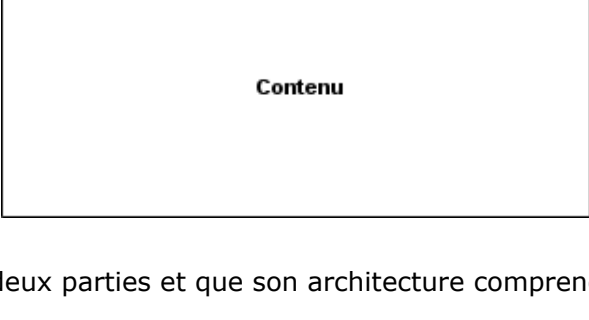


Exemple 1 : XHTML

Commençons par une architecture simple dont voici le schéma.



Nous constatons que son squelette est découpé en deux parties et que son architecture comprend :

- un entête ;
- un contenu.

Mise en page : index.html

L'écriture d'un document XHTML doit être dotée de sens afin de tendre vers le Web sémantique [1].

Les moteurs de recherche tels que Google, Yahoo, Bing ou autres analysent le code HTML afin d'extraire, d'analyser et d'interpréter le sens d'un document et ainsi d'en déduire et de générer un classement pertinent pour les internautes.

Utiliser un balisage sémantique présente aussi d'autres intérêts notamment pour l'accessibilité et le traitement d'un document.

Reprenons notre architecture et écrivons le document XHTML décrivant au mieux celle-ci d'un point de vue sémantique.

Ce code sera écrit dans un fichier **index.html** lui même positionné dans un répertoire **Html** [2].

```
<!DOCTYPE html>

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

<head>

<meta charset="utf-8" />

<title>Xhtml 1</title>

</head>


<body>


<header>

<p>Entête</p>

</header>


<div id="content">

<p>Contenu</p>

</div><!--id="content"-->


</body>

</html>
```

Analysons chaque élément de l'élément corps du document **<body>** de ce document XHTML.

<header>

L'élément **<header>** comme son nom l'indique correspond à l'entête d'une portion d'un document XHTML.

Pour notre exemple, cela correspond à l'entête général de l'élément corps du document **<body>**.

L'élément entête **<header>** est un élément de type **block**.

```
<header>

<p>Entête</p>

</header>
```

<div id="content">

L'élément **<content>**, symbolisant le bloc principal d'un site, n'existe malheureusement pas.

L'élément **<div>** représente une division du document. C'est un élément générique de type **block** servant à découper des parties d'un document et conçu pour être formaté, positionné et stylisé à l'aide d'une feuille de styles.

```
<div id="content">

<p>Contenu</p>

</div><!--id="content"-->
```

L'attribut **id="content"** identifie l'élément **<div>**. Le nom de cet identifiant **id** doit être **unique** dans tout le document XHTML.

L'élément **<div>** avec l'attribut **id="content"** représentera, dans l'ensemble des cours, le contenant renfermant le contenu principal d'un site.

Nous avons ajouté, à la fin de la balise de fermeture **</div>**, le commentaire **<!--id="content"-->** permettant de visualiser rapidement la fin de l'élément **<div id="content">**. Cette technique très utile permet de repérer la fin d'un élément, surtout lorsqu'il y a un nombre important d'éléments identiques imbriqués.

<p>

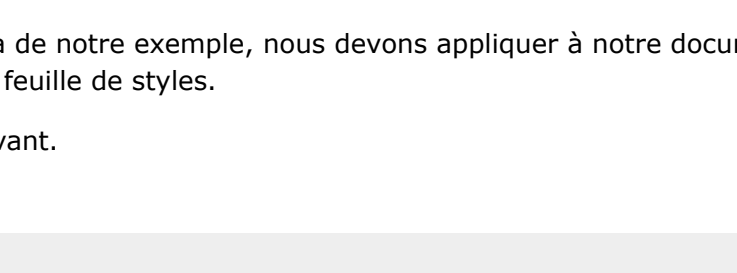
La balise **<p>** marque le début d'un nouveau paragraphe. Le texte contenu entre les balises **<p>** et **</p>** sera affiché dans la fenêtre du navigateur.

L'élément **<p>** est un élément de type **block**, son contenu est de type texte. Il peut aussi contenir des éléments de type **inline** : **
, **<a>, ****, ****, etc.

L'élément **<p>** ne peut pas contenir d'élément de type **block**.

Visualisation du document XHTML

A partir de ce document XHTML, nous obtenons un affichage minimaliste de la page HTML, sans "look and design".



📄 Téléchargement (.zip)

Si nous voulons obtenir l'apparence du schéma de notre exemple, nous devons appliquer à notre document une mise en forme et un positionnement des éléments de l'architecture en utilisant une feuille de styles.

C'est ce que nous verrons dans le chapitre suivant.

Bibliographie

- 1 http://fr.wikipedia.org/wiki/Web_sémantique
- 2 Le fichier **index.html** est le fichier répertorié par défaut dans le fichier **httpd.conf** de configuration du serveur Apache.



Exemple 1 : CSS

Ecrivons à présent nos premières feuilles de styles.

Feuille de styles CSS3 : layout.css

Dans un premier temps, nous délimiterons la dimension occupée par la page html, puis nous préciserons sa typographie.

Et dans un deuxième temps, nous décrirons la mise en page du site, c'est à dire la position de ses différents éléments par rapport à l'architecture du document XHTML.

La feuille de styles de la mise en page sera écrite dans un fichier **layout.css** positionné dans le répertoire **Css**.

Hauteur – Largeur du document
Définissons la hauteur et la largeur occupée par la page html et par le corps du document :

```
html, body

{

    height: 100%;

    width: 100%;

    overflow: auto;

}
```

En donnant une hauteur et une largeur de 100% aux éléments **<body>** et **<html>**, l'élément corps du document **<body>** occupe toute la page.

La propriété **overflow: auto** indique aux navigateurs que si la place des éléments est insuffisante dans la fenêtre celle-ci peut positionner automatiquement des barres de défilement horizontal et/ou vertical.

Typographie

La couleur de fond et de la police, et le choix de la typographie sont décrits pour l'ensemble du corps du document dans le sélecteur **body** :

```
body

{

    background-color: #fff;

    color: #000;

    font-family: Verdana, Arial, Helvetica, sans-serif;

    font-size: 0.75em;

}
```

Les couleurs sont définies en notation hexadécimale. Le symbole **#** définit une valeur RGB (Red, Green, Blue) codée en hexadécimal. Les trois valeurs sont regroupées sur six caractères (ex : #ffff00 correspond au jaune).

Quand les trois valeurs correspondent à trois paires jumelles (ex : #aacc00 et #88bb22), nous pouvons écrire la couleur en notation hexadécimale courte en écrivant qu'un seul caractère de la paire (ex: #ac0 et #8b2).

Il est recommandé d'utiliser pour les sorties écrans des polices de taille relative plutôt que celles de taille fixe appropriées pour les sorties papiers. On utilisera la taille relative en pourcentage **%** ou en **em** (cadratin) qui correspond à la largeur du caractère M (majuscule) dans la police de référence.



Dans la propriété **font-family** ne pas oublier la police par défaut (**serif** ou **sans-serif**) qui est obligatoire.

- La police **serif** correspond aux polices avec empattement telles que : **Times, Georgia, ...** ;
- La police **sans-serif** le contraire telles que : **Arial, Helvetica,**

Positionnement et apparence de l'entête

Dans le sélecteur d'entête **header**, on spécifie :

- son type de positionnement : **position: absolute** ;
- son positionnement par rapport au sommet : **top: 0** et à gauche : **left: 0** ;
- sa largeur : **width: 100%** ;
- sa hauteur : **height: 2em** ;
- sa couleur de fond : **background-color: #f00** (rouge) ;
- sa couleur de police : **color: #fff** (blanche).

```
header

{

    position: absolute;

    top: 0;

    left: 0;

    width: 100%;

    height: 2em;

    background-color: #f00;

    color: #fff;

}
```

Positionnement et apparence du contenu

Utilisons le sélecteur d'identifiant **#content** pour décrire les propriétés de l'élément **<div id="content">**.

Décalons cet élément d'une marge par rapport supérieure de : **margin-top: 2em**, correspondant à la hauteur de l'élément entête **<header>**.

La largeur **width: 100%** correspond à la largeur de la page.

Nous utilisons la propriété **overflow: auto** afin d'autoriser les barres de défilement horizontal et vertical dans l'élément **<div>**.

```
#content

{

    margin-top: 2em;

    width: 100%;

    overflow: auto;

}
```

Positionnement et apparence des paragraphes

Les textes dans les paragraphes sont centrés : **text-align: center**, et la graisse de leur police est mise à gras : **font-weight: bold**.

```
p

{

    text-align: center;

    font-weight: bold;

}
```

Le texte dans le paragraphe de l'élément **<header>** est centré verticalement. En mettant la propriété hauteur de ligne (**line-height**) à la même valeur que la hauteur de l'élément **<header>** le texte sera automatiquement centré verticalement : **line-height: 2em**.

```
header p

{

    line-height: 2em;

}
```

Le texte dans le paragraphe de l'élément **<div>** est décalé du sommet de 10em : **margin-top: 10em**;

```
#content p

{

    margin-top: 10em;

}
```

Feuille de styles CSS3 : exo.css

Le fichier maître **exo.css** appelle tous les fichiers des feuilles de style dont à besoin l'application.

Le nom du fichier de la feuille de style maître devra être explicite et le plus souvent correspondre au nom de l'application ou au nom du répertoire principal.

```
@charset "UTF-8";

@import "normalisation.css" screen;

@import "layout.css" screen;
```

Le fichier maître commence par encoder les fichiers via la règle **@charset** suivi de l'encodage.

L'appel aux différents fichiers de feuilles de styles se fait par la règle **@import**.

Il est recommandé d'indiquer sur quel média la règle s'applique, dans notre exemple screen (écran).

Le fichier maître **exo.css** sera positionné dans le répertoire **Css**.

Mise en page opérationnel pour le design : index.html

Nous pouvons ajouter le fichier maître **exo.css** en utilisant l'élément **<link>** et écrire notre document XHTML opérationnel pour le design.

```
<!DOCTYPE html>

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

<head>

    <meta charset="utf-8" />

    <title>Xhtml 1</title>

    <link rel="stylesheet" type="text/css" href="../Css/exo.css" />

</head>

<body>

    <header>

        <p>Entête</p>

    </header>

    <div id="content">

        <p>Contenu</p>

    </div><!--id="content"-->

</body>

</html>
```

Visualisation et téléchargement

A partir de ce document XHTML et de l'appel aux feuilles de style de l'application, nous obtenons l'affichage de la page HTML.



Téléchargement (.zip)



Exemple 2

Continuons par une nouvelle architecture simple dont voici le schéma.



Document XHTML5 : index.html

Pour cet architecture nous avons encore un découpage du site en deux parties :

- un menu ;
- un contenu.

Ce qui donne comme document XHTML :

```
<!DOCTYPE html>

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

<head>

<meta charset="utf-8" />

<title>Xhtml 3</title>

<link rel="stylesheet" type="text/css" href="../Css/exo.css" />

</head>

<body>

<nav>

<p>Menu</p>

</nav>

<div id="content">

<p>Contenu</p>

</div><!--id="content"-->

</body>

</html>
```

<nav>

L'élément entête **<header>** a été remplacée par l'élément navigation **<nav>**, comme son nom l'indique, cet élément correspond à la navigation dans le site, c'est à dire à un menu.

L'élément navigation **<nav>** est un élément de type **block**.

Feuille de styles CSS3 : layout.css

Positionnement et apparence du menu

Nous supprimons le sélecteur d'entête **header** et nous le remplaçons par celui de navigation **nav** où nous spécifions :

- son type de positionnement de l'élément : **position: absolute** ;
- son positionnement par rapport au sommet : **top: 0** et à gauche : **left: 0** ;
- sa largeur : **width: 10em** ;
- sa hauteur : **height: 100%** ;
- sa couleur de fond : **background-color: #00f** (bleu);

```
nav

{

position: absolute;

top: 0;

left: 0;

width: 10em;

height: 100%;

background-color: #00f;

}
```

Positionnement et apparence du contenu

L'élément **<div id="content">** est décalé avec une marge gauche correspondant à la largeur de l'élément **<nav>** : **margin-left: 10em**.

La hauteur **height: 100%** correspond à la hauteur de la page.

Nous utilisons la propriété **overflow: auto** afin d'autoriser les barres de défilement horizontal et vertical dans l'élément **<div>**.

```
#content

{

margin-left: 10em;

height: 100%;

overflow: auto;

}
```

Positionnement et apparence des paragraphes

Les textes dans les paragraphes sont centrés : **text-align: center**, la graisse de la police est mise à gras : **font-weight: bold**, et le texte est décalé du sommet de 10em : **margin-top: 10em**.

```
p

{

margin-top: 10em;

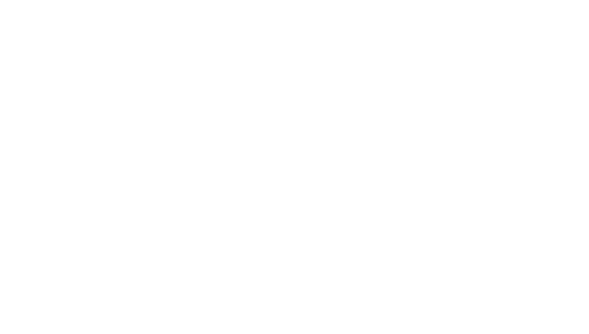
text-align: center;

font-weight: bold;

}
```

Visualisation du document XHTML

A partir de ce document XHTML et de l'appel aux feuilles de style de l'application, nous obtenons l'affichage de la page HTML.



📄 Téléchargement (.zip)



Exemple 3

Continuons par une nouvelle architecture combinant les deux architectures précédentes dont voici le schéma.



Document XHTML5 : index.html

Pour cet architecture nous avons un découpage du site en trois parties :

- un entête ;
- un menu ;
- un contenu.

Ce qui donne comme document XHTML :

```
<!DOCTYPE html>

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

<head>

<meta charset="utf-8" />

<title>Xhtml 4</title>

<link rel="stylesheet" type="text/css" href="../Css/exo.css" />

</head>

<body>

<header>

<p>Entête</p>

</header>

<nav>

<p>Menu</p>

</nav>

<div id="content">

<p>Contenu</p>

</div><!--id="content"-->

</body>

</html>
```

Feuille de styles CSS3 : layout.css

Positionnement et apparence
Dans cet exercice on fusionne les fichiers **layout.css** des deux exemples précédents.
Pour l'élément paragraphe **<p>** contenu dans l'élément entête **<header>** nous ajoutons une marge à gauche de : **margin-left: 10em** afin d'éviter un recouvrement par l'élément de navigation **<nav>** (largeur de 10em).

Nous annulons la marge supérieure : **margin-top: 0** pour l'élément paragraphe **<p>** contenu dans l'élément entête **<header>**.

```
header p

{

margin-top: 0;

margin-left: 10em;

line-height: 2em;

}
```

Nous modifions le sélecteur d'identifiant **#content** afin de tenir compte de la nouvelle architecture.

Nous décalons l'élément **<div id="content">** en mettant :

- une marge supérieure correspondant à la hauteur de l'élément **<header>** : **margin-top: 2em** ;
- une marge gauche correspondant à la largeur de l'élément **<nav>** : **margin-left: 10em**.

```
#content

{

margin-top: 2em;

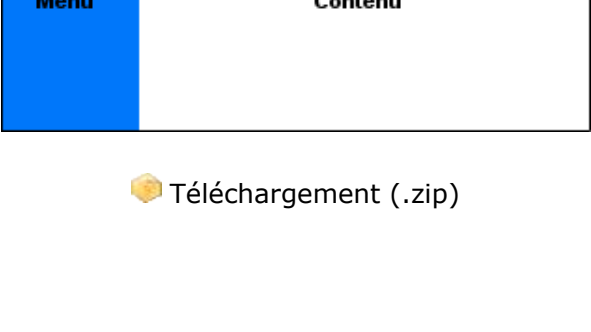
margin-left: 10em;

overflow: auto;

}
```

Visualisation du document XHTML

A partir de ce document XHTML et de l'appel aux feuilles de styles de l'application, nous obtenons l'affichage de la page HTML.

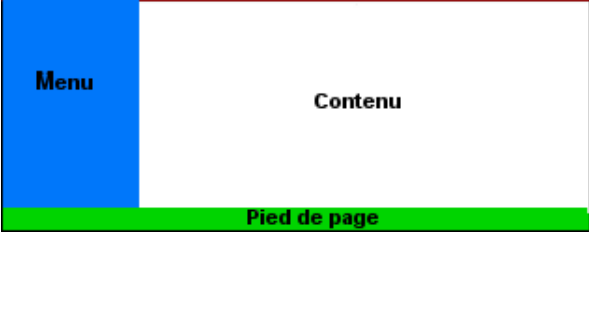


📄 Téléchargement (.zip)



Exemple 4

Continuons par une dernière architecture simple dont voici le schéma.



Document XHTML5 : index.html

Pour cet architecture nous avons un découpage du site en quatre parties :

- un entête ;
- un menu ;
- un contenu ;
- un pied de page.

Ce qui donne comme document XHTML :

```
<!DOCTYPE html>

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

<head>

<meta charset="utf-8" />

<title>Xhtml 4</title>

<link rel="stylesheet" type="text/css" href="../Css/exo.css" />

</head>

<body>

<header>

<p>Entête</p>

</header>

<nav>

<p>Menu</p>

</nav>

<div id="content">

<p>Contenu</p>

</div><!-- id="content" -->

<footer>

<p>Pied de page</p>

</footer>

</body>

</html>
```

Nous avons ajouté l'élément **<footer>** qui sémantiquement correspond à la partie pied de page.

L'élément **<footer>** est un élément de type **block**.

Feuille de styles CSS3 : layout.css

Positionnement et apparence du pied de page

Dans cet exercice nous ajoutons le sélecteur **footer** où nous spécifions :

- son type de positionnement de l'élément : **position: absolute** ;
- son positionnement par rapport au bas de page : **bottom: 0** et à gauche : **left: 0** ;
- sa largeur : **width: 100%** ;
- sa hauteur : **height: 4em** ;
- sa couleur de fond : **background-color: #0f0** (vert).

```
footer

{

position: absolute;

bottom: 0;

left: 0;

width: 100%;

height: 4em;

background-color: #0f0;

}
```

Pour l'élément paragraphe **<p>** contenu dans l'élément pied de page **<footer>** nous annulons la marge du sommet : **margin-top: 0**, comme pour l'élément entête **<header>**.

Le texte dans le paragraphe de l'élément **<footer>** est centré verticalement. En mettant la propriété hauteur de ligne (**line-height**) à la même valeur que la hauteur de l'élément **<footer>** le texte sera automatiquement centré : **line-height: 4em**.

```
footer p

{

margin-top: 0;

line-height: 4em;

}
```



L'élément pied de page **<footer>** recouvre l'élément de navigation **<nav>**. Si nous souhaitons le contraire, il suffit de rajouter sur le sélecteur **nav** la propriété : **z-index: 1**.

Visualisation du document XHTML

A partir de ce document XHTML, nous obtenons l'affichage de la page HTML.



Téléchargement (.zip)

