HTML

##### **Les balises en paires**

**Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :**

<titre>Ceci est un titre</titre>

**Une balise orpheline s'écrit comme ceci :**

<image />

* **le HTML définit le fond (contenu, logique des éléments) ;**
* **le CSS définit la forme (apparence).**

# Structure de base d'une page HTML5

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Titre</title>

</head>

<body>

</body>

</html>

# ORGANISER SON TEXTE

##### **L'encodage (charset)**

<meta charset="utf-8" />

**Cette balise indique l'encodage utilisé dans votre fichier .html.**

#### Insérer un commentaire

<!-- Ceci est un commentaire -->

* <p> </p> : pour organiser son texte en paragraphes ;
* <br/> : pour aller à la ligne.
* <h1> </h1> : signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.
* <h2> </h2> : signifie « titre important ».
* <h3> </h3> <h4> </h4> :<h5> </h5> <h6> </h6> (moin important etc…)

 la balise <em> </em> a pour conséquence de mettre le texte en italique.

La balise <strong> ne signifie pas « mettre en gras » mais « important ».

<mark> a pour effet de surligner le texte.\*

<ul>

<li>Fraises</li>

<li>Framboises</li>

<li>Cerises</li>

</ul>

 <ul></ul> délimite toute la liste ;

 <li></li> délimite un élément de la liste (une puce).

Une liste ordonnée fonctionne de la même façon, seule une balise change :

il faut remplacer <ul></ul>par <ol></ol>.  
L'ordre dans lequel vous placez les éléments de la liste est important. Le premier <li></li>sera l'élément n° 1, le second sera le n°2 etc…

# CRÉER DES LIENS

Pour faire un lien, la balise que nous allons utiliser est très simple à retenir : <a>. Il faut cependant lui ajouter un attribut, href, pour indiquer vers quelle page le lien doit conduire.

<p>Bonjour. Souhaitez-vous visiter le <a href="http://www.siteduzero.com">Site du Zéro</a> ?<br />

C'est un bon site ! ;-)</p>

vous devrez remplacer tous les « & » par « &amp; »

Voici le code que nous allons utiliser dans nos fichiers page1.html et page2.html se trouvant dans le même dossier.

<p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la page 2</a> ?</p>

S'il y avait plusieurs sous-dossiers, on écrirait ceci :

<a href="contenu/autredossier/page2.html">

Si votre fichier cible est placé dans un dossier qui se trouve « plus haut » dans l'arborescence, il faut écrire deux points comme ceci :

<a href="../page2.html">

Pour créer une ancre, il suffit de rajouter l'attribut id à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise, un titre par exemple.  
Utilisez l'attribut id pour donner un nom à l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre. Par exemple :

<h2 id="mon\_ancre">Titre</h2>

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut href contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

<a href="#mon\_ancre">Aller vers l'ancre</a>

Normalement, si vous cliquez sur le lien, cela vous amènera plus bas dans la même page (à condition que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement).

Voici une page qui contient trois liens, chacun amenant vers une des ancres de la page de l'exemple précédent :

<h1>Le Mégamix</h1>

<p>

Rendez-vous quelque part sur une autre page :<br />

<a href="ancres.html#cuisine">La cuisine</a><br />

<a href="ancres.html#rollers">Les rollers</a><br />

<a href="ancres.html#arc">Le tir à l'arc</a><br />

</p>

La bulle d'aide peut être utile pour informer le visiteur avant même qu'il n'ait cliqué sur le lien.  
Voici comment reproduire ce résultat :

<p>Bonjour. Souhaitez-vous visiter le <a href="http://www.siteduzero.com" title="Réservé aux débutants">Site du Zéro</a> ?</p>

#### Un lien qui ouvre une nouvelle fenêtre

Il est possible de « forcer » l'ouverture d'un lien dans une nouvelle fenêtre. Pour cela, on rajouteratarget="\_blank" à la balise <a> :

<p>Bonjour. Souhaitez-vous visiter le <a href="http://www.siteduzero.com" target="\_blank">Site du Zéro</a> ?<br />

Le site s'affichera dans une autre fenêtre.</p>

#### Un lien pour envoyer un e-mail

Si vous voulez que vos visiteurs puissent vous envoyer un e-mail, vous pouvez utiliser des liens de typemailto. Rien ne change au niveau de la balise, vous devez simplement modifier la valeur de l'attributhref comme ceci :

<p><a href="mailto:votrenom@bidule.com">Envoyez-moi un e-mail !</a></p>

Par exemple, supposez que vous vouliez faire télécharger monfichier.zip. Placez simplement ce fichier dans le même dossier que votre page web (ou dans un sous-dossier) et faites un lien vers ce fichier :

<p><a href="monfichier.zip">Télécharger le fichier</a></p>

# LES IMAGES

#### Le JPEG = Photo

#### Le PNG = Diverse image

#### Le GIF = Animé

Quelle est la fameuse balise qui va nous permettre d'insérer une image ? Il s'agit de… <img /> !

La balise doit être accompagnée de deux attributs obligatoires :

* src : il permet d'indiquer où se trouve l'image que l'on veut insérer. Vous pouvez soit mettre un chemin absolu (ex. : http://www.site.com/fleur.png), soit mettre le chemin en relatif (ce qu'on fait le plus souvent). Ainsi, si votre image est dans un sous-dossier images, vous devrez taper : src="images/fleur.png"
* alt : cela signifie « texte alternatif ». On doit *toujours* indiquer un texte alternatif à l'image, c'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée (cela arrive. Cela aide aussi les robots des moteurs de recherche pour les recherches d'images. Pour la fleur, on mettrait par exemple : alt="Une fleur".

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe (<p></p>). Voici un exemple d'insertion d'image :

<p>

Voici une photo que j'ai prise lors de mes dernières vacances à la montagne :<br />

<img src="images/montagne.jpg" alt="Photo de montagne" title="C'est beau les Alpes quand même !" />

</p>

#### Ajouter une infobulle

L'attribut permettant d'afficher une bulle d'aide est le même que pour les liens : il s'agit de title. Cet attribut est facultatif (contrairement à alt).

Survolez la photo avec la souris pour voir l'infobulle apparaître.

Je les place toutes les deux dans un dossier appelé par exemple img. J'affiche la versionmontagne\_mini.jpg sur ma page et je fais un lien vers montagne.jpg pour que l'image agrandie s'affiche lorsqu'on clique sur la miniature.

Voici le code HTML que je vais utiliser pour cela :

<p>

Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus !<br />

<a href="img/montagne.jpg"><img src="img/montagne\_mini.jpg" alt="Photo de montagne" title="Cliquez pour agrandir" /></a>

</p>

* Si elle n'apporte aucune information (c'est juste une illustration pour décorer) : placez l'image dans un paragraphe.
* Si elle apporte une information : placez l'image dans une figure.
* <p>Connaissez-vous le logiciel Bloc-Notes ? On peut faire des sites web avec !</p>
* <figure>
* <img src="images/blocnotes.png" alt="Bloc-Notes" />
* <figcaption>Le logiciel Bloc-Notes</figcaption>
* </figure>

CSS

* En CSS, on sélectionne quelles portions de la page HTML on veut modifier et on change leur présentation avec des propriétés CSS :

balise1

{

propriete1: valeur1;

propriete2: valeur2;

}

En fait, vous devez écrire un nom qui sert à identifier la balise. Ce que vous voulez, du moment que le nom commence par une lettre.

Par exemple, je vais associer la classe introduction à mon premier paragraphe (ligne 12) :

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

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

<title>Premiers tests du CSS</title>

</head>

<body>

<h1>Mon super site</h1>

<p class="introduction">Bonjour et bienvenue sur mon site !</p>

<p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>

</body>

</html>

Maintenant que c'est fait, votre paragraphe est identifié. Il a un nom : introduction. Vous allez pouvoir réutiliser ce nom dans le fichier CSS pour dire : « *Je veux que seules les balises qui ont comme nom 'introduction' soient affichées en bleu* ».

Pour faire cela en CSS, indiquez le nom de votre classe en commençant par un point, comme ci-dessous :

.introduction

{

*color*: blue;

}

<img src="images/logo.png" alt="Logo du site" id="logo" />

Si vous utilisez des id, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'id par un dièse (#) :

#logo

{

/\* Indiquez les propriétés CSS ici \*/

}

<span> </span> : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte

On la met autour de « bienvenue », on lui ajoute une classe (du nom qu'on veut), on crée le CSS et c'est gagné !

<p>Bonjour et <span class="salutations">bienvenue</span> sur mon site !</p>

.salutations

{

*color*: blue;

}

<div> </div> : c'est une balise de type **block**, qui entoure un bloc de texte

##### \* : sélecteur universel

\*

{

}

Sélectionne toutes les balises sans exception. On l'appelle le sélecteur universel.

Et plein d’autre …

# FORMATAGE DU TEXTE

## La taille

Voici un exemple d'utilisation (placez ce code dans votre fichier .css) :

p

{

*font-size*: 14px; /\* Paragraphes de 14 pixels ou changer de valeur au choix \*/

}

p

{

*font-size*: small; /\*(x)(xx)small,medium,(x)(xx)large,\*/

}

p

{

*font-size*: 0.8em; /\*valeur 1 taille normal\*/

}

## La police

em

{

*font-style*: normal; /\*oblique,italic\*/

}

h1

{

*font-weight*: bold; (gras) /\*normal\*/

}

h1

{

*text-decoration*: blink; /\*underline, line-through, overline, none\*/

}

h1

{

*text-align*: center; /\*left, right, justify\*/

}

<p><img src="flash.gif" class="imageflottante" alt="Image flottante" /> Ceci est un texte normal de paragraphe, écrit à la suite de l'image et qui l'habillera car l'image est flottante.</p>

Vous devez placer l'élément flottant en premier dans le code HTML.

Si vous placez l'image après le paragraphe, l'effet ne fonctionnera pas.

Voici le seul bout de code CSS qu'on ait besoin de taper, qui permet de faire flotter l'image à gauche :

.imageflottante

{

*float*: *left*;

}

Il existe en fait une propriété CSS qui permet de dire : « *Stop, ce texte doit être en-dessous du flottant et non plus à côté* ». C'est la propriété clear, qui peut prendre ces trois valeurs :

* left : le texte se poursuit en-dessous après un float: left;
* right : le texte se poursuit en-dessous après un float: right;
* both : le texte se poursuit en-dessous, que ce soit après un float: left; ou après unfloat: right;.

Pour simplifier, on va utiliser tout le temps le clear: both, qui marche après un flottant à gauche et après un flottant à droite (cela fonctionne donc à tous les coups). Pour illustrer son fonctionnement, on va prendre ce code HTML :

<p><img src="flash.gif" class="imageflottante" alt="Image flottante" /></p>

<p>Ce texte est écrit à côté de l'image flottante.</p>

<p class="dessous">Ce texte est écrit sous l'image flottante.</p>

Et ce code CSS :

.imageflottante

{

*float*: *left*;

}

.dessous

{

*clear*: both;

}

# LA COULEUR ET LE FOND

h1

{

*color*: maroon; /\*white, silver, gray, black, red, marroon,

lime, green, yellow, olive, blue, navy, fuchsia,

purple, aqua, teal\*/

}

p

{

color: #FFFFFF; /\*000000 et autre code de ce genre \*/

}

p

{

color: rgb(240,96,204); /\*couleur récupérable sur Paint\*/

}

/\* On travaille sur la balise body, donc sur TOUTE la page \*/

body

{

*background-color*: black; /\* Le fond de la page sera noir \*/

*color*: white; /\* Le texte de la page sera blanc \*/

}

la balise <mark> qui permet de mettre en valeur certains mots ? Utilisons-la à nouveau ici :

<h1>Qui a éteint la lumière ?</h1>

<p>Brr, il fait tout noir sur ce site, c'est un peu <mark>inquiétant</mark> comme ambiance non vous trouvez pas ?</p>

Par défaut, le texte s'affiche sur un fond jaune. Vous pouvez changer ce comportement en CSS :

body

{

*background-color*: black;

*color*: white;

}

mark

{

/\* La couleur de fond prend le pas sur celle de toute la page \*/

*background-color*: red;

*color*: black;

}

body

{

*background-image*: url("neige.png");

*background-attachment*: fixed; /\* Le fond restera fixe, scroll (défile avec le texte)\*/

}

body

{

*background-image*: url("soleil.png");

*background-repeat*: no-repeat; /\*no-repeat, repeat-x, repeat-y\*/

}

background-position: 30px 50px;

background-position: top right; /\*top, bottom, left, center, right\*/

Combiner les propriétés

Si vous utilisez beaucoup de propriétés en rapport avec le fond (comme c'est le cas sur ce dernier exemple), vous pouvez utiliser une sorte de « super-propriété » appelée background dont la valeur peut combiner plusieurs des propriétés vues précédemment : background-image, background-repeat,background-attachment et background-position.

On peut donc tout simplement écrire :

body

{

*background*: url("soleil.png") fixed no-repeat *top* *right*;

}

 il est possible de donner plusieurs images de fond à un élément. Pour cela, il suffit de séparer les déclarations par une virgule, comme ceci :

body

{

*background*: url("soleil.png") fixed no-repeat top right, url("neige.png") fixed;

}

p

{

*opacity*: 0.6; /\*de 0 jusqu’à 1\*/

}

p

{

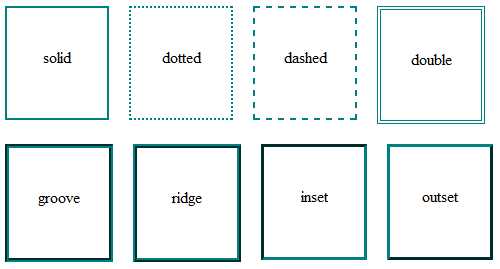
*background-color*: rgba(255, 0, 0, 0.5); /\* Fond rouge(255,0,0) à moitié transparent(0.5) \*/

}

# LES BORDURES ET LES OMBRES

Pour border on peut utiliser jusqu'à trois valeurs pour modifier l'apparence de la bordure :

* **La largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px).
* **La couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur (black, red,…), soit une valeur hexadécimale (#FF0000), soit une valeur RGB (rgb(198, 212, 37)).
* **Le type de bordure** : là, vous avez le choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc. Voici les différentes valeurs disponibles :
* **Le placement :** border-top, border-bottom, border-left, border-right



p

{

*border-radius*: 10px; /\*bordure arrondie\*/

}

p

{

*border-radius*: 10px 5px 10px 5px; /\*HG, HD, BD, BG\*/

}

p

{

border-radius: 20px / 10px; /\*raffinée\*/

}

p

{

*box-shadow*: 6px 6px 0px black inset; /\*decalage horizontale de l’ombre, Vertical,

Adoucissement du dégradé, couleur, ombre interieur \*/

}

p

{

*text-shadow*: 2px 2px 4px black;

}

# CRÉATION D'APPARENCES DYNAMIQUES

Voici un exemple de présentation des liens, mais n'hésitez pas à inventer le vôtre :

a /\* Liens par défaut (non survolés) \*/

{

*text-decoration*: none;

*color*: red;

*font-style*: italic;

}

A ou p:hover /\* Apparence au survol des liens \*/

{

*text-decoration*: underline;

*color*: green;

}

a:active /\* Quand le visiteur clique sur le lien \*/

{

*background-color*: #FFCC66;

}

a:focus /\* Quand le visiteur sélectionne le lien \*/

{

*background-color*: #FFCC66;

}

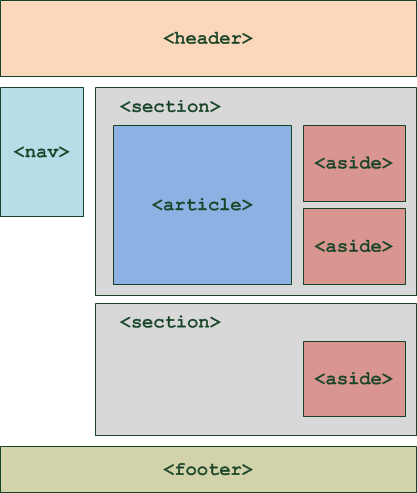
a:visited /\* Quand le visiteur a déjà vu la page concernée \*/

{

*color*: #AAA; /\* Appliquer une couleur grise \*/

}

# STRUCTURER SA PAGE



 <header> : en-tête ;

 <footer> : pied de page ;

 <nav> : liens principaux de navigation ;

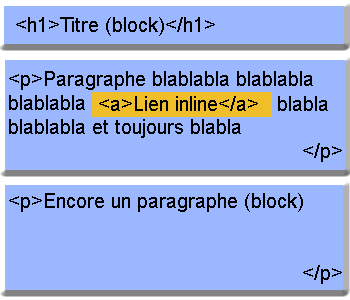
 <section> : section de page ;

 <aside> : informations complémentaires ;

 <article> : article indépendant.

* Le code JavaScript HTML5shiv permet de faire en sorte que ces balises soient reconnues pour les versions d'Internet Explorer antérieures à IE9.

# LE MODÈLE DES BOÎTES



Différence entre une balise inline et une balise block

* Sur fond bleu, vous avez tout ce qui est de type block.
* Sur fond jaune, vous avez tout ce qui est de type inline.

#### Quelques exemples

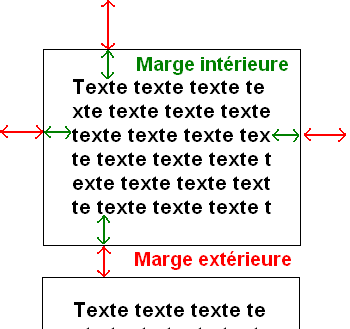
| **Balises block** | **Balises inline** |
| --- | --- |
| <p> | <em> |
| <footer> | <strong> |
| <h1> | <mark> |
| <h2> | <a> |
| <article> | <img /> |
| … | … |

## Les dimensions

Nous allons ici travailler uniquement sur des balises de type block.

* width : c'est la largeur du bloc. À exprimer en pixels (px) ou en pourcentage (%).
* height : c'est la hauteur du bloc. Là encore, on l'exprime soit en pixels (px), soit en pourcentage (%).
* p
* { *width*: 50%;
* Ou width: 250px;
* }
* p
* { *width*: 50%;
* *min-width*: 400px; <!-- min max, height,width 🡪
* }

## Les marges



Marges extérieure et intérieure

 padding : indique la taille de la marge intérieure. À exprimer en général en pixels (px).

 margin : indique la taille de la marge extérieure. Là encore, on utilise le plus souvent des pixels.

p

{

*width*: 350px;

*border*: 1px solid black;

*text-align*: justify;

*padding*: 12px;

*margin*: 50px; /\* Marge extérieure de 50px \*/

}

margin-top/ bottom/ left/ right

padding-top/ bottom/ left/ right

Pour centrer, il faut respecter les règles suivantes :

* donnez une largeur au bloc (avec la propriété width) ;
* indiquez que vous voulez des marges extérieures automatiques, comme ceci : margin: auto;.
* p
* {
* *width*: 350px; /\* On a indiqué une largeur (obligatoire) \*/
* *margin*: auto; /\* On peut donc demander à ce que le bloc soit centré avec auto \*/
* *border*: 1px solid black;
* *text-align*: justify;
* *padding*: 12px;
* *margin-bottom*: 20px;
* }

Si vous voulez que le texte ne dépasse pas des limites du paragraphe, il va falloir utiliser la propriété overflow. Voici les valeurs qu'elle peut accepter :

* visible (par défaut) : si le texte dépasse les limites de taille, il reste visible et sort volontairement du bloc.
* hidden : si le texte dépasse les limites, il sera tout simplement coupé. On ne pourra pas voir tout le texte.
* scroll : là encore, le texte sera coupé s'il dépasse les limites. Sauf que cette fois, le navigateur mettra en place des barres de défilement pour qu'on puisse lire l'ensemble du texte. C'est un peu comme un cadre à l'intérieur de la page.
* auto : c'est le mode « pilote automatique ». En gros, c'est le navigateur qui décide de mettre ou non des barres de défilement (il n'en mettra que si c'est nécessaire). C'est la valeur que je conseille d'utiliser le plus souvent.

Avec overflow: hidden; le texte est donc coupé (on ne peut pas voir la suite), comme sur la figure suivante.

#### word-wrap : couper les textes trop larges

p

{

*word-wrap*: break-word;

}

# LE POSITIONNEMENT EN CSS

nav

{*float*: *left*;

*width*: 150px;

*border*: 1px solid black;

}

section

{*margin-left*: 170px;

*border*: 1px solid blue;

}



* Le positionnement inline-block consiste à affecter un type inline-block à nos éléments grâce à la propriété display. Ils se comporteront comme des inlines (placement de gauche à droite) mais pourront être redimensionnés comme des blocs (avec width et height). Cette technique est à préférer au positionnement flottant.
* Le positionnement absolu permet de placer un élément où l'on souhaite sur la page, au pixel près.
* Le positionnement fixe est identique au positionnement absolu mais l'élément restera toujours visible même si on descend plus bas dans la page.
* Le positionnement relatif permet de décaler un bloc par rapport à sa position normale.
* Un élément A positionné en absolu à l'intérieur d'un autre élément B (lui-même positionné en absolu, fixe ou relatif) se positionnera par rapport à l'élément B, et non par rapport au coin en haut à gauche de la page.
* strong
* {
* *background-color*: red;
* *color*: yellow;
* *position*: relative /\* position: absolute ; \*/
* *left*: 55px; /\* top,left,right, bottom \*/
* *top*: 10px;
* }

# Arretee a G3 CREATION SITE