

Le WEB (World Wide Web) est un **ensemble de documents** (ou pages) reliés entre eux par des **liens hypertexte**. Cela signifie que ces documents peuvent être parcourus dans n'importe quel ordre, en "sautant" d'un document à l'autre à l'aide des liens hypertextes.

## 1 HTML

Le langage **HTML** (HyperText Markup Language) permet de **décrire le contenu** et la **structure** d'une page WEB.

Pour coder en HTML il suffit d'un **éditeur de texte**, par exemple Visual Studio Code, Notepad++, Bracket... et d'un **navigateur** internet pour l'affichage, comme Mozilla, Chrome... Une connexion internet n'est pas nécessaire.

On utilise des **balises** pour décrire les différents éléments de la page.

Une qui ouvre `<p>` et l'autre qui ferme `</p>`.

*Exemple* : fichier1.html

- **Structure de base d'un document HTML :**

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <title>Bienvenue</title>
</head>
<body>
  ....
  ....
</body>
</html>
```

Dans la balise `<head>` on place les informations générales et dans la balise `<body>` ce qui s'affichera à l'écran.

*Exemple* : fichier1.html

- **Des balises de structure :**

`<h1> </h1>`, puis `<h2> </h2>`, `<h3> </h3>`, ..., `<h6> </h6>` pour les titres et sous-titres.

`<p> </p>` permet d'écrire des paragraphes ;

`<ul>` pour les listes ;

`<li>....</li>`

`<li>....</li>`

`</ul>`

`<div> </div>` une balise neutre sans rôle particulier.

- **Des balises de texte (elles modifient le texte) :**

`<strong> </strong>` pour écrire en gras ;

`<em> </em>` pour écrire en italique ;

`<u> </u>` pour souligner ;

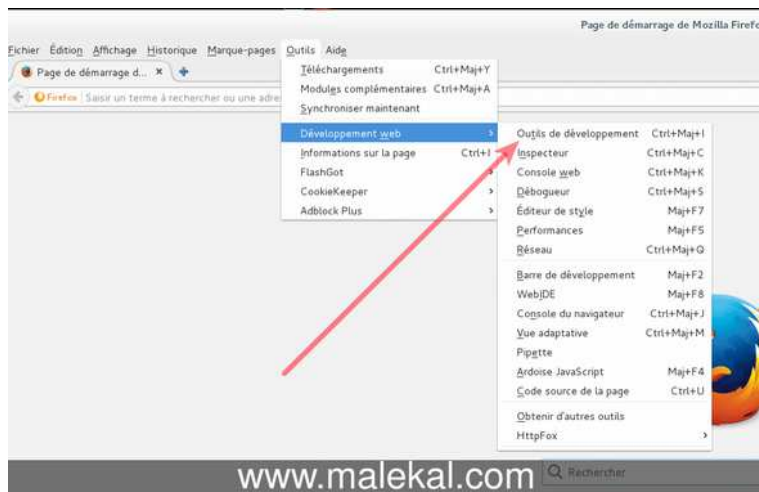
`<mark> </mark>` pour surligner

`<span> </span>` une balise neutre sans rôle particulier.

Exemple : fichier2.html

Remarques :

- On écrit des commentaires entre les balises `<!-- commentaires -->`
- L'**indentation** (décalage) est nécessaire pour une meilleur lisibilité du code. On décale de deux espaces.
- On ouvre sa page web en ouvrant le fichier avec un navigateur, par exemple Firefox.
- On peut examiner le code d'une page Web grâce au navigateur ; dans **Développement Web - Outil de développement** avec Firefox :



En utilisant l'**inspecteur** on peut suivre le code, voir directement sur la page ce qu'il représente et même le modifier en direct.

### Exercice 1 :

A vous d'essayer ! Commencez par ouvrir l'éditeur de texte **Visual Studio code**.

Recopiez la structure de base et sauvegardez le fichier dans un dossier WEB en le nommant **base.html**. Cela vous évitera de tout réécrire à chaque fois.

Créez un second fichier du nom de exercice1.html :

Placez un titre dans votre page, un ou deux paragraphes, une liste...

Testez votre code en exécutant le fichier (choisissez de l'ouvrir avec Mozilla).

### - Insérer des images :

Dans un paragraphe :

`<p>`

``

`</p>`

ou bien dans une figure :

`<figure>`

``

`<figcaption>légende de l'illustration</figcaption>`

`</figure>`

Remarques :

- Les images doivent être dans le **même dossier que le fichier html** ou bien vous devez indiquer le **chemin d'accès** à l'image ;

- L'attribut **alt** est un texte qui s'affiche sur la page web lorsque la photo ne peut s'afficher correctement ;
- Les noms des fichiers photos ne doivent pas être en majuscule, ni comporter d'espace ;
- Vous trouverez des images libres de droits sur le site *pixabay.com* ;
- On peut redimensionner une image sur le site *resizeimage.net*.

#### - Du son et de la vidéo :

- On peut ajouter de la musique sur la page :

```
<audio src="nom-fichier.mp3" controls>
```

#### Remarques :

- L'attribut **controls** fait apparaître sur la page une barre permettant de déclencher la lecture du fichier.
- On peut aussi mettre l'attribut **autoplay** à la place de controls qui déclenche la lecture du fichier audio dès que la page ouvre.

- On peut ajouter des vidéos sur la page :

```
<video width="320" height="240" poster="image.jpg" controls>  
<source src="nom-fichier.mp4" type="video/mp4">  
</video>
```

#### Remarques :

- On peut ajouter les attributs **width="200"** et **height="120"** pour choisir la taille de la fenêtre vidéo en pixels ; ainsi que **poster="nom-image"** qui met une image dans la fenêtre avant le début de la vidéo.
- Il existe différents formats vidéo qui ne sont pas tous pris en compte par les différents navigateurs. En cas de besoin on peut convertir les fichiers vidéo sur **microvideoconverter.com**.
- Vous devez utiliser des fichiers **libres de droits** pour pouvoir les afficher sur votre page sous peine que l'on vous demande de les retirer...

*Exemple :* fichier3.html

#### Exercice 2 :

Créez une nouvelle page Web contenant une photo, une musique et une vidéo.

#### - Créer des liens hypertextes dans la page ou vers d'autres pages :

- *Un lien vers une autre page Web :*

```
<a href="https :adresse de la page à visiter">Nom du lien</a>
```

#### Remarque :

Pour que la vidéo s'ouvre dans un nouvel onglet, on ajoute l'attribut **target="\_blank"**.

- *Se déplacer dans la page :*

On place le lien (en bas de page par exemple) :

```
<a href="#mot clé">nom du lien</a>
```

puis l'ancre **id** à l'endroit où l'on veut se rendre dans la page (en haut de la page par exemple) :

```
<a id="mot clé"></a>
```

- *Se déplacer d'une page à l'autre d'un site :*

```
<a href="page2.html">une autre page du site</a>
```

*Exemple* : fichier4.html

### Exercice 3 :

Créez une nouvelle page Web permettant d'accéder à une page d'Internet, mais aussi à la page de l'exercice 2.

### Remarque :

Pour chaque balise d'une page HTML, il est possible d'ajouter deux *attributs* **id** et **class**.

L'attribut **id** représente un identifiant et doit être unique dans le fichier, alors que l'attribut **class** permet de créer un groupe de balises reconnu par un nom.

```
<h1 id="titre1">Titre 1</h1>
<li class="first">Premier point</li>
```

*Exemple* : fichier5.html

### Remarques :

- Ces ajouts ne modifient pas la page HTML qui s'affiche.
- Le **W3C** (World Wide Web Consortium) est une communauté internationale qui fabrique les standards du web (les règles communes de programmation). Sur leur site on trouve les références en HTML 5 :

<https://www.w3.org/TR/html5>

Il propose également un validateur en ligne qui permet de vérifier si un fichier HTML est conforme au standard :

<https://validator.w3.org/>

### Exercice 4 :

Reprenez vos fichiers précédents et testez-les pour voir leur conformité ; corrigez-les si nécessaire.

## 2 CSS

Le langage **CSS** (Cascading style Sheets) permet de définir les **propriétés graphiques** des éléments HTML constituant une page Web.

*Exemple* : fichier6.html

### **Un fichier CSS :**

Nous allons créer un nouveau fichier pour la mise en forme de la page web, et nous ferons le lien en insérant le code suivant dans le fichier HTML, avec les instructions générales dans **<head>** :

```
<link rel="stylesheet" href="nom-du-fichier.css"/>
```

- voici des exemples de syntaxe en CSS :

```
h1{
    color : red ;
    font-size : 50px ;
}
```

Cela permet de mettre les titres h1 en rouge et d'ajuster leur taille.

Chaque règle de mise en forme est composée d'un **sélecteur** (ex : h1) et d'un **bloc** entre accolades, qui contient des **déclarations** séparées par des points-virgule.

Chaque déclaration est composée d'une **propriété** (ex : color) et d'une **valeur** (ex : red).

### Exercice 5 :

Créez un fichier CSS et faire le lien avec une de vos pages HTML.

Changez la couleur et la taille des textes de votre page et observez les changements en actualisant la page sur le navigateur.

### On peut conjuguer les sélecteurs :

- On peut choisir de travailler sur tous les sélecteurs de la page :

```
*{  
}
```

- On peut en affecter plusieurs à la fois en utilisant une virgule (ici tous les titres h1 et h2) :

```
h1,h2{  
}
```

- On peut n'affecter que le paragraphe qui suit les titres h1 :

```
h1+p{  
    action  
}
```

- On peut également choisir de travailler sur un seul élément parmi un certain type de balise en utilisant les balises avec un identifiant (**id**) ou une classe (**class**).

Par identifiant :

```
#nom{  
}
```

Par classe :

```
.nom{  
}
```

### Remarques :

- On peut écrire des commentaires dans le programme en CSS, avec la syntaxe `/* commentaires */` ;
- Si on applique un style à une balise, toutes les balises qui se trouvent à l'intérieur prendront le même.

### - Diverses propriétés :

**font-weight : bold** ; permet d'écriture le texte en gras ;

**font-family : Arial, Verdana** ; permet de choisir la police de caractère.

### Remarques :

Les polices les plus reconnues par les navigateurs sont Arial, Verdana, Trebuchet MS.

- On en place plusieurs au cas où la première ne serait pas reconnue.
- On peut aller plus loin et en télécharger d'autres, par exemple sur le site **fontquirrel.com**.

**text-decoration :underline** ; permet de souligner

**text-align :center** ; permet de centrer le texte

**background-color :lightyellow** ; met le fond de la page en jaune clair

**color :blue** ; met le texte en bleu

**background-image :url("nom-image.jpg")** ; mettre une image en fond

### Remarques :

- On peut ajouter des options comme la position de l'image, si on la répète ou non.... avec **background-position**, **background-repeat**.

Par exemple :

**background-repeat :no-repeat ;** ne pas répéter l'image

**background-position :center ;** placer l'image au centre du bloc

- On peut écrire tout cela avec l'unique commande :

**background :url("nom-image.jpg") no-repeat center ;** (l'ordre n'a pas d'importance)

**border :8px black solid ;** permet d'encadrer en donnant une largeur pour le trait, sa couleur (noire) et en traits pleins

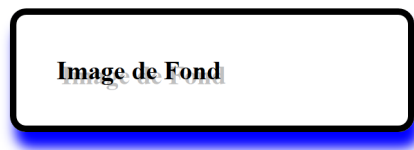
### Remarques :

- On peut ne faire un trait que sur un seul côté avec **border-left ;**

- On peut arrondir les angles avec **border-radius : "20" ;**

- On peut créer une ombre bleue sous le cadre avec **box-shadow :6px 20px 15px bleu ;**

**text-shadow :6px 6px 6px silver ;** on fait apparaître une ombre argentée sous le texte



**list-style-type :none** on enlève les puces de nos listes

### Remarque :

On peut remplacer les puces des listes par des images.

Exemple : **list-style-image :url("nom-fichier.jpg") ;**

**p :hover{**

**font-style :italic ;}** en passant la souris sur les paragraphes ceux-ci s'écrivent en italique.

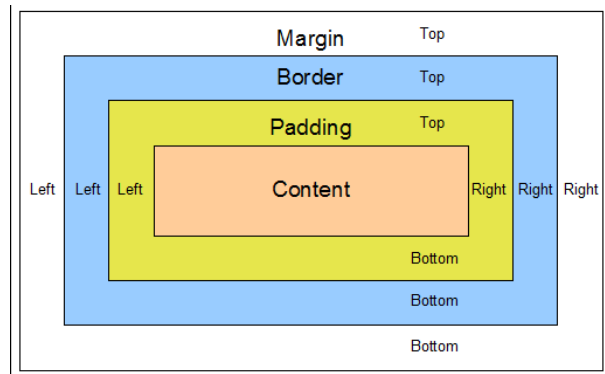
### Remarque :

La commande **hover** fait changer un élément de la page lorsque la souris passe dessus.

**a :visited{**

**color :red}** après qu'un lien ait été sélectionné il devient rouge

Tout élément HTML possède une boîte rectangulaire qui délimite son contenu :



- On peut donner aux divers **blocs** des dimensions, les agencer avec des marges, gérer leur contenu...

**width : "200" ;** on choisit la largeur du bloc

**height : "50" ;** on choisit la hauteur du bloc

**margin : "50" ;** on place une marge entre le bloc et ce qui est à l'extérieur

**padding : "12" ;** on place une marge intérieure entre le bord (border) du bloc et son contenu (content)

#### Remarque :

- on peut spécifier si ces distances s'appliquent au haut (top), au bas (bottom), à la gauche (left) ou à la droite (right) du bloc.

**margin : 20px 5px 10px 1px ;** permet d'indiquer les différentes marges dans l'ordre suivant : haut droite bas gauche

**margin : auto ;** centrage horizontal automatique, suivant la taille de la page

#### Exercice 6 :

Testez les différentes propriétés proposées : effectuez l'encadrement d'un élément de votre page ; placez une image dans le fond de la page ; jouez sur les marges pour centrer les éléments...

#### **Les balises de structure dans la page HTML :**

Voici de nouvelles balises à utiliser dans votre code HTML ; il faut prendre l'habitude de les utiliser pour encadrer les différentes parties de vos pages car elles permettent une meilleure mise en forme en CSS.

**<header> </header>** pour l'en-tête ;

**<section> </section>** pour une partie de page ;

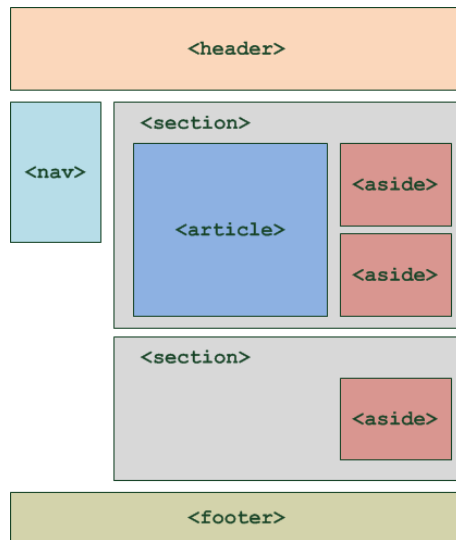
**<nav> </nav>** pour un menu de navigation ;

**<article> </article>** pour un article dans la page ;

**<aside> </aside>** pour un complément d'information ;

**<footer> </footer>** pour le pied de page.

Voici un exemple de structure de page que l'on pourra obtenir :



La propriété CSS **display** est la plus importante pour contrôler la mise en page.

Les valeurs par défaut de la plupart des éléments sont **block** ou **inline**.

**display :inline ;** place les éléments en ligne

#### Exercice 7 :

Ajoutez les balises de structures proposées dans votre code HTML, puis utilisez-les dans le style CSS pour essayer de mettre en forme la structure de votre page.

#### **Utiliser Flexbox :**

Avec **flexbox** on va organiser les blocs les uns par rapport aux autres.

On place tous les blocs dans un conteneur général dans le code HTML.

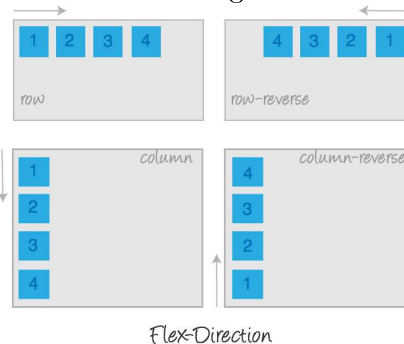
Dans le code CSS on peut alors agencer les différents blocs du conteneur les uns par rapport aux autres :

```
conteneur{
  display : flex;
}
```

#### Remarque :

Par défaut les blocs seront placés en colonne ; pour changer cela on utilise la propriété :

**flex-direction :row ;** les blocs sont alignés

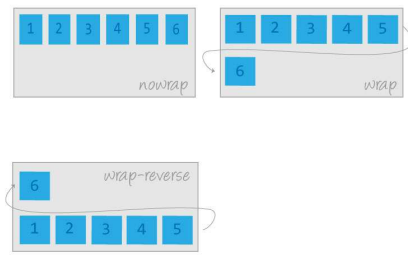


#### Remarque :

Les blocs se resserrent autant qu'ils le peuvent.

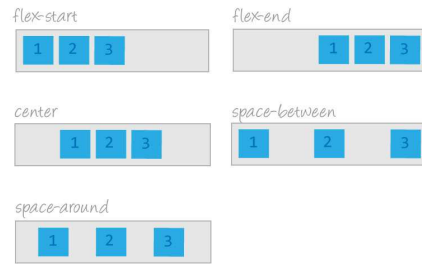
**flex-wrap :wrap ;** les blocs sont renvoyés à la ligne quand il n'y a plus de place





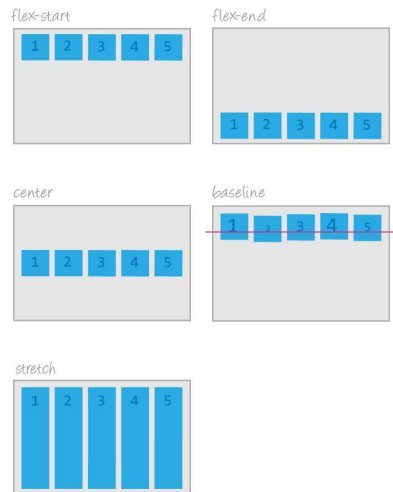
*FlexWrap*

**justify-content :flex-start;** les blocs sont alignés au début



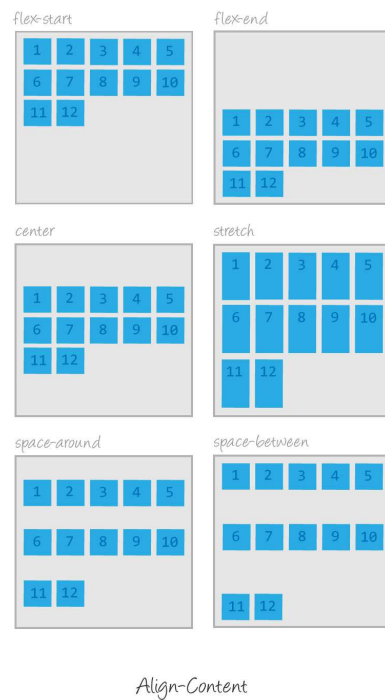
*Justify-Content*

**align-items :stretch;** les blocs vont s'étirer pour prendre toute la place disponible



*Align-Items*

**align-content :space-around;** s'il y a plusieurs lignes de blocs, on les espaces régulièrement



### Exercice 8 :

Testez l'utilisation de flexbox.

-----

Quelques liens pour approfondir :

[https ://developer.mozilla.org/fr/](https://developer.mozilla.org/fr/)

[http ://www.w3schools.com/](http://www.w3schools.com/)