

# HTML, XHTML et CSS

HTML est le langage de base permettant de définir des pages Web. Il est composé de balises (aussi appelées marqueurs ou tags) qui décrivent et mettent en forme des contenus.

XHTML est un langage assez proche d'HTML, à quelques nuances près : toutes les balises doivent avoir un parent, et le parent de tous les parents est <html>. Voici la structure minimale d'une page XHTML :

```
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

## Quelques restrictions :

- pas de majuscules dans les noms et attributs des balises ;
- balises de fin obligatoires (les balises HTML non terminées prennent un slash de fin, comme par exemple `<br />` ou `<img />`) ;
- les balises doivent être correctement imbriquées. A titre d'exemple, seule la seconde ligne de code est correcte en XHTML :

```
<b><i>Ce texte est gras et italique</b></i>
```

```
<b><i>Ce texte est gras et italique</i></b>
```

Le langage HTML ou XHTML est utilisé pour définir le contenu des pages Web.

Le langage CSS est utilisé pour définir la mise en forme et la mise en page des contenus HTML. Il peut être utilisé directement dans les balises HTML/XHTML, dans l'en-tête des pages ou dans un fichier externe appelé "feuille de styles".



## Les moteurs de rendu



Outre leur interface, les navigateurs se distinguent par leur moteur de rendu. C'est lui qui détermine comment sont interprétées puis affichées les instructions HTML, CSS et JavaScript. Pour en savoir plus à ce sujet, consultez les pages Wikipedia relatives aux navigateurs Web classés par moteur de rendu. Ces pages sont accessibles ici :

[http://fr.wikipedia.org/wiki/Cat%C3%A9gorie:Navigateur\\_web\\_par\\_moteur\\_de\\_rendu](http://fr.wikipedia.org/wiki/Cat%C3%A9gorie:Navigateur_web_par_moteur_de_rendu) (ou <http://goo.gl/octwhx>) :

Ainsi, vous pouvez voir que les trois navigateurs vedettes utilisent trois moteurs de rendu différents :

Navigateur	Moteur de rendu
Mozilla Firefox	Gecko
Google Chrome et Safari	Webkit
Internet Explorer	Trident

Consultez la page [http://en.wikipedia.org/wiki/Comparison\\_of\\_layout\\_engines\\_\(ECMAScript\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(ECMAScript)) (soit <http://goo.gl/CQ6SMi> en raccourci) pour prendre connaissance du niveau de compatibilité des moteurs de rendu avec la norme ECMAScript :

The screenshot shows a web browser window displaying a comparison chart titled "ECMAScript version support". The chart is a grid where rows represent ECMAScript features and columns represent rendering engines. The features listed are: Name of ECMAScript Engine, ECMAScript Edition 3, ECMAScript Edition 5, JavaScript 1.5 extensions, JavaScript 1.6 extensions (excluding E4X), JavaScript 1.7 extensions, JavaScript 1.8 extensions, and JavaScript 1.9. The engines listed are: Trident, Gecko, WebKit, Presto, and Blink. The chart uses color coding to indicate support levels: green for full support, yellow for partial support, and red for no support.

	Trident	Gecko	WebKit	Presto	Blink
Name of ECMAScript Engine	JScript/Chakra	Spidermonkey/TraceMonkey/JägerMonkey/IonMonkey	JavaScriptCore/SquirrelFish Extreme	Linear B/Futhark/Carakan	V8
ECMAScript Edition 3	Yes	0.6	Yes	1.0	Yes
ECMAScript Edition 5	5.0	2.0	Partial	Yes	Yes
JavaScript 1.5 extensions	No	0.6	Yes	1.0	Yes
JavaScript 1.6 extensions (excluding E4X)	No	1.8	Partial	Partial	Partial
JavaScript 1.7 extensions	No	1.8.1	No	Partial	No
JavaScript 1.8 extensions	No	1.9	Partial	No	Partial
JavaScript 1.9	...	1.9.1	...	...	...

## Editeurs HTML/XHTML/CSS

La saisie du code HTML peut se faire dans un simple éditeur de texte, comme le Bloc-Notes de Windows. Mais vous lui préférerez un éditeur de code à coloration syntaxique.

Plusieurs éditeurs de ce type sont diffusés sous la forme de freewares. Vous utiliserez par exemple NotePad ++ ou Sublime Text 2.

# Histoire du HTML

HTML (*HyperText Markup Language*) est le langage utilisé pour représenter les pages Web. Il permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédia et des formulaires. Il est souvent utilisé conjointement aux langages CSS (*Cascading Style Sheet*) et JavaScript.

Le premier document portant l'extension `html` a été publié à l'attention du public l'a été en août 1991 par Tim Berners-Lee, lorsque ce dernier a annoncé la naissance du World Wide Web.

Les différentes versions du langage :

HTML 2.0 : fin 1995 : tables, expressions mathématiques

HTML 3.2 : Janvier 1997 : styles CSS, scripts

HTML 4.0 : Décembre 1997 : séparation entre structure et présentation, variantes *strict* (*qui exclut les éléments de présentation*), *transitional* (*qui reprend les éléments de présentation de HTML 3.2*) et *frameset* (*qui permet la création de pages par l'assemblage de frames*) du langage

HTML 4.01 : Décembre 1999 : format de transition XHTML 1.0

HTML 5.0 : Octobre 2014 : accent sur le côté sémantique du langage

# Les organismes de normalisation W3C et WHATWG

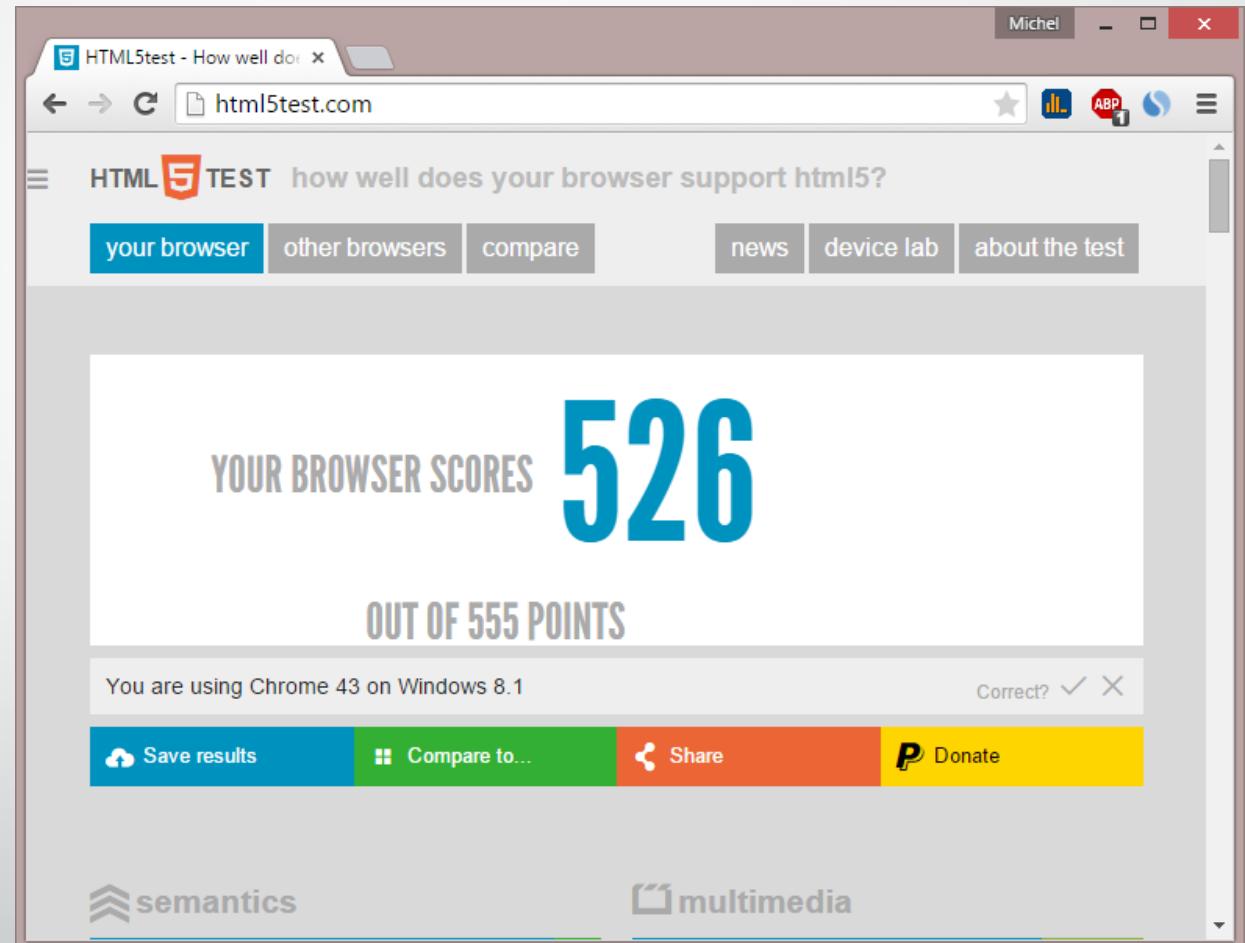
Le *World Wide Web Consortium* (W3C) est un organisme de normalisation à but non lucratif, fondé en Octobre 1994. Son but : promouvoir la compatibilité des technologies du Web telles que HTML5, HTML, XHTML, XML, RDF, SPARQL, CSS, XSL, PNG, SVG et SOAP.

Site officiel du W3C : <http://www.w3.org/>

# Compatibilité avec les navigateurs

Les nouveaux éléments propres au langage HTML5 ne sont pas encore entièrement implémentés sur les principaux navigateurs.

Pour connaître le taux de compatibilité de votre navigateur, rendez-vous à la page <http://html5test.com/>.



Testez la compatibilité de vos navigateurs

# Compatibilité avec les navigateurs

Les principaux navigateurs utilisent un préfixe pour (pré)implémenter les nouvelles propriétés CSS3 :

- -moz pour les navigateurs Mozilla (Firefox) ;
- -webkit pour les navigateurs Webkit (Safari, OmniWeb, Midori, etc.) ;
- -khtml pour les navigateurs Konqueror ;
- -o pour les navigateurs Opera ;
- -ms pour le navigateur Internet Explorer 9 et supérieur.

Tant que la spécification du langage n'a pas atteint au moins le statut de *recommandation candidate*, vous devrez utiliser plusieurs préfixes dans les propriétés CSS3 pour assurer la plus grande compatibilité possible. À terme, ces préfixes ne devraient plus avoir cours et une seule instruction devrait être interprétée à l'identique dans tous les navigateurs.

# Différences HTML5/prédécesseurs

HTML5 est rétrocompatible

La déclaration de type de document (DTD) est simplifiée à son extrême : <!DOCTYPE html>

Le jeu de balises propre à HTML5 introduit une nouvelle logique de formulation, plus sémantique et plus intuitive : la balisage ne décrit plus le contenu du document. Il détaille sa structure et désigne le rôle de chaque section.

Enfin, les développeurs web peuvent désormais utiliser des API JavaScript gérées nativement par les navigateurs. Le but est d'obtenir des niveaux de performances comparables à ceux des applications Desktop, mais aussi de tirer parti des possibilités offertes par les périphériques mobiles, qui vont prendre une part de plus en plus importante dans le paysage Internet.

# Les bases du HTML5/CSS3

Dans cette partie :

- Déclaration de type de document (DTD)
- Syntaxes HTML et XHTML
- Jeux de caractères
- Définir la langue dans un document HTML5
- Structure d'un document HTML5
- Valider du code HTML5
- Styles CSS dans les balises
- Feuille de styles interne
- Feuille de styles externe
- Éléments, sélecteurs, propriétés et valeurs CSS
- Sélecteurs CSS 2.1
- Sélecteurs CSS3
- id ou class ?
- Pseudo-classes et pseudo-éléments
- Unités CSS

# Déclaration de type de document

La balise <!DOCTYPE> est utile pour que les navigateurs adaptent le rendu d'un document HTML.

La balise <!DOCTYPE> des documents écrits en HTML5 est réduite à sa plus simple expression :

```
<!DOCTYPE html>
```

En HTML4.01, les trois variantes généralement utilisées étaient les suivantes :

- HTML 4.01 strict (mode rigoureux) :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

- HTML 4.01 transitionnel (langage XHTML utilisé) :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

- HTML 4.01 frameset (la page contient des frames) :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

# DTD

Voici le squelette d'un document écrit en HTML5 :

```
<!DOCTYPE html>

<html>

  <head>

    </head>

  <body>

    </body>

  </html>
```

# Jeu de caractères

Les *charsets* (jeux de caractères) sont associés à la notion de claviers nationaux. Pour indiquer aux navigateurs dans quel jeu de caractères vous travaillez, vous pouvez insérer une balise dans l'en-tête de votre document :

```
<meta charset="jeu-à-utiliser">
```

Vous utiliserez le jeu de caractères :

- **ISO-8859-1** pour accéder directement à la majorité des caractères des langues occidentales (français, anglais, allemand, espagnol, etc.).
- **UTF-8** pour afficher sur une même page des caractères issus de plusieurs langues (français et japonais par exemple). Consultez la page [www.columbia.edu/kermit/utf8.html](http://www.columbia.edu/kermit/utf8.html) pour vous rendre compte des immenses possibilités du jeu de caractères UTF-8.

# Squelette standard HTML5

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta ...>
    ...
    <meta ...>
    <title>...</title>
    <link ...>
    <style>...</style>
  </head>
  <body>
    </body>
</html>
```

Les éléments meta, s'ils sont présents, permettent, entre autres, d'associer un jeu de caractères, une description et un nom d'auteur au document. Par exemple :

```
<meta charset="UTF-8" />

<meta name="description" content="Description de la page en quelques phrases." />

<meta name="author" content="Michel Martin" />
```

link lie le document courant à un autre document (HTML, CSS, image, XML). Par exemple, pour définir une feuille de styles externe :

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

Pour faire apparaître du texte dans une page Web, il suffit de le taper tel qu'il doit apparaître, entre les balises <body> et </body> :

```
<!DOCTYPE html>
<html lang="fr" >
  <head>
    ...
  </head>
  <body>
    Ce texte apparaît dans le navigateur.
  </body>
</html>
```

Le texte est affiché tel qu'il est saisi dans le navigateur. Si nécessaire, vous pouvez effectuer un passage à la ligne avec la balise `<br>` :

Texte sur la ligne 1  
Texte sur la ligne 2

Vous pouvez aussi définir des paragraphes avec la balise `<p> </p>` :

<p>Ceci est un paragraphe</p>  
<p>Ceci est un autre paragraphe</p>

Quelle est la différence entre `<br>` et `<p>` ?

## ***Des commentaires dans le texte***

Si vous concevez des pages longues et/ou complexes, vous voudrez peut-être y insérer des commentaires pour faciliter leur maintenance. Pour cela, vous utiliserez le marqueur `<!--` :

```
<!-- Commentaires -->
```

Le texte placé dans le marqueur n'est pas affiché dans le navigateur : il est uniquement visible dans le code source.

# ***Titres et sous-titres***

Vous pouvez affecter jusqu'à six niveaux de titre à vos documents HTML. Pour cela, vous utiliserez les marqueurs <hx> et </hx>, où x est un chiffre compris entre 1 et 6.

## **Exemple :**

Ce petit exemple montre comment utiliser les marqueurs <hx>. Remarquez qu'un marqueur de fin de titre (</hx>) provoque un passage à la ligne. Il est donc inutile de faire appel aux marqueurs <br> ou <p>.

```
<html>
<head>
<title>Les balises de titre</title>
</head>
<body>
<h1>titre de niveau 1</h1>
<h2>titre de niveau 2</h2>
<h3>titre de niveau 3</h3>
<h4>titre de niveau 4</h4>
<h5>titre de niveau 5</h5>
<h6>titre de niveau 6</h6>
texte normal
</body>
</head>
</html>
```

# ***Liens hypertexte***

Une grande part de la magie inhérente au Web est liée à la présence de liens hypertexte dans les pages HTML : en cliquant sur certains mots ou sur certaines images, on peut avoir accès à d'autres parties du document en cours de consultation, à d'autres pages du même site ou d'un autre site, ou encore à d'autres services Internet (mail, ftp, gopher, telnet).

Avant de passer à la pratique, vous devez savoir que l'intérêt d'un site tient à la qualité et à la quantité de ses liens.

Au niveau de la qualité, il est nécessaire de placer les liens à des endroits logiques et de bien spécifier vers quoi ils pointent. N'oubliez pas que les informations qui se trouvent dans vos pages sont consultées on-line. La rapidité d'accès aux données recherchées est donc essentielle.

Au niveau quantité, veillez à ne pas submerger l'utilisateur sous une profusion de liens. Un document aéré est toujours plus facile à lire. Sauf cas particuliers (tableau, sommaire, etc.), limitez-vous à quelques liens par page et révisez-les fréquemment pour que votre site soit toujours proche de l'actualité (c'est un des intérêts majeurs d'Internet).

## ***Lien dans la même page***

Si une de vos pages HTML vous semble un peu longue, vous pouvez définir quelques liens pour en faciliter la lecture.

Pour chaque lien, vous procéderez en deux étapes :

1. Définition d'un signet pour marquer la partie à accéder.
2. Définition d'un lien hypertexte vers ce signet.

Pour placer un signet dans le document courant, vous utiliserez une variante de la balise <a> :

```
<a name="Signet">  
    Texte  
</a>
```

où Signet est le nom du marqueur et Texte le texte vers lequel vous désirez pointer.

Pour créer un lien hypertexte vers le signet que vous venez de définir, vous utiliserez une autre variante de la balise <a> :

```
<a href = "#Signet">  
    Texte  
</a>
```

où Signet est le nom du marqueur à pointer et Texte le texte qui servira de lien.

## Exercice

Définissez un document qui contient plusieurs paragraphes de texte. Ce texte pourra être pris sur le site <http://fr.lipsum.com/>.

Placez ce texte dans des paragraphes. Définissez des signets pour chacun d'entre eux et des liens hypertextes pour accéder aux différents paragraphes.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Liens et signets</title>
  </head>
  <body>
    <a href="#par1">Paragraphe 1</a> <a href="#par2">Paragraphe 2</a> <a href="#par3">Paragraphe
3</a> <a href="#par4">Paragraphe 4</a> <a href="#par5">Paragraphe 5</a>
<a name="par1"><p>Paragraphe 1 : Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
<a name="par2"><p>Paragraphe 2 : Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
<a name="par3"><p>Paragraphe 3 : Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
<a name="par4"><p>Paragraphe 4 : Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
<a name="par5"><p>Paragraphe 5 : Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>

  </body>
</html>
```

## ***Lien vers un document local***

Comme nous venons de le voir, les signets facilitent l'accès à un passage déterminé dans un même document. Lorsqu'un lien doit donner accès à des informations qui ne se trouvent pas dans le document courant mais dans un autre document du site, les paramètres de la balise `<a>` doivent être légèrement différents :

```
<a href = "fichier">
```

Texte

```
</a>
```

où Fichier est le nom du fichier HTML auquel accéder et Texte le texte qui servira de lien.

## ***Lien vers une partie d'un document local***

Vous savez maintenant définir un lien hypertexte vers un signet du même document et vers un autre document HTML. En combinant les deux syntaxes, il est possible d'accéder à un signet d'un autre document :

```
<a href = "fichier#signet">  
    Texte  
</a>
```

où Fichier est le nom du fichier HTML auquel accéder, Signet le nom du signet dans le fichier HTML, et Texte le texte qui servira de lien.

# **Lien vers un document distant**

Jusqu'à présent, vous avez appris à utiliser un lien hypertexte pour pointer vers un passage du même document, un autre document du site ou un passage d'un autre document du site. En utilisant une syntaxe légèrement différente, il est possible d'accéder à un document d'un autre site :

```
<a href = "http://adresse_url">  
    Texte  
</a>
```

où http://Adresse\_URL est l'adresse URL de la page HTML et Texte le texte qui servira de lien.

Un URL (*Uniform Ressource Locator*) est une adresse permettant d'identifier de manière unique un objet Internet. Sa syntaxe "façon UNIX" détermine le nom du serveur, le chemin d'accès dans le serveur et le nom du fichier HTML à accéder : **http://serveur/chemin/document#signet**

## ***Lien vers un fichier***

Pour donner accès à un fichier, il suffit de préciser son nom dans le chemin URL. Par exemple, pour accéder au document ebook.pdf dans le dossier documents du serveur [www.monsite.com](http://www.monsite.com), vous utiliserez ce lien :

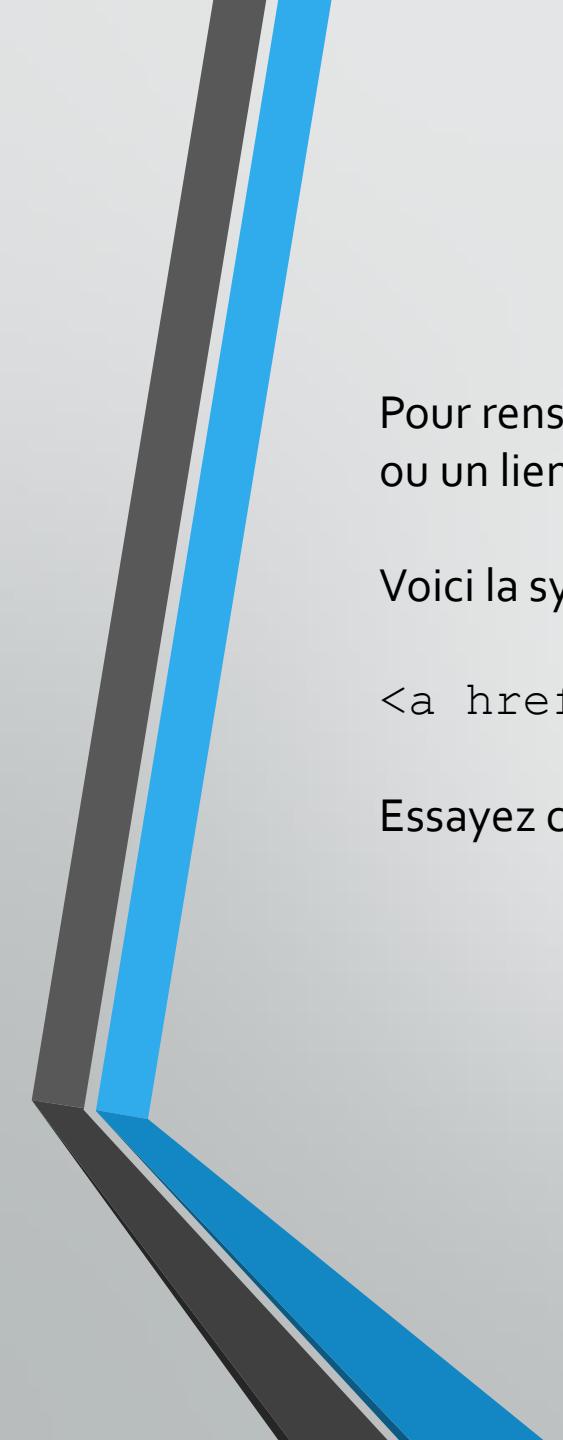
```
<a href="http://www.monsite.com/documents/ebook.pdf">Cliquez ici</a>
```

## **Accéder à un service Internet**

Dans les lignes précédentes, vous avez appris à accéder à une page Web située sur un site distant. Comme le montre ce tableau, il suffit de modifier la première partie de l'adresse pour accéder tout aussi simplement à un autre service Internet.

<b>Service</b>	<b>Marqueur</b>
Site ftp	<a href = "ftp://adresse ">Texte</a>
Serveur e-mail	<a href = "mailto:adresse ">Texte</a>
Gopher	<a href = "gopher://adresse ">Texte</a>
Telnet	<a href = "telnet://adresse ">Texte</a>

Dans chacun des marqueurs précédents, adresse désigne l'adresse URL du service auquel accéder et texte le texte utilisé pour réaliser le lien.



Pour renseigner le visiteur de votre site, vous pouvez ajouter des informations concernant une image ou un lien via une infobulle.

Voici la syntaxe à utiliser pour un lien :

```
<a href="adresse" title="Texte de l'infobulle">Texte du lien</a>
```

Essayez ce code

# Tableaux

Certaines pages Web contiennent des tableaux. Ceux-ci ont été réalisés à l'aide de balises `<table></table>`.

La syntaxe du marqueur `<table>` est assez complexe :

```
<table  
    [border = "bordure"]  
    [cellpadding = "espace"]  
    [cellspacing = "epaisseur"]  
    [width = "largeur"]  
    [height = "hauteur"]>  
</table>
```

Bordure est un entier qui définit l'épaisseur de la bordure autour du tableau. Si BORDER n'apparaît pas dans le marqueur ou si Bordure vaut 0, le tableau n'a pas de bordure.

Espace définit l'espace entre le contenu des cellules et la bordure gauche de la cellule, en pixels.

Epaisseur définit l'épaisseur du trait entre les cellules, en pixels.

Largeur force la largeur du tableau, en pixels. Si ce paramètre n'est pas spécifié, la largeur du tableau est calculée automatiquement.

Hauteur force la hauteur du tableau, en pixels. Si ce paramètre n'est pas spécifié, la hauteur du tableau est calculée automatiquement.

Vous utiliserez le marqueur <tr> </tr> pour créer chacune des lignes du tableau :

```
<tr  
    [align = alignement]>  
</tr>
```

où alignement définit l'alignement des données sur la ligne. Il peut prendre la valeur left, center ou right selon l'effet recherché. Si ce paramètre n'est pas spécifié, les données sont alignées à gauche.

Enfin, pour définir les caractéristiques d'une cellule, vous utiliserez le marqueur <td> </td> dont voici la syntaxe :

```
<td  
    [colspan = "nbcol"]  
    [rowspan = "nblig"]  
    [align = "alignhoriz"]  
    [valign = "alignvert"]  
    [width = "largeur"]>  
    élément  
</td>
```

NbCol est le nombre de colonnes occupées par la cellule.

NbLig est le nombre de lignes occupées par la cellule.

AlignHoriz indique l'alignement horizontal du texte dans la cellule. Ce paramètre peut prendre la valeur LEFT, CENTER ou RIGHT. S'il n'est pas spécifié, les données sont alignées à gauche.

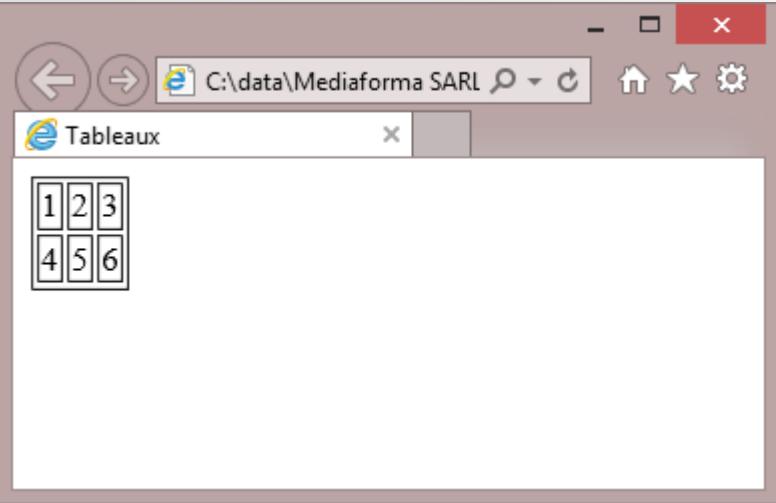
AlignVert indique l'alignement vertical du texte dans la cellule. Ce paramètre peut prendre la valeur TOP, MIDDLE ou BOTTOM. S'il n'est pas spécifié, les données sont alignées sur la partie supérieure de la cellule.

Largeur indique la largeur de la cellule, en pixels.

Elément décrit le contenu de la cellule. Il peut s'agir d'un texte ou d'une image (marqueur IMG).

## Exercice 1

Définissez le tableau suivant :



A screenshot of a web browser window titled "Tableaux". The address bar shows "C:\data\Mediaforma SARL". The main content area displays a 2x3 grid of numbers:

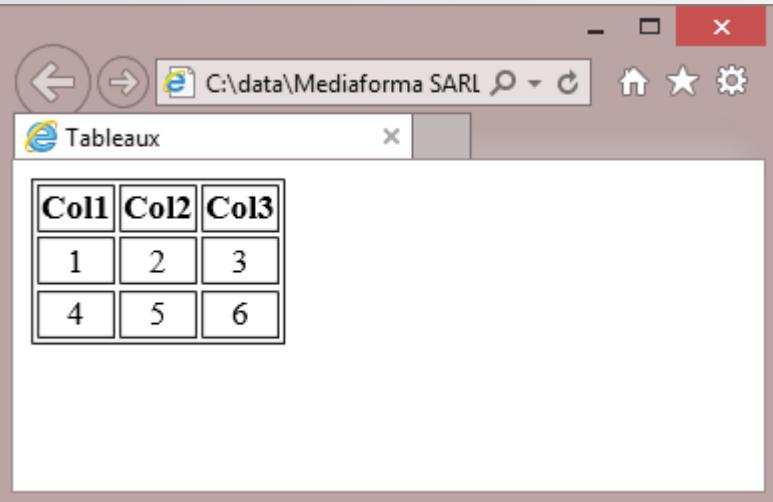
1	2	3
4	5	6

## Solution

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Tableaux</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <td>1</td><td>2</td><td>3</td>
      </tr>
      <tr>
        <td>4</td><td>5</td><td>6</td>
      </tr>
    </table>
  </body>
</html>
```

## Exercice 2

Définissez le tableau suivant :



The screenshot shows a web browser window with the title 'Tableaux'. The address bar displays 'C:\data\Mediaforma SARL'. The main content area contains a 2x3 grid of numbers:

Col1	Col2	Col3
1	2	3
4	5	6

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Tableaux</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <th>Col1</th><th>Col2</th><th>Col3</th>
      </tr>
      <tr>
        <td align="center">1</td><td align="center">2</td><td align="center">3</td>
      </tr>
      <tr>
        <td align="center">4</td><td align="center">5</td><td align="center">6</td>
      </tr>
    </table>
  </body>
</html>
```

## Exercice 3

Définissez le tableau suivant :

The screenshot shows a browser window with the title bar "C:\data\Mediaforma SARL\Formation\Déjà données\2015-06-08 au 12 HTM". The main content area is titled "Tableaux" and displays a table with the following structure:

Titre centré sur les 3 colonnes		
Col1	Col2	Col3
1	2	3
4	5	6

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Tableaux</title>
  </head>
  <body>
    <table border="1">
      <tr><th width="500" colspan="3" align="center">Titre centré sur les 3 colonnes</th></tr>
      <tr><th>Col1</th><th>Col2</th><th>Col3</th></tr>
      <tr><td align="center">1</td><td align="center">2</td><td align="center">3</td></tr>
      <tr><td align="center">4</td><td align="center">5</td><td align="center">6</td></tr>
    </table>
  </body>
</html>
```

## Exercice : Définissez le tableau suivant

The screenshot shows a web browser window with the following details:

- Title Bar:** C:\data\Mediaforma SARL\Formation\Déjà données\2015-06-08 au 12 HTML5 CSS3 JavaScript Xerox Clermo
- Tab:** Titres dans un tableau
- Content:** A table titled "Recettes budgétaires par région".

**Table Data:**

	En millions d'Euros		2015 en €/hab.
	2014	2015	
Ile-de-france	399258	415920	178,8
Midi-pyrénées	73961	63826	219,4
Centre	43638	55802	185,9

```
<!DOCTYPE html>
<html>
  <head>
    <title>Tableaux</title>
  </head>
  <body>
    <center>
      Recettes budgétaires par région
      <table border>
        <th></th>
        <th colspan = "2">En millions d'Euros</th>
        <th rowspan = "2">2015 en €/hab.</th>
        <tr>
          <th></th>
          <th>2014</th>
          <th>2015</th>
        </tr>
        <tr align = "center">
          <td>Ile-de-france</td><td>399258</td><td>415920</td><td>178,8</td>
        </tr>
        <tr align = "center">
          <td>Midi-pyrénées</td><td>73961</td><td>63826</td><td>219,4</td>
        </tr>
        <tr align = "center">
          <td>Centre</td><td>43638</td><td>55802</td><td>185,9</td>
        </tr>
      </table>
    </body>
  </html>
```

# ***Insertion d'une image dans un document HTML***

L'insertion d'une image dans un document HTML s'effectue en trois étapes :

1. Définition ou choix de l'image.
2. Sauvegarde au format GIF (.GIF), JPEG (.JPG) ou PNG (.PNG) et stockage dans le même dossier que le fichier HTML.
3. Insertion d'un marqueur <img> dans le document HTML.

Téléchargez l'application PhotoFiltre 7 sur  
<http://www.photofiltre-studio.com/news.htm>

Entraînez-vous à créer des images au format GIF, JPG (avec plusieurs formats de compression) et PNG et comparez leurs tailles en octets

Lorsque l'image a été sauvegardée au bon format, insérez un marqueur <IMG> dans la page HTML, à l'endroit où vous désirez l'afficher. La syntaxe du marqueur est la suivante :

```
<img  
    src = "nom de l'image"  
    [align = {top|bottom|middle|left|right} ]  
    [border = épaisseur]  
    [alt = "texte"]>
```

1. Nom de l'image est le nom complet de l'image (nom du fichier et extension).
2. ALIGN définit l'alignement de l'image par rapport au texte : TOP, BOTTOM et MIDDLE alignent le texte respectivement par rapport à la partie supérieure, à la partie inférieure et au centre de l'image ; LEFT et RIGHT alignent l'image respectivement à gauche et à droite du texte.
3. Epaisseur définit l'épaisseur du cadre autour de l'image, en pixels.
4. Texte est un court texte qui apparaît à la place de l'image lorsque l'affichage des images a été désactivé.

## Rendre une image cliquable

Pour insérer un lien sur une image, il suffit de l'entourer d'une balise `<a></a>` :

```
<a href="http://www.site.com"></a>
```

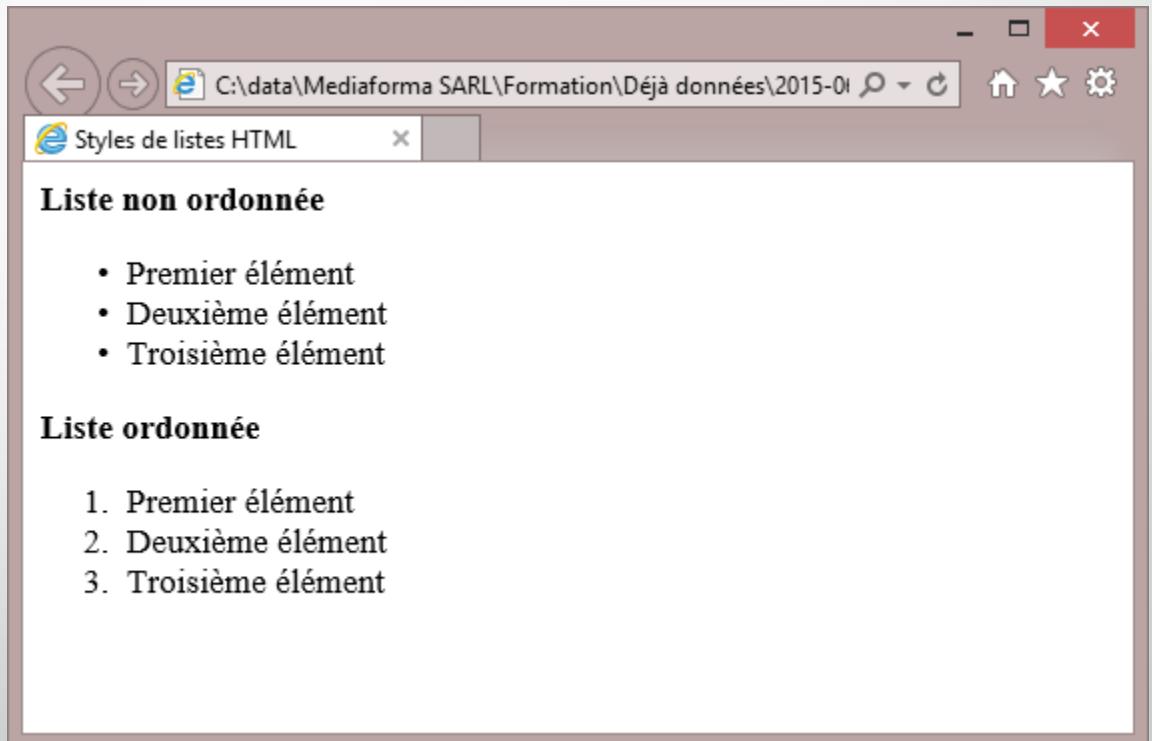
# Listes

Le langage HTML compte trois types de listes :

- non ordonnées ou listes à puces : éléments `ul` et `li` ;
- ordonnées : éléments `ol` et `li` ;

Cet exemple illustre ces deux types de listes :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Styles de listes HTML</title>
  </head>
  <body>
    <b>Liste non ordonnée</b>
    <ul>
      <li>Premier élément</li>
      <li>Deuxième élément</li>
      <li>Troisième élément</li>
    </ul>
    <b>Liste ordonnée</b>
    <ol>
      <li>Premier élément</li>
      <li>Deuxième élément</li>
      <li>Troisième élément</li>
    </ol>
  </body>
</html>
```



Entre les balises <form> et </form>, vous utiliserez des balises <input> pour définir les données à saisir :

```
<input  
[type = {"text|password|checkbox|radio|submit|reset|hidden|image"}]  
name = "nom"  
[value = "valeur"]  
[size = "taille1[,taille2]"]  
[maxlength = "longueur"]  
[checked]>
```

Le paramètre optionnel TYPE définit le type de l'objet :

Valeur	Effet
TEXT	Champ de saisie
PASSWORD	Mot de passe
CHECKBOX	Case à cocher
RADIO	Bouton radio
SUBMIT	Bouton d'envoi, pour envoyer les données du formulaire au serveur
RESET	Bouton de réinitialisation du formulaire
HIDDEN	Champ caché qui n'apparaît pas sur le formulaire mais qui est envoyé lors de l'appui sur le bouton SUBMIT ou sur une image cliquable

Exercice :  
Définissez un formulaire pour obtenir le résultat suivant :

The screenshot shows a web browser window titled "Un premier formulaire". The address bar displays the path "C:\data\Mediaforma SARL\Formation\Déjà données\2015-01". The main content area contains the following form elements:

- "Entrez votre nom" followed by an empty input field.
- "Entrez votre adresse e-mail" followed by an empty input field.
- A question "Etes-vous client de la fnac ?" with two radio button options: "OUI" (selected) and "NON".
- Two buttons at the bottom: "Envoyer" and "Annuler".

## Solution

```
<!DOCTYPE html>
<html>
    <head>
        <title>Un premier formulaire</title>
    </head>
    <body>
        <form action = "mailto:bertrand.oscar@oracle.fr" method = "post">
            Entrez votre nom <input type = "text" name = nom><br>
            Entrez votre adresse e-mail <input type = "text" name = nom><br>
            Etes-vous client de la fnac ?
            OUI <input type = "radio" name = "client" value = "oui" checked>
            NON <input type = "radio" name = "client" value = "non" ><br>
            <input type = "submit" value = "Envoyer">
            <input type = "reset" value = "Annuler">
        </form>
    </body>
</html>
```

## Exercice

Utilisez un tableau pour améliorer le formulaire comme ceci :

The screenshot shows a Windows application window titled "Un premier formulaire". The window has a standard title bar with icons for back, forward, search, and other system functions. The main content area contains the following elements:

- A label "Entrez votre nom" followed by a text input field.
- A label "Entrez votre adresse e-mail" followed by a text input field.
- A question "Etes-vous client de la fnac ?" followed by two radio buttons: "OUI" (with a checked circle) and "NON" (with an empty circle).
- Two buttons at the bottom: "Envoyer" on the left and "Annuler" on the right.

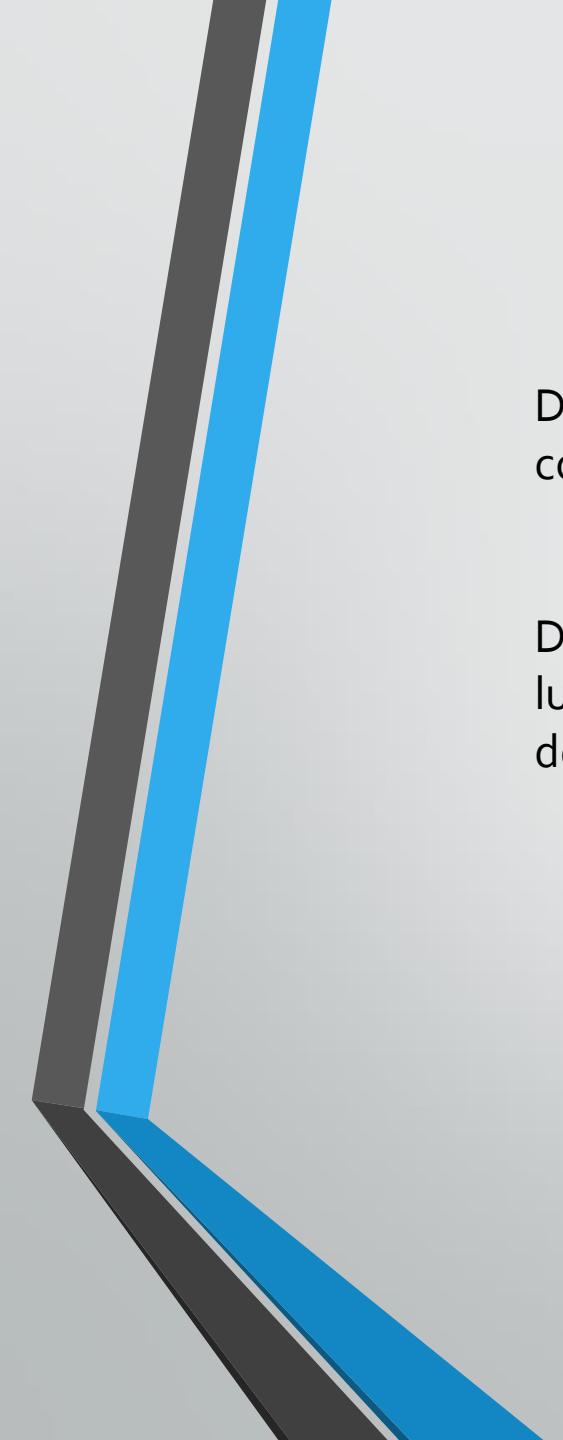
## Solution

```
<!DOCTYPE html>
<html>
  <head>
    <title>Un premier formulaire</title>
  </head>
  <body>
    <form action = "mailto:bertrand.oscar@oracle.fr" method = "post">
      <table>
        <tr><td>Entrez votre nom</td><td><input type = "text" name = nom></td></tr>
        <tr><td>Entrez votre adresse e-mail</td><td><input type = "text" name = nom></td></tr>
        <tr><td>Etes-vous client de la fnac ?</td>
          <td>OUI <input type = "radio" name = "client" value = "oui" checked>
          NON <input type = "radio" name = "client" value = "non" ></td></tr>
        <tr><td><input type = "submit" value = "Envoyer"></td>
          <td><input type = "reset" value = "Annuler"></td></tr>
      </table>
    </form>
  </body>
</html>
```

## Une liste déroulante dans un formulaire

Pour définir une liste déroulante, vous utiliserez la balise <select>. Chacun des éléments de la liste sera placé dans une balise enfant <option>

```
<form>
  <select name="nom">
    <option>Lundi</option>
    <option>Mardi</option>
    <option>Mercredi</option>
    <option>Jeudi</option>
    <option>Vendredi</option>
  </select>
</form>
```



Dans la balise **<select>**, vous pouvez définir l'attribut **size** pour indiquer combien d'éléments sont visibles dans la liste (par défaut, **size** vaut **1**).

Dans l'une des balises **<option>** vous pouvez définir l'attribut **selected**, sans lui affecter aucune valeur. L'élément correspondant est alors sélectionné par défaut dans la liste.

Outre les boutons de formulaires **submit** et **reset** :

```
<input type="submit">  
<input type="reset">
```

Vous pouvez utiliser de simples boutons à l'intérieur ou à l'extérieur d'un formulaire avec la balise **<button>** :

```
<button>Texte</button>
```

# Découpage d'une page

Pour découper une page en blocs, vous utiliserez les balises `<div>` et `</div>`. Ces balises sont de type **block**. Elles s'affichent automatiquement l'une sous l'autre, sans qu'il soit nécessaire d'utiliser une balise `<br>` pour passer à la ligne ou de les insérer dans une balise `<p> </p>`.

Les balises `<div>` peuvent être personnalisées en utilisant des instructions CSS. Nous étudierons ces instructions dans la suite de la formation.

## Partie 2

Mise en forme en CSS



Vous utiliserez le langage CSS pour mettre en forme les balises d'un ou de plusieurs documents, à travers des attributs ou des feuilles de styles.

Les prochaines diapositives vont vous montrer comment utiliser ce langage.

# Styles CSS dans les balises

style=""

où :

- balise est un nom de balise : <p> ou <h1> par exemple.
- propriété1 sont des propriétés de style de la balise.
- valeur1 sont les valeurs affectées aux propriétés.

Cette technique n'est pas optimale, car limitée au seul élément qui l'utilise. Dans la mesure du possible, privilégiez l'utilisation d'une feuille de styles externe, d'extension .css et reliée au document à l'aide d'un élément link. Cette même feuille de styles pourra être utilisée dans tous les documents apparentés pour uniformiser leur apparence.

# Styles CSS dans les balises

À titre d'exemple, pour affecter la couleur jaune à l'arrière-plan d'un élément p, vous utiliserez le code suivant :

```
<p style="background-color:yellow;>  
Ce texte a un arrière-plan jaune  
</p>
```

Ou encore, pour utiliser la police Verdana corps 18 dans un titre h2, vous utiliserez le code suivant :

```
<h2 style="font-family:Verdana; font-size: 18px;">  
Ce titre est en Verdana corps 18  
</h2>
```

Essayez ces deux codes

# Feuille de styles interne

Pour étendre la portée des définitions CSS, vous pouvez les regrouper dans l'en-tête du document HTML.

Voici la syntaxe permettant d'affecter des propriétés à une balise :

élément {propriété1:valeur1; ... propriétéN:valeurN}

où :

- élément est un nom d'élément : p ou h1 par exemple.
- propriété1 sont des propriétés de style de l'élément.
- valeur1 sont les valeurs affectées aux propriétés.

Ces lignes de code doivent être insérées entre les balises <style> et </style>, à l'intérieur de l'élément head.

# Exercice

Ecrivez une feuille de styles interne pour :

- Définir dans tout le document un arrière-plan de couleur jaune pour les éléments `p`
- Utiliser la police Verdana corps 18 dans tous les titres `h2`

# Solution

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Une feuille de styles interne</title>
    <style>
      p {background-color:yellow; }
      h2 {font-family:Verdana; font-size: 18px;}>
    </style>
  </head>
  <body>
    ...
  </body>
</html>
```

# Feuille de styles externe

Pour augmenter encore le champ d'action des styles définis dans l'en-tête d'un document, vous pouvez les stocker dans un fichier CSS. Dans ce cas, plusieurs documents HTML peuvent référencer cette "feuille de styles" (le fichier CSS) pour utiliser les styles qui y sont définis.

L'opération se fait avec un élément link (ici, la feuille de styles a pour nom *moncss.css*) :

```
<link rel="stylesheet" href="moncss.css" />
```

# Exercice

Définissez une feuille de styles externe qui reprend les règles définies dans l'exercice précédent et faites référence à cette feuille de styles dans un document HTML.

# Solution

```
p {background-color:yellow;}  
h2 {font-family:Verdana; font-size: 18px;}
```

La feuille de styles  
moncss.css

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset=UTF-8" />  
<title>Une feuille de styles externe</title>  
<link rel="stylesheet" href="moncss.css" />  
</head>  
<body>  
<!-- Les instructions HTML5 peuvent utiliser les styles --&gt;<br/><!-- définis dans la feuille de styles moncss.css --&gt;<br/></body>  
</html>
```

Le document HTML

# Sélecteurs, propriétés et valeurs CSS

La syntaxe à utiliser est la suivante :

```
sélecteur {propriété1: valeur1; ... propriétéN: valeurN;}  
élément {propriété1: valeur1; ... propriétéN: valeurN;}
```

Pour définir le style d'un élément, il suffit d'entrer le nom de l'élément, d'ouvrir des accolades, d'énumérer des propriétés et de leur affecter les valeurs souhaitées.

# Exercice

Définissez un document HTML5 qui contient une liste à puces.

Définissez une feuille de styles internet qui donne les caractéristiques de la liste à puces :

- Couleur : blue
- Arrière-plan : #FF4040
- Marges : 14px
- Style des puces : square

# Solution

code001.htm

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Style des li</title>
    <style>
      li
      {
        color: blue;
        background: #FF4040;
        margin: 14px 14px 14px 14px;
        list-style: square;
      }
    </style>
  </head>
  <body>
    Une liste à puces :
    <ul>
      <li>Premier</li>
      <li>Deuxième</li>
      <li>Troisième</li>
      <li>Quatrième</li>
    </ul>
  </body>
</html>
```

# Sélecteurs CSS 2.1

Le sélecteur universel étoile (\*) s'adresse à tous les éléments. Vous l'utiliserez essentiellement pour modifier le style de tous les éléments de la page. Par exemple, la police.

```
* { font-family: Verdana; }
```

Le sélecteur de type permet de changer le style d'un élément. Ici, par exemple, le style de tous les éléments h1 :

```
h1 { font-size: 120%; }
```

Le sélecteur de classe est représenté par un point (.). Il cible les éléments en fonction de la valeur de leur attribut class. Par exemple, le style ci-après s'applique à tous les éléments p dont l'attribut class vaut info.

```
p.info { background: yellow; }
```

Si aucun élément ne restreint le sélecteur de classe, tous les éléments dont l'attribut class a la valeur spécifiée sont concernés. Ici, par exemple, il s'agira de tous les éléments dont l'attribut class vaut info :

```
.info { background: yellow; }
```

# Sélecteurs CSS 2.1

Il est possible d'affecter plusieurs classes à un élément (l'attribut class peut recevoir plusieurs valeurs séparées par des espaces). On parle alors de "sélecteur de classes multiples". Ici par exemple, seuls les éléments p dont l'attribut class vaut info et premier sont affectés :

```
p.info.premier {background: yellow;}
```

Voici un exemple d'élément p concerné par cette règle CSS :

```
<p class="info premier">texte</p>
```

# Sélecteurs CSS 2.1

Le sélecteur d'identificateur est représenté par le caractère dièse (#). Il cible l'élément dont l'attribut id a la valeur spécifiée. Ici, seul l'élément dont la propriété id vaut corne est concerné :

```
#corne {font-family: Serif;}
```

Vous pouvez restreindre le ciblage en précisant un élément devant le sélecteur d'id. Ici, par exemple, seul l'élément p dont le marqueur id vaut corne est concerné :

```
p#corne {font-family: Serif;}
```

Un sélecteur descendant est construit avec deux sélecteurs traditionnels (ou plus) séparés par un espace. Il cible les enfants du premier sélecteur. Par exemple, pour affecter un arrière-plan jaune à tous les éléments li enfants d'un élément div, vous utiliserez le sélecteur descendant ci-après :

```
div li {color: yellow;}
```

# Sélecteurs CSS 2.1

Un sélecteur d'attribut cible les éléments en fonction de la présence d'attributs et/ou de leurs valeurs. Par exemple, pour affecter une taille de 120% à tous les éléments qui possèdent un attribut perso, vous utiliserez le sélecteur suivant :

```
[perso] {font-size: 120%;}
```

Pour cibler les seuls éléments dont l'attribut perso a pour valeur vert, vous utiliserez le sélecteur suivant :

```
[perso=vert] {font-size: 120%;}
```

# Sélecteurs CSS 2.1

Enfin, pour sélectionner les éléments **el2** immédiatement précédés par une élément **el1**, vous utiliserez le sélecteur suivant :

`el1+el2`

# Exercice

- 1) Affectez la couleur rouge aux balises p de classe rouge
- 2) Affectez la police Courrier aux balises div de classe code ou exemple
- 3) Affectez la couleur d'arrière-plan jaune aux balises <li> enfants de la balise <ol>

Voici le code HTML sur lequel appliquer ces règles :

```
<body>
    <p class="rouge">Ceci est un paragraphe de classe rouge</p>
    <p class="vert">Ceci est un paragraphe de classe vert</p>
    <p class="noir">Ceci est un paragraphe de classe noir</p>
    <div class="code">Ce texte est affiché dans une balise div de classe code</div>
    <div>Ce texte est affiché dans une balise div sans classe</div>
    <div class="exemple">Ce texte est affiché dans une balise div de classe
        exemple</div>
    Une liste UL
    <ul>
        <li>Premier</li>
        <li>Deuxième</li>
        <li>Troisième</li>
    </ul>
    Une liste OL
    <ol>
        <li>Premier</li>
        <li>Deuxième</li>
        <li>Troisième</li>
    </ol>
</body>
```

# Exercice

Et voici le résultat à obtenir :

Ceci est un paragraphe de classe rouge

Ceci est un paragraphe de classe vert

Ceci est un paragraphe de classe noir

Ce texte est affiché dans une balise div de classe code

Ce texte est affiché dans une balise div sans classe

Ce texte est affiché dans une balise div de classe exemple

Une liste UL

- Premier
- Deuxième
- Troisième

Une liste OL

1. Premier
2. Deuxième
3. Troisième

# Solution

code002.htm

```
<style>
    p.rouge {color: red; }
    div.code {font-family:Courier New; }
    div.exemple {font-family:Courier New; }
    ol li {background: yellow; }
</style>
```

# id ou class ?

L'attribut id est généralement utilisé :

- Sur les éléments qui structurent le document et qui sont désignés comme uniques : par exemple, les éléments header, container, content, nav, footer, etc. Une fois ces éléments identifiés avec l'attribut id, ils peuvent être stylés avec des propriétés CSS.
- Sur d'autres éléments qui doivent être accessibles en JavaScript.
- Pour accéder à une ancre.

```
<header id="hperso">  
  
<nav id="nperso">  
  
<p id="position1">, puis <a href="#position1">Aller au repère 1</a>
```

L'attribut class est utilisé dans tous les autres cas.

```
<div class="perso">  
  
<p class="perso">
```

# Sélecteurs CSS3

Tout ce qui a été dit sur les sélecteurs à la section précédente reste valable en CSS3. Cependant, de nouveaux sélecteurs sont maintenant disponibles, qui permettent de cibler encore plus précisément les éléments auxquels vous vous adressez.

# Sélecteurs CSS3

Syntaxe	Signification
<code>nom élément[attr^="valeur"]</code>	Tout élément <u>nom élément</u> dont la valeur de l'attribut <u>attr</u> commence exactement par la chaîne <u>valeur</u>
<code>nom élément[attr\$="valeur"]</code>	Tout élément <u>nom élément</u> dont la valeur de l'attribut <u>attr</u> finit exactement par la chaîne <u>valeur</u>
<code>nom élément[attr*="valeur"]</code>	Tout élément <u>nom élément</u> dont la valeur de l'attribut <u>attr</u> contient la sous-chaîne <u>valeur</u>
<code>:root</code>	Un élément <u>nom élément</u> , racine du document
<code>nom élément:nth-child(n)</code>	Un élément <u>nom élément</u> qui est le n-ième enfant de son parent
<code>nom élément:nth-last-child(n)</code>	Un élément <u>nom élément</u> qui est le n-ième enfant de son parent en comptant depuis le dernier enfant
<code>nom élément:nth-of-type(n)</code>	Un élément <u>nom élément</u> qui est le n-ième enfant de son parent et de ce type
<code>nom élément:nth-last-of-type(n)</code>	Un élément <u>nom élément</u> qui est le nième enfant de son parent et de ce type en comptant depuis le dernier enfant
<code>nom élément:last-child</code>	Un élément <u>nom élément</u> , dernier enfant de son parent
<code>nom élément:first-of-type</code>	Un élément <u>nom élément</u> , premier enfant de son type

# Sélecteurs CSS3

Syntaxe	Signification
<code>nom élément:last-of-type</code>	Un élément <u>nom élément</u> , dernier enfant de son type
<code>nom élément:only-child</code>	Un élément <u>nom élément</u> , seul enfant de son parent
<code>nom élément:only-of-type</code>	Un élément <u>nom élément</u> , seul enfant de son type
<code>nom élément:empty</code>	Un élément <u>nom élément</u> qui n'a aucun enfant
<code>nom élément:target</code>	Un élément <u>nom élément</u> qui est la cible de l'URL d'origine
<code>nom élément:enabled</code>	Un élément d'interface utilisateur <u>nom élément</u> qui est actif ou inactif
<code>nom élément:checked</code>	Un élément d'interface utilisateur <u>nom élément</u> qui est coché ou dont l'état est indéterminé (bouton radio ou case à cocher par exemple)
<code>nom élément:contains("attr")</code>	Un élément <u>nom élément</u> dont le contenu textuel concaténé contient la sous-chaîne <u>attr</u>
<code>nom élément:selection</code>	La partie d'un élément <u>nom élément</u> qui est actuellement sélectionnée par l'utilisateur
<code>nom élément:not(sel)</code>	Un élément <u>nom élément</u> qui n'est pas représenté par le sélecteur simple <u>sel</u>
<code>nom élément ~ F</code>	Un élément <u>F</u> précédé par un élément <u>nom élément</u>

# Pseudo-classes, pseudo-éléments

Le sélecteur de pseudo-classe est représenté par le caractère deux-points (`:`).

Vous ferez appel aux pseudo-classes pour cibler des éléments en fonction de caractéristiques inaccessibles aux sélecteurs traditionnels : premier enfant ou focus par exemple.

La pseudo-classe `:first-child` permet de cibler le premier enfant d'un élément. Par exemple, pour mettre en gras le premier élément `p` enfant de l'élément `div`, vous utiliserez le sélecteur ci-après :

```
div p:first-child {font-weight: bold;}
```

Les pseudo-classes `:link` et `:visited` ciblent les éléments `a` dont (respectivement) le lien n'a pas été visité/a été visité. Les deux lignes suivantes définissent la couleur des liens :

```
:link {color: fuchsia}
```

```
:visited {color: navy}
```

# Pseudo-classes, pseudo-éléments

La pseudo-classe `:focus` cible les éléments qui ont le focus (par défaut, les liens et les éléments de formulaires). Elle permet très simplement de modifier la couleur d'arrière-plan (ou un autre style) d'un élément. Ici, par exemple, nous affectons un arrière-plan rouge à l'élément `input` de type `text` qui a le focus :

```
input[type=text]:focus {background: red;}
```

La pseudo-classe `:hover` cible les éléments dont le contenu est survolé. Cela permet par exemple de changer la bordure d'une image lorsqu'elle est pointée par la souris :

```
img:hover {border-style: dotted;}
```

La pseudo-classe `:lang` permet de définir un style en fonction de la langue du document. Ici, par exemple, nous définissons les guillemets à utiliser pour la langue française :

```
:lang(fr) {quotes: '«' '»';}
```

# Pseudo-classes, pseudo-éléments

CSS3 s'enrichit de plusieurs pseudo-classes :

Pseudo-classe	Signification
<u>:nth-child(position)</u>	Éléments repérés par leur position dans le parent
<u>:nth-last-child(position)</u>	Dernier élément du parent
<u>type:nth-of-type(position)</u>	Éléments du type spécifié repérés par leur position dans le parent
<u>type:nth-last-of-type(N)</u>	Dernier élément du type spécifié du parent
<u>:last-child</u>	Dernier élément du parent
<u>type:first-of-type</u>	Premier élément du type spécifié
<u>:last-of-type</u>	Dernier élément du type spécifié
<u>:only-child</u>	Élément qui est l'enfant unique du parent
<u>type:only-of-type</u>	Élément qui est l'enfant unique du type spécifié du parent
<u>:root</u>	Éléments à la racine du document
<u>:empty</u>	Éléments sans enfant
<u>:target</u>	Élément dont l'identifiant est l'ancre spécifiée dans l'URL de la page
<u>:enabled</u>	Éléments validés (enabled)
<u>:disabled</u>	Éléments dévalidés (disabled)
<u>:checked</u>	Éléments (cases à cocher ou boutons radio) cochés
<u>:not(Sélecteur)</u>	Éléments qui sont différents du sélecteur

# Exercice

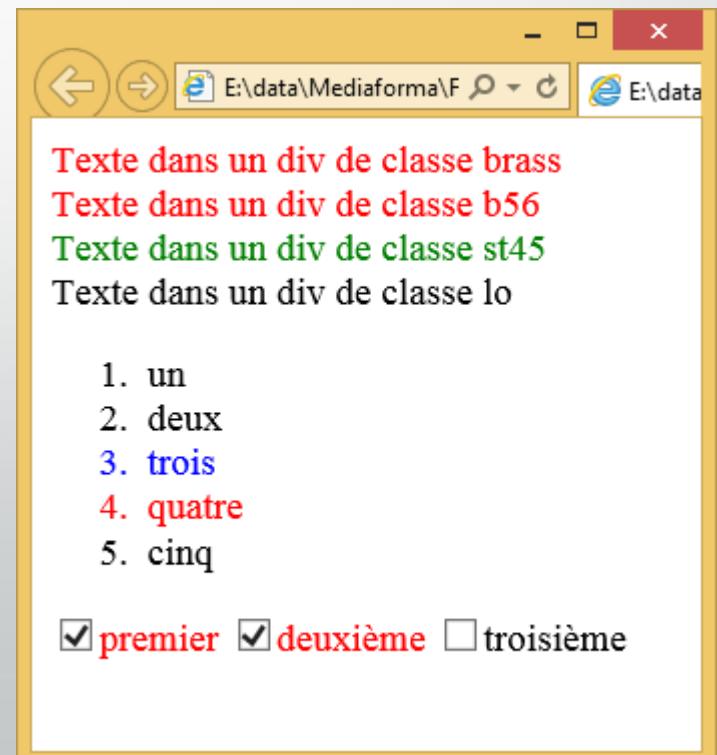
Quelle syntaxe utiliser pour s'adresser :

- 1) Aux éléments div dont l'attribut class commence par "br"
- 2) Aux éléments div dont l'attribut class contient la lettre "t"
- 3) L'avant-dernier élément li enfant de ol
- 4) Le troisième enfant li de l'élément ol de classe "t"
- 5) Les éléments checkbox qui sont cochés

# Solution

- 1) div[class^="br"]
- 2) div[class\*="t"]
- 3) ol li:nth-last-child(2)
- 4) ol li:nth-child(3)
- 5) input[type=checkbox]:checked+label

Définissez le code HTML5 et CSS3 nécessaire pour obtenir ce résultat :



# Solution

code003.htm

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset=UTF-8" />
    <style>
      div[class^="br"] {color: red;}
      div[class*="t"] {color: green;}
      ol li:nth-last-child(2) {color: red;}
      ol li:nth-child(3) {color: blue;}
      input[type=checkbox]:checked+label {color:red; }
    </style>
  </head>
  <body>
    <div class="brass">Texte dans un div de classe brass</div>
    <div class="br56">Texte dans un div de classe b56</div>
    <div class="st45">Texte dans un div de classe st45</div>
    <div class="lo">Texte dans un div de classe lo</div>
    <ol>
      <li>un</li>
      <li>deux</li>
      <li>trois</li>
      <li>quatre</li>
      <li>cinq</li>
    </ol>
    <form>
      <input type="checkbox" checked><label>premier</label>
      <input type="checkbox" checked><label>deuxième</label>
      <input type="checkbox"><label>troisième</label>
    </form>
  </body>
</html>
```

# Unités CSS

Voici la liste des unités exprimées de façon absolue :

- cm : centimètre ;
- in : pouce (2,54 cm) ;
- mm : millimètre ;
- pt : point (1/62 inch) ;
- pc : pica (12 points).

À titre d'information, 1 in = 2,54 cm = 25,4 mm = 72 pt = 6 pc.

# Unités CSS

Il existe également des unités exprimées de façon relative, qui dépendent de leur élément parent :

- em : cadratin. Un em correspond grossièrement à la hauteur de la lettre "C" de l'élément parent (taille du texte définie dans la propriété font-size).
- ex : fraction de la hauteur des caractères. Dans les polices occidentales, ex correspond à la hauteur de la lettre minuscule "x".
- px : pixel. Cette unité dépend de la résolution du périphérique d'affichage.
- % : pourcentage de la taille de l'élément ou de son parent.

L'unité em n'est pas à exclure. Elle permet en effet aux utilisateurs finaux de redimensionner le texte proportionnellement à la taille de police, tout en conservant la mise en page intacte. C'est la raison pour laquelle je ne saurais trop vous recommander de l'utiliser !

# Ossature d'un document HTML

Dans cette partie :

- Nouvelle organisation des documents HTML
- Sections et niveaux de titre
- address

# Organisation des documents HTML5

Bien souvent, les pages web sont constituées d'un en-tête et d'un pied de page, d'un menu, d'une zone réservée au contenu (article, images, vidéos, etc.) et, éventuellement, d'une zone annexe, n'ayant aucun rapport direct avec le contenu de la page.

Le langage HTML5 a de nouvelles balises pour répondre à ces besoins.

# HTML5

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset=UTF-8" />
        <title>Un document HTML5 avec en-tête, bas de page, menu, zone annexe et Zone de contenu</title>
        <link rel="stylesheet" href="styles004.css">
        <script src="Creation-elements-HTML5.js"></script>
    </head>
    <body>
        <header>
            <h1>En-tête du document</h1>
            Texte de l'en-tête
        </header>
        <nav>
            <h2>Menu</h2>
            <ul>
                <li><a href="page1.htm">Page 1</a></li>
                <li><a href="page2.htm">Page 2</a></li>
                <li><a href="page3.htm">Page 3</a></li>
            </ul>
            <br><br><br>
        </nav>
        <aside>
            Texte affiché dans la partie droite de la page avec la balise &lt;aside&gt;<br><br>
        </aside>
        <article>
            <h2>Premier article</h2>
            <p>Texte du premier article</p>
        </article>
        <article>
            <h2>Deuxième article</h2>
            <p>Texte du deuxième article</p>
        </article>
        <footer>
            <p>Copyright et e-mail du Webmaster</p>
        </footer>
    </body>
</html>
```

Les balises <div> n'ont pas disparu du langage HTML5. Elles sont simplement moins souvent utilisées, car plusieurs nouvelles balises plus spécialisées viennent lui prêter main-forte.

Saisissez ce code et sauvegardez-le sous le nom [code004.htm](#)

# CSS3

Saisissez ce code et sauvegardez-le sous le nom styles004.css

```
header, nav, article, section, footer, aside
{
    display: block;
}
```

Le rendu block (`display: block;`) est affecté aux nouveaux éléments HTML5 pour assurer le support de ces nouveaux éléments dans les navigateurs peu ou pas compatibles.

```
header
{
    background-color: red;
}
```

header défini l'en-tête d'une section ou d'une page web.

```
nav
{
    float:left;
    width:20%;
    background-color:yellow;
}
```

nav est destiné à contenir des liens de navigation dans le site.

```
article
{
    background-color: #66FFFF;
    width:80%;
}
```

article correspond à une entité autonome du document : par exemple, un article (ou un ensemble d'articles) extrait d'un blog ou d'un forum.

```
aside
{
    float:right;
    width:20%;
    background-color:yellow;
}
```

aside définit un contenu qui n'a pas de rapport direct avec la page. Il peut être utilisé pour définir un menu ou pour donner accès aux archives du site.

```
footer
{
    clear:both;
    background-color: #99FF66;
}
```

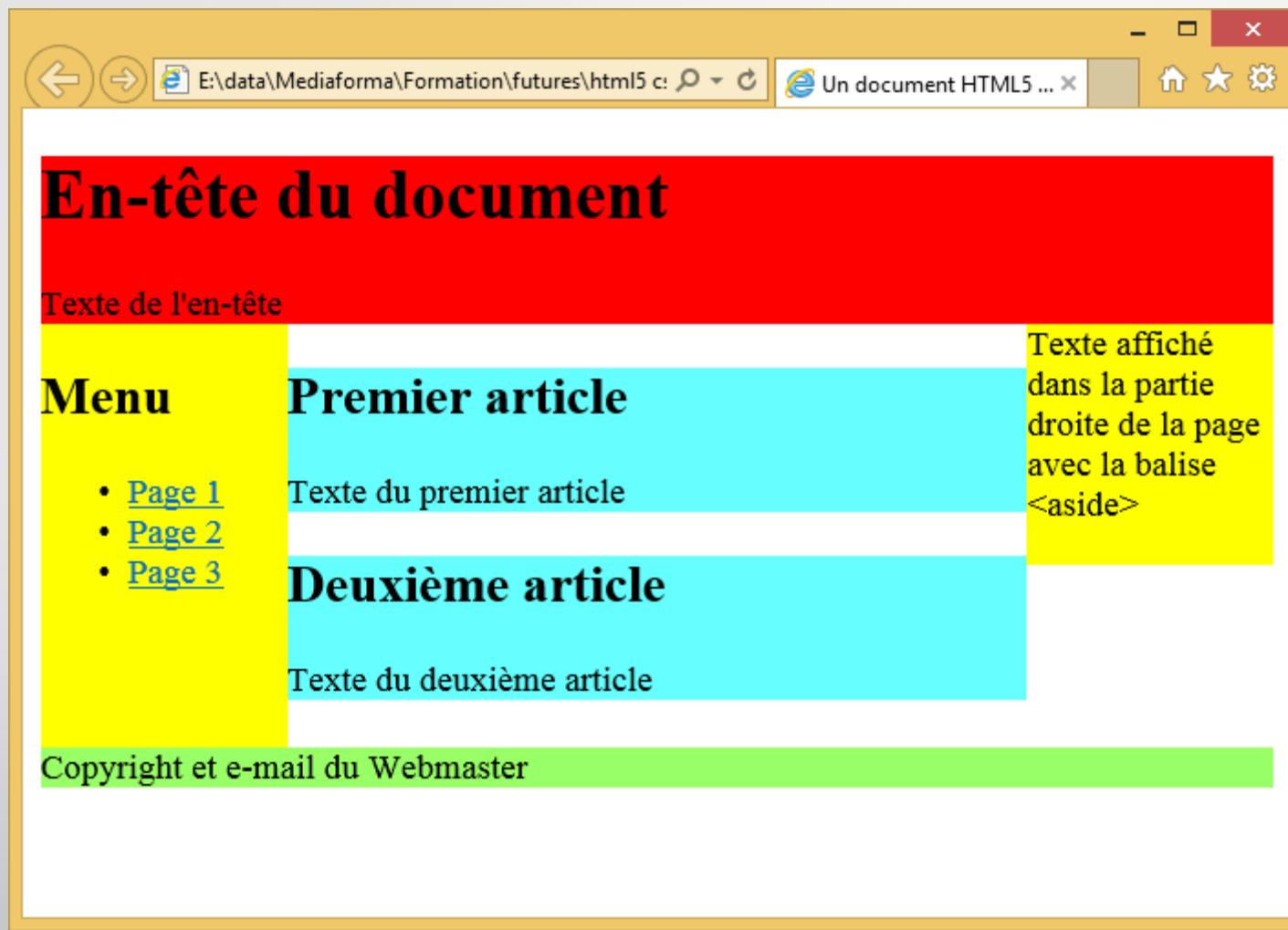
footer définit le pied de page d'une section ou d'une page web.

Les navigateurs Internet Explorer 8 et inférieur ne sont pas en mesure d'affecter un style aux nouveaux éléments HTML5. En effet, puisqu'ils ne les connaissent pas, ils doivent les intégrer au DOM *via* la méthode `createElement`. Pour ce faire, quelques lignes de JavaScript sont nécessaires :

```
document.createElement("footer");  
  
document.createElement("header");  
  
document.createElement("article");  
  
document.createElement("nav");  
  
document.createElement("aside");
```

Saisissez ce code et  
sauvegardez-le sous le nom  
Creation-elements-HTML5.js

Exécutez le code HTML. Voici ce que vous devriez obtenir :



# Sections et niveaux de titre

Les éléments header, nav, aside, article et footer ne sont pas les seuls à faire leur apparition dans le langage HTML5. Vous pouvez également utiliser l'élément section pour définir des sections dans un document.

Il sert généralement pour segmenter une page ou un article en plusieurs sujets.

Il peut être employé conjointement aux éléments de titre h1 à h6 pour définir la structure du document.

Il correspond plus ou moins à l'ancien élément div du HTML 4.0x/XHTML 1.x, à ceci près qu'il y ajoute une signification sémantique. En effet, il regroupe des éléments qui traitent d'un même sujet.

# Sections et niveaux de titre

Ici, par exemple, nous utilisons un élément section pour englober plusieurs articles au sein du document :

```
<section>

    <article>
        Contenu du premier article
    </article>

    ...
    <article>
        Contenu du dernier article
    </article>

</section>
```

# Sections et niveaux de titre

Vous pouvez également inclure plusieurs contenus dans un article en utilisant plusieurs sections.

```
<article>

    <section>
        Contenu de la première section
    </section>

    ...
    <section>
        Contenu de la dernière section
    </section>

</article>
```

# Sections et niveaux de titre

Prenons l'exemple d'un blog, constitué d'un empilement de plusieurs articles (ou *posts*), non nécessairement liés les uns aux autres. Ces composants pourraient être insérés dans des éléments article, eux-mêmes insérés dans un élément section :

```
<section>
  <article>
    <h3>Titre de la première section</h3>
    Texte ou autre contenu qui présente la section
  </article>
  ..
  <article>
    <h3>Titre de la dernière section</h3>
    Texte ou autre contenu qui présente la section
  </article>
</section>
```

# Sections et niveaux de titre

Si vous avez du mal à faire la différence entre `<section>` et `<article>`, dites-vous qu'une section peut être composée d'un ou de plusieurs articles qui traitent d'un même sujet.

Par exemple, si une page web contient plusieurs sujets sur les baleines blanches et plusieurs autres sur les arbres fruitiers, les sujets sur les baleines blanches seront regroupés dans une même section et décomposés en autant d'articles que nécessaire.

De même en ce qui concerne les sujets sur les arbres fruitiers.



```
<section>
  <h1>Articles sur les baleines blanches</h1>
  <article>
    <h2>Les baleines blanches, habitat</h2>
    <p>texte de l'article</p>
  </article>
  <article>
    <h2>Les baleines blanches, population</h2>
    <p>texte de l'article</p>
  </article>
  <article>
    <h2>Les baleines blanches, une espèce menacée</h2>
    <p>texte de l'article</p>
  </article>
</section>
<section>
  <h1>Articles sur les arbres fruitiers</h1>
  <article>
    <h2>Les arbres fruitiers, catégories</h2>
    <p>texte de l'article</p>
  </article>
  <article>
    <h2>Les arbres fruitiers, taille</h2>
    <p>texte de l'article</p>
  </article>
  <article>
    <h2>Les arbres fruitiers, récolte</h2>
    <p>texte de l'article</p>
  </article>
</section>
```

# Sections et niveaux de titre

Un ou plusieurs niveaux de titre `<h1>` à `<h6>` peuvent être utilisés à l'intérieur d'un élément `section`, mais aussi `article`, `aside` ou `nav`. Dans le code suivant, des niveaux de titre `<h2>` sont utilisés dans les sections et un niveau de titre `<h1>` dans l'article qui englobe les sections :

```
<article>
  <h1>Titre de l'article</h1>
  Texte ou autre contenu qui présente les grandes lignes de l'article
  <section>
    <h2>Titre de la section 1</h2>
    <p>Contenu texte pour la section 1.</p>
  </section>
  ...
  <section>
    <h2>Titre 2 pour dernière section </h2>
    <p>Contenu texte pour la dernière section</p>
  </section>
</article>
```

# address

L'élément address permet de définir des informations permettant aux visiteurs d'une page d'entrer en contact avec son initiateur. Vous pouvez ainsi renvoyer vers les contributeurs d'une page, les adresses e-mail en rapport avec une page de produit ou encore l'adresse postale de la société qui est à l'origine du site.

Par exemple, pour renvoyer vers les pages web des personnes qui assurent la maintenance d'un site, vous pourriez utiliser le code suivant (ce code pourrait être placé dans l'élément footnote) :

Ce site est maintenu par les personnes suivantes :

```
<address>
  <a href="http://www.site.com/nom1">Nom première personne</a>,
  <a href="http://www.site.com/nom2">Nom deuxième personne</a>,
  <a href="http://www.site.com/nom3">Nom troisième personne</a>
  ...
</address>
```

# address

Il est possible de combiner plusieurs types d'informations dans un élément address :

```
<address>
    Ecrit par Michel Martin <br />
    <a href="mailto:lem.korp@free.fr">Ecrivez moi</a><br />
    Address: 5, rue des Formateurs, 75005 Paris<br />
    Téléphone: 01 23 45 67 89
</address>
```

# Mise en forme d'un document en HTML5 et CSS3

Dans cette partie :

- Polices exotiques
- Couleur et image d'arrière-plan
- Gradients linéaires
- Gradient radiaux
- Masquer une image avec un gradient
- RGBA() et HSLA()
- Regrouper des éléments
- etc..

# Polices exotiques

CSS3 permet d'utiliser des polices qui ne sont pas installées sur les "postes clients". C'est la propriété CSS3 @font-face qui autorise cette prouesse.

```
@font-face
{
    font-family: "nom-police";
    [font-style: style-police;]
    [font-variant: petites-capitales;]
    [font-weight: graisse-police;]
    [font-stretch: condensé-étendu;]
    [font-size: taille-police;]
    src: [local('nom-local'),] url('url-police') [format(format-police)];
}
```

- nom-police est le nom de la police, tel qu'il sera utilisé dans la feuille de styles.
- style-police définit le style de la police : normal, italic ou oblique.
- Font-variant: indique la casse des caractères : normal ou small-caps.
- largeur-police indique la largeur de la police : normal, bold, 100, 200, 300, 400, 500, 600, 700, 800 ou 900.
- condensé-étendu définit l'état de compression/extension horizontale de la police : all, normal, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded ou ultra-expanded.
- taille-police définit les tailles possibles pour la police : all pour toutes les polices proportionnelles, une ou deux tailles pour les polices bitmap ou devant être rendues dans une taille ou une fourchette de tailles spécifiques.
- nom-local définit le nom local de la police, au cas où cette dernière serait installée sur l'ordinateur.
- url-police définit l'adresse URL de la police accessible sur le serveur.
- format-police définit le format de la police : truedoc-pfr (TrueDoc™ Portable Font Resource, .pfr), embedded-opentype (Embedded OpenType, .eot), type-1 (PostScript™ Type 1, .pfb ou .pfa), truetype (TrueType, .ttf) ou opentype (OpenType, .ttf).

# Police exotiques

Pour implémenter la police Swiss 721, disponible sur le serveur dans le dossier fonts, sous le nom swiss721lt.pfr, en style Normal et italique et en graisse 200, vous utiliserez les instructions suivantes :

```
@font-face
{
    font-family: "Swiss 721";
    src: url("fonts/swiss721lt.pfr");
    font-style: normal, italic;
    font-weight: 200;
}
```

# Exercice

Définissez le code CSS pour implémenter la police DeliciousRoman, disponible localement sous le nom "Delicious Roman" et sur le site web <http://www.font-face.com/fonts/delicious/>, sous le nom Delicious-Roman.otf, en largeur 400.

# Solution

```
@font-face
{
    font-family: DeliciousRoman;
    src: local('Delicious Roman'), url(http://localhost.../Delicious-Roman.otf);
    font-weight:400;
}
```

# Exercice

Définissez le code CSS pour implémenter la police SketchRockwell, disponible sur le serveur, dans le dossier courant, sous le nom SketchRockwell.ttf.

# Solution

```
@font-face
{
    font-family: SketchRockwell;
    src: url('SketchRockwell.ttf');
}
```

# Polices exotiques

Pour utiliser une police implémentée *via* @font-face, il suffit d'affecter la propriété font-family définie dans @font-face à un élément ou une classe quelconque. Voici un exemple complet qui implémente la police know\_your\_product.ttf côté serveur.

Le code CSS mis en œuvre est le suivant :

```
@font-face
{
    font-family: KnowYourProduct;
    src: url ('know_your_product.ttf');
}

h1
{
    font-family: 'KnowYourProduct';
```

La propriété @font-face définit la police serveur KnowYourProduct (font-family). Le fichier correspondant se trouve dans le dossier courant sous le nom *know\_your\_product.ttf*

La propriété font-family du style h1 est redéfinie pour que les éléments de style h1 appliquent la police KnowYourProduct.

Saisissez ce code et sauvegardez-le dans le fichier font-face.css dans le dossier c:\data\html5css3

# Police exotiques

Pour utiliser ce code CSS3, il suffit de créer un document HTML5 conventionnel et d'utiliser un titre de niveau 1 dans le corps du document.

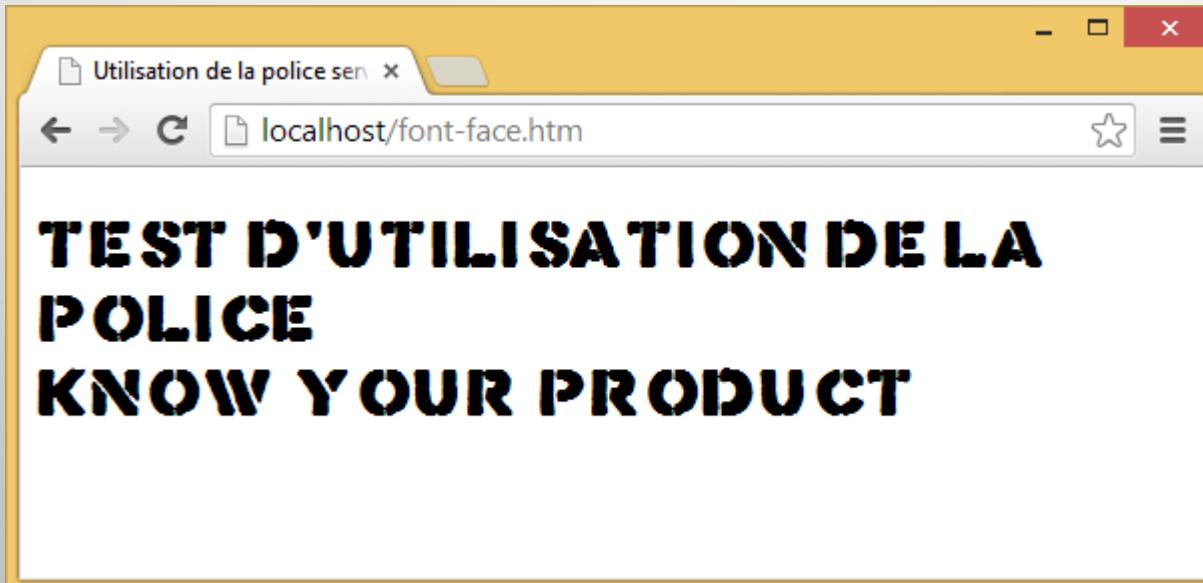
```
<!doctype html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Utilisation de la police serveur know_your_product.ttf</title>
    <link rel="stylesheet" href="font-face.css">
  </head>
  <body>
    <h1>Test d'utilisation de la police know_your_product</h1>
  </body>
</html>
```

Saisissez ce code et sauvegardez-le dans le fichier font-face.htm  
dans le dossier c:\data\html5css3

# Polices exotiques

Insérez le fichier **know\_product.ttf** dans le dossier c:\data\html5css3, puis exécutez le code font-face.htm dans Google Chrome ou Firefox en tapant cette URL :  
<http://localhost/font-face.htm>

Voici ce que vous devriez obtenir :



# Polices exotiques

Affichez ce document dans Internet Explorer. Le résultat est bien décevant :



Cela ne vient pas d'une incompatibilité d'Internet Explorer, mais plutôt d'un mauvais format de police : le navigateur n'est pas en mesure d'exploiter les polices au format TTF. Pour qu'il produise le résultat escompté, la police doit être spécifiée au format EOT.

# Polices exotiques

Modifiez le fichier font-face.css en ajoutant une ligne src: dans la définition @font-face :

```
@font-face
{
    font-family: KnowYourProduct;
    src: url('know_your_product.ttf');
src: url('know_your_product.eot');
}

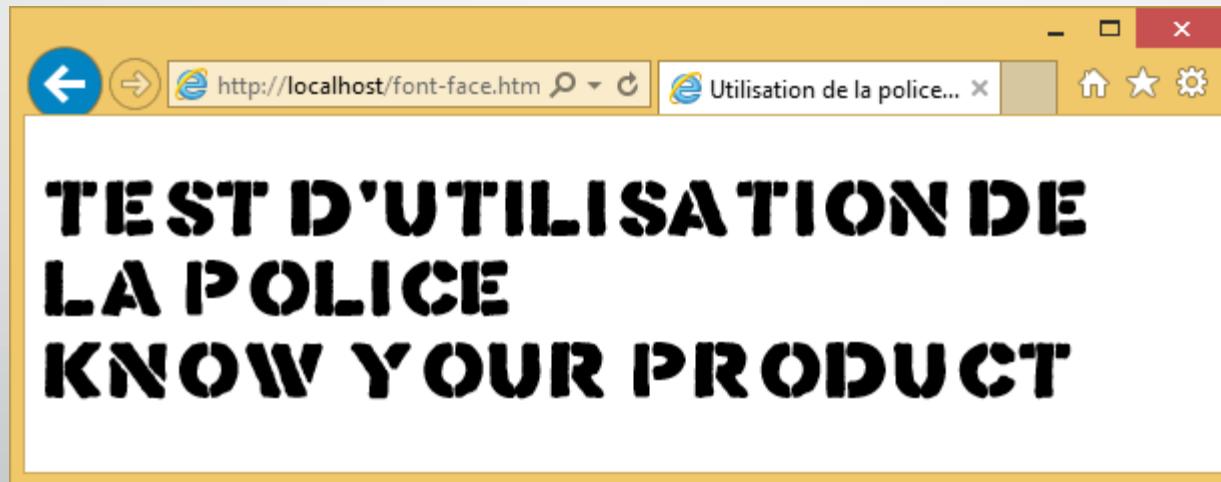
h1
{
    font-family: 'KnowYourProduct';
}
```

# Polices exotiques

Insérez le fichier **know\_your\_product.eot** dans le dossier c:\data\html5css3, puis exécutez le code font-face.htm dans Internet Explorer en tapant cette URL :

<http://localhost/font-face.htm>

Voici ce que vous devriez obtenir :



# Police exotiques

Pour assurer la plus grande compatibilité possible, le mieux est de mettre à disposition les versions EOT, TTF, SVG et WOFF de la police. Il vous suffit alors de référencer ces quatre variantes dans le code CSS :

```
@font-face
{
    font-family: KnowYourProduct;
    src: url('know_your_product.ttf');
    src: url('know_your_product.eot');
    src: url('know_your_product.svg');
    src: url('know_your_product.woff');
}
```

# Où trouver des polices exotiques ?

Pour trouver des polices librement téléchargeables et utilisables, je vous conseille d'aller faire un tour sur le site Font Squirrel, à l'adresse <http://www.fontsquirrel.com/fonts/>, où plusieurs milliers de polices libres de droits vous attendent

Choisissez un filtre et un style dans la partie droite du site. Les polices correspondantes sont affichées dans la partie centrale. Cliquez sur le lien Download correspondant à la police de votre choix et procédez comme indiqué à la section précédente. A titre d'exercice, téléchargez la police script **Alex Brush**.



# Où trouver des polices exotiques ?

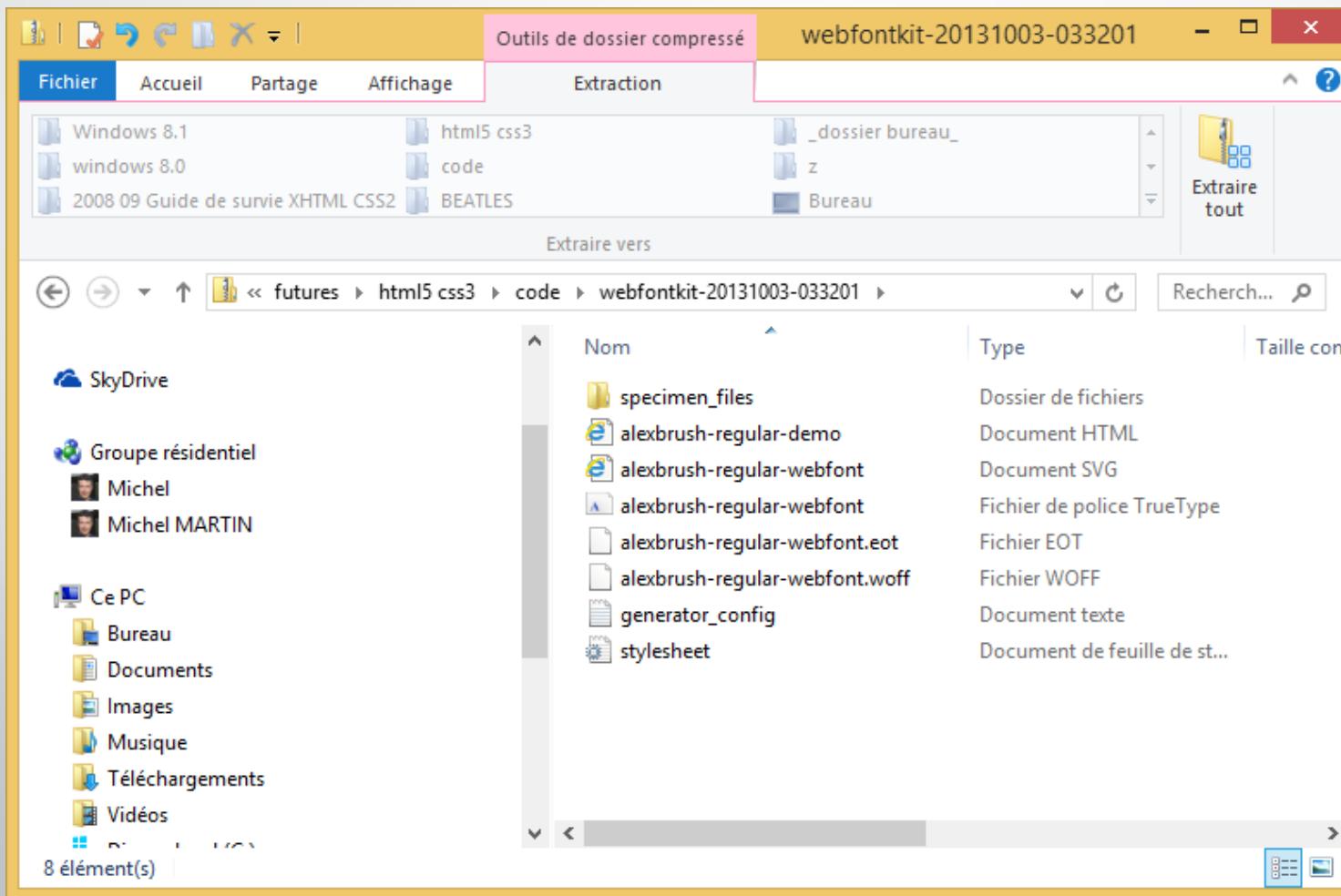
Si vous voulez assurer une compatibilité maximale avec les différents navigateurs disponibles sur le marché, le mieux est de proposer les versions TTF, EOT, SVG et WOFF de vos polices. Là encore, le site Font Squirrel est digne d'intérêt. Cliquez sur **WEBFONT GENERATOR** dans le menu, uploadez une police, générez et téléchargez le kit correspondant en cliquant sur **DOWNLOAD YOUR KIT** :

The screenshot shows the 'WEBFONT GENERATOR' interface. At the top, it says 'Usage: Click the "Add Fonts" button, check the agreement and download your fonts. If you need more fine-grain control, choose the Expert option.' Below this, there's a table for the 'Alex Brush Regular' font. The table includes columns for the font name, file type (TTF), number of glyphs (258 glyphs), and file size (48 KB). There's also a delete icon (an 'X'). Underneath the table, there are three optimization options: 'BASIC', 'OPTIMAL' (which is selected), and 'EXPERT...', each with a brief description. At the bottom, there's an 'Agreement:' section with a checked checkbox stating 'Yes, the fonts I'm uploading are legally eligible for web embedding.' A note below it says 'Font Squirrel offers this service in good faith. Please honor the EULAs of your fonts.' Finally, a large blue button at the bottom right says 'DOWNLOAD YOUR KIT'.

Add Fonts				
Alex Brush Regular	TTF	258 glyphs	48 KB	X
<input type="radio"/> BASIC Straight conversion with minimal processing.	<input checked="" type="radio"/> OPTIMAL Recommended settings for performance and speed.	<input type="radio"/> EXPERT... You decide how best to optimize your fonts.		
Agreement:	<input checked="" type="checkbox"/> Yes, the fonts I'm uploading are legally eligible for web embedding. Font Squirrel offers this service in good faith. Please honor the EULAs of your fonts.			
				DOWNLOAD YOUR KIT

# Où trouver des polices exotiques ?

Le kit téléchargé contient tous les formats nécessaires pour assurer une bonne compatibilité dans les navigateurs.



# Exercice

Ecrivez le code nécessaire pour interfaçer la police alexbrush regular webfont sur tous les navigateurs.

# Solution

```
<!doctype html>
<html>
  <head>
    <meta charset=UTF-8" />
    <title>Utilisation de la police serveur Alexbrush</title>
    <link rel="stylesheet" href="alexbrush.css">
  </head>
  <body>
    <h1>Test d'utilisation de la police Alexbrush</h1>
  </body>
</html>
```

```
@font-face
{
  font-family: Alexbrush;
  src: url('alexbrush-regular-webfont.eot');
  src: url('alexbrush-regular-webfont.ttf');
  src: url('alexbrush-regular-webfont.woff');
}

h1
{
  font-family: 'Alexbrush';
}
```



# Couleur et image d'arrière-plan

La couleur d'arrière-plan d'un élément est définie par la propriété background-color. Voici sa syntaxe :

```
background-color: couleur;
```

La valeur couleur peut être spécifiée :

- "En dur", c'est-à-dire sous la forme d'un nom normalisé : yellow ou skyblue par exemple.
- À l'aide d'un code hexadécimal sur 6 digits : #RRVVBB (où RR, VV et BB sont les composantes rouge, vert et bleu de la couleur, codées en hexadécimal entre 00 et FF).
- À l'aide de la fonction RGB(R,V,B), où R, V et B sont des valeurs comprises entre 0 et 255.

L'image d'arrière-plan d'un élément est définie par la propriété background-image. Voici sa syntaxe :

```
background-image: url (image);
```

Où image est l'adresse URL de l'image. Par exemple img/fond.jpg, ou encore  
http://monsite.com/image/fond.png.

Il n'est pas recommandé d'utiliser des guillemets pour délimiter l'adresse URL de l'image d'arrière-plan car ils peuvent provoquer des dysfonctionnements dans les navigateurs anciens.

# Couleur et image d'arrière-plan

Une image d'arrière-plan peut être répétée horizontalement et/ou verticalement. Pour cela, on donne la valeur repeat (répétition horizontale et verticale), repeat-x (répétition horizontale seulement) ou repeat-y (répétition verticale seulement), à la propriété CSS background-repeat.

La position d'une image d'arrière-plan peut être définie horizontalement et/verticalement : affectez la position horizontale (left, center ou right) et éventuellement la position verticale (top, center ou bottom) à la propriété CSS background-position.

# Couleur et image d'arrière-plan

Dans cet exemple (code005.htm), du texte est affiché au-dessus d'une image en arrière-plan. Nous modifions tour à tour la couleur d'arrière-plan d'un lien hypertexte, d'un paragraphe, puis d'un élément div. Les styles correspondants sont directement définis dans les éléments concernés :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Couleur d'arrière-plan</title>
  </head>
  <body>
    <div style="background-image: url(chat.jpg); font-size: 40px; color: black;">
      <p>La couleur d'arrière-plan de ce texte n'est pas définie. Par contre, <a href="#" style="background-color: red">ce lien a un arrière-plan de couleur rouge</a>.</p>
      <p style="background-color: yellow;">Cette ligne a un arrière-plan de couleur jaune</p>
      <div style="background-color: skyblue;">et celle-ci un arrière-plan bleu ciel.</div>
    </div>
  </body>
</html>
```

