site internet.

Ainsi on pourra changer l'apparence de différents éléments que l'on visualise un même site sur ordinateur, ou sur smartphone.

En CSS2 on pouvait déjà cibler le média de destination. mais cela était limité :

- **screen** Écrans
- **handheld** Périphériques mobiles ou de petite taille
- **print** Impression
- **aural** (CSS 2.0) / **speech** (CSS 2.1) Synthèses vocales
- **braille** Plages braille
- **embossed** Imprimantes braille
- **projection** Projecteurs (ou présentations avec slides)

- `tty` Terminal/police à pas fixe
- `tv` Téléviseur
- `all` Tous les précédents

Avec les `media-queries`, on va pouvoir cibler les tailles d'écran soit directement dans l'appel de la feuille de style :

```
<link rel="stylesheet" media="screen and (max-width: 640px)" href="style.css">
```

Mais on préférera tout regrouper dans la feuille de style.

```
@media screen and (max-width: 640px) {  
  .bloc {  
    display: block;  
    clear: both;  
  }  
}  
  
@media screen and (max-width: 640px) {  
  .bloc {  
    display: block;  
    clear: both;  
  }  
}
```

<http://www.alsacreations.com/article/lire/930-css3-media-queries.html>

Les spécifications pour les mobiles et tablettes peuvent être trouvée ici :  
<http://stephen.io/mediaqueries/>

## Exercice

**Reproduire :**

<http://www.boulanger.com>



## IX. Construction et bonnes pratiques HTML5 / CSS3

(HTML5 Boilerplate, Bootstrap Twitter, Foundation,...)

### Boilerplate

Il s'agit d'un template HTML + CSS + JS contenant toutes les bonnes pratiques du moment en terme de performances, javascript et compatibilité inter-navigateurs.

Vous pouvez générer une structure de document basée sur Bootstrap et contenant :

- Des documents HTML5 ready
- Compatibilité cross-browser (IE6 inclus)
- Directives pour les navigateurs mobiles
- Préparation pour l'utilisation de règles [@font-face](#)
- Dégradation gracieuse et amélioration progressive
- Classes spécifiques à IE pour un contrôle plus fin
- Classes [.js](#) et [.no-js](#)
- Profiling JavaScript dans IE6 et IE7
- Reset CSS et normalisation des polices
- Feuille de style impression (media print)
- Squelette HTML/CSS adaptable pour iOS, Android, Opera Mobile
- Fichier [.htaccess](#) pour utilisation des fonctionnalités HTML5 et optimisation du chargement du contenu, règles pour le cache (dates d'expiration) et la compression
- Fix PNG pour IE
- etc...



### Bootstrap

Un des framework, Open Source, les plus utilisé aujourd'hui. Ce framework vous permet (entre autre) de poser les bases d'un site responsive, en deux secondes.

<http://getbootstrap.com>



Foundation  
Start here, build everywhere.

### ZURB Foundation

Est aussi un framework qui permet de créer rapidement des sites internet responsive.

## Le responsive

Est une technique de développement web qui utilise les grille (grid).

Très simplement , les framework considère qu'un écran d'au minimum 1200px de large diffusera idéalement un site de 1170px de large.

Dans cet espace, On trouvera 12 colonnes d'environ 97px + 15px de padding de chaque côté.

Ainsi dans une page on pourra aligner 12 div avec une classe `.col-md-1`

Ou 2 div avec la classe `.col-md-6`

Ou 3 div avec la classe `.col-md-4`

Ou 1 div `.col-md-6` et 2 div `.col-md-3`

etc. ...

En fonction de la taille d'écran du périphérique qui visitera le site, les div se réduiront ou passeront les uns sous les autres.

## X. Introduction au JavaScript

JavaScript : langage de programmation côté client.

Le JavaScript est un langage qui est lu et exécuté par votre navigateur – le client – (donc directement sur votre ordinateur) et qui permet de créer des réactions en réponse à des événements sur la page ou à des actions de l'utilisateur. Par exemple :

- un clic sur un lien qui affiche un texte masqué progressivement
- un compteur qui défile
- l'apparition renouvelée d'actualités sans que le navigateur ne demande à nouveau la page au serveur
- etc.

Auparavant, le code JavaScript était souvent intégré au code HTML, mais là encore, on préconise aujourd'hui la séparation des deux langages pour des raisons d'organisation du code, mais aussi d'accessibilité du site.

Le JavaScript est capable lui-même d'écrire du code HTML ou CSS, et donc de modifier totalement la structure d'une page Web.

De la même façon que la feuille de style, le javascript peut être appelé dans un fichier externe \*.js ou être écrit inline.

La flexibilité des navigateurs web nous permet de sortir de la balise <head> pour écrire du javascript.

```
<div id="maDiv"></div>
<script>
  var container = document.getElementById("maDiv");
  container.innerHTML = "Un nouveau contenu !";
</script>
```

Si vous codez directement en javascript "pur", voici ce que cela donnera en HTML :

Ou

```
<!doctype html>
<html lang="fr">
<head>
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css" media="screen">
  <script>
    var container = document.getElementById("maDiv");
    container.innerHTML = "Un nouveau contenu !";
  </script>
</head>
<body>
  <div id="maDiv"></div>
</body>
</html>
```

Si vous faites appel à jQuery et un plugin :

```
<!doctype html>
<html lang="fr">
<head>
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css" media="screen">
  <script type="text/javascript" src="js/jquery.min.1.10.2.js"></script>
  <script src="jquery.aw-showcase.js"></script>
  <script>
    $(document).ready(function()
    {
      $("#showcase").awShowcase(
      {
        content_width: 700,
        content_height: 470,
        fit_to_parent: false,
        auto: true,
        ...
      });
    });
  </script>
</head>
<body>
  <div id="showcase"></div>
</body>
</html>
```

### Erreurs à ne pas commettre :

- Ne pas appeler jQuery
- Appeler une version de jQuery différente des préconisations du développeur du script
- Appeler les script dans un autre ordre que celui-ci
- Ne pas appeler "l'appel" inline ou dans un fichier externe.

## XI. UI (User Interface) - UX (User eXperience)

L'UI et L'UX sont deux concepts permettant de prendre en compte la relation entre le site internet et les utilisateurs de celui-ci.

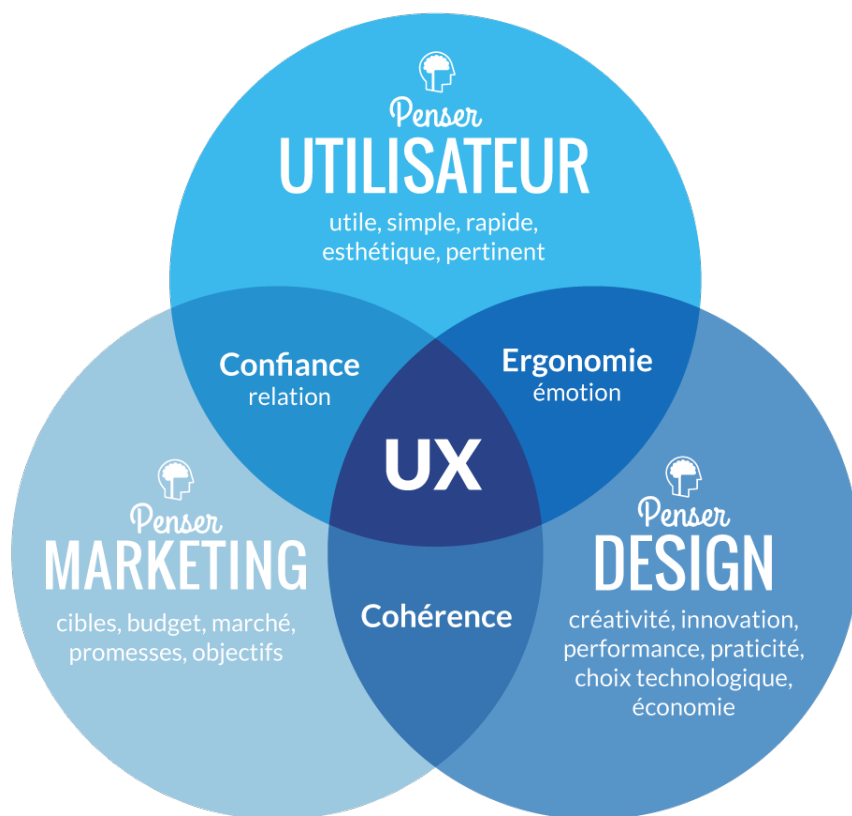
Définition de Wikipedia :

[https://fr.wikipedia.org/wiki/Expérience\\_utilisateur](https://fr.wikipedia.org/wiki/Expérience_utilisateur)

L'**expérience de l'utilisateur** (en anglais, **user experience**), souvent contracté **expérience utilisateur** ou abrégé en **UX** (*User eXperience*), est un concept apparu dans les années 2000 pour tenter de qualifier le résultat (bénéfice) et le ressenti de l'utilisateur (expérience) lors d'une manipulation (utilisation provisoire ou récurrente) d'un objet fonctionnel ou d'une interface homme-machine (via une interface utilisateur) de manière heuristique par un ensemble de facteurs.

Contrairement à l'utilisabilité, ce concept n'est pas strictement pragmatique dans le sens où il sous-entend un impact émotionnel cumulé à un bénéfice rationnel, il est entendu que la démarche est bien de créer une expérience agréable. Ainsi les définitions respectives des deux termes permettent de relativement bien appréhender la notion.

...



### 11.1) L'expérience utilisateur

Il ne faut pas confondre **UX** et **UI** :

- **UX Design** = *User Experience Design* = Design de l'expérience utilisateur
- **UI Design** = *User Interface Design* = Design d'interface utilisateur

Là où l'**UI** va définir l'apparence et l'ergonomie de votre site internet ou application, l'**UX** va se poser les questions qui vont déterminer quel chemin on va faire prendre à l'utilisateur pour qu'il arrive à ses fins sans même s'être poser de questions.

**UI Design**

De quelle couleur doit-être le bouton pour que l'utilisateur le voit immédiatement?

**UX Design**

A-t-on besoin d'un bouton ?