

Site de référence : <https://www.w3schools.com>

Pour créer des sites web, on utilise deux langages informatiques :

- **HTML** : permet d'écrire et organiser le **contenu** de la page (paragraphe, titres. . .) ;
- **CSS** : permet de mettre en **forme** la page (couleur, taille, disposition. . .).

Le navigateur web est un programme qui permet d'afficher des sites web. Il lit les langages HTML et CSS pour savoir ce qu'il doit afficher. Chaque navigateur affiche un site web de manière légèrement différente des autres navigateurs. Nous utiliserons de préférence le navigateur Mozilla Firefox.

1. Principes de base du langage HTML

Pour créer une page web, il faut utiliser un éditeur de texte comme **notepad++** et créer un fichier texte qui aura pour extension .html. Exemple : `essai.html`.

Voilà le code HTML minimal d'une page web édité avec notepad++ :

```
<!DOCTYPE html >
<html>
  <head>
    <!-- En tête -->
    <meta charset = "utf-8" />
    <title > page minimale </title>
  </head>

  <body>
    <!-- Corps de la page -->
  </body>
</html>
```

On utilisera ce code minimal pour débiter l'édition d'une page web.

Le langage HTML (HyperText **Markup** Language) décrit le contenu de la page et utilise des **balises**.

Les balises peuvent avoir deux formes :

- `<balise> </balise>` : elles s'ouvrent et se ferment pour délimiter le contenu (début et fin d'un titre, par exemple).
- `<balise />` : balises orphelines (on ne les insère qu'en un seul exemplaire), elles permettent d'insérer un élément à un endroit précis (par exemple une image ou un saut de ligne).

Les balises sont parfois accompagnées d'**attributs** pour donner des indications supplémentaires. (Exemple : `src` est un attribut de la balise `img` : ``).

La syntaxe complète d'une balise avec un attribut est : `<balise attribut="valeur"> ... </balise>`

Il est possible d'utiliser plusieurs attributs, séparés par un espace, dans une même balise :

`<balise attribut1="valeur1" attribut2="valeur2"> ... </balise>`.

Les balises doivent être correctement imbriquées :

- `<a><c>contenu</c>` est correct.
- `<a><c>contenu</c>` est incorrect.

Une page web est constituée de deux sections principales : un en-tête (`<head>`) et un corps (`<body>`).

2. Le texte

La balise `<p>` permet de délimiter des **paragraphe** :

- `<p>` marque le début du paragraphe
- `</p>` marque la fin du paragraphe

Notez qu'après un paragraphe, le navigateur saute une ligne.

La balise orpheline `
` sert à indiquer qu'on doit **aller à la ligne**. Elle doit obligatoirement être mise à l'intérieur d'un paragraphe.

La balise `<h>` permet de délimiter des titres. On peut utiliser six niveaux de titres différents :

```

<html>
  <head>
    <meta charset ="utf-8" />
    <title>Niveaux de titres </title >
  </head>

  <body >
    <h1 >Titre très important de niveau 1</h1 >
    <h2 >Titre important de niveau 2</h2 >
    <h3 >Titre un peu moins important (sous - titre)</h3 >
    <h4 >Titre encore moins important </h4 >
    <h5 >Titre pas important </h5 >
    <h6 >Titre de niveau 6 </h6 >
  </body >
</html >

```

La balise **** permet de mettre un peu en **valeur** certains mots.

Pour mettre encore plus en valeur, on utilise la balise ****.

La balise **<mark>** permet de faire **ressortir visuellement** une portion de texte.

Une **liste non ordonnée** est délimitée par les balises **** et ****.

Une **liste ordonnée** est délimitée par les balises **** et ****.

A l'intérieur d'une liste, chaque **élément** est délimité par les balises **** et ****.

Les différentes balises présentées précédemment permettent de créer une page qui aura l'apparence suivante dans le navigateur :

Bonjour et bienvenue sur mon site !

Vous trouverez ci-dessous quelques **langages** que je vais étudier ainsi que la *démarche* utilisée pour créer cette page.

Dans quelques heures d'apprentissage, j' aurai appris un peu plus de choses et **mon site sera très présentable !**

Langages informatiques

- C++
- HTML
- CSS

Pour faire ma page web

1. J'ouvre notepad++.
2. Je tape mon code html.
3. J'enregistre.
4. J'ouvre avec mon navigateur.

Voici le code html correspondant à la page précédente :

```
<html >
  <head >
    <meta charset ="utf-8" />
    <title >Synthèse balises de texte </title >
  </head >
  <body >
    <p>Bonjour et bienvenue sur mon site !<br />
    Vous trouverez ci-dessous quelques <strong> langages </strong> que je vais étudier
    ainsi que la <em> démarche </em> utilisée pour créer cette page.</p>
    <p>Dans quelques heures d'apprentissage, j' aurai appris un peu plus de choses
    et <mark> mon site sera très présentable ! </mark>

    <h1 >Langages informatiques</h1 >
    <ul >
      <li >C++ </li >
      <li >HTML </li >
      <li >CSS </li >
    </ul >

    <h1 >Pour faire ma page web</h1 >
    <ol >
      <li >J'ouvre notepad++.</li >
      <li >Je tape mon code html.</li >
      <li >J'enregistre.</li >
      <li >J'ouvre avec mon navigateur.</li >
    </ol >
    <p>
  </body >
</html >
```

3. Les liens

Un site est généralement constitué de plusieurs pages et pour passer de l'une à l'autre, on utilise des liens. La balise permettant de faire des **liens** est la balise **<a>**. Il faut lui ajouter un attribut, **href**, pour indiquer vers quelle page le lien doit conduire.

On peut faire un **lien absolu** vers un autre site web en mettant son URL comme valeur de l'attribut href.

On peut faire un **lien relatif** vers une autre page du site située dans le même répertoire. Il suffit pour cela de donner comme valeur de l'attribut **href** le nom du fichier cible :

<p>Bonjour. Souhaitez-vous consulter la page 2 ?</p>

Si le fichier cible est dans un autre dossier, on indiquera le chemin pour y accéder :

<p>Bonjour. Souhaitez-vous consulter la page 2 ?</p>

Si le fichier cible est dossier parent, on écrira :

Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut **id** pour marquer un endroit dans la page, puis faire un lien vers l'ancre comme ceci :

.

Exemple de lien vers un titre :

On crée l'ancre en mettant l'attribut id dans la balise de titre h2 : <h2 id="mon_ancre">Titre </h2 >

On crée le lien en mettant le nom de l'ancre précédé d'un dièse en valeur de l'attribut href de la balise a :

Aller vers l'ancre

On peut aussi faire un lien vers une ancre d'une autre page :

Pour afficher une **infobulle** (texte descriptif) lorsqu'on survole le lien, on utilise l'**attribut title** :

<p>En cliquant ici, vous irez à la page2. </p>

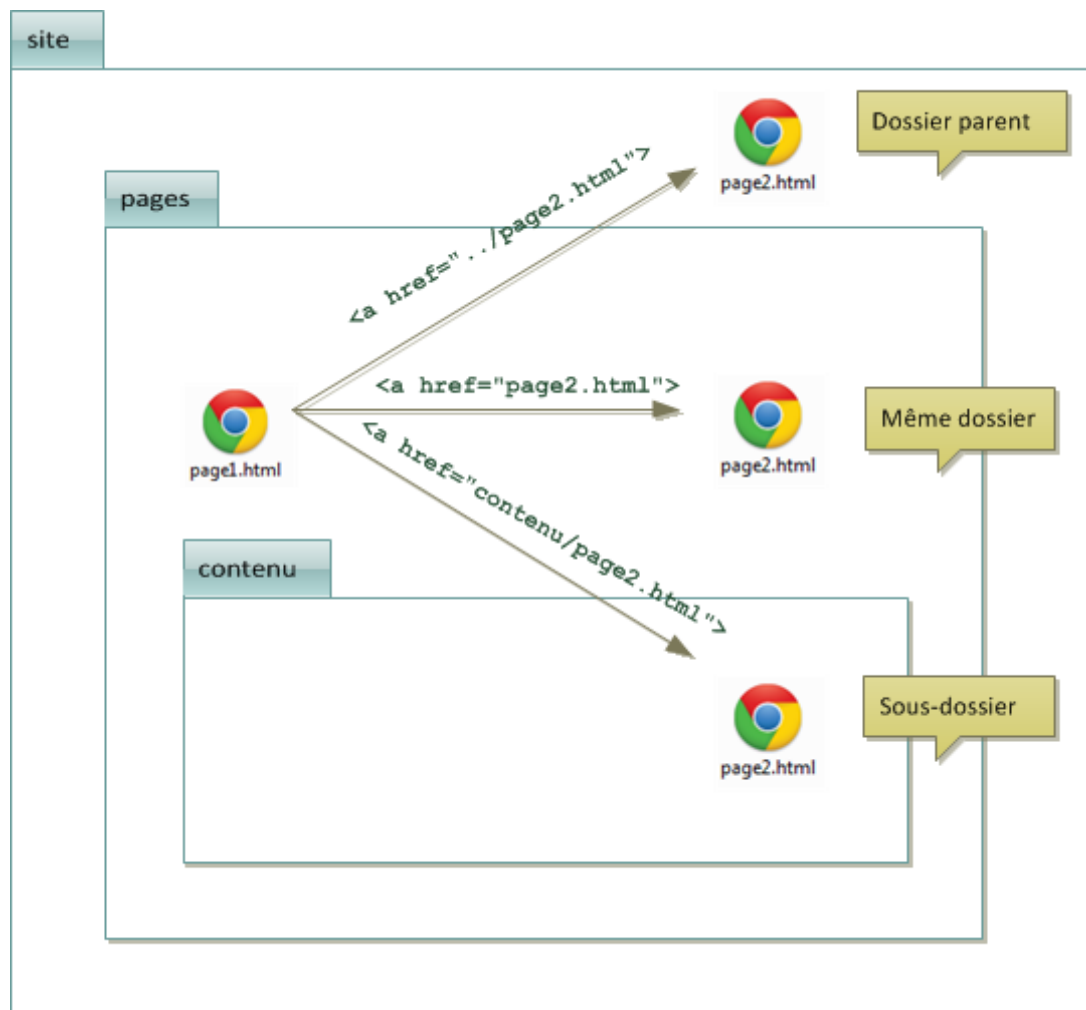
En mettant l'attribut target="_blank" à la balise <a>, on peut forcer l'ouverture d'un lien dans un **nouvel onglet**.

Pour faire un lien permettant de **télécharger un fichier**, il suffit de mettre le nom du fichier en valeur de l'attribut href. Le navigateur voyant que ce n'est pas un fichier html lancera la procédure de téléchargement :

`<p>Télécharger le fichier </p>`

Pour faire un lien permettant d'envoyer un e-mail :

`<p>Envoyez-moi un e-mail !</p>`



Les liens relatifs

4. Les images

Il existe plusieurs formats d'images adaptées au Web :

- JPEG : pour les photos ;
- PNG : pour toutes les autres illustrations ;
- GIF : similaire au PNG, plus limité en nombre de couleurs mais qui peut être animé.

On insère une image avec la balise orpheline ****. Elle doit obligatoirement comporter au moins ces deux attributs :

- **src** : on indique un chemin absolu ou relatif vers le fichier image.
- **alt** : on donne une courte description de l'image sous forme de texte de substitution qui s'affichera si l'image ne peut être téléchargée et qui peut également servir aux moteurs de recherche.

On peut également ajouter une infobulle avec l'attribut facultatif **title**.

Une balise `` peut être incluse dans une balise lien `<a>`.

Si une image illustre le texte (et n'est pas seulement décorative), il est conseillé de la placer au sein d'une balise **<figure>**. La balise **<figcaption>** placée à l'intérieur de la balise `<figure>` permet d'écrire la légende de l'image.

Exemple de code :

```
<!DOCTYPE html >
<html >
  <head>
    <meta charset ="utf-8" />
    <title >Images </title >
  </head >

  <body>
    <h1> Différents types d'images </h1>
    <h2> Images png </h2>
    <p> Ces images sont simples (logos, schémas, ...)</p>
    

    <h2> Images gif </h2>
    <p> Ces images peuvent êtres animées</p>
    

    <h2> Images jpeg </h2>
    <p> Ces images compressées sont utilisées pour avoir plus de qualité </p>
    Cliquez sur la vignette ci-contre si vous aimez les uns et les zéros <a href="images/univnum.jpg">
 </a></br>

    <p> La figure ci-dessous apporte une information car elle permet de savoir quels sont les
composants de la carte FPGA </p>
    <figure >
      
      <figcaption >La carte DE2 </figcaption >
    </figure >
  </body >
</html >
```

5. Les tableaux

Les **tableaux** sont nécessaires lorsqu'on veut structurer la présentation des informations.

Pour délimiter le début et la fin d'un tableau, on utilise respectivement les balises **<table>** et **</table>**.

- **<tr> </tr>** : indique le début et la fin d'une **ligne** du tableau ;
- **<td> </td>** : indique le début et la fin du contenu d'une **cellule** ;
- **<th> </th>** : indique le début et la fin d'une cellule d'en-tête.

Si le tableau est assez gros, on peut le découper en plusieurs parties.

Pour cela, il existe des balises HTML facultatives qui permettent de définir les trois zones du tableau :

- **l'en-tête** (en haut) : il se définit avec les balises **<thead></thead>** ;
- **le corps** (au centre) : il se définit avec les balises **<tbody></tbody>** ;
- **le pied** du tableau (en bas) : il se définit avec les balises **<tfoot></tfoot>**.

Pour fusionner des cellules **horizontalement**, on applique à la balise cellule, l'attribut **colspan** (fusion de colonnes). La valeur de cet attribut correspond au nombre de cellules à fusionner.

Pour fusionner des cellules **verticalement**, on applique à la balise cellule, l'attribut **rowspan** (fusion de lignes). La valeur de cet attribut correspond au nombre de cellules à fusionner.

L'exemple suivant utilise les différentes balises décrites précédemment.
Le langage HTML a permis de définir la structure et le contenu du tableau. Pour l'apparence (le style) de ce dernier, il faut utiliser le langage CSS.

```
<!DOCTYPE html >
<html >
  <head >
    <meta charset ="utf-8" />
    <title >Tableau classement ligue 1 </title >
  </head >

  <body >
    <table>
      <caption >Classement ligue 1 </caption >
      <thead>
        <tr >
          <th >Catégorie </th >
          <th >Rang </th >
          <th >Equipe </th >
          <th >Points </th >
        </tr >
      </thead>
      <tbody>
        <tr >
          <td rowspan="3">Européens </td >
          <td >1 </td >
          <td >A.S. Saint-Etienne </td >
          <td >84 </td >
        </tr >
        <tr >
          <td >2 </td >
          <td >F.C. Nantes </td >
          <td >76 </td >
        </tr >
        <tr >
          <td >3 </td >
          <td >Paris S.G. </td >
          <td >74 </td >
        </tr >
        <tr >
          <td rowspan="3">Relégués </td >
          <td >18 </td >
          <td >F.C. Lorient </td >
          <td >38 </td >
        </tr >
        <tr >
          <td >19 </td >
          <td >O. Marseille </td >
          <td >14 </td >
        </tr >
        <tr >
          <td >20 </td >
          <td >O. Lyonnais </td >
          <td >11 </td >
        </tr >
      </tbody>
    </table>
  </body >
</html >
```

Résultat affiché :

Classement ligue 1			
Catégorie	Rang	Equipe	Points
Européens	1	A.S. Saint-Etienne	84
	2	F.C. Nantes	76
	3	Paris S.G.	74
	18	F.C. Lorient	38
Relégués	19	O. Marseille	14
	20	O. Lyonnais	11

1. Appliquer un style

CSS (Cascading Style Sheets) est un langage qui vient compléter le HTML. Son rôle est de mettre en **forme** (style) la page web.

Quand un navigateur ne connaît pas une instruction de mise en forme, il l'ignore.

On peut écrire le code CSS à trois endroits différents :

- Le plus conseillé est de créer un fichier séparé portant l'extension .css (exemple : style.css). Ainsi, une modification du style dans le fichier css se répercutera dans toutes les pages du site. La liaison entre le fichier html et la feuille de style css se fait via une balise <link /> dans l'en-tête de la page html : `<link rel="stylesheet" type="text/css" href="style.css" />`
- Autre méthode moins conseillée : dans l'en tête (head), on peut décrire toute une série de spécifications de style entre <style> et </style>

Exemple :

```
<style>
p { color : green ; }
</style>
```

- Dernière méthode encore moins conseillée : via un attribut style directement dans les balises.

Exemple :

```
<p style="color:red; background-color:yellow">
```

2. Règles d'écriture

En CSS, on sélectionne quelles portions de la page HTML on veut modifier à l'aide de **sélecteurs** et on change leur présentation avec des **propriétés** CSS selon cette syntaxe :

```
sélecteur
{
    propriete1 : valeur1 ;
    propriete2 : valeur2 ;
}
```

Exemple :

```
h1, p
{
    color : blue ; /* les titres 1 et les paragraphes seront en bleu */
}
em
{
    color : red ; /* les textes importants seront en rouge */
}
```

Il existe de nombreuses façons de sélectionner (à l'aide de sélecteurs) la portion de la page que l'on veut mettre en forme. Par exemple, on peut viser :

- toutes les balises d'un même type, en écrivant simplement leur nom (h1 par exemple) ;
- certaines balises spécifiques, auxquelles on a donné des noms à l'aide des attributs de balise class ou id (.nomclasse ou #nomid) ; plusieurs balises peuvent avoir le même nom d'attribut class. Un nom d'id doit en revanche être unique dans la page HTML.
- * : toutes les balises sans exception (* sélecteur universel)
- A B : uniquement les balises B qui se trouvent à l'intérieur d'autres balises A (h3 em).
- A+B : une balise A qui en suit une autre B (h3 +p)
- A[attribut] : une balise A qui possède un attribut (a[title])
- A[attribut="Valeur"] : une balise, un attribut et une valeur exacte (a[title =" Cliquez ici "])
- A[attribut*="Valeur"] : une balise, un attribut et une valeur contenue dans l'attribut (a[title *= " ici "])
- Etc...(consulter le site du W3C pour une liste exhaustive de tous les sélecteurs existants)

Si l'on veut appliquer un style à un mot ou un groupe de mots qui ne sont pas entourés de balises, on les entoure par des balises universelles :

- : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement.

- `<div> </div>` : c'est une balise de type **block** (elle provoque obligatoirement un retour à la ligne), qui entoure un bloc de texte.

Exemple d'attribut classe (ici introduction) :

Fichier css :

```
h1,.introduction
{
    color : blue ; /* les titres de niveau 1 et les paragraphes de type introduction
seront en bleu */
}
em
{
    color : red ;
}
```

Fichier html :

```
<! DOCTYPE html >
<html >
  <head>
    <meta charset ="utf-8" />
    <link rel=" stylesheet " href =" style.css " />
    <title > Test du CSS </title >
  </head >

  <body>
    <h1 >Présentation du site </h1 >

    <p class ="introduction"> Bonjour et bienvenue sur mon site !</p>
    <p> C'est décidé, je commence mon site dédié au <em >numérique </em >.
    Vous allez voir le boulot !</p>
  </body >
</html >
```

Exemples :

➤ Attribut id :

```

```

dans le fichier html, à associer à :

```
# logo
{
  /* Indiquez les propriétés CSS ici */
}
```

dans le fichier css.

➤ Sélecteurs avancés

- `<h3>Titre avec texte important </h3>`

```
h3 em
{
}
```

- `<h3>Titre </h3>`
`<p>Paragraphe </p>`

```
h3 + p
{
}
```

- ``

```
a[ title ]
{
}
```

- ``

```
a[ title =" Cliquez ici "]
{
}
```

■ ``

```
a[ title *= " ici " ]
{
}
```

3. Formatage du texte

On modifie la **taille** du texte avec la propriété CSS **font-size**.

On peut indiquer la taille **absolue** en pixels (16px) : **font-size** : 16px ;

On peut aussi indiquer la taille **relative** en em (1 em = normal) , en pourcentage (110%), etc.

```
p
{
    font-size : 0.8em; /* taille des lettres de paragraphe à 80% de la taille
normale*/
}
h1
{
    font-size : 1.3em; /* taille des lettres de titre 1 à 130% de la taille normale*/
}
```

On change la **police** du texte avec **font-family**.

```
sélecteur
{
    font-family : police ;
}
```

Dans l'exemple suivant, on met plusieurs polices, la première disponible sera utilisée :

```
p
{
    font-family : Impact , " Arial Black " , Arial , Verdana , serif ;
}
```

Attention, seules quelques polices sont connues par tous les ordinateurs. On peut cependant utiliser une police personnalisée avec la directive **@font-face** : cela forcera les navigateurs à télécharger la police de notre choix.

www.fontsquirrel.com est un site qui permet de télécharger des polices libres de droit et surtout des packs prêts à l'emploi pour CSS 3 permettant au navigateur de choisir le format adéquat.

Exemple :

```
@font-face {
    font-family: myFirstFont;
    src: url(sansation_light.woff);
}

div {
    font-family: myFirstFont;
}
```

De nombreuses propriétés de mise en forme du texte existent :

- Avec la propriété **font-style** qui peut prendre les valeurs **italic**, **oblique** ou **normal**

```
em
{
    font-style : normal ; /* les textes importants ne seront pas en italique mais en
normal */
}
```

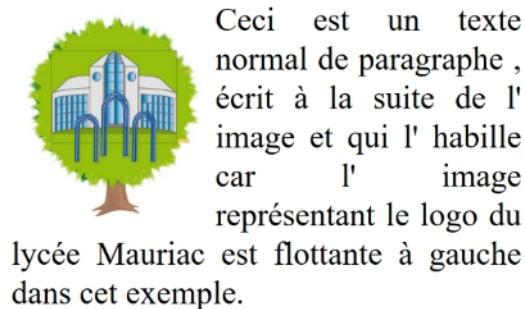
- Avec la propriété **font-weight** qui peut prendre les valeurs **bold** (texte en gras) ou **normal**

```
h1
{
    font-weight : bold ; /* les titres 1 seront en caractères gras */
}
```

- Avec la propriété **text-decoration** qui peut prendre les valeurs **underline** (souligné), **line-through** (barré), **overline** (ligne au-dessus), **blink** (clignotant), **none** (normal).

```
h1
{
  text-decoration : blink ; /* attention, ne fonctionne pas avec tous les navigateurs
*/
}
.souligne
{
  text-decoration : underline ;
}
```

- Avec la propriété **text-align** qui peut prendre les valeurs **left** (gauche), **center** (centré), **right** (droite), **justify** (justifié).
- Avec la propriété **float** qui peut prendre les valeurs **left** (gauche), **et right** (droite). Ceci permet par exemple de faire flotter une image à gauche ou à droite et le texte entoure cette image :



```
<body>
  <div>
    <p> Ceci est un texte normal de paragraphe , écrit à la suite de l' image
et qui l' habille car l' image représentant le logo du lycée Mauriac est flottante à
gauche dans cet exemple.</p>
  </div>
</body >
```

```
.imageflottante
{
  float : left;
}
div
{
  width : 250px;
  text-align : justify;
}
```

Pour stopper le **flottant**, on utilise la propriété **clear** qui peut prendre comme valeur **left**, **right** ou **both** (les deux)

4. La couleur et le fond

On a vu que la propriété **color** permet de régler la couleur d'un texte. Elle peut prendre les 16 valeurs standard suivantes : white, silver, grey, black, red, maroon, lime, green, yellow, olive, blue, navy, fushia, purple, aqua, teal.

On peut pour choisir parmi plus de 16 millions de couleurs utiliser la notation hexadécimale dans laquelle les 3 couleurs de base Rouge Vert Bleu sont chacune codée sur un octet.

```
color : # FFFFFFFF ; /* correspond au blanc et #000000 correspond au noir */
```

On peut également utiliser la notation RGB (Red Green Blue) :

```
color : rgb(240 ,96 ,204); /* avec des valeurs comprises entre 0 et 255! */
```

Pour trouver les valeurs, utilisez internet ou colorpicker ou Paint.

Pour modifier la couleur du fond de la page, il faut utiliser la propriété CSS nommée **background-color** et l'appliquer à la balise **<body>**.

Attention ! : Si l'on ne précise rien au niveau des balises contenues dans la balise body (h1, p par exemple), celles-ci (titres, paragraphes...) hériteront des propriétés CSS de la balise body qui les contient.

C'est la propriété de l'élément le plus précis qui a la priorité.

Cette notion d'héritage est importante en CSS.

On peut également appliquer la propriété **background-color** à des titres, paragraphes ou mots. Ils apparaîtront alors surlignés.

Pour appliquer une image de fond, il faut utiliser la propriété **background-image** :

```
body
{
    background-image : url("logoMauriac.jpg");
}
```

L'adresse indiquant où se trouve l'image de fond peut être écrite en absolu (http://...) ou en relatif (fond.png).

Attention, lorsqu'on écrit une adresse en relatif dans le fichier CSS, l'adresse de l'image doit être indiquée par rapport au fichier .css et non pas par rapport au fichier .html.

Pour fixer le fond, on utilise la propriété **background-attachment** qui a deux valeurs possibles :

- **fixed** : l'image de fond reste fixe ;
- **scroll** : l'image de fond défile avec le texte (par défaut).

Pour répéter le fond, on utilise la propriété **background-repeat** qui a quatre valeurs possibles :

- **no-repeat** : le fond ne sera pas répété. L'image sera donc unique sur la page.
- **repeat-x** : le fond sera répété uniquement sur la première ligne, horizontalement.
- **repeat-y** : le fond sera répété uniquement sur la première colonne, verticalement.
- **repeat** : le fond sera répété en mosaïque (par défaut).

Pour positionner le fond, on utilise la propriété **background-position** qui a cinq valeurs possibles :

- **top** : en haut
- **bottom** : en bas
- **left** : à gauche
- **center** : au centre
- **right** : à droite

ou bien une combinaison comme par exemple :

```
background-position : top right ;
```

On peut aussi donner deux valeurs en pixel par rapport au coin supérieur gauche :

```
background-position : 30px 50px ; /* fond placé à 30 pixels de la gauche et 50 du haut */
```

On peut combiner les différentes propriétés de fond : background-image, background-repeat, background-attachment et background-position en une seule "super propriété" **background** comme ceci:

```
body
{
background : url("logoMauriac.jpg") fixed no-repeat top right ; /* l'ordre et le nombre
n'ont pas d'importance*/
}
```

On peut rendre une portion de la page transparente avec la propriété **opacity** ou avec la notation **RGBA** (identique à la notation RGB, avec une quatrième valeur indiquant le niveau de transparence).

Exemple de code :

```
<! DOCTYPE html >
<html >
  <head>
    <meta charset ="utf-8" />
    <link rel="stylesheet" href ="style.css" />
    <title >Formatage CSS </title >
  </head >

  <body>
    <p class ="introduction"> Bonjour et bienvenue sur mon site !</p>
    <h1 >Définition de l'intelligence artificielle </h1 >
    <p></p>
    <p> <em>L'intelligence artificielle</em> est un champ extraordinairement
vaste, et donc difficile à définir. La définition la plus "simple" de l'IA qu'on peut
trouver dans un dictionnaire serait la suivante: <span
class="definition">"L'intelligence artificielle est la recherche de moyens susceptibles
```

```

de doter les systèmes informatiques de capacités intellectuelles comparables à celles
des êtres humains".</span></p>
    <p class ="dessous"> </br></p>
    
<p>Intéressant ce site!</p>
    </body >
</html >

```

```

body
{
    background-image:url("images/ciel.jpg");
    background-attachment : scroll ;
    background-repeat : repeat;
}
h1
{
    color : yellow ; /* les titres de niveau 1 seront en jaune... */
    border : 5px yellow outset; /*...avec bordure en relief...*/
    border-radius : 20px ; /*...et coins arrondis de rayon 20 pixels*/
    background-color:rgba(255 , 0, 0, 0.5);
}
.introduction
{
    color : white; /* les paragraphes de type introduction seront en vert */
    font-size : 20px;
    font-weight : bold;
    box-shadow : 6px 6px 6px white;
}
.definition
{
    font-style : italic;
}
h1
{
    font-family : Verdana, Impact , " Arial Black ", Arial , serif;
}
em
{
    color : red ;
}
p
{
    font-size : 1.8em;
    text-align : justify;
}
.imageBleue
{
    float : left ;
}
.imageSmiley
{
    float : left ;
    opacity : 0.2;
}
.dessous
{
    clear : both ;
    font-size : 2.2em;
    font-family : Verdana, Impact , " Arial Black ", Arial , serif ;
}

```

5. Bordure et ombres

On peut appliquer une bordure à un élément avec la super propriété **border** qui regroupe **border-width**, **border-color**, **border-style**. Il faut indiquer la largeur de la bordure, sa couleur et son type (trait continu, pointillés. . .).

```
h1
{
    border : 3px blue dashed ;
}
```

Valeurs disponibles pour le type de bordure:

- **none** : pas de bordure (par défaut) ;
- **solid** : un trait simple ;
- **dotted** : pointillés ;
- **dashed** : tirets ;
- **double** : bordure double ;
- **groove** : en relief ;
- **ridge** : autre effet relief ;
- **inset** : effet 3D global enfoncé ;
- **outset** : effet 3D global surélevé.

Si on ne veut travailler que sur une des 4 bordures, on peut préciser laquelle en utilisant à la place de **border** :

border-top, **border-bottom**, **border-right** ou **border-left**.

On peut arrondir les coins des bordures avec **border-radius**. Exemple :

```
p
{
    border-radius : 10px ; /* la taille de l'arrondi est de 10 pixels */
}
```

On peut préciser des arrondis différents pour les 4 coins : **border-radius** : 10px 5px 10px 5px;

On peut réaliser des arrondis elliptiques en donnant un couple de valeurs : **border-radius** : 20px / 10px ;

On peut ajouter une ombre aux blocs de texte avec **box-shadow**. On doit indiquer dans l'ordre : le décalage vertical et horizontal de l'ombre, le niveau d'adoucissement du dégradé et sa couleur.

```
p
{
    box-shadow : 6px 6px 6px black inset ;/* la cinquième valeur inset est facultative
et déplace l'ombre à l'intérieur */
}
```

Le texte peut lui aussi avoir une ombre avec **text-shadow** (même principe).

6. Création d'apparences dynamiques

En CSS, on peut modifier l'apparence de certaines sections dynamiquement, après le chargement de la page, lorsque certains événements se produisent. On utilise pour cela les **pseudo-formats**. Un pseudo format est une information que l'on rajoute après le nom de la balise (ou de la classe).

Le pseudo-format **:hover** permet de changer l'apparence au survol (par exemple : **a:hover** pour modifier l'apparence des liens lorsque la souris pointe dessus).

```
a /* Liens par défaut (non survolés) */
{
    text-decoration : none ;
    color : red;
}
a:hover /* Apparence au survol des liens */
{
    text-decoration : underline ;
    color : green ;
}
```

Le pseudo-format **:active** modifie l'apparence des liens au moment du clic.

Le pseudo-format **:focus** permet de modifier l'apparence d'un élément sélectionné.

On peut régler l'apparence d'un lien déjà visité avec le pseudo-format **:visited**. Par défaut, le navigateur colore le lien d'une page déjà visitée en violet. Ceci est pratique sur Google pour voir les sites déjà consultés. Certains sites préfèrent garder l'aspect initial du lien après la visite.

7. Formatage de tableau

Reprenons le tableau du cours HTML. Appliquons-lui la feuille CSS suivante :

```
td /* Toutes les cellules des tableaux ... */
{
    border : 1px solid black ; /* auront une bordure de 1px */
}
```

On obtient :

Catégorie	Rang	Equipe	Points
Européens	1	A.S. Saint-Etienne	84
	2	F.C. Nantes	76
	3	Paris S.G.	74
Relégués	18	F.C. Lorient	38
	19	O. Marseille	14
	20	O. Lyonnais	11

Il existe une propriété CSS spécifique aux tableaux, **border-collapse**, qui signifie coller les bordures entre elles. Cette propriété peut prendre deux valeurs :

- collapse : les bordures seront collées entre elles,
- separate : les bordures seront dissociées (valeur par défaut)

On peut changer la **position du titre** avec la propriété CSS **caption-side** qui peut prendre quatre valeurs : top (titre au-dessus , valeur par défaut) ; bottom (titre en dessous) ; left (titre à gauche) ; right (titre à droite) En appliquant la feuille de style ci-dessous, on donne au tableau une apparence plus agréable :

```
td , th /* Toutes les cellules des tableaux ... */
{
    border : 2px blue solid; /* auront une bordure de 1px */
}
table
{
    border-collapse : collapse ;
    font-family : Arial , Verdana , serif ;
    caption-side : bottom;
}
#verts
{
    background-color : rgb(0,220,0);
    color : white;
}
.point
{
    text-align : center;
}
.rang
{
    text-align : right;
}
```

Catégorie	Rang	Equipe	Points
Européens	1	A.S. Saint-Etienne	84
	2	F.C. Nantes	76
	3	Paris S.G.	74
Relégués	18	F.C. Lorient	38
	19	O. Marseille	14
	20	O. Lyonnais	11

Classement ligue 1

Pour ça, on a mis l'attribut id= "verts" à la balise cellule contenant notre équipe de cœur et modifié les couleurs de cette cellule dans le fichier css. Il a fallu mettre des attributs class="rang" pour les cellules de la colonne rang afin de justifier le texte à droite et des attributs class="point" pour les cellules de la colonne points afin de justifier au centre. Mettez un lien redirigeant vers le site de l'ASSE lorsqu' on clique dessus.

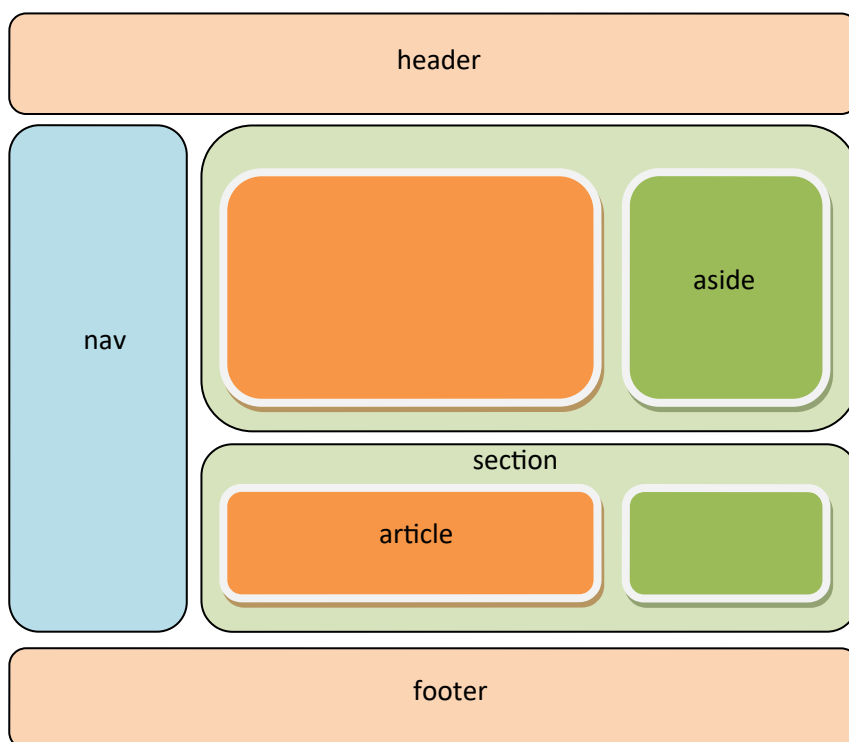
1. Structure d'une page web

HTML5 propose plusieurs balises pour délimiter les différentes zones qui constituent une page web :

- `<header>` : en-tête ;
- `<footer>` : pied de page ;
- `<nav>` : liens principaux de navigation ;
- `<section>` : section de page ;
- `<aside>` : informations complémentaires ;
- `<article>` : article indépendant.

Ces balises peuvent être imbriquées les unes dans les autres. Ainsi, une section peut avoir son propre en-tête. Ces balises ne s'occupent pas de la mise en page. Elles servent seulement à indiquer le sens du texte qu'elles contiennent.

Exemple de structure de page :



2. Le modèle des boîtes

En HTML, la plupart des balises peuvent se ranger dans l'une ou l'autre de ces deux catégories :

- Les balises **block** : Blocs avec retour à la ligne.
exemple : paragraphes `<p></p>`
- Les balises **inline** : Inclues de façon continue (en ligne) dans un bloc
Exemple : liens `<a>`.

```
<h1>Moteur de recherche </h1>
<p> Si vous voulez chercher sur internet, cliquez <a href="https://www.google.fr/">ici</a> et
puis c'est tout.</p><p>J'ai tout compris.</p>
```

Page affichée dans le navigateur :

Moteur de recherche

Si vous voulez chercher sur internet, cliquez [ici](https://www.google.fr/) et puis c'est tout.

J'ai tout compris.

Les deux balises universelles :

- `` (inline) ;
- `<div></div>` (block).

On peut modifier la taille d'une balise de type block avec les propriétés CSS **width** (largeur) et **height** (hauteur) en pixels (px) ou en pourcentage (%).

On peut définir des minima et maxima autorisés pour la largeur et la hauteur avec les propriétés :

- `min-width` : largeur minimale ;
- `min-height` : hauteur minimale ;
- `max-width` : largeur maximale ;
- `max-height` : hauteur maximale.

Exemple à tester :

```
p
{
  width : 50%;
  min-width : 400px ;
}
```

Tout bloc possède deux types de **marges** :

- intérieures
- extérieures

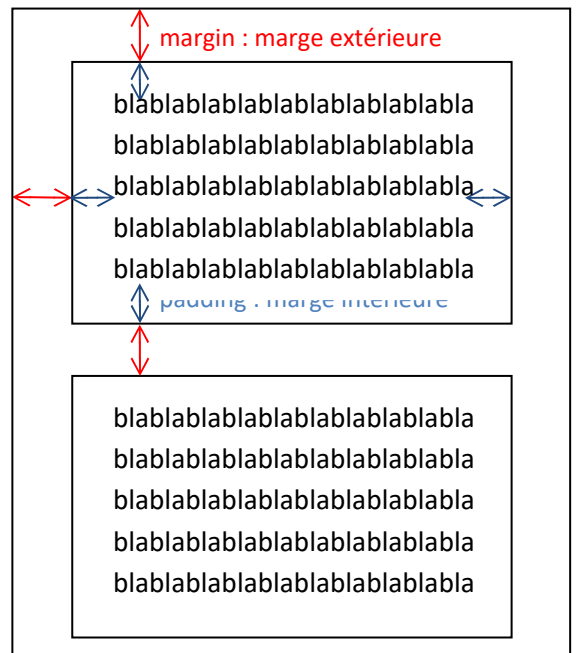
Les deux propriétés CSS suivantes permettent de modifier la taille des marges :

- **padding** (en px) pour la marge intérieure
- **margin** pour la marge extérieure

Si on veut différencier les marges, on peut utiliser :

- **margin-top** : marge extérieure en haut ;
- **margin-bottom** : marge extérieure en bas ;
- **margin-left** : marge extérieure à gauche ;
- **margin-right** : marge extérieure à droite.

Idem pour **padding**.



Pour **centrer** des blocs, 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;`

S'il y a trop de texte à l'intérieur d'un bloc de dimensions fixes, il y a un risque de débordement. Dans ce cas, il peut être judicieux de rajouter des barres de défilement avec la propriété **overflow** (valeur **auto**) ou de forcer la césure avec **word-wrap** (valeur **break-word**).

3. Positionnement en CSS

La mise en page d'un site web s'effectue en CSS. Voici plusieurs techniques de positionnement.

- Le positionnement flottant (avec la propriété **float**) permet par exemple de placer un menu à gauche ou à droite de la page. Néanmoins, cette propriété n'a pas été conçue pour cela et il est préférable d'éviter cette technique.
- 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.

Exemple de code CSS permettant de positionner côte à côte un menu et une section :

```

nav
{
display : inline-block ;
width : 150px ;
border : 1px solid black ;
vertical-align : top ;
}

section
{
display : inline-block ;
border : 1px solid blue ;
vertical-align : top ;
}

```

La propriété **vertical-align** peut prendre les valeurs suivantes :

- **baseline** : alignement de la base de l'élément avec celle de l'élément parent (par défaut) ;
- **top** : aligne en haut ;
- **middle** : centre verticalement ;
- **bottom** : aligne en bas ;
- (valeur en px ou %) : aligne à une certaine distance de la ligne de base (baseline).

- Le **positionnement absolu** permet de placer un élément où l'on souhaite sur la page, au pixel près.

Exemple

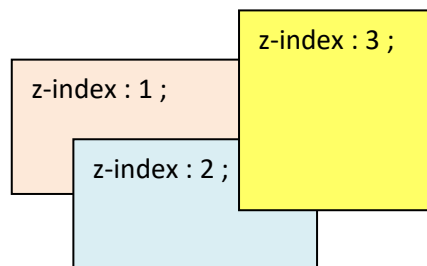
```

element
{
position : absolute ;
right : 0px;
bottom : 0px;
}

```

Il existe évidemment des propriétés **left** et **top**.

Si plusieurs éléments sont amenés à se chevaucher, on peut utiliser la propriété **z-index** comme ceci :



- Le **positionnement fixe** (position : fixed ;) est identique au positionnement absolu mais l'élément restera toujours visible même si on descend plus bas dans la page.
- Le **positionnement relatif** (position : relative ;) permet de décaler un bloc par rapport à sa position normale.

IMPORTANT : Un élément A positionné en absolu à l'intérieur d'un autre élément B positionné en relatif se positionnera par rapport à l'élément B, et non par rapport au coin en haut à gauche de la page.