

# Créer un site web avec HTML5 et CSS3





# Table des matières

- ▶ **Partie 1 - Découvrir les bases de HTML5**
  - ▶ **Le fonctionnement des sites web**
  - ▶ **Créer votre première page web en HTML**
  - ▶ **Organiser votre texte avec des balises.**
  - ▶ **Créer des liens absolus, relatifs, vers des ancres**
  - ▶ **Insérer des images**



- **Partie 2 - Mettre en forme vos pages avec CSS**

- **Mettre en place la feuille de style**
- **Formater du texte**
- **Ajouter de la couleur et un fond**
- **Créer des bordures et des ombres**
- **Créer des apparences dynamiques**



- **Partie 3 - Mettre en page votre site**

- **Structurer votre page**

- **Le modèle des boîtes**

- **Faire une mise en page avec Flexbox**

- **Découvrir d'autres techniques de mise en page : flottants.**

- **TP : Créer un site à partir d'une maquette.**



## ➤ **Partie 4 - Découvrir des fonctionnalités évoluées**

- **Ajouter des tableaux**
- **Créer des formulaires**
- **Insérer de la vidéo et de l'audio**
- **Utiliser le responsive design avec les Media Queries**
- **TP : Adapter son site en responsive**



# Partie 1 : Découvrir les bases du HTML

- ▶ Nous découvrirons dans ce cours les langages **HTML5** et **CSS3** que l'on utilise pour concevoir des sites web.
- ▶ Ils permettent notamment :
  - Insérer du texte, des images et des vidéos
  - Faire des liens entre vos pages
  - Mettre en forme votre site en modifiant la couleur, le fond, les types de police et leur taille...
  - Positionner les éléments du site comme bon vous semble : en-tête, menus, bas de page...

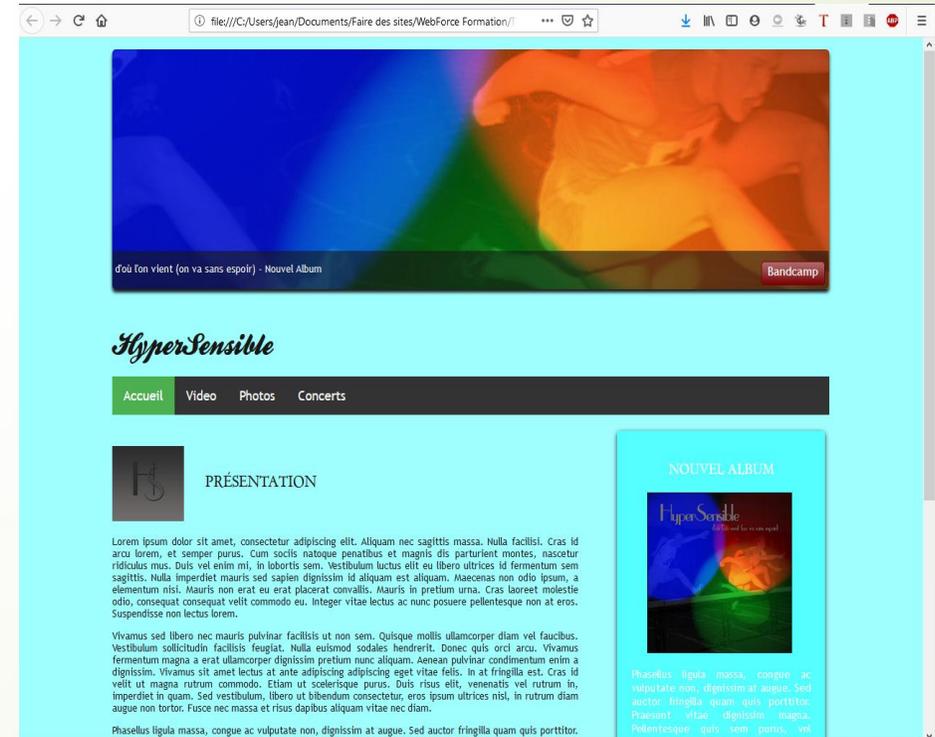
# Le fonctionnement des sites WEB

## ➤ Le Navigateur

Pour consulter une site web, on lance un programme appelé **navigateur**.

Le navigateur traduit le code HTML/CSS et affiche le site.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <link rel="stylesheet" href="style.css" />
6 <!--[if lt IE 9]>
7 <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
8 <![endif-->
9 <title>HyperSensible</title>
10 </head>
11 <!--[if IE 6]><body class="ie6 old_ie"><![endif-->
12 <!--[if IE 7]><body class="ie7 old_ie"><![endif-->
13 <!--[if IE 8]><body class="ie8"><![endif-->
14 <!--[if IE 9]><body class="ie9"><![endif-->
15 <!--[if !IE]><!--><body<!--<![endif-->
16 <div id="bloc_page">
17
18
19 <div id="banniere_image">
20 <div id="banniere_description">
21 d'ou l'on vient (on va sans espoir) - Nouvel Album
22 <a href="#" class="bouton_rouge">Bandcamp</a>
23 </div>
24 </div>
25
26 <header>
27 <div id="titre_principal">
28 <h1>HyperSensible</h1>
29 </div>
30
31 <div class="topnav">
32 <a class="active" href="index.html">Accueil</a>
33 <a href="video.html">Video</a>
34 <a href="photos.html">Photos</a>
35 <a href="concerts.html">Concerts</a>
36 </div>
37 </header>
38
39 <section>
40 <article>
41 <h2>Présentation</h2>
42 <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aliquam nec sagittis massa. Nulla facilisi. Cras id arcu lorem, et se
43 <p>Vivamus sed libero nec mauris pulvinar facilisis ut non sem. Quisque mollis ullamcorper diam vel faucibus. Vestibulum sollicit
44 <p>Phasellus ligula massa, congue ac vulputate non, dignissim at augue. Sed auctor fringilla quam quis porttitor. Praesent vitae
45 <p><a href="#">Télécharger dossier de presse</a></p>
46 </article>
47 <aside>
48 <h3>Nouvel album</h3>
49 <p><a href="#"></a></p>
50 <p>Phasellus ligula massa, congue ac vulputate non, dignissim at augue. Sed auctor fringilla quam quis porttitor. Praesent vitae
```



# Le fonctionnement des sites WEB

## ► Différents navigateurs

Il existe différents navigateurs qui n'affichent pas tous exactement le même site de la même façon car ils ne connaissent pas toujours les dernières fonctionnalités HTML et CSS. Il faudra donc vérifier que votre site fonctionne correctement sur la plus part des navigateurs.



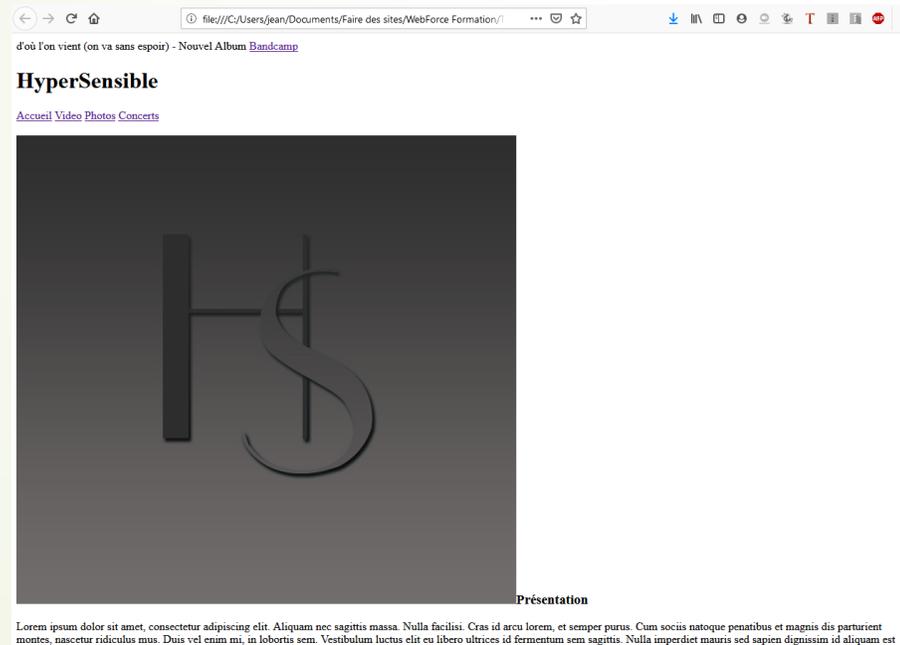


# Le fonctionnement des sites WEB

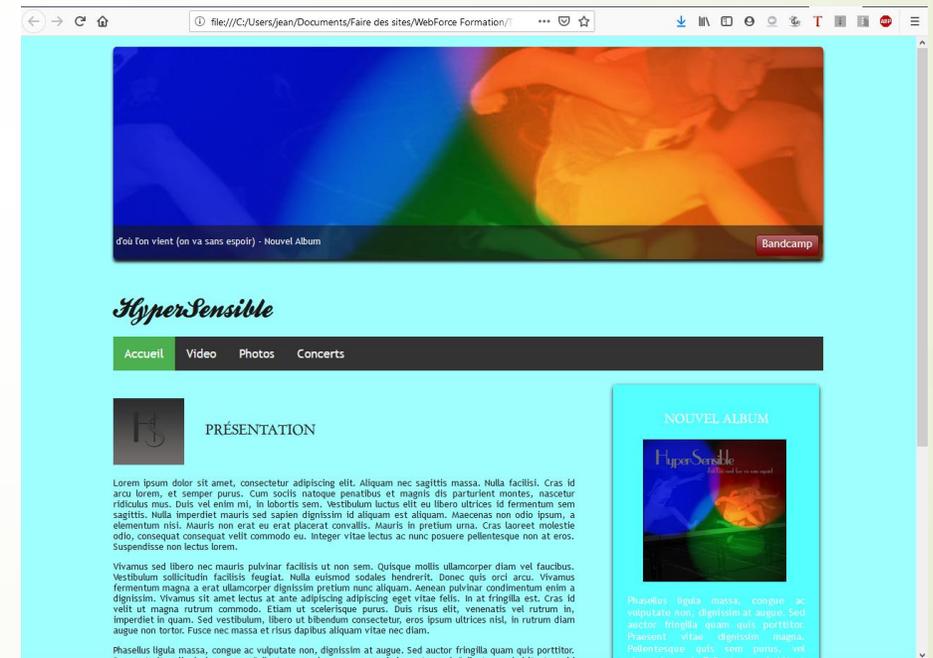
- Rôles du HTML et du CSS
- **HTML** (*HyperText Markup Language*) :
  - Il a fait son apparition dès 1991 lors du lancement du Web.
  - Son rôle est de gérer et organiser le contenu.
  - C'est donc en HTML que vous écrirez ce qui doit être affiché sur une page web : du texte, des liens, des images etc..
- **CSS** (*Cascading Style Sheets*, aussi appelées *Feuilles de style*) :
  - Son rôle est de gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...).
  - Ce langage est venu compléter le HTML en 1996.

# Le fonctionnement des sites WEB

## HTML sans CSS



## HTML avec CSS





# Le fonctionnement des sites WEB

➤ Les différentes versions du HTML et du CSS

➤ **HTML**

- **HTML 1** : Première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : Deuxième version du HTML apparue en 1994 et prend fin en 1996 avec l'apparition du HTML 3.0. C'est cette version qui posera en fait les bases des versions suivantes du HTML. Les règles et le fonctionnement de cette version sont donnés par le W3C (tandis que la première version avait été créée par un seul homme).
- **HTML 3** : apparue en 1996, cette nouvelle version du HTML rajoute de nombreuses possibilités au langage comme les tableaux, le positionnement du texte autour des images, etc.



# Le fonctionnement des sites WEB

## ➤ HTML

- **HTML 4** : cette version aura été utilisée un long moment durant les années 2000. Elle apparaît pour la première fois en 1998 et propose l'utilisation de frames (qui découpent une page web en plusieurs parties), des tableaux plus complexes, des améliorations sur les formulaires, etc. Mais surtout, cette version permet pour la première fois d'exploiter des feuilles de style, le CSS !
- **HTML 5** : c'est la dernière version. De plus en plus répandue, elle apporte de nombreuses améliorations comme la possibilité d'inclure facilement des vidéos, un meilleur agencement du contenu, de nouvelles fonctionnalités pour les formulaires, etc.



# Le fonctionnement des sites WEB

## ➤ CSS

- **CSS 1** : dès 1996, on dispose de la première version du CSS. Elle pose les bases de ce langage qui permet de présenter sa page web, comme les couleurs, les marges, les polices de caractères, etc.
- **CSS 2** : apparue en 1999 puis complétée par CSS 2.1, cette nouvelle version de CSS rajoute de nombreuses options. On peut désormais utiliser des techniques de positionnement très précises, qui nous permettent d'afficher des éléments où on le souhaite sur la page.
- **CSS 3** : c'est la dernière version, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, les animations etc.



# Le fonctionnement des sites WEB

- C'est quoi le W3C et le WHATWG ?
- Le W3C (World Wide Web Consortium) est l'organisation qui s'occupe de la standardisation du web. Elle réfléchit à l'évolution des standards tels que l'HTML et le CSS. Discute des bonnes pratiques à employer pour écrire son code HTML, ou encore de nouvelles balises qu'il serait intéressant d'ajouter.
- Le WHATWG (Web Hypertext Application Technology Working Group) est un second groupe au sein du W3C. Ce groupe est constitué principalement de développeurs des navigateurs tels que Mozilla, Opera ou Apple. L'approche est ici totalement différente puisque ce groupe est beaucoup plus ouvert et surtout a pour objectif d'accélérer la standardisation (ou du moins la mise en place de standards pour les navigateurs).



# Le Fonctionnement des sites WEB

- ▶ Editeurs de texte : Logiciels dédiés à la création de sites web. On peut les classer en deux catégories :
- ▶ **Les WYSIWYG** (*What You See Is What You Get // Ce Que Vous Voyez Est Ce Que Vous Obtenez*) : ce sont des programmes qui se veulent très faciles d'emploi, ils permettent de créer des sites web sans apprendre de langage particulier. Parmi les plus connus : Mozilla Composer, Microsoft Expression Web, Dreamweaver et Word ! Leur principal défaut est la qualité souvent assez mauvaise du code HTML et CSS qui est automatiquement généré. Et comme plus on a de maîtrise sur les moyens, plus on a de liberté sur le but, il nous faudra connaître les langages HTML et CSS, plutôt que de recourir à l'usage de ces outils.
- ▶ **Les éditeurs de texte** : ce sont des programmes dédiés à l'écriture de code. On peut en général les utiliser pour d'autres langages que le HTML et le CSS tel que Javascript, PHP etc. Ils sont les outils indispensables pour la création de sites web.



# Créer votre première page web en html.

- Choisir et Télécharger son éditeur de texte. On passe à la pratique !
  - Sous Windows : Sublime Text, Notepad++...
  - Sous Mac : Sublime Text, TextWrangler...
  - Sous Linux : Sublime Text...



# Créer votre première page web en html

- Les **balises** sont la base du langage html.
- Elles ne s'affichent pas dans la page mais donne une instruction au navigateur : « ceci est un titre », « ceci est un paragraphe », « ceci est une image » etc.
- Il existe deux types de balises :

Les balises qui marchent par paires : `<balise>Texte du site</balise>`

Les balises orphelines : `<br/>`



# Créer votre première page web en html

- Les **attributs** complètent les balises en donnant des informations supplémentaires :
- `<balise attribut="valeur">Texte du site</balise>`
- `<image nom="photo.jpg" />`



# Créer votre première page web en html

## Structure de base d'une page HTML5 :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre du site</title>
  </head>
  <body>

  </body>
</html>
```



# Créer votre première page web en html

```
<!-- Insérer des commentaires-->
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <!-- En-tête de la page -->
```

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

```
    <title>Titre</title>
```

```
  </head>
```

```
  <body>
```

```
    <!-- Corps de la page -->
```

```
  </body>
```

```
</html>
```

# Organiser son texte à l'aide des balises

## ► Les paragraphes

`<p>Bonjour et bienvenue sur mon site !</p>`

## ► Aller à la ligne dans un paragraphe

`<p>Bonjour et bienvenue sur mon site !<br/>Bonne visite</p>`

## ► Mise en valeur du texte

Section de texte importante : `<strong>Mon texte</strong>`

Section de texte un peu moins importante : `<em>Mon texte</em>`

Attention à ne pas confondre fond (HTML) et forme (CSS). Ne pas pervertir la destination de ces balises, qui est de souligner l'importance d'un texte pour les moteurs de recherche, en les utilisant pour mettre en gras ou italique. Ce qui doit être fait en CSS.



# Organiser son texte à l'aide des balises

## ► Les titres

`<h1>Titre</h1>` : signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.

`<h2>Titre</h2>` : signifie « titre important ».

`<h3>Titre</h3>` : titre un peu moins important.

`<h4>Titre</h4>` : titre encore moins important.

`<h5>Titre</h5>` : titre pas important.

`<h6>Titre</h6>` : titre pas important du tout.

# Organiser son texte à l'aide des balises

## ► Liste à puces

<ul>

<li>Premier élément de la liste</li>

<li> Second élément de la liste</li>

<li> Troisième élément de la liste</li>

</ul>

## ► Liste ordonnée

<ol>

<li>Premier élément de la liste</li>

<li> Second élément de la liste</li>

<li> Troisième élément de la liste</li>

</ol>

# Organiser son texte à l'aide des balises

## ► Lien vers un autre site (lien absolu)

```
<a href="https://adresse.com">Lien vers le site internet</a>
```

Exemple d'usage dans un paragraphe :

```
<p>Visitez le site <a href="https://adresse.com">Nom du site</a> pour plus d'informations.</p>
```

## ► Lien vers une page de votre site (lien relatif)

Si la page vers laquelle pointe le lien est dans le même dossier :

```
<a href="page2.html">Page2</a>
```

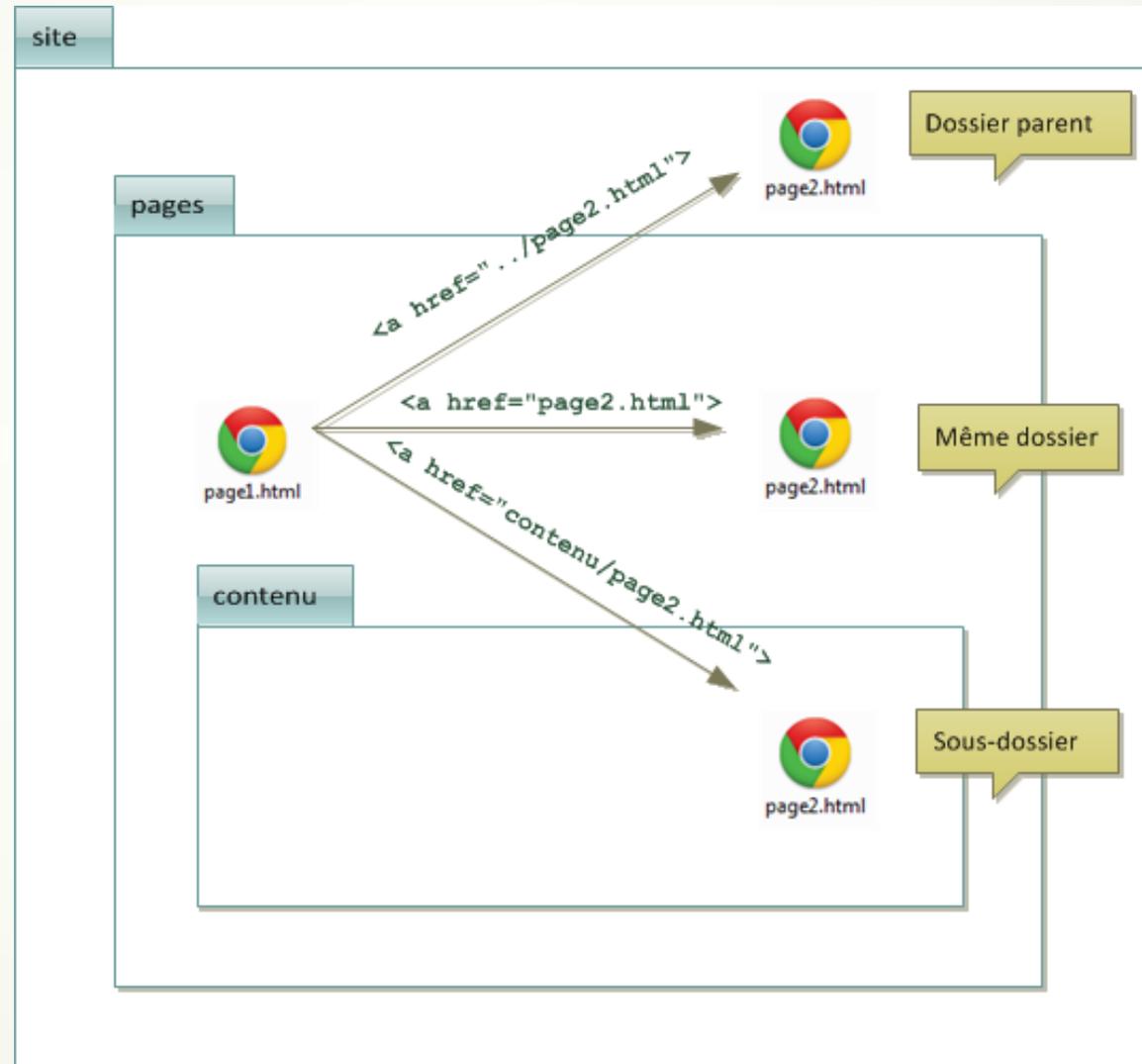
Si la page vers laquelle pointe le lien est dans un sous-dossier :

```
<a href="dossier/page2.html">Page2</a>
```

Si la page vers laquelle pointe le lien se trouve plus haut dans l'arborescence :

```
<a href="../page2.html">Page2</a>
```

# Organiser son texte à l'aide des balises



# Organiser son texte à l'aide des balises

## ► Liens vers une ancre

Une ancre est un point de repère sur la page. Un titre par exemple. En cliquant sur le lien on va directement aller sur la partie de la page qui nous intéresse. Il faudra donc 1) identifier l'ancre c'est-à-dire le point d'arrivée et 2) spécifier ce point dans la balise du lien.

Identifier l'ancre grâce à l'attribut id :

```
<h2 id="ancre">Titre</h2>
```

Créer le lien vers l'ancre :

```
<a href="#ancre">Aller vers l'ancre</a>
```

## ► Liens vers une ancre située dans une autre page

```
<a href="ancres.html#concerts">
```

# Organiser son texte à l'aide des balises

## ■ Exemple d'usage des ancres

```
<h1>Titre du paragraphe</h1>
```

```
<p>
```

```
  Aller directement à la partie traitant de :<br />
```

```
  <a href="#musique">Musique</a><br />
```

```
  <a href="#concert">Concerts</a><br />
```

```
  <a href="#video">Video</a><br />
```

```
</p>
```

```
<h2 id="musique">Musique</h2>
```

```
<p>... (beaucoup de texte) ...</p>
```

```
<h2 id="concert">Concerts</h2>
```

```
<p>... (beaucoup de texte) ...</p>
```

```
<h2 id="video">Video</h2>
```

```
<p>... (beaucoup de texte) ...</p>
```

# Organiser son texte à l'aide des balises

## ➤ Attributs supplémentaires des liens

- Lien qui affiche une infobulle au survol :

Attribut **'title'** : `<a href="https://adresse.com" title="visitez le site">Le site</a>`

- Lien qui s'ouvre dans une nouvelle fenêtre :

Attribut **'target'** : `<a href="https://adresse.com" title="Visitez le site" target="_blank">Le site</a>`

- Lien pour envoyer un mail :

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

- Lien pour télécharger un fichier :

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



# Insérer des images

## ► Les formats d'images :

- Le poids (en Ko, voire en Mo) de l'image sera plus ou moins élevé selon le format choisi et la qualité de l'image va changer.
- Le format JPEG. Il réduit le poids des photos en les compressant. C'est le format le plus intéressant pour les photos. L'extension JPEG est .jpg ou .jpeg
- Le format PNG. C'est le format le mieux adapté à tout ce qui n'est pas une photo. Son fond peut être rendu transparent, ce qui le rend intéressant. L'extension PNG est .png
- Le format GIF. Il reste intéressant dans la seule mesure où il peut être animé.
- Nommez vos photos sans accent ni espace.

# Insérer des images

- ▶ Insertion d'une image grâce à la balise `<img/>` et les attributs `src` et `alt`.
  - ``
  - Si le lien vers l'image est absolu, on entre l'adresse de l'image :  
`src="http://www.site.com/image.png"`
  - Si l'image est dans un sous dossier « images », on écrira : `src="images/image.jpg"`
  - `alt` signifie « texte alternatif ». Ce texte s'affiche à la place de l'image si celle-ci n'est pas chargée, et il permet au moteur de recherche de savoir de quoi il s'agit.
  - La balise `<img/>` est une balise inline au même titre que la balise `<a>` qui doit être insérée à l'intérieur d'un paragraphe `<p><img/></p>`.
  - Rajouter une infobulle au passage de la souris grâce à l'attribut « title » : ``

# Insérer des images

## ► Agrandir une miniature :

- `<a href="image.jpg"></a>`

## ► Illustrer une image avec la balise `<figure>` :

- `<figure>`
- ``
- `<figcaption>Texte Illustrant votre image</figcaption>`
- `</figure>`



## Partie 2 : Mettre en forme vos pages avec le CSS

- Gérer la mise en forme du site grâce au fichier CSS



# Mettre en place la feuille de style

- Trois manières d'inclure le CSS au HTML :
  - Dans un fichier séparé qui sera appelé dans le <head> du fichier html.
  - Directement dans le <head> du fichier html.
  - Directement dans les balises du fichier html grâce à l'attribut style.
- La première des trois méthodes reste la meilleure :
  - On crée alors une page ou fichier css : style.css
  - On appelle ce fichier dans le head du fichier html grâce à la balise `<link rel="stylesheet" href="style.css"/>`



# Formater du texte

- De quoi se compose un fichier CSS ?

```
p {  
color: red;  
}
```

**p** = Nom de la balise dont on veut modifier l'apparence.

**color** = Propriété à modifier.

**blue** = La valeur que l'on attribut à la propriété.

⇒ Ce code CSS signifie que tous les textes des paragraphes du fichier HTML seront rouges.



# Formater du texte

- Si deux balises doivent avoir la même propriété :

```
P, h1 {  
color: red;  
}
```

=> Tous les paragraphes et tous les titres h1 seront écrits en rouge.

- Ecrire un commentaire dans le fichier CSS :

```
/*ceci est un commentaire*/
```



# Formater du texte

- Spécifier les modifications grâce à des attributs spéciaux.

Jusqu'ici dans notre code CSS nous avons modifier tous les paragraphes sans distinction. Comment n'en modifier que certains ?

En les spécifiant grâce aux attributs `class` et `id`.

On peut mettre ces attributs sur toutes les balises html afin de les distinguer des autres, de leur donner un nom.

- Différence entre les attributs `class` et `id` :

Ils sont à peu près identiques et on la même fonction. Mais si `class` définit un ensemble d'élément de même classe, `id` définit un seul élément en le distinguant de tous les autres.

Il peut y avoir plusieurs balises de même classe mais un seul portera le nom de l'id.

La différence est pertinente dans l'usage des liens ancre par exemple.

# Formater du texte

- L'attribut **id** porte un nom unique.

```
<h1 id="nom">Titre</h1>
```

- L'attribut **class** porte un nom que l'on peut attribuer à plusieurs balises qui auront les mêmes propriétés.

```
<p class="intro">Texte1</p>
```

```
<p class="intro">Texte2</p>
```

- Pour modifier les balises ayant le même nom de class on écrit dans le fichier CSS :

```
.intro {color: red;}
```

- Pour modifier les balises ayant un attribut id on écrit dans le fichier CSS :

```
#nom {color: red;}
```



# Formater du texte

- Il est possible d'attribuer un attribut `class` et un attribut `id` à une même balise :

```
<h1 class="intro" id="nom">Titre</h1>
```

- Il est possible de donner plusieurs nom à l'attribut `class` d'une même balise :

```
<p class="intro texte">Texte</p>
```

# Formater du texte

## ► Les balises HTML universelles <div> et <span>

Si on veut modifier un élément qui n'est pas entouré de balises on va faire appel aux balises universelles <div> et <span> afin de leur attribuer un attribut class ou un attribut id permettant de les modifier en CSS. Ce sont des balises qui ne servent à rien d'autre qu'à insérer un attribut.

<span> est une balise inline au même titre que les balises <strong> ou <em>, elle s'insère au milieu d'un paragraphe pour spécifier un bout de texte à modifier.

<div> est une balise block au même titre que les balises <p> ou <h1>, elle crée un bloc avec retour à la ligne. Elles sont utiles pour structurer sa page.

Exemple :

HTML :

```
<p>Bonjour et <span class="salut">bienvenu</span> sur mon site</p>
```

CSS :

```
.salut {color: pink;}
```



# Formater du texte

## ► Les sélecteurs avancés

On a vu jusqu'alors :

- `p` `h1` `p,h1` `.class` `#id`

On a aussi :

- `*` qui est un sélecteur universel. Il sélectionne toutes les balises sans distinction.
- `h3 em` balise imbriquée qui signifie « toutes les balises em dans les balises h3 ».
- `h3+p` sélectionne la première balise p située après toutes les balises h3.

## ► D'autres sélecteurs existent

- Nous n'avons vu qu'une partie des sélecteurs existants, et nous en faisons d'autres. Si vous voulez une liste complète, vous pouvez vous renseigner directement à la source sur le site du W3C.



# Formater du texte

## ► Les propriétés de texte

- Taille du texte :

**font-size:** 16px; (taille absolue)

font-size: 1em; (taille normale) 1.3em (agrandit relativement) 0.8 (réduit relativement).

- Police :

**font-family:** nom\_de\_la\_police;

font-family: police1, police2, police3, serif; (le navigateur sélectionne la police qu'il possède en dernière occurrence on mettra « serif » qui indique la police par défaut si aucune autre police n'a été trouvée.)



# Formater du texte

- ▶ Police personnalisée :
- ▶ Avec CSS 3, il est possible d'utiliser n'importe quelle police sur son site. Cela fonctionne bien avec la plupart des navigateurs.
- ▶ Il faudra que le navigateur de vos visiteurs *télécharge* automatiquement le fichier de la police, dont le poids peut atteindre, voire dépasser 1 Mo...  
La plupart des polices sont soumises au droit d'auteur, il n'est donc *pas légal* de les utiliser sur son site. Mais il existe des sites comme [fontsqwirrel.com](http://fontsqwirrel.com) et [dafont.com](http://dafont.com) qui proposent en téléchargement un certain nombre de polices libres de droits. Je recommande en particulier [fontsqwirrel.com](http://fontsqwirrel.com) car il permet de télécharger des packs prêts à l'emploi pour CSS 3.  
Il existe *plusieurs formats* de fichiers de polices et ceux-ci ne fonctionnent pas sur tous les navigateurs.

# Formater du texte

- Voici les différents formats de fichiers de polices qui existent et qu'il faut connaître :

*.ttf* : *TrueType Font*. Fonctionne sur IE9 et tous les autres navigateurs.

*.eot* : *Embedded OpenType*. Fonctionne sur Internet Explorer uniquement, toutes versions. Ce format est propriétaire, produit par Microsoft.

*.otf* : *OpenType Font*. Ne fonctionne pas sur Internet Explorer.

*.svg* : *SVG Font*. Le seul format reconnu sur les iPhones et iPads pour le moment.

*.woff* : *Web Open Font Format*. Nouveau format conçu pour le Web, qui fonctionne sur IE9 et tous les autres navigateurs.

- En CSS, pour définir une nouvelle police, vous devez la déclarer comme ceci :

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

Le fichier de police (ici *NomDeLaPolice.eot*) doit ici être situé dans le même dossier que le fichier CSS (ou dans un sous-dossier, si vous utilisez un chemin relatif).

# Formater du texte

- Comme le format .eot ne fonctionne que sur Internet Explorer. L'idéal est de proposer plusieurs formats pour la police : le navigateur téléchargera celui qu'il sait lire. Voici comment indiquer plusieurs formats :

```
@font-face {  
  font-family: 'MaPolice';  
  src: url('MaPolice.eot') format('eot'),  
        url('MaPolice.woff') format('woff'),  
        url('MaPolice.ttf') format('truetype'),  
        url('MaPolice.svg') format('svg');  
}  
  
/* Utilisation de la police qu'on vient de définir sur les titres */  
h1 {  
  font-family: 'cac_champagneregular', Arial, serif;  
}
```

# Formater du texte

## ► Mettre en italique

On utilise la propriété : *font-style*

Avec les valeurs : *italic, oblique, normal*.

`em` /\*permet de supprimer l'italique sur les balises de texte `<em>*/`

```
{  
  font-style: normal;  
}
```

## ► Mettre en gras

On utilise la propriété : *font-weight*

Avec les valeurs : *bold, normal*.

`strong`/\*permet de supprimer le gras sur les balises de texte `<strong>*/`

```
{  
  font-weight: normal;  
}
```

# Formater du texte

## ▀ Soulignement et autres décorations

On utilise la propriété : *text-decoration*

Avec les valeurs : *underline* (souligné), *line-through* (barré), *overline* (ligne au dessus), *blink* (clignotant), *none* (normal).

a/\* permet de supprimer le soulignement sur les balises de lien <a>\*/

```
{  
  text-decoration: none;  
}
```

## ▀ L'alignement

On utilise la propriété : *text-align*

Avec les valeurs : *left* (le texte sera aligné à gauche, c'est la valeur par défaut), *center* (le texte sera centré), *right* (le texte sera aligné à droite), *justify* (le texte sera justifié).

# Formater du texte

## ► Les flottants

Le CSS nous permet de faire flotter un élément autour du texte. On dit aussi qu'on fait un « habillage ».



Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Donec  
vitae lorem imperdiet lacus molestie  
molestie. Cum sociis natoque penatibus  
et magnis dis parturient montes, nascetur ridiculus  
mus. Donec eu purus. Phasellus metus lorem,  
blandit et, posuere quis, tincidunt vitae, ante.  
Vivamus consequat mauris a diam. Vivamus nibh  
erat, hendrerit nec, aliquet ut, hendrerit quis, nunc.  
Vestibulum et turpis et elit tempor euismod.

# Formater du texte

On utilise la propriété : *float*

Avec les valeurs : *left* (l'élément flottera à gauche) ou *right* (l'élément flottera à droite).

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.

Html :

```
<p> Ceci est un texte normal de  
paragraphe, écrit à la suite de l'image et qui l'habillera car l'image est flottante.</p>
```

Css :

```
.imageflottante {  
    float: left;  
}
```

# Formater du texte

- Stopper un flottant

Comment faire si vous voulez qu'au bout d'un moment le texte continue *en dessous* du flottant ?



Ce texte entoure l'image. Bla  
bla bla bla bla bla bla

Texte écrit en-dessous

# Formater du texte

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* (après un `float: right`) ou *both* (le texte se poursuit en dessous que ce soit après un `float: left` ou un `float: 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).

## Html:

```
<p></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>
```

## Css:

```
.imageflottante {
```

```
    float: left;
```

```
}
```

```
.dessous {
```

```
    clear: both;
```

```
}
```



# Ajouter de la couleur et un fond

- Les couleurs
- Il existe plusieurs façons de déclarer la valeur d'une couleur en CSS :
  - Soit on tape son nom en anglais mais il n'existe que seize couleurs dites « standard » ce qui reste pauvre.
  - Soit on tape sa valeur hexadécimale du type #ffffff pour le blanc.
  - Soit on tape la valeur RGB (Red Green Blue). Toute couleur est une combinaison des trois couleurs rouge, vert, bleu selon certaines quantités de rouge, de vert ou de bleu.
  - Exemple pour le blanc, on peut écrire : white, #ffffff ou rgb(255, 255, 255).
  - Télécharger la boîte à couleur : <http://www.colovid.be/ColorBox.htm>

# Ajouter de la couleur et un fond

- Changer la couleur du texte grâce à la propriété *color* : `p{color: rgb(255, 255, 255)}`
- Changer la couleur du fond grâce à la propriété *background-color* : `body {background-color: black;}` Avec la balise `body` c'est toute la page qui sera noire, on peut appliquer la propriété à n'importe quelle balise. Ici il faudra aussi changer la couleur du texte en blanc par exemple pour qu'il apparaisse. Tout le texte à l'intérieur de la balise `body` sera blanc, c'est ce qu'on appelle l'héritage.

```
body {  
    background-color: black;  
    color: white;  
}
```

- Mettre une image de fond grâce à la propriété *background-image* :

```
body {  
    background-image: url("nom_image.png");  
}
```

# Ajouter de la couleur et un fond

- Options disponibles pour l'image de fond afin de changer le comportement de l'image :

- background-attachment: fixed* (l'image reste fixe), *scroll* (l'image défile avec le texte);

```
body {  
    background-image: url("nom_image.png");  
    background-attachment: fixed; /* Le fond restera fixe */  
}
```

- background-repeat : 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).

```
body {  
    background-image: url("nom_image.png");  
    background-repeat: fixed; /* Le fond restera fixe */  
}
```

# Ajouter de la couleur et un fond

- *background-position* : Vous devez donner à *background-position* deux valeurs en pixels pour indiquer la position du fond par rapport au coin supérieur gauche de la page (ou du paragraphe, si vous appliquez le fond à un paragraphe).

*background-position: 30px 50px;* votre fond sera placé à 30 pixels de la gauche et à 50 pixels du haut. Il est aussi possible d'utiliser ces valeurs en anglais : *top bottom left center right*.

```
body {  
    background-image: url("nom_image.png");  
    background-position: top right;  
    background-repeat: no-repeat;  
}
```

Cette propriété n'est intéressante que si elle est combinée avec *background-repeat: no-repeat;*(un fond qui ne se répète pas).

# Ajouter de la couleur et un fond

## ► Combiner les propriétés

```
body
{
  background: url("soleil.png") fixed no-repeat top right;
}
```

- L'ordre des valeurs n'a pas d'importance. Vous pouvez combiner les valeurs dans n'importe quel ordre.
- Vous n'êtes pas obligés de mettre toutes les valeurs. Ainsi, si vous ne voulez pas écrire `fixed`, vous pouvez l'enlever sans problème.

## ► Plusieurs images de fond

- Depuis CSS3, 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;
}
```

- La première image de cette liste sera placée par-dessus les autres.

# Ajouter de la couleur et un fond

## ➤ La transparence

### ➤ La propriété : opacity.

- Avec une valeur de 1, l'élément sera totalement opaque : c'est le comportement par défaut.
- Avec une valeur de 0, l'élément sera totalement transparent.

```
p
{
  opacity: 0.6;
}
```

## ➤ La notation RGBa.

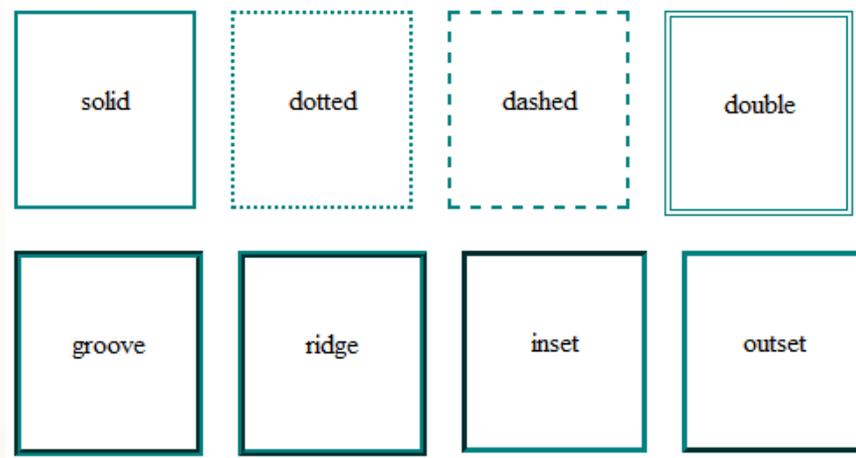
- Si vous appliquez la propriété `opacity` à un élément de la page, *tout* le contenu de cet élément sera rendu transparent (même les images, les autres blocs à l'intérieur, etc.). Si vous voulez juste rendre la couleur de fond transparente, utilisez la notation `RGBa`. Avec une valeur de 1, le fond sera totalement opaque : c'est le comportement par défaut. Avec une valeur de 0, le fond sera totalement transparent.

```
p
{
  background-color: rgba(255, 0, 0, 0.5); /* Fond rouge à moitié transparent */
}
```

# Créez des bordures et des ombres

## ➤ Bordures

- La « super » propriété : border.
- La largeur : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px). Avec une valeur de 0, l'élément sera totalement transparent.
- La couleur.
- 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.



# Créez des bordures et des ombres

- Ainsi, pour avoir une bordure bleue, en tirets, épaisse de 3 pixels autour de mes titres, je vais écrire :

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

- En haut, à droite, à gauche, en bas...

- Il est possible de modifier un seul des cotés de la bordure avec les propriétés : border-top: bordure du haut ; border-bottom: bordure du bas ; border-left: bordure de gauche ; border-right: bordure de droite.

```
p
{
  border-left: 2px solid black;
  border-right: 2px solid black;
}
```

# Créez des bordures et des ombres

## ➤ Bordures arrondies

- La propriété `border-radius` va nous permettre d'arrondir facilement les angles de n'importe quel élément. Il suffit d'indiquer la taille (« l'importance ») de l'arrondi en pixels :

```
p
{
  border-radius: 10px;
}
```

- On peut aussi préciser la forme de l'arrondi pour chaque coin. Dans ce cas, indiquez quatre valeurs :

```
p
{
  border-radius: 10px 5px 10px 5px;
}
```

# Créez des bordures et des ombres

## ► Les ombres

### ► Les ombres sur les boîtes.

- Avec la propriété `box-shadow`. Elle prend 4 valeurs : le décalage horizontal de l'ombre ; le décalage vertical de l'ombre ; l'adoucissement du dégradé ; la couleur de l'ombre. On peut rajouter la valeur `inset` à la fin pour placer l'ombre à l'intérieur de la boîte.

```
p
{
  box-shadow: 6px 6px 6px black;
}
```

### ► Les ombres sur les textes.

```
p
{
  text-shadow: 2px 2px 4px black;
}
```



# Créez des apparences dynamiques

- ▶ Le CSS nous permet aussi de modifier l'apparence des éléments de façon dynamique, c'est-à-dire que des éléments peuvent changer de forme une fois que la page a été chargée. Nous allons faire appel à une fonctionnalité puissante du CSS : les pseudo-formats.
- ▶ Nous verrons comment changer l'apparence :
- ▶ au survol ;
- ▶ lors du clic ;
- ▶ lors du focus (élément sélectionné) ;
- ▶ lorsqu'un lien a été consulté.

# Créez des apparences dynamiques

## ► Au survol

Nous allons découvrir dans ce chapitre plusieurs pseudo-formats CSS. Le premier que je vais vous montrer s'appelle `:hover`. Comme tous les autres pseudo-formats que nous allons voir, c'est une information que l'on rajoute après le nom de la balise (ou de la classe) dans le CSS, comme ceci :

```
a:hover  
{  
  
}
```

`:hover` signifie « survoler ». `a:hover` peut donc se traduire par : « Quand la souris est sur le lien » (quand on pointe dessus).

# Créez des apparences dynamiques

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

- On a défini ici deux versions des styles pour les liens :
- pour les liens par défaut (non survolés) ;
- pour les liens au survol.

# Créez des apparences dynamiques

## ► Au clic et lors de la sélection

Vous pouvez interagir encore plus finement en CSS. Nous allons voir ici que nous pouvons changer l'apparence des éléments lorsque l'on clique dessus et lorsqu'ils sont sélectionnés !

## ► `:active` : au moment du clic

Le pseudo-format `:active` permet d'appliquer un style particulier *au moment du clic*. En pratique, il n'est utilisé que sur les liens.

Le lien gardera cette apparence très peu de temps : en fait, le changement intervient lorsque le bouton de la souris est enfoncé.

```
a:active /* Quand le visiteur clique sur le lien */  
{  
  background-color: #FFCC66;  
}
```

# Créez des apparences dynamiques

- ▶ `:focus:` lorsque l'élément est sélectionné

Le pseudo-format `:focus` applique un style *lorsque l'élément est sélectionné*.

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

```
{
```

```
background-color: #FFCC66;
```

```
}
```

# Créez des apparences dynamiques

## ► Lorsque le lien a déjà été consulté

Il est possible d'appliquer un style à un lien vers une page qui a déjà été vue. Par défaut, le navigateur colore le lien en un violet.

Vous pouvez changer cette apparence avec `:visited` (qui signifie « visité »). En pratique, sur les liens consultés, on ne peut pas changer beaucoup de choses à part la couleur (figure suivante).

```
a:visited /* Quand le visiteur a déjà vu la page concernée */  
{  
  color: #AAA; /* Appliquer une couleur grise */  
}
```



## Partie 3 Mettre en page votre site

- ▶ En général, une page web est constituée d'un en-tête (tout en haut), de menus de navigation (en haut ou sur les côtés), de différentes sections au centre... et d'un pied de page (tout en bas).
- ▶ Dans ce chapitre, nous allons nous intéresser aux nouvelles balises HTML dédiées à la structuration du site. Ces balises ont été introduites par HTML5 (elles n'existaient pas avant) et vont nous permettre de dire : « Ceci est mon en-tête », « Ceci est mon menu de navigation », etc.
- ▶ Pour le moment, nous n'allons pas encore faire de mise en page. Nous allons en fait préparer notre document HTML pour pouvoir découvrir la mise en page dans les prochains chapitres.



# Structurer votre page

## ► Les balises structurantes de HTML5

### ► <header>: l'en-tête

- La plupart des sites web possèdent en général un en-tête, appelé *header* en anglais. On y trouve le plus souvent un logo, une bannière, le slogan de votre site...

### ► <footer>: le pied de page

- À l'inverse de l'en-tête, le pied de page se trouve en général tout en bas du document. On y trouve des informations comme des liens de contact, le nom de l'auteur, les mentions légales, etc.

### ► <nav>: principaux liens de navigation

- La balise <nav> doit regrouper tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site. Généralement, le menu est réalisé sous forme de liste à puces à l'intérieur de la balise <nav>.



# Structurer votre page

- **Les balises structurantes de HTML5**

- `<section>`: une section de page

- La balise `<section>` sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

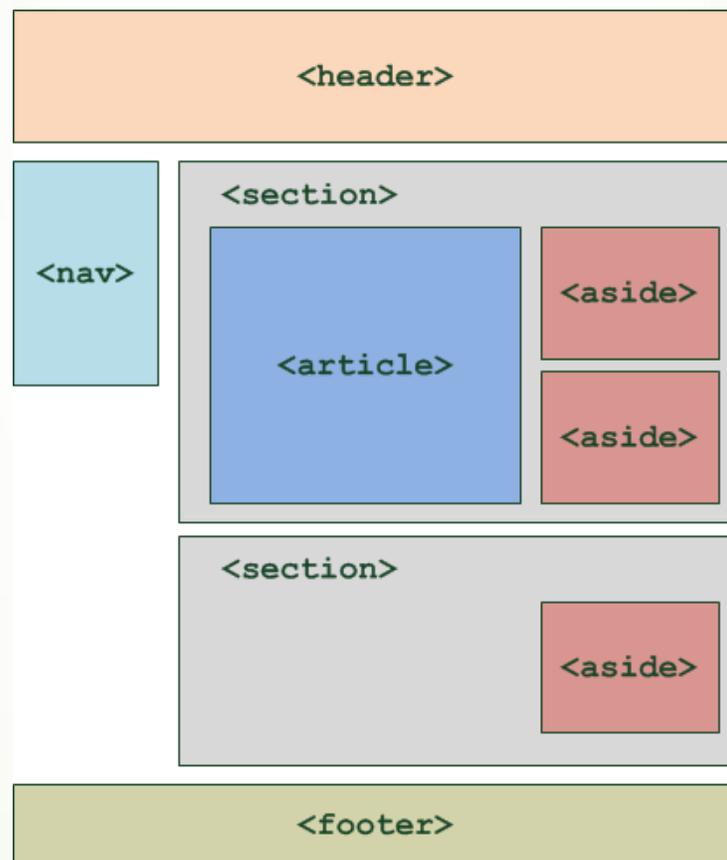
- `<aside>`: informations complémentaires

- La balise `<aside>` est conçue pour contenir des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté (bien que ce ne soit pas une obligation).

- `<article>`: un article indépendant

- La balise `<article>` sert à englober une portion généralement autonome de la page. C'est une partie de la page qui pourrait ainsi être reprise sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

# Structurer votre page



# Structurer votre page

## Exemple concret d'utilisation des balises

```
<body>
  <header>
    <h1>HyperSensible</h1>
    <h2>Un groupe de musique</h2>
  </header>
  <nav>
    <ul>
      <li><a href="#">Accueil</a></li>
      <li><a href="#">Concert</a></li>
      <li><a href="#">Video</a></li>
    </ul>
  </nav>
  <section>
    <aside>
      <h1>Nouvel Album</h1>
      <p>Phasellus ligula massa, congue ac vulputate non, dignissim at augue.</p>
    </aside>
    <article>
      <h1>Présentation</h1>
      <p>Phasellus ligula massa, congue ac vulputate non, dignissim at augue.</p>
    </article>
  </section>
  <footer>
    <p>Copyright HyperSensible- Tous droits réservés<br />
    <a href="#">Me contacter !</a></p>
  </footer>
</body>
```



# Les modèles de boîtes

- ▶ Une page web peut être vue comme une succession et un empilement de boîtes, qu'on appelle « blocs ». La plupart des éléments vus au chapitre précédent sont des blocs :<header>,<article>,<nav>... Mais nous connaissions déjà d'autres blocs : les paragraphes<p>, les titres<h1>...
- ▶ Dans ce chapitre, nous allons apprendre à manipuler ces blocs comme de véritables boîtes. Nous allons leur donner des dimensions, les agencer en jouant sur leurs marges, mais aussi apprendre à gérer leur contenu... pour éviter que le texte ne dépasse de ces blocs !
- ▶ Ce sont des notions fondamentales dont nous allons avoir besoin pour mettre en page notre site web.

# Les modèles de boîtes

- Les balises de type block et inline
- En HTML, la plupart des balises peuvent se ranger dans l'une ou l'autre de deux catégories : block ou inline.
- **block** : une balise de type block sur votre page web crée automatiquement un retour à la ligne avant et après. Il suffit d'imaginer tout simplement un bloc. Votre page web sera en fait constituée d'une série de blocs les uns à la suite des autres. Mais vous verrez qu'en plus, il est possible de mettre un bloc à l'intérieur d'un autre, ce qui va augmenter considérablement nos possibilités pour créer le design de notre site ! Exemples : `<p><footer>h1</footer></p><article>`
- **inline** : une balise de type inline se trouve obligatoirement à l'intérieur d'une balise block. Une balise inline ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne (c'est pour cela que l'on parle de balise « en ligne »). Exemples : `<em><strong><a><img/>`

# Les modèles de boîtes

- ▶ Les balises de type block et inline
- ▶ Sur fond bleu, vous avez tout ce qui est de type block.
- ▶ Sur fond jaune, vous avez tout ce qui est de type inline.

```
<h1>Titre (block)</h1>
```

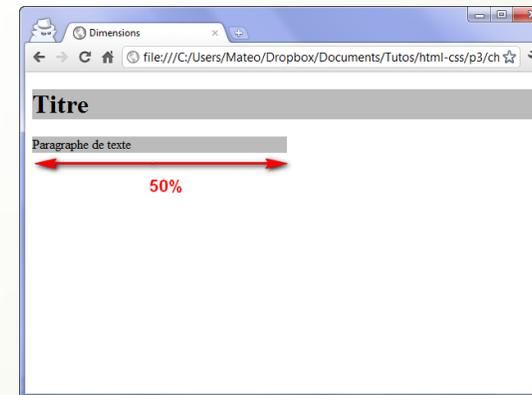
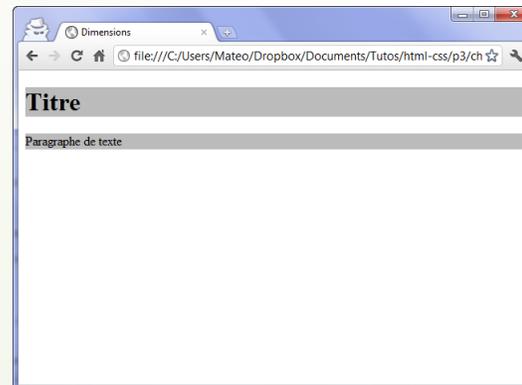
```
<p>Paragraphe blablabla blablabla  
blablabla <a>Lien inline</a> blabla  
blablabla et toujours blabla  
</p>
```

```
<p>Encore un paragraphe (block)  
</p>
```

# Les modèles de boîtes

- Les balises universelles
- `<span></span>` (**inline**) ;
- `<div></div>` (**block**).
- Les dimensions 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%;  
}
```



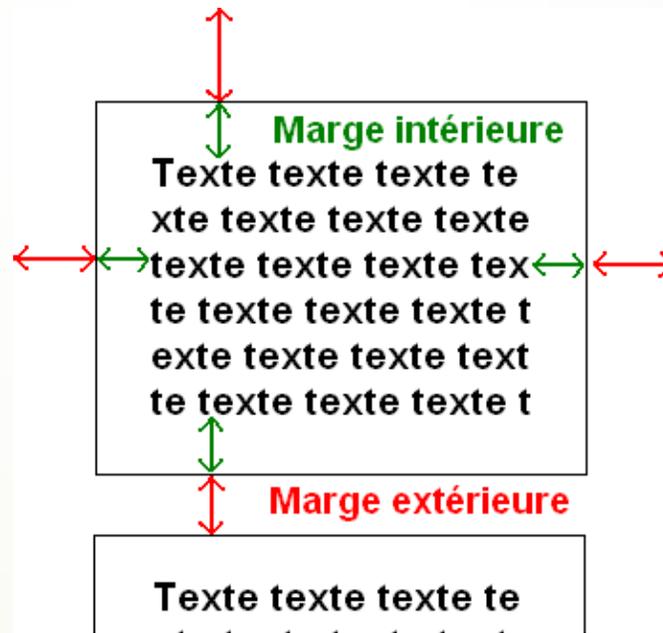
# Les modèles de boîtes

- Les dimensions des balises de type block.
- Minimum et maximum
- On peut demander à ce qu'un bloc ait des dimensions minimales et maximales. C'est très pratique car cela nous permet de définir des dimensions « limites » pour que notre site s'adapte aux différentes résolutions d'écran de nos visiteurs :
- min-width: largeur minimale ;
- min-height: hauteur minimale ;
- max-width: largeur maximale ;
- max-height: hauteur maximale.
- Par exemple, on peut demander à ce que les paragraphes occupent 50% de la largeur *et* exiger qu'il fassent au moins 400 pixels de large dans tous les cas :

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

# Les modèles de boîtes

- Les marges sur les balise block et inline.
- Il faut savoir que tous les blocs possèdent des marges. Il existe *deux types de marges* :
- les marges intérieures ;
- les marges extérieures.



# Les modèles de boîtes

- Les marges sur les balise block et inline.
- En CSS, on peut modifier la taille des marges avec les deux propriétés suivantes :
- 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.
- Exemple : `margin: 2px 0 3px 1px;` signifie « 2 px de marge en haut, 0 px à droite (le px est facultatif dans ce cas), 3 px en bas, 1 px à gauche ».
- Si l'on ne veut modifier qu'une seule marge :
- margin-top: marge extérieure en haut ;
- margin-bottom: marge extérieure en bas ;
- margin-left: marge extérieure à gauche ;
- margin-right: marge extérieure à droite.
  
- padding-top: marge intérieure en haut ;
- padding-bottom: marge intérieure en bas ;
- padding-left: marge intérieure à gauche ;
- padding-right: marge intérieure à droite.



# Les modèles de boîtes

- ▶ Centrer des blocs
- ▶ Il est tout à fait possible de centrer des blocs. C'est même très pratique pour réaliser un design centré quand on ne connaît pas la résolution du visiteur.
- ▶ 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;*
- ▶ Quand ça dépasse...
- ▶ Lorsqu'on commence à définir des dimensions précises pour nos blocs, comme on vient de le faire, il arrive qu'ils deviennent trop petits pour le texte qu'ils contiennent.
- ▶ Les propriétés CSS que nous allons voir ici ont justement été créées pour contrôler les dépassements... et décider quoi faire si jamais cela devait arriver.

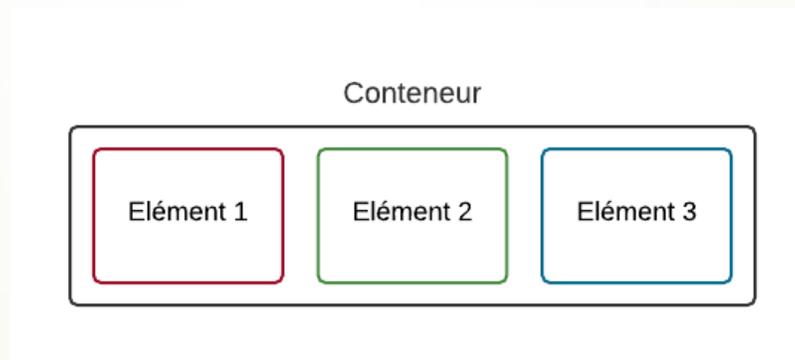
# Les modèles de boîtes

- **overflow: couper un bloc**
- 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.
- **word-wrap: couper les textes trop larges**
- Cette propriété permet de forcer la césure des très longs mots (généralement des adresses un peu longues).

```
p  
{  
  word-wrap: break-word;  
}
```

# Faites votre mise en page avec Flexbox

- ▶ Le principe de la mise en page avec Flexbox est simple : vous définissez un conteneur, et à l'intérieur vous placez plusieurs éléments. Imaginez un carton dans lequel vous rangez plusieurs objets : c'est le principe !
- ▶ Sur une même page web, vous pouvez sans problème avoir plusieurs conteneurs (plusieurs cartons si vous préférez ). Ce sera à vous d'en créer autant que nécessaire pour obtenir la mise en page que vous voulez.
- ▶ Commençons par étudier le fonctionnement d'un conteneur.



# Faites votre mise en page avec Flexbox

- ▶ Le conteneur est une balise HTML, et les éléments sont d'autres balises HTML à l'intérieur :

```
<div id="conteneur">  
  <div class="element 1">Element 1</div>  
  <div class="element 1">Element 2</div>  
  <div class="element 3">Element 3</div>  
</div>
```

- ▶ Avec un tel code html les éléments se mettront les uns en dessous des autres.
- ▶ Avec la propriété css flex appliquée au conteneur, les blocs se placent les uns à côté des autres.

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

# Faites votre mise en page avec Flexbox

- La direction
- Flexbox nous permet d'agencer ces éléments dans le sens que l'on veut. Avec flex-direction, on peut les positionner verticalement ou encore les inverser. Il peut prendre les valeurs suivantes :
- row : organisés sur une ligne (par défaut)
- column : organisés sur une colonne
- row-reverse : organisés sur une ligne, mais en ordre inversé
- column-reverse : organisés sur une colonne, mais en ordre inversé

```
#conteneur  
{  
  display: flex;  
  flex-direction: row-reverse;  
}
```

# Faites votre mise en page avec Flexbox

- Le retour à la ligne
- Par défaut, les blocs essaient de rester sur la même ligne s'ils n'ont pas la place (ce qui peut provoquer des bugs de design parfois). Si vous voulez, vous pouvez demander à ce que les blocs aillent à la ligne lorsqu'ils n'ont plus la place avec `flex-wrap` qui peut prendre ces valeurs :
- `nowrap` : pas de retour à la ligne (par défaut)
- `wrap` : les éléments vont à la ligne lorsqu'il n'y a plus la place
- `wrap-reverse` : les éléments vont à la ligne lorsqu'il n'y a plus la place en sens inverse.

```
#conteneur
{
  display: flex;
  flex-wrap: wrap;
}
```



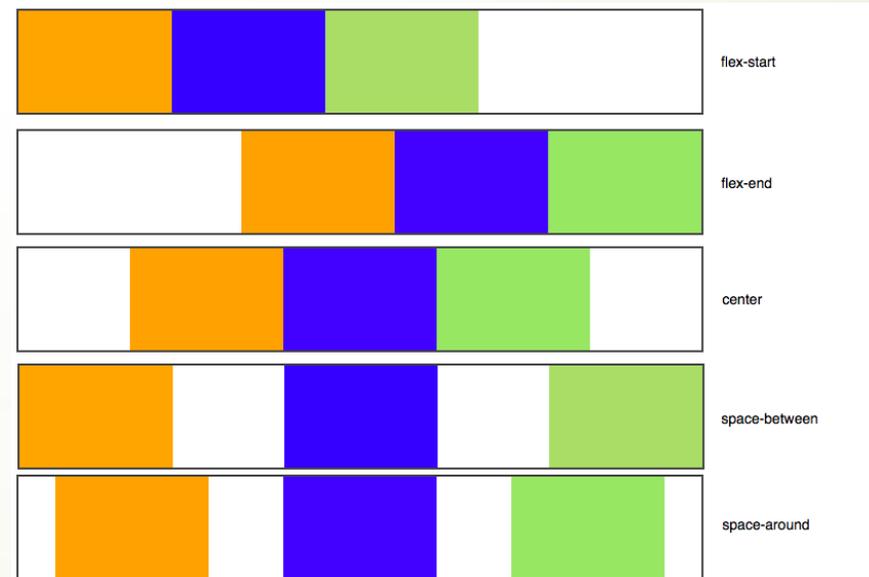
# Faites votre mise en page avec Flexbox

- Aligner les éléments
- Les éléments sont organisés soit horizontalement (par défaut), soit verticalement. Cela définit ce qu'on appelle **l'axe principal**. Il y a aussi un axe secondaire :
- Si vos éléments sont organisés horizontalement, l'axe secondaire est l'axe vertical.
- Si vos éléments sont organisés verticalement, l'axe secondaire est l'axe horizontal.
- Nous allons découvrir comment aligner nos éléments sur l'axe principal *et* sur l'axe secondaire.

# Faites votre mise en page avec Flexbox

- **Aligner sur l'axe principal**
- Pour faire simple, partons sur des éléments organisés horizontalement (c'est le cas par défaut).
- Pour changer leur alignement, on va utiliser *justify-content*, qui peut prendre ces valeurs :
- flex-start : alignés au début (par défaut)
- flex-end : alignés à la fin
- center : alignés au centre
- space-between : les éléments sont étirés sur tout l'axe (il y a de l'espace entre eux)
- space-around : idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités

```
#conteneur
{
  display: flex;
  justify-content: space-around;
}
```

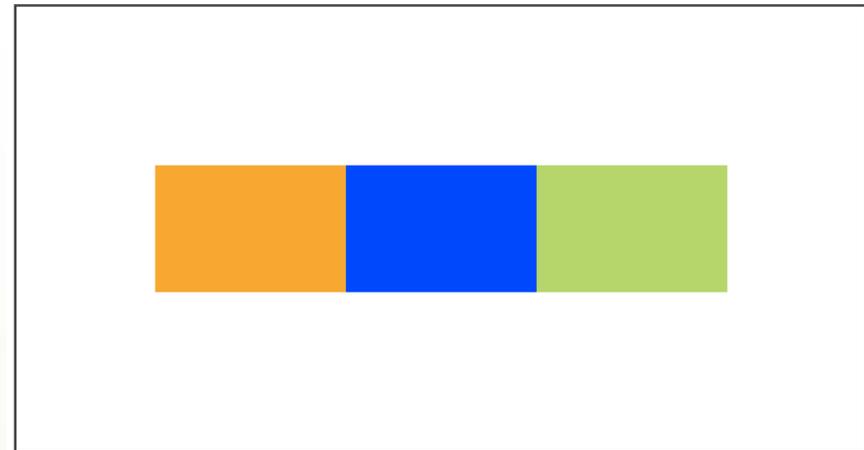


# Faites votre mise en page avec Flexbox

## ➤ Aligner sur l'axe secondaire

- Avec *align-items*, nous pouvons changer leur alignement sur l'axe secondaire. Il peut prendre ces valeurs :
- *stretch* : les éléments sont étirés sur tout l'axe (valeur par défaut)
- *flex-start* : alignés au début
- *flex-end* : alignés à la fin
- *center* : alignés au centre
- *baseline* : alignés sur la ligne de base (semblable à *flex-start*)

```
#conteneur  
{  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```



# Faites votre mise en page avec Flexbox

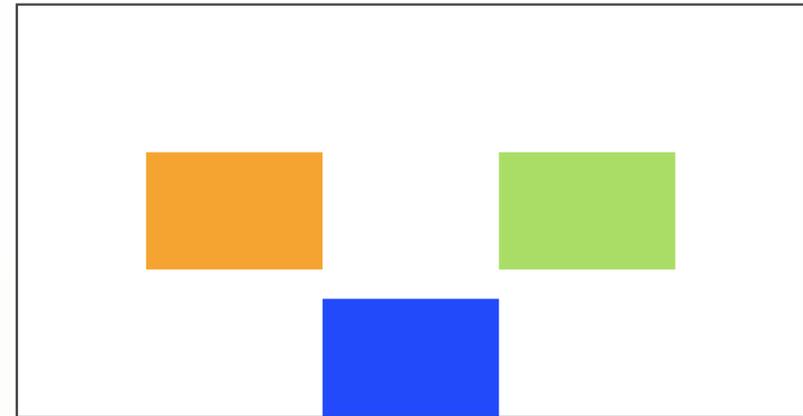
## ➤ Aligner un seul élément

- Pour cibler un seul élément de du conteneur, on va utiliser la pseudo classe `:nth-child` à laquelle on attribut la propriété `align-self`

```
#conteneur
{
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
}

.element:nth-child(2) /* On prend le deuxième bloc élément */
{
  background-color: blue;
  align-self: flex-end; /* Seul ce bloc sera aligné à la fin */
}

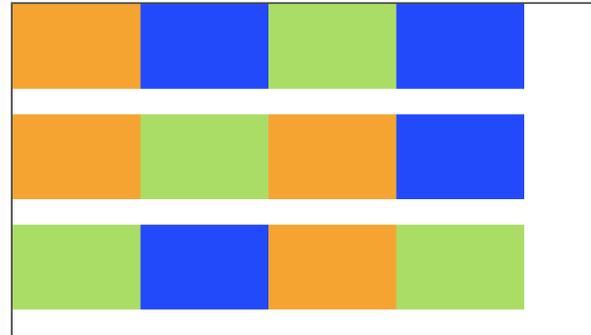
/* ... */
```





# Faites votre mise en page avec Flexbox

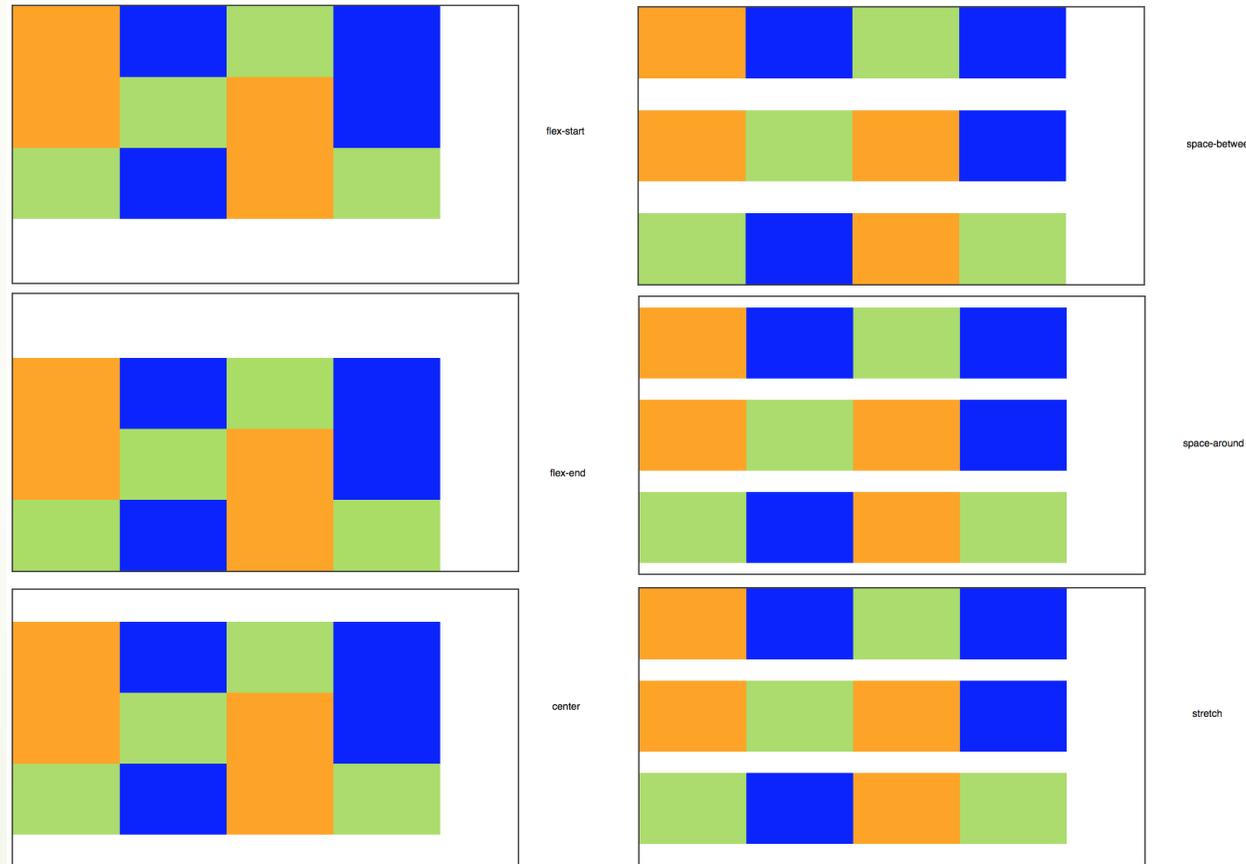
- Répartir plusieurs lignes
- Voici ce que l'on obtient



- Avec la propriété align-content il est possible de répartir les lignes différemment grâce aux valeurs :
- flex-start : les éléments sont placés au début
- flex-end : les éléments sont placés à la fin
- center : les éléments sont placés au centre
- space-between : les éléments sont séparés avec de l'espace entre eux
- space-around : idem, mais il y a aussi de l'espace au début et à la fin
- stretch (par défaut) : les éléments s'étirent pour occuper tout l'espace

# Faites votre mise en page avec Flexbox

- Répartir plusieurs lignes
- Voici ce que l'on obtient :



# Faites votre mise en page avec Flexbox

## ► Change l'ordre des éléments

- Sans changer le code HTML, nous pouvons modifier l'ordre des éléments en CSS grâce à la propriété `order` et la pseudo class `:nth-child`. Indiquez simplement un nombre, et les éléments seront triés du plus petit au plus grand nombre.
- Si on aligne trois éléments. Ici le 1 sera en 3, le 2 en 1 et le 3 en 2 :

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

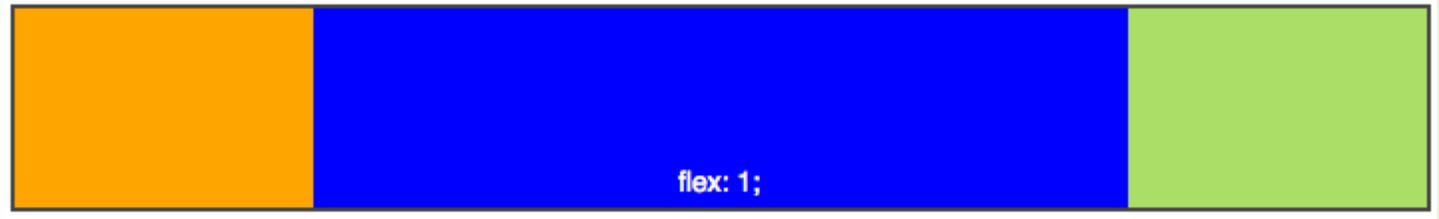
- Il est possible de changer l'ordre des éléments :

```
.element:nth-child(1)  
{  
  order: 3;  
}  
.element:nth-child(2)  
{  
  order: 1;  
}  
.element:nth-child(3)  
{  
  order: 2;  
}
```

# Faites votre mise en page avec Flexbox

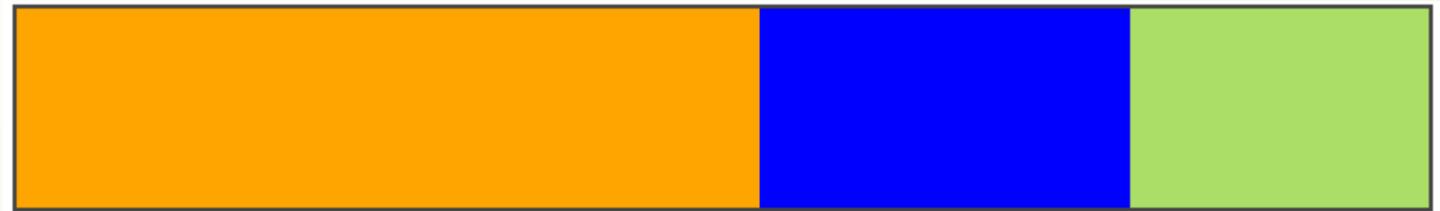
- **Changer la largeur des éléments**
- Avec la propriété *flex*, nous pouvons permettre à un élément de grossir pour occuper tout l'espace restant.

```
.element:nth-child(2)  
{  
  flex: 1;  
}
```



- Le nombre que vous indiquez à la propriété *flex* indique dans quelle mesure il peut grossir par rapport aux autres.

```
.element:nth-child(1)  
{  
  flex: 2;  
}  
.element:nth-child(2)  
{  
  flex: 1;  
}
```



# Autres techniques de mise en page

- Flexbox est une nouveauté du css3. Si vous gérez des sites plus anciens il peut être utile de connaître les anciennes techniques de positionnement.
- **Le positionnement flottant**
- On va vu la propriété *float* pour permettant de faire flotter une image autour du texte. On peut aussi s'en servir pour la mise en page.

```
<nav>
  <ul>
    <li><a href="#">Accueil</a></li>
    <li><a href="#">Musique</a></li>
    <li><a href="#">Concert</a></li>
  </ul>
</nav>
```

```
<section>
  <aside>
    <h1>Album</h1>
    <p> Phasellus ligula massa, congue ac vulputate non, dignissim at augue. </p>
  </aside>
  <article>
    <h1>Présentation</h1>
    <p>Bla bla bla bla (texte de l'article)</p>
  </article>
</section>
```

# Autres techniques de mise en page

- Nous allons essayer de placer le menu à gauche et le reste du texte à droite. Pour cela, nous allons faire flotter le menu à gauche et laisser le reste du texte se placer à sa droite.
- Nous voulons que le menu occupe 150 pixels de large. Nous allons aussi rajouter une bordure noire autour du menu et une bordure bleue autour du corps (à la balise<section>) pour bien les distinguer.

```
nav
{
  float: left;
  width: 150px;
  border: 1px solid black;
}

section
{
  border: 1px solid blue;
}
```

- On s'aperçoit que le texte touche la bordure du menu et que le texte passe sous le menu.
- Pour résoudre ces deux problèmes d'un seul coup, il faut ajouter une marge extérieure à gauche de notre<section>, marge qui doit être par ailleurs *supérieure* à la largeur du menu. Si notre menu fait 150 px, nous allons par exemple donner une marge extérieure gauche de 170 px à notre section de page.

# Autres techniques de mise en page

## ► Le positionnement inline-block

- La propriété *display* permet de transformer un élément inline en élément block et inversement. Si on applique la propriété *display* à un balise de lien `a{display: block;}`, les liens vont se positionner les uns en dessous des autres.
- La propriété *display* peut prendre 4 valeurs : **inline** (Éléments d'une ligne. Se placent les uns à côté des autres), **block** (Éléments en forme de blocs. Se placent les uns en-dessous des autres et peuvent être redimensionnés), **inline-block** (Éléments positionnés les uns à côté des autres - comme les inlines- mais qui peuvent être redimensionnés -comme les blocs-), **none** (Élément ni affichés).
- Les éléments block comme le menu et le corps de notre page transformés en inline-block peuvent être mis côte à côte (comme une balise inline)et redimensionnés (comme une balise block).

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

```
section
{
  display: inline-block;
  border: 1px solid blue;
}
```

# Autres techniques de mise en page

- Pour finir. Les éléments inline-block se positionnent sur une même ligne de base en bas (baseline). La propriété *vertical-align* permet de modifier verticalement les éléments grâce aux valeurs :
- **baseline**: aligne 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 ; ou (valeur en **px** ou **%**) : aligne à une certaine distance de la ligne de base (baseline).

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

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

# Autres techniques de mise en page

## ► Les positionnements absolu, fixe et relatif

- La propriété *position* permet de positionner avec précision des éléments sur la page grâce aux valeurs :
- *absolute* : positionnement absolu, il nous permet de placer un élément n'importe où sur la page (en haut à gauche, en bas à droite, tout au centre, etc.).
- *fixed* : positionnement fixe, identique au positionnement absolu mais, cette fois, l'élément reste toujours visible, même si on descend plus bas dans la page. C'est un peu le même principe que `background-attachment: fixed`;
- *relative* : positionnement relatif, permet de décaler l'élément par rapport à sa position normale.
- Il faut ensuite indiquer où doit se positionner l'élément grâce à ces quatre propriétés :
- *left*: position par rapport à la gauche de la page ; *right*: position par rapport à la droite de la page ; *top*: position par rapport au haut de la page ; *bottom*: position par rapport au bas de la page. Auxquels on donne une valeur en px ou en %.

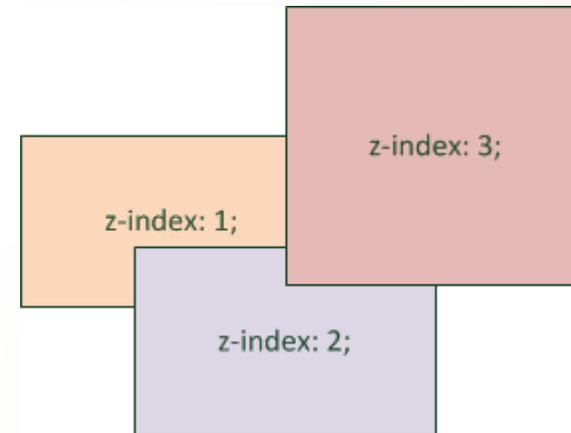
```
element
{
  position: absolute;
  right: 0px;
  bottom: 0px;
}
```

# Autres techniques de mise en page

## ► Les positionnements absolu, fixe et relatif

- Positionnement absolu et fixe :
- Les éléments positionnés en *absolute* ou *fixed* sont placés par-dessus le reste des éléments de la page ! Par ailleurs, si vous placez deux éléments en absolu vers le même endroit, ils risquent de se chevaucher. Dans ce cas, utilisez la propriété *z-index* pour indiquer quel élément doit apparaître au-dessus des autres.

```
element
{
  position: absolute;
  right: 0px;
  bottom: 0px;
  z-index: 1;
}
element2
{
  position: absolute;
  right: 30px;
  bottom: 30px;
  z-index: 2;
}
```



Le positionnement absolu ne se fait pas forcément toujours par rapport au coin en haut à gauche de la fenêtre ! Si vous positionnez en absolu un bloc A qui se trouve dans un autre bloc B, lui-même positionné en absolu (ou fixe ou relatif), alors votre bloc A se positionnera par rapport au coin supérieur gauche du bloc B.

# Autres techniques de mise en page

## ► Les positionnements absolu, fixe et relatif

### ► Positionnement relatif :

- L'élément est décalé par rapport à sa position initiale. Prenons par exemple un texte important, situé entre deux balises <strong>. Pour commencer, je le mets sur fond rouge pour qu'on puisse mieux le repérer :

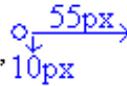
```
strong
{
  background-color: red; /* Fond rouge */
  color: yellow; /* Texte de couleur jaune */
}
```



Pas de doute, **ce texte est important** si on veut comprendre corre

- Ici le point de coordonnées (0, 0) ne se trouve plus en haut à gauche de votre fenêtre comme c'était le cas tout à l'heure. Elle se trouve en haut à gauche... de la position actuelle de votre élément.
- Donc, si vous faites un `position: relative;` et que vous appliquez une des propriétés `top`, `left`, `right` ou `bottom`, le texte sur fond rouge va se déplacer par rapport à la position où il se trouve.

# Autres techniques de mise en page

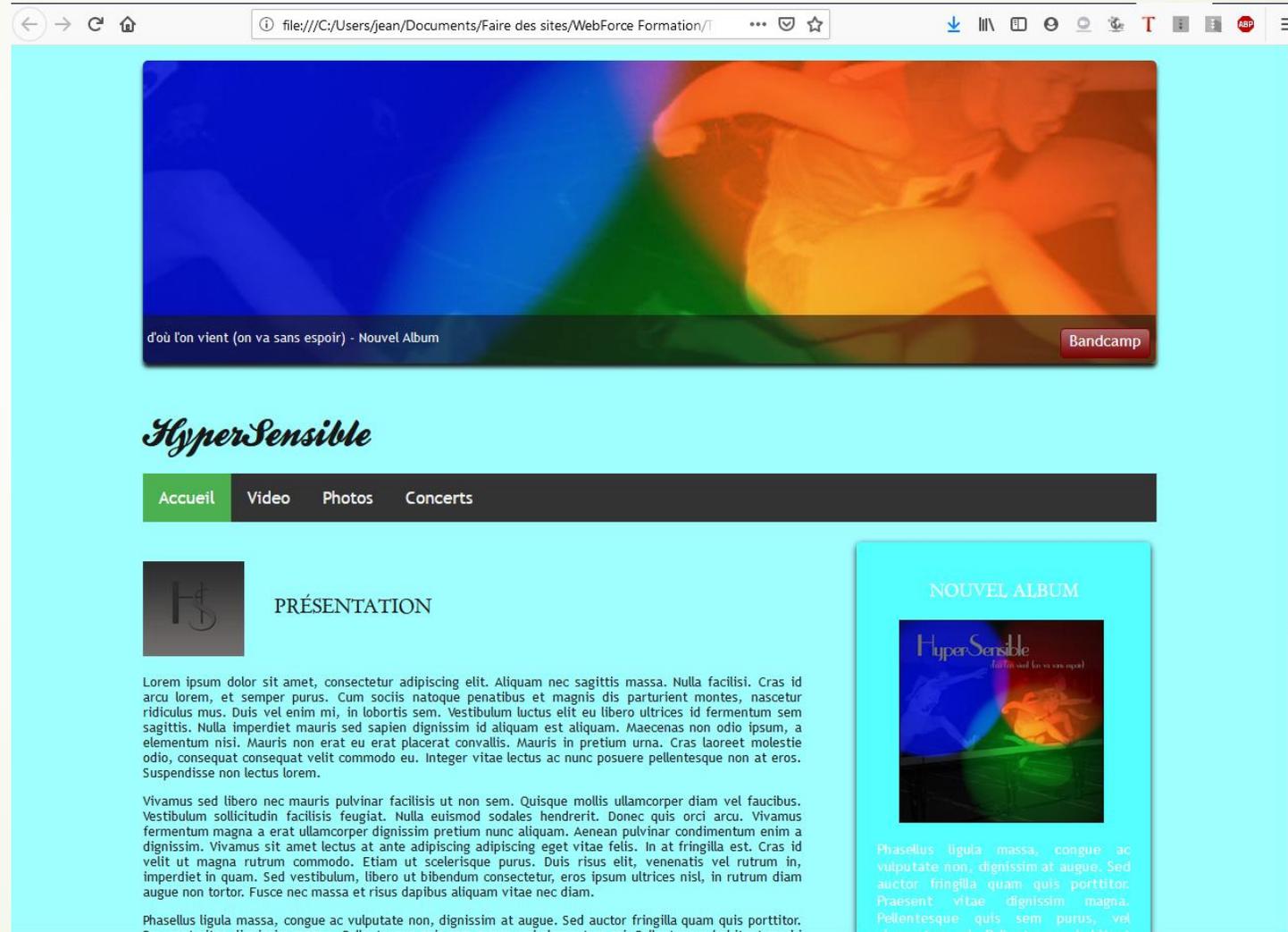
- Les positionnements absolu, (Pas de doute,  ce texte est important) si on veut com

```
strong
{
  background-color: red;
  color: yellow;

  position: relative;
  left: 55px;
  top: 10px;
}
```

- Le texte est alors décalé par rapport à sa position initiale, comme illustré à la figure suivante.

# Créer un site à partir d'une maquette



# Partie 4 : Découvrir des fonctionnalités évoluées

## ➤ Ajoutez des tableaux

### ➤ Un tableau simple

- La première balise html à connaître est `<table>` `</table>`. C'est cette balise qui permet d'indiquer le début et la fin d'un tableau. C'est une balise bloc.
- A l'intérieur de ces balises on insère les balises `<tr>` `</tr>` qui indique le début et la fin d'une ligne du tableau ; et `<td>` `</td>` qui indique le début et la fin du contenu d'une cellule.

```
<table>
<tr>
  <td>Carmen</td>
  <td>33 ans</td>
  <td>Espagne</td>
</tr>
<tr>
  <td>Michelle</td>
  <td>26 ans</td>
  <td>États-Unis</td>
</tr>
</table>
```

The diagram shows a table with four rows and three columns. The first row contains headers: 'Nom', 'Age', and 'Pays'. The second row contains 'Anne', '27 ans', and 'France'. The third row contains 'Carmen', '33 ans', and 'Espagne'. The fourth row contains 'Michelle', '26 ans', and 'États-Unis'. The fifth row contains 'Ogasaku Nyagatosoka', '18 ans', and 'Japon'. A red box highlights the third row, labeled 'Tr (ligne)'. Green boxes highlight the three cells of the third row, labeled 'Td (cellules)'. Green lines connect the label 'Td (cellules)' to each of the three cells in the third row.

Nom	Age	Pays
Anne	27 ans	France
Carmen	33 ans	Espagne
Michelle	26 ans	États-Unis
Ogasaku Nyagatosoka	18 ans	Japon

# Ajoutez des tableaux

- Par défaut un tableau est sans bordure. Pour en rajouter :

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

- Pour coller les bordures entre elles :

```
table  
{  
  border-collapse: collapse; /* Les bordures du tableau seront collées*/  
}  
td  
{  
  border: 1px solid black;  
}
```

# Ajoutez des tableaux

- ▶ Ajouter une ligne d'entête :
- ▶ La ligne d'en-tête se crée avec un <tr> comme on l'a fait jusqu'ici, mais les cellules qu'elle contient sont, cette fois, encadrées par des balises <th> et non pas <td> !

```
<table>
  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>

  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>

  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

# Ajoutez des tableaux

- Donner un titre au tableau `<caption>` :

```
<table>
  <caption>Passagers du vol 377</caption>

  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

# Ajoutez des tableaux

- Structurer un tableau :
- balises HTML 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>`.

<b>En-tête du tableau</b>	<b>Nom</b>	<b>Age</b>	<b>Pays</b>	<b>&lt;thead&gt;</b>
	Carmen	33 ans	Espagne	
	Michelle	26 ans	Etats-Unis	
<b>Corps du tableau</b>	François	43 ans	France	<b>&lt;tbody&gt;</b>
	Martine	34 ans	France	
	Jonathan	13 ans	Australie	
	Xu	19 ans	Chine	
<b>Pied du tableau</b>	<b>Nom</b>	<b>Age</b>	<b>Pays</b>	<b>&lt;tfoot&gt;</b>

# Ajoutez des tableaux

```
<table>
  <caption>Passagers du vol 377</caption>

  <thead> <!-- En-tête du tableau -->
    <tr>
      <th>Nom</th>
      <th>Âge</th>
      <th>Pays</th>
    </tr>
  </thead>

  <tfoot> <!-- Pied de tableau -->
    <tr>
      <th>Nom</th>
      <th>Âge</th>
      <th>Pays</th>
    </tr>
  </tfoot>

  <tbody> <!-- Corps du tableau -->
    <tr>
      <td>Carmen</td>
      <td>33 ans</td>
      <td>Espagne</td>
    </tr>
    <tr>
      [...]
    </tr>
  </tbody>
</table>
```



# Ajoutez des tableaux

- Fusionner les colonnes et les lignes :
- Pour effectuer une fusion, on rajoute un attribut à la balise `<td>`. Il faut savoir qu'il existe deux types de fusion :
- **La fusion de colonnes** : La fusion s'effectue horizontalement. On utilisera l'attribut `colspan`.
- **La fusion de lignes** : La fusion s'effectuera verticalement. On utilisera l'attribut `rowspan`.
- On indique le nombre de cellules à fusionner.

# Ajoutez des tableaux

```
<table>
<tr>
  <th>Titre du film</th>
  <th>Pour enfants ?</th>
  <th>Pour adolescents ?</th>
</tr>
<tr>
  <td>Massacre à la tronçonneuse</td>
  <td>Non, trop violent</td>
  <td>Oui</td>
</tr>
<tr>
  <td>Les bisounours font du ski</td>
  <td>Oui, adapté</td>
  <td>Pas assez violent...</td>
</tr>
<tr>
  <td>Lucky Luke, seul contre tous</td>
  <td colspan="2">Pour toute la famille !</td>
</tr>
</table>
```

<b>Titre du film</b>	<b>Pour enfants ?</b>	<b>Pour adolescents ?</b>
Massacre à la tronçonneuse	Non, trop violent	Oui
Les bisounours font du ski	Oui, adapté	Pas assez violent...
Lucky Luke, seul contre tous	Pour toute la famille !	

# Ajoutez des tableaux

```
<table>
<tr>
  <th>Titre du film</th>
  <td>Massacre à la tronçonneuse</td>
  <td>Les bisounours font du ski</td>
  <td>Lucky Luke, seul contre tous</td>
</tr>
<tr>
  <th>Pour enfants ?</th>
  <td>Non, trop violent</td>
  <td>Oui, adapté</td>
  <td rowspan="2">Pour toute la famille !</td>
</tr>
<tr>
  <th>Pour adolescents ?</th>
  <td>Oui</td>
  <td>Pas assez violent...</td>
</tr>
</table>
```

<b>Titre du film</b>	Massacre à la tronçonneuse	Les bisounours font du ski	Lucky Luke, seul contre tous
<b>Pour enfants ?</b>	Non, trop violent	Oui, adapté	Pour toute la famille !
<b>Pour adolescents ?</b>	Oui	Pas assez violent...	

# Créer des formulaires

- ▶ A partir de la balise <form>, il est possible de créer des formulaires en HTML permettant de saisir du texte, sélectionner des options, créer des boutons de validations etc. Mais ces informations ne pourront pas être traitées en HTML il vaudra faire appel à un langage permettant d'interagir avec le visiteur du site tel le PHP. En bref le HTML ne permettra que de mettre en forme vos formulaires, c'est la limite du HTML.
- ▶ Supposons que votre visiteur vienne de taper un commentaire dans votre formulaire, par exemple un message qu'il aimerait publier sur vos forums. Ce message doit être *envoyé* pour que vous puissiez le recevoir et l'afficher pour vos autres visiteurs. On est donc face à deux problèmes.  
**Problème n°1** : comment envoyer le texte saisi par le visiteur ? Par quel moyen ? **Problème n°2** : une fois que les données ont été envoyées, comment les traiter ? Souhaitez-vous recevoir le message automatiquement par mail ou préférez-vous qu'un programme se charge de l'enregistrer quelque part, puis de l'afficher sur une page visible par tout le monde ?

# Créer des formulaires

- ▶ Pour répondre à ces deux problèmes, vous devez ajouter deux attributs à la balise <form>:
- ▶ **method** : cet attribut indique par quel moyen les données vont être envoyées (réponse au **problème n°1**). Il existe deux solutions pour envoyer des données sur le Web :
  - ▶ `method="get"`: c'est une méthode en général assez peu adaptée car elle est limitée à 255 caractères. La particularité vient du fait que les informations seront envoyées dans l'adresse de la page (`http://...`), mais ce détail ne nous intéresse pas vraiment pour le moment. La plupart du temps, je vous recommande d'utiliser l'autre méthode : `post`.
  - ▶ `method="post"`: c'est la méthode la plus utilisée pour les formulaires car elle permet d'envoyer un grand nombre d'informations. Les données saisies dans le formulaire ne transitent pas par la barre d'adresse.
- ▶ **action** : c'est l'adresse de la page ou du programme qui va *traiter* les informations (réponse au **problème n°2**). Cette page se chargera de vous envoyer un e-mail avec le message si c'est ce que vous voulez, ou bien d'enregistrer le message avec tous les autres dans une base de données.

Cela ne peut pas se faire en HTML et CSS, on utilisera en général un autre langage dont vous avez peut-être entendu parler : PHP.

```
<form method="post" action="traitement.php">  
  <p>Texte à l'intérieur du formulaire</p>  
</form>
```

Pour le moment, on ne sait pas ce qu'il se passe à l'intérieur de la page `traitement.php`. Notre priorité, pour le moment, est de découvrir en HTML/CSS comment faire pour insérer des zones de texte, des boutons et des cases à cocher dans votre page web. C'est ce que nous allons voir maintenant.

# Créer des formulaires

- Les zones de saisie basiques
- Il existe deux zones de texte différentes :
- **La zone de texte monoligne** : comme son nom l'indique, on ne peut y écrire qu'une seule ligne. Elle sert à saisir des textes courts, par exemple un pseudo.
- **La zone de texte multiligne** : cette zone de texte permet d'écrire une quantité importante de texte sur plusieurs lignes, par exemple une dissertation sur l'utilité du HTML dans le développement des pays d'Asie du Sud-Est (ce n'est qu'une suggestion hein...).
- **Zone monoligne**
- Pour insérer une zone de texte dans une ligne, on va utiliser la balise `<input>` avec l'attribut « type ». Ici on veut une zone de texte, on écrira donc : `<input type="text" />`. Pour finir on donne un nom à cette balise par l'attribut « name ». S'il s'agit d'enregistrer son pseudo, on écrira : `<input type="text" name="pseudo" />`.

```
<form method="post" action="traitement.php">
  <p><input type="text" name="pseudo" /></p>
</form>
```

- Rajouter des libellés pour préciser ce que le visiteur doit renseigner dans la balise `<form>` grâce à la balise `<label>`. On doit ensuite lier le label grâce à l'attribut « for » à la zone de texte que l'on identifie par un « id » (unique). On aura alors :

```
<form method="post" action="traitement.php">
  <p>
    <label for="pseudo">Votre pseudo</label> : <input type="text" name="pseudo" id="pseudo" />
  </p>
</form>
```

# Créer des formulaires

## ▀ Quelques attributs supplémentaires

- ▀ On peut ajouter un certain nombre d'autres attributs à la balise `<input />` pour personnaliser son fonctionnement :
- ▀ On peut agrandir le champ avec **size**.
- ▀ On peut limiter le nombre de caractères que l'on peut saisir avec **maxlength**.
- ▀ On peut pré-remplir le champ avec une valeur par défaut à l'aide de **value**.
- ▀ On peut donner une indication sur le contenu du champ avec **placeholder**. Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.
- ▀ Dans l'exemple suivant, la zone de texte contient une indication permettant de comprendre ce qu'il faut saisir ; le champ fait 30 caractères de long mais on ne peut écrire que 10 caractères maximum à l'intérieur :

```
<form method="post" action="traitement.php">
  <p>
    <label for="pseudo">Votre pseudo :</label>
    <input type="text" name="pseudo" id="pseudo" placeholder="Ex : Zozor" size="30" maxlength="10" />
  </p>
</form>
```

- ▀ De la même manière on peut rajouter une zone de mot de passe (zone où l'on ne voit pas les caractères saisis) en modifiant le nom de l'attribut « type » par `password`.

```
<form method="post" action="traitement.php">
  <p>
    <label for="pseudo">Votre pseudo :</label>
    <input type="text" name="pseudo" id="pseudo" />

    <br />
    <label for="pass">Votre mot de passe :</label>
    <input type="password" name="pass" id="pass" />

  </p>
</form>
```

# Créer des formulaires

## ■ Zone de texte multiligne

- Pour créer une zone de texte multiligne, on change de balise `<input/>` par la balise `<textarea> </textarea>`.

```
<form method="post" action="traitement.php">
  <p>
    <label for="message">Laissez moi un message</label><br />
    <textarea name="message" id="message"></textarea>
  </p>
</form>
```

- Il est possible de modifier la taille du champ en css grâce aux propriétés `width` et `height` appliqués à la balise `<textarea>` ou en ajoutant directement dans le html les attributs `rows` (indique le nombre de lignes) et `cols` (indique le nombre de colonnes) à la balise `<textarea>`

```
<form method="post" action="traitement.php">
  <p>
    <label for="message">
      Laissez moi un message
    </label>
    <br />
    <textarea name="message" id="message" rows="10" cols="50"></textarea>
  </p>
</form>
```

# Créer des formulaires

## ► Les zones de saisie enrichies

► Tous les types de champs sont modifiables en changeant la valeur de l'attribut « type » dans la balise `<input />`

► **E-mail** : `<input type="email" />`

► **URL** : `<input type="url" />`

► **Numéro de tel** : `<input type="tel" />`

► **Nombre** : `<input type="number" />`

► Vous pouvez personnaliser le fonctionnement du champ avec les attributs suivants :

► *min*: valeur minimale autorisée.

► *max*: valeur maximale autorisée.

► *step*: c'est le « pas » de déplacement. Si vous indiquez un pas de 2, le champ n'acceptera que des valeurs de 2 en 2 (par exemple 0, 2, 4, 6...).

► **Curseur** : `<input type="range" />`



► **Choisir une couleur** : `<input type="color" />`

► **Date** : `<input type="date" />` (time pour l'heure (13:37 par exemple) ; week pour la semaine ; month pour le mois ; datetime pour la date et l'heure avec gestion du décalage horaire ; datetime-local pour la date et l'heure sans gestion du décalage horaire.

► **Champ de recherche** : `<input type="search" />`

# Créer des formulaires

## ► Les élément d'option

- Les cases à cocher `<input type="checkbox" name="choix" />` Remarque : il est possible avec l'attribut `checked` qu'une case soit cochée par défaut `<input type="checkbox" name="choix" checked />`

```
<form method="post" action="traitement.php">
  <p>
    Cochez les aliments que vous aimez manger :<br />
    <input type="checkbox" name="frites" id="frites" /> <label for="frites">Frites</label><br />
    <input type="checkbox" name="steak" id="steak" /> <label for="steak">Steak haché</label><br />
    <input type="checkbox" name="epinards" id="epinards" /> <label for="epinards">Epinards</label><br />
    <input type="checkbox" name="huitres" id="huitres" /> <label for="huitres">Huitres</label>
  </p>
</form>
```

Cochez les aliments que vous aimez manger :

- Frites
- Steak haché
- Epinards
- Huitres

# Créer des formulaires

- ▶ **Les élément d'option**
- ▶ Les zones d'options permettent de faire un choix (et un seul) parmi une liste de possibilités. Elles ressemblent un peu aux cases à cocher mais il y a une petite difficulté supplémentaire : elles doivent être organisées en groupes. Les options d'un même groupe possèdent le même nom (name), mais chaque option doit avoir une valeur (value) différente.
- ▶ La balise à utiliser est toujours un `<input />`, avec cette fois la valeur radio pour l'attribut type.

Cochez les aliments que vous aimez manger :

- Frites
- Steak haché
- Epinards
- Huitres

# Créer des formulaires

## ➤ Les élément d'option

- Zone d'option (la valeur de l'attribut name doit être la même dans chaque balise <input/>)

```
<form method="post" action="traitement.php">
  <p>
    Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :<br />
    <input type="radio" name="age" value="moins15" id="moins15" /> <label for="moins15">Moins de 15 ans</label><br />
    <input type="radio" name="age" value="medium15-25" id="medium15-25" /> <label for="medium15-25">15-25 ans</label><br />
    <input type="radio" name="age" value="medium25-40" id="medium25-40" /> <label for="medium25-40">25-40 ans</label><br />
    <input type="radio" name="age" value="plus40" id="plus40" /> <label for="plus40">Encore plus vieux que ça ?!</label>
  </p>
</form>
```

**Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :**

- Moins de 15 ans
- 15-25 ans
- 25-40 ans
- Encore plus vieux que ça ?!

# Créer des formulaires

## ► Les élément d'option

- Les listes déroulantes, elles permettent de faire un choix parmi plusieurs possibilités. On va utiliser la balise `<select>` `</select>` qui indique le début et la fin de la liste déroulante. On ajoute l'attribut *name* à la balise pour donner un nom à la liste. Puis, à l'intérieur du `<select>` `</select>`, nous allons placer plusieurs balises `<option>` `</option>` (une par choix possible). On ajoute à chacune d'elles un attribut *value* pour pouvoir identifier ce que le visiteur a choisi.

```
<form method="post" action="traitement.php">
  <p>
    <label for="pays">Dans quel pays habitez-vous ?</label><br />
    <select name="pays" id="pays">
      <option value="france">France</option>
      <option value="espagne">Espagne</option>
      <option value="italie">Italie</option>
      <option value="royaume-uni">Royaume-Uni</option>
      <option value="canada">Canada</option>
      <option value="etats-unis">États-Unis</option>
      <option value="chine">Chine</option>
      <option value="japon">Japon</option>
    </select>
  </p>
</form>
```

Dans quel pays habitez-vous ?



A screenshot of a web form element. It shows a label "Dans quel pays habitez-vous ?" followed by a dropdown menu. The dropdown menu is currently open, showing a list of countries: Espagne, France, Italie, Royaume-Uni, Canada, États-Unis, Chine, and Japon. The "Espagne" option is highlighted in blue, indicating it is the selected value.

# Créer des formulaires

## ► Les élément d'option

- Les listes déroulantes. Vous pouvez aussi grouper vos options avec la balise <optgroup> </optgroup>. Dans notre exemple, pourquoi ne pas séparer les pays en fonction de leur continent ?

```
<form method="post" action="traitement.php">
  <p>
    <label for="pays">Dans quel pays habitez-vous ?</label><br />
    <select name="pays" id="pays">
      <optgroup label="Europe">
        <option value="france">France</option>
        <option value="espagne">Espagne</option>
        <option value="italie">Italie</option>
        <option value="royaume-uni">Royaume-Uni</option>
      </optgroup>
      <optgroup label="Amérique">
        <option value="canada">Canada</option>
        <option value="etats-unis">Etats-Unis</option>
      </optgroup>
      <optgroup label="Asie">
        <option value="chine">Chine</option>
        <option value="japon">Japon</option>
      </optgroup>
    </select>
  </p>
</form>
```

Dans quel pays habitez-vous ?



Espagne	▼
<b>Europe</b>	
France	
Espagne	
Italie	
Royaume-Uni	
<b>Amérique</b>	
Canada	
Etats-Unis	
<b>Asie</b>	
Chine	
Japon	



# Créer des formulaires

- **Finaliser et envoyer le formulaire**

- Regrouper les champs : Si votre formulaire grossit et comporte beaucoup de champs, il peut être utile de les regrouper au sein de plusieurs balises `<fieldset>`. Chaque `<fieldset>` peut contenir une légende avec la balise `<legend>`

# Créer des formulaires

- Regrouper les champs :

```
<form method="post" action="traitement.php">
  <fieldset>
    <legend>Vos coordonnées</legend> <!-- Titre du fieldset -->
    <label for="nom">Quel est votre nom ?</label>
    <input type="text" name="nom" id="nom" />
    <label for="prenom">Quel est votre prénom ?</label>
    <input type="text" name="prenom" id="prenom" />
    <label for="email">Quel est votre e-mail ?</label>
    <input type="email" name="email" id="email" />
  </fieldset>
  <fieldset>
    <legend>Votre souhait</legend> <!-- Titre du fieldset -->
    <p>
      Faites un souhait que vous voudriez voir exaucé :
      <input type="radio" name="souhait" value="riche" id="riche" /> <label for="riche">Etre riche</label>
      <input type="radio" name="souhait" value="celebre" id="celebre" /> <label for="celebre">Etre célèbre</label>
      <input type="radio" name="souhait" value="intelligent" id="intelligent" /> <label for="intelligent">Etre
      <strong>encore</strong> plus intelligent</label>
      <input type="radio" name="souhait" value="autre" id="autre" /> <label for="autre">Autre...</label>
    </p>
    <p>
      <label for="precisions">Si "Autre", veuillez préciser :</label>
      <textarea name="precisions" id="precisions" cols="40" rows="4"></textarea>
    </p>
  </fieldset>
</form>
```

# Créer des formulaires

- Regrouper les champs :

Vos coordonnées

Quel est votre nom ?

Quel est votre prénom ?

Quel est votre e-mail ?

Votre souhait

Faites un souhait que vous voudriez voir exaucé :

- Etre riche
- Etre célèbre
- Etre **encore** plus intelligent
- Autre...

# Créer des formulaires

- ▶ **Sélectionner automatiquement un champ.** Vous pouvez placer automatiquement le curseur dans l'un des champs de votre formulaire avec l'attribut *autofocus* :
  - `<input type="text" name="prenom" id="prenom" autofocus />`
- ▶ **Rendre un champ obligatoire.** Vous pouvez faire en sorte qu'un champ soit obligatoire en lui donnant l'attribut *required*.
  - `<input type="text" name="prenom" id="prenom" required />`
  - On dispose de pseudo-formats en CSS pour changer le style des éléments requis (:required) et invalides (:invalid). N'oubliez pas non plus que vous disposez du pseudo-format :focus pour changer l'apparence d'un champ lorsque le curseur se trouve à l'intérieur.

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

# Créer des formulaires

- ▶ **Le bouton d'envoi.** Il ne nous reste plus qu'à créer le bouton d'envoi. Là encore, la balise `<input />` vient à notre secours. Elle existe en quatre versions :
  - `type="submit"` : le principal bouton d'envoi de formulaire. C'est celui que vous utiliserez le plus souvent. Le visiteur sera conduit à la page indiquée dans l'attribut `action` du formulaire.
  - `type="reset"` : remise à zéro du formulaire.
  - `type="image"` : équivalent du bouton submit, présenté cette fois sous forme d'image. Rajoutez l'attribut `src` pour indiquer l'URL de l'image.
  - `type="button"` : bouton générique, qui n'aura (par défaut) aucun effet. En général, ce bouton est géré en JavaScript pour exécuter des actions sur la page. Nous ne l'utiliserons pas ici.
- ▶ On peut changer le texte affiché à l'intérieur des boutons avec l'attribut `value`.
  - `<input type="submit" value="Envoyer" />`
- ▶ Lorsque vous cliquez sur le bouton « Envoyer », le formulaire vous amène alors à la page indiquée dans l'attribut `action`. Celle que nous avons imaginé, la page fictive : `traitement.php`. Ainsi pour récupérer et traiter ces informations il faudra apprendre le langage PHP.



# Insérer de la vidéo et de l'audio

## ➤ Les formats audio

- **MP3** : vous ne pouvez pas ne pas en avoir entendu parler ! C'est l'un des plus vieux, mais aussi l'un des plus compatibles (tous les appareils savent lire des MP3), ce qui fait qu'il est toujours très utilisé aujourd'hui.
- **AAC** : utilisé majoritairement par Apple sur iTunes, c'est un format de bonne qualité. Les iPod, iPhone et autres iPad savent les lire sans problème.
- **OGG** : le format Ogg Vorbis est très répandu dans le monde du logiciel libre, notamment sous Linux. Ce format a l'avantage d'être libre, c'est-à-dire qu'il n'est protégé par aucun brevet.
- **WAV (format non compressé)** : évitez autant que possible de l'utiliser car le fichier est très volumineux avec ce format. C'est un peu l'équivalent du Bitmap (BMP) pour l'audio.

# Insérer de la vidéo et de l'audio

## ► Les formats vidéo

- Le stockage de la vidéo est autrement plus complexe. On a besoin de trois éléments :
- **Un format conteneur** : c'est un peu comme une boîte qui va servir à contenir les deux éléments ci-dessous. On reconnaît en général le type de conteneur à l'extension du fichier : AVI, MP4, MKV...
- **Un codec audio** : c'est le format du son de la vidéo, généralement compressé. Nous venons de les voir, on utilise les mêmes : MP3, AAC, OGG...
- **Un codec vidéo** : c'est le format qui va compresser les images. C'est là que les choses se corsent, car ces formats sont complexes et on ne peut pas toujours les utiliser gratuitement. Les principaux à connaître pour le Web sont :
  - **H.264** : l'un des plus puissants et des plus utilisés aujourd'hui... mais il n'est pas 100% gratuit. En fait, on peut l'utiliser gratuitement dans certains cas (comme la diffusion de vidéos sur un site web personnel), mais il y a un flou juridique qui fait qu'il est risqué de l'utiliser à tout va.
  - **Ogg Theora** : un codec gratuit et libre de droits, mais moins puissant que H.264. Il est bien reconnu sous Linux mais, sous Windows, il faut installer des programmes pour pouvoir le lire.
  - **WebM** : un autre codec gratuit et libre de droits, plus récent. Proposé par Google, c'est le concurrent le plus sérieux de H.264 à l'heure actuelle.

# Insérer de la vidéo et de l'audio

## ► Insertion d'un élément audio

► `<audio src="musique.mp3"></audio>`

► Vous pouvez compléter la balise des attributs suivants :

► *controls*: pour ajouter les boutons « Lecture », « Pause » et la barre de défilement.

► *width*: pour modifier la largeur de l'outil de lecture audio.

► *loop*: la musique sera jouée en boucle.

► *autoplay*: la musique sera jouée dès le chargement de la page. Évitez d'en abuser, c'est en général irritant d'arriver sur un site qui joue de la musique tout seul !

► *preload*: indique si la musique peut être préchargée dès le chargement de la page ou non. Cet attribut peut prendre les valeurs :

► *auto*(par défaut) : le navigateur décide s'il doit précharger toute la musique, uniquement les métadonnées ou rien du tout.

► *metadata*: charge uniquement les métadonnées (durée, etc.).

► *none*: pas de préchargement. Utile si vous ne voulez pas gaspiller de bande passante sur votre site.

► `<audio src="musique.mp3" controls></audio>`

► Pour proposer plusieurs versions du même fichier :

```
<audio controls>
  <source src="hype_home.mp3">
  <source src="hype_home.ogg">
</audio>
```

# Insérer de la vidéo et de l'audio

## ► Insertion d'une vidéo

► `<video src="sintel.webm"></video>`

► Vous pouvez compléter la balise des attributs suivants :

- `poster`: image à afficher à la place de la vidéo tant que celle-ci n'est pas lancée. Par défaut, le navigateur prend la première image de la vidéo mais, comme il s'agit souvent d'une image noire, vous pouvez tout simplement faire une capture d'écran d'un moment de la vidéo.
- `controls`: pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript. En ce qui nous concerne, ce sera largement suffisant !
- `width`: pour modifier la largeur de la vidéo.
- `height`: pour modifier la hauteur de la vidéo.
- `loop`: la vidéo sera jouée en boucle.
- `autoplay`: la vidéo sera jouée dès le chargement de la page. Là encore, évitez d'en abuser, c'est en général irritant d'arriver sur un site qui lance quelque chose tout seul !
- `preload`: indique si la vidéo peut être préchargée dès le chargement de la page ou non.
- `<video src="film.webm" controls poster="sintel.jpg" width="600"></video>`

# Insérer de la vidéo et de l'audio

## ► Insertion d'une vidéo

► `<video src="sintel.webm"></video>`

► Vous pouvez compléter la balise des attributs suivants :

- `poster`: image à afficher à la place de la vidéo tant que celle-ci n'est pas lancée. Par défaut, le navigateur prend la première image de la vidéo mais, comme il s'agit souvent d'une image noire, vous pouvez tout simplement faire une capture d'écran d'un moment de la vidéo.
- `controls`: pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript. En ce qui nous concerne, ce sera largement suffisant !
- `width`: pour modifier la largeur de la vidéo.
- `height`: pour modifier la hauteur de la vidéo.
- `loop`: la vidéo sera jouée en boucle.
- `autoplay`: la vidéo sera jouée dès le chargement de la page. Là encore, évitez d'en abuser, c'est en général irritant d'arriver sur un site qui lance quelque chose tout seul !
- `preload`: indique si la vidéo peut être préchargée dès le chargement de la page ou non.
- `<video src="film.webm" controls poster="sintel.jpg" width="600"></video>`

# Insérer de la vidéo et de l'audio

## ► Insertion d'une vidéo

- Comme avec l'audio (mais je ne l'avais pas précisé), il est possible d'afficher un message entre les balises `<video></video>` qui s'affiche si le navigateur ne reconnaît pas les balises.

```
<video src="sintel.webm" controls poster="sintel.jpg" width="600">  
  Il est temps de mettre à jour votre navigateur !  
</video>
```

- Vous utiliserez la balise `<source>` à l'intérieur de la balise `<video>` pour proposer différents formats. Le navigateur prendra celui qu'il reconnaît :

```
<video controls poster="sintel.jpg" width="600">  
  <source src="sintel.mp4">  
  <source src="sintel.webm">  
  <source src="sintel.ogv">  
</video>
```



# Utilisez le responsive design avec les Media Queries

- ▶ Les media queries sont des règles à appliquer pour changer le design d'un site en fonction des caractéristiques de l'écran ! Grâce à cette technique, nous pourrions créer un design qui s'adapte automatiquement à l'écran de chaque visiteur !
- ▶ Les media queries font partie des nouveautés de CSS3. Il ne s'agit pas de nouvelles propriétés mais de *règles* que l'on peut appliquer dans certaines conditions. Concrètement, vous allez pouvoir dire « Si la résolution de l'écran du visiteur est inférieure à tant, alors applique les propriétés CSS suivantes ». Cela vous permet de changer l'apparence du site dans certaines conditions : vous pourrez augmenter la taille du texte, changer la couleur de fond, positionner différemment votre menu dans certaines résolutions, etc.

# Utilisez le responsive design avec les Media Queries

- **Appliquer une media query**
- **Chargement d'une feuille de style différente.** On peut lui ajouter un attribut *media*, dans lequel on va écrire la règle qui doit s'appliquer pour que le fichier soit chargé :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="style.css" /> <!-- Pour tout le monde -->
    <link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" /> <!-- Pour ceux qui ont une
    résolution inférieure à 1280px -->
    <title>Media queries</title>
  </head>
```

- **Chargement des règles directement dans la feuille de style.** Technique plus pratique qui consiste à écrire ces règles dans le même fichier css que d'habitude. On écrit la règle dans le fichier :

```
@media screen and (max-width: 1280px)
{
  /* Rédigez vos propriétés CSS ici */
}
```



# Utilisez le responsive design avec les Media Queries

## ➤ Les règles principales disponibles

- color: gestion de la couleur (en bits/pixel).
- height: hauteur de la zone d'affichage (fenêtre).
- width: largeur de la zone d'affichage (fenêtre).
- device-height: hauteur du périphérique.
- device-width: largeur du périphérique.
- orientation: orientation du périphérique (portrait ou paysage).
- media: type d'écran de sortie. Quelques-unes des valeurs possibles :
  - screen: écran « classique » ;
  - handheld: périphérique mobile mais qui n'est reconnu que par Opera mobile;
  - print: impression ;
  - tv: télévision ;
  - projection: projecteur ;
  - all: tous les types d'écran.

# Utilisez le responsive design avec les Media Queries

- ▶ On peut rajouter le préfixe min- ou max- devant la plupart de ces règles. Ainsi, min-width signifie « Largeur minimale », max-height « Hauteur maximale », etc.  
La différence entre width et device-width se perçoit surtout sur les navigateurs mobiles des smartphones, nous en reparlerons plus loin.
- ▶ Les règles peuvent être combinées à l'aide des mots suivants :
- ▶ only: « uniquement » ;
- ▶ and: « et » ;
- ▶ not: « non ».
- ▶ Voici quelques exemples de media queries pour vous aider à bien comprendre le principe :

```
/* Sur les écrans, quand la largeur de la fenêtre fait au maximum 1280px */  
@media screen and (max-width: 1280px)
```

```
/* Sur tous types d'écran, quand la largeur de la fenêtre est comprise entre 1024px et 1280px */  
@media all and (min-width: 1024px) and (max-width: 1280px)
```

```
/* Sur les téléviseurs */  
@media tv
```

```
/* Sur tous types d'écrans orientés verticalement */  
@media all and (orientation: portrait)
```

# Utilisez le responsive design avec les Media Queries

## ► Media queries et navigateurs mobiles

- Comme vous le savez sûrement, les écrans des smartphones sont beaucoup moins larges que nos écrans habituels (seulement quelques centaines de pixels de large). Pour s'adapter, les navigateurs mobiles affichent le site en « dézoomant », ce qui permet d'avoir un aperçu de l'ensemble de la page. La zone d'affichage simulée est appelée le **viewport** : c'est la largeur de la fenêtre du navigateur sur le mobile.
- En CSS, avec les media queries, si vous ciblez l'écran avec `max-width` sur un mobile, celui-ci va comparer la largeur que vous indiquez avec celle de son viewport. Le problème, c'est que le viewport change selon le navigateur mobile utilisé ! Opera Mobile : 850 px, iPhone Safari : 980px; Android : 800px ; windows Phone : 1024px.
- Pour cibler les smartphones, plutôt que d'utiliser `max-width`, il peut être intéressant de recourir à `max-device-width` : c'est la largeur du périphérique. Les périphériques mobiles ne dépassant pas 480 px de large, on pourra viser uniquement les navigateurs mobiles avec cette media query :  
`@media all and (max-device-width: 480px)`
- Vous pouvez modifier la largeur viewport du navigateur mobile avec une balise `meta` à insérer dans l'en-tête (`<head>`) du document : `<meta name="viewport" content="width=320" />`
- Vous pouvez utiliser cette balise pour modifier la façon dont le contenu de votre page s'organise sur les mobiles. Pour obtenir un rendu facile à lire, sans zoom, vous pouvez demander à ce que le viewport soit le même que la largeur de l'écran : `<meta name="viewport" content="width=device-width" />`

# Memento Balises HTML

- Balises de premier niveau qui structurent une page HTML et en constituent le code minimal.

Balise	Description
<html>	Balise principale
<head>	En-tête de la page
<body>	Corps de la page

# Memento Balises HTML

## ► Balises d'en-tête

Balises	Description
<link/>	Liaison avec une feuille de style
<meta/>	Métadonnées de la page web (charset, mots-clés, etc.)
<script>	Code JavaScript
<style>	Code CSS
<title>	Titre de la page

# Memento Balises HTML

- Balise de structuration du texte

Balise	Description
<abbr>	Abréviation
<blockquote>	Citation (longue)
<cite>	Citation du titre d'une œuvre ou d'un évènement
<q>	Citation (courte)
<sup>	Exposant
<sub>	Indice
<strong>	Mise en valeur forte
<em>	Mise en valeur normale

# Memento Balises HTML

Balise	Description
<mark>	Mise en valeur visuelle
<h1> <h2> <h3> <h4> <h5> <h6>	Titre selon leur niveau d'importance
<img/>	Image
<figure>	Figure (image, code, etc.)
<figcaption>	Description de la figure
<audio>	Son
<video>	Vidéo
<source>	Format source pour les balises<audio> et <video>
<a>	Lien hypertexte

# Memento Balises HTML

Balise	Description
 	Retour à la ligne
<p>	Paragraphe
<hr/>	Ligne de séparation horizontale
<adress>	Adresse de contact
<del>	Texte supprimé
<ins>	Texte inséré
<dfn>	Définition
<kbd>	Saisie clavier
<pre>	Affichage formaté (pour les codes sources)
<progress>	Barre de progression
<time>	Date ou heure

# Memento Balises HTML

## ► Balises de listes

Balises	Description
<ul>	Liste à puces, non numérotée
<ol>	Liste numérotée
<li>	Élément de la liste à puces
<dl>	Liste de définitions
<dt>	Terme à définir
<dd>	Définition du terme

# Memento Balises HTML

## ► Balises de Tableaux

Balise	Description
<table>	Tableau
<caption>	Titre du tableau
<tr>	Ligne de tableau
<th>	Cellule d'en-tête
<td>	Cellule
<thead>	Section de l'en-tête du tableau
<tbody>	Section du corps du tableau
<tfoot>	Section du pied du tableau

# Memento Balises HTML

## ► Balises de formulaires

Balise	Description
<form>	Formulaire
<fieldset>	Groupe de champs
<legend>	Titre d'un groupe de champs
<label>	Libellé d'un champ
<input />	Champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.)
<textarea>	Zone de saisie multiligne
<select>	Liste déroulante
<option>	Élément d'une liste déroulante
<optgroup>	Groupe d'éléments d'une liste déroulante

# Memento Balises HTML

- ▶ Balises sectionnantes permettant de construire l'architecture de notre site.

Balise	Description
<header>	En-tête
<nav>	Liens principaux de navigation
<footer>	Pied de page
<section>	Section de page
<article>	Article (contenu autonome)
<aside>	Informations complémentaires

# Memento Balises HTML

- ▶ Balises génériques ou universelles. Elles n'ont pas de sens sémantique.
- ▶ Ces balises n'ont d'intérêt que si on leur associe un attribut class, id ou style.
  - **class** : indique le nom de la classe à utiliser. Plusieurs balises peuvent avoir le même nom de classe.
  - **id** : donne un nom unique à la balise permettant de l'identifier. L'attribut id est utile pour les ancres, l'identification d'une modification CSS ou pour des manipulations JavaScript.
  - **style** : permet d'indiquer directement le style CSS à appliquer dans la balise HTML.

Balise	Description
<span>	Balise générique de type inline
<div>	Balise générique de type block

# Memento CSS

## ▀ Typographie

Propriété	Valeur	Description
font-family	police1, police2, police3, serif, sans-serif, monospace	Nom de police
@font-face	@font-face { font-family: 'FontAwesome'; src: url('fonts/fontawesome-webfont.woff2'); }	Police personnalisée
font-size	1.3em, 16px, 120%	Taille du texte
font-weight	italic, oblique, normal	Italique...
text-decoration	underline, overline, line-through, blink, none	Soulignement, ligne au-dessus, barré ou clignotant

# Memento CSS

## ▀ Typographie

Propriété	Valeur	Description
font-variant	small-caps, normal	Petites capitales
text-transform	capitalize, lowercase, uppercase	Capitales
font	font: italic small-caps bolder 16px/3 cursive;	Super propriété de police. Combine : font-weight, font-style, font-size, font-variant, font-family...
text-align	left, center, right, justify	Alignement horizontal
vertical-align		Alignement vertical (cellules de tableau ou éléments inline-block uniquement)

# Memento CSS

## ▀ Typographie

Propriété	Valeur	Description
line-height	18px, 120%, normal...	Hauteur de ligne
text-indent	25px	Alinéa
white-space	pre, nowrap, normal	Césure
word-wrap	break-word, normal	Césure forcée
text-shadow	5px 5px 2px blue <i>(horizontale, verticale, fondu, couleur)</i>	Ombre de texte

# Memento CSS

## ▀ Couleurs et fonds

Propriété	Valeur	Description
color	#FFF rgb(255, 255, 255) rgba(255, 255, 255, 0.5)	Exemple pour le blanc. la valeur a est l'opacité.
background-color	id color	Couleur de fond
background-image	url(localisation du fichier image)	Image de fond
background-attachment	fixed, scroll	Fixe ou défilant
background-repeat	repeat-x, repeat-y, repeat, no-repeat	Répétition du fond

# Memento CSS

## ■ Couleurs et fonds

Propriété	Valeur	Description
background-position	background-position: (top, bottom, left, right, center) (25% 75%) (10px 20px);	Position du fond
background		Super propriété de fond

# Memento CSS

## Propriétés des boîtes

Propriété	Valeur	Description
width	150px, 80%;	Largeur
height	150px, 80%;	Hauteur
min-width	150px, 80%;	Largeur minimale
max-width	150px, 80%;	Largeur maximale
min-height	150px, 80%;	Hauteur minimale
max-height	150px, 80%;	Hauteur maximale

# Memento CSS

## Propriétés des boîtes

Propriété	Valeur	Description
margin-top	10px, 8%;	Marge en haut
margin-left	10px, 8%;	Marge à gauche
margin-right	10px, 8%;	Marge à droite
margin-bottom	10px, 8%;	Marge en bas
margin	23 px 5px 23px 5px (haut, droite, bas, gauche)	Super-propriété de marge. Combine : margin- top, margin-right, margin-bottom, margin-left.

# Memento CSS

## Propriétés des boîtes

Propriété	Valeur	Description
padding-top	10px, 8%;	Marge intérieure en haut
padding-left	10px, 8%;	Marge intérieure à gauche
padding-right	10px, 8%;	Marge intérieure à droite
padding-bottom	10px, 8%;	Marge intérieure en bas
padding	23 px 5px 23px 5px (haut, droite, bas, gauche)	Combine : padding-top, padding-right, padding-bottom, padding-left.

# Memento CSS

## ► Propriétés des boîtes

Propriété	Valeur	Description
border-width	10px;	Épaisseur de bordure
border-color	<i>pink</i> , rgb(rouge,vert,bleu),r gba(rouge,vert,bleu,transparence), #CF1A20...	Couleur de bordure
border-style	solid, dotted, dashed, double, groove, ridge, inset, outset;	Type de bordure

# Memento CSS

## Propriétés des boîtes

Propriété	Valeur	Description
border	3px solid black;	Super-propriété de bordure. Combine border-width, border-color, border-style. Existe aussi en version border-top, border-right, border-bottom, border-left.
border-radius	5px; 5px 10px 5px 10px;	Bordure arrondie
box-shadow	6px 6px 0px black; <i>(horizontale, verticale, fondu, couleur)</i>	Ombre de boîte

# Memento CSS

## ► Propriétés des boîtes

Propriété	Valeur	Description
background-clip	border-box; padding-box; content-box; text;	Limitation du fond dans la boîte. Sous la bordure, dans le contenu y compris le padding, dans le contenu, dans le texte.

# Memento CSS

## ► Propriétés de positionnement et d'affichage

Propriété	Valeur	Description
display	block, inline, inline-block, table, table-cell, flex, none;	Type d'élément (block, inline, inline-block, flex, none...)
visibility	visible, hidden;	Visibilité
clip	rect (0px, 60px, 30px, 0px);	Affichage d'une partie de l'élément
overflow	auto, scroll, visible, hidden;	Comportement en cas de dépassement

# Memento CSS

## Propriétés de positionnement et d'affichage

Propriété	Valeur	Description
float	left, right, none;	Flottant
clear	left, right, both, none;	Arrêt d'un flottant
position	relative, absolute, static;	Positionnement
top, bottom, left, right	20px;	Position par rapport au haut, au bas, la gauche, la droite
z-index	10;	Ordre d'affichage en cas de superposition. La plus grande valeur est affichée par-dessus les autres.

# Memento CSS

## Propriétés des listes

Propriété	Valeur	Description
list-style-type	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none;	Type de liste
list-style-position	inside, outside;	Position en retrait
list-style-image	url('puce.png');	Puce personnalisée
list-style		Super-propriété de liste. Combine list-style-type, list-style-position, list-style-image.

# Memento CSS

## ▀ Propriétés des tableaux

Propriété	Valeur	Description
border-collapse	collapse, separate;	Fusion des bordures
empty-cells	hide, show;	Affichage des cellules vides
caption-side	bottom, top;	Position du titre du tableau