

# Sommaire

Comment utiliser HTML5 et CSS3, les deux langages de programmation à la base de tous les sites web.

## Partie 1 - Les bases de HTML5

1. Comment fait-on pour créer des sites Web ?
2. Votre première page web en HTML
3. Organiser son texte
4. Pratiquez !
5. Créer des liens
6. Les images

Quiz

Activité : Structurer son CV

## Partie 2 - Les joies de la mise en forme avec CSS

1. Mettre en place le CSS
2. Formatage du texte
3. La couleur et le fond
4. Les bordure et les ombres
5. Création d'apparences dynamiques

Quiz

Activité : Mettre en forme son CV

## Partie 3 - Mise en page du site

1. Structurer sa page
2. Le modèle des boîtes
3. La mise en page avec Flexbox
4. Quelques autres techniques de mise en page
5. TP: création d'un site pas à pas

Quiz

Activité : Organiser son CV

## Partie 4 - Fonctionnalités évoluées

1. Les tableaux
2. Les formulaires
3. La vidéo et l'audio
4. Le responsive design avec les Media Queries
5. Aller plus loin

Quiz

Activité : Adapter son CV en responsive design

## Partie 5 - Annexes

1. Envoyez votre CV sur le Web
2. Mémento des balise HTML
3. Mémento des propriétés CSS

# Partie 1 - Les bases de HTML5

## Comment fait-on pour créer des sites Web ?

Le fonctionnement des sites web

HTML et CSS : deux langages pour créer un site web

L'éditeur de texte

Les navigateurs

### En résumé

- Le Web a été inventé par Tim Berners-Lee au début des années 1990.
- Pour créer des sites web, on utilise deux langages informatiques :
- HTML : permet d'écrire et organiser le contenu de la page (paragraphes, titres...) ;
- CSS : permet de mettre en forme la page (couleur, taille...).
- Il y a eu plusieurs versions des langages HTML et CSS. Les dernières versions sont HTML5 et CSS3.
- Le navigateur web est un programme qui permet d'afficher des sites web. Il lit les langages HTML et CSS pour savoir ce qu'il doit afficher.
- Il existe de nombreux navigateurs web différents : Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera... Chacun affiche un site web de manière légèrement différente des autres navigateurs.
- Dans ce cours, nous allons apprendre à utiliser les langages HTML et CSS. Nous travaillerons dans un programme appelé « éditeur de texte » (Sublime Text, Notepad++, jEdit, vim...).

# Votre première page web en HTML

Créer une page web avec l'éditeur

Les balises et leurs attributs

Les balises

**<balise>**

Les balises en paires

**<title>Ceci est un titre</title>**

Les balises orphelines

**<image /> <br />**

Les attributs

**<balise attribut="valeur">**

**<image nom="photo.jpg" />**

Structure de base d'une page HTML5

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

  <body>

  </body>
</html>
```

## Les 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>
```

## En résumé

- On utilise l'éditeur de texte (Sublime Text, Notepad++, jEdit, vim...) pour créer un fichier ayant l'extension .html (par exemple : test.html). Ce sera notre page web.
- Ce fichier peut être ouvert dans le navigateur web simplement en faisant un double-clic dessus.
- À l'intérieur du fichier, nous écrivons le contenu de notre page, accompagné de balises HTML.
- Les balises peuvent avoir plusieurs formes :
- <balise> </balise> : elles s'ouvrent et se ferment pour délimiter le contenu (début et fin d'un titre, par exemple).
- <balise /> : balises orphelines (on ne les insère qu'en un seul exemplaire), elles permettent d'insérer un élément à un endroit précis (par exemple une image).
- Les balises sont parfois accompagnées d'attributs pour donner des indications supplémentaires (exemple : <image nom="photo.jpg" />).
- Une page web est constituée de deux sections principales : un en-tête (<head>) et un corps (<body>).
- On peut afficher le code source de n'importe quelle page web en faisant un clic droit puis en sélectionnant Afficher le code source de la page.

# Organiser son texte

Les paragraphes

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

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>Paragraphes</title>
```

```
  </head>
```

```
  <body>
```

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

```
  </body>
```

```
</html>
```

Sauter une ligne

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>Paragraphes</title>
```

```
  </head>
```

```
  <body>
```

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

```
    Ceci est mon premier test alors soyez indulgents s'il vous plaît, j'apprends petit à petit comment cela marche.</p>
```

```
    <p>Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours quand j'aurai appris un peu plus de choses, je vous assure que vous allez être surpris !</p>
```

```
  </body>
```

```
</html>
```

Il existe <br />

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Sauts de ligne</title>
  </head>

  <body>
    <p>Bonjour et bienvenue sur mon site !<br />
    Ceci est mon premier test alors soyez indulgents s'il vous plaît, j'apprends petit à petit comment cela marche.</p>

    <p>Pour l'instant c'est un peu vide, mais revenez dans 2-3 jours quand j'aurai appris un peu plus de choses, je vous assure que vous allez être surpris !</p>
  </body>
</html>
<p> </p> : pour organiser son texte en paragraphes ;
<br /> : pour aller à la ligne.

```

#### Les titres

```

<h1> </h1> : signifie « titre très important ». En général, on s'en sert pour afficher le titre de la page au début de celle-ci.
<h2> </h2> : signifie « titre important ».
<h3> </h3> : pareil, c'est un titre un peu moins important (on peut dire un « sous-titre » si vous voulez).
<h4> </h4> : titre encore moins important.
<h5> </h5> : titre pas important.
<h6> </h6> : titre vraiment, mais alors là vraiment pas important du tout.

```

```

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

  <body>
    <h1>Titre super important</h1>
    <h2>Titre important</h2>
    <h3>Titre un peu moins important (sous-titre)</h3>

```

```
<h4>Titre pas trop important</h4>
<h5>Titre pas important</h5>
<h6>Titre vraiment pas important du tout</h6>
</body>
</html>
```

La mise en valeur

Mettre un peu en valeur : `<em>` `</em>`.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>Emphase</title>
```

```
</head>
```

```
<body>
```

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

```
Ceci est mon premier test alors <em>soyez indulgents</em> s'il vous plaît, j'apprends petit à petit comment cela marche.</p>
```

```
</body>
```

```
</html>
```

Mettre bien en valeur `<strong>` `</strong>`

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>Forte emphase</title>
```

```
</head>
```

```
<body>
```

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

```
Ceci est mon premier test alors <strong>soyez indulgents</strong> s'il vous plaît, j'apprends petit à petit comment cela marche.</p>
```

```
</body>
```

```
</html>
```

Marquer le texte <mark> </mark>  
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="utf-8" />  
  <title>Marquage du texte</title>  
</head>

<body>  
  <p>Bonjour et bienvenue sur mon site !<br />  
  Ceci est mon premier test alors <mark>soyez indulgents</mark> s'il vous plaît, j'apprends petit à petit comment cela marche.</p>  
</body>  
</html>

N'oubliez pas : HTML pour le fond, CSS pour la forme  
le HTML définit le fond (contenu, logique des éléments) ;  
le CSS définit la forme (apparence).

Les listes

Liste non ordonnée

<ul></ul>

<ul>  
  <li>Fraises</li>  
  <li>Framboises</li>  
  <li>Cerises</li>  
</ul>

<ul></ul> délimite toute la liste ;

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

Liste ordonnée

```
<ol></ol>
```

```
<h1>Ma journée</h1>
```

```
<ol>
```

```
<li>Je me lève.</li>
```

```
<li>Je mange et je bois.</li>
```

```
<li>Je retourne me coucher.</li>
```

```
</ol>
```

## En résumé

- Le HTML comporte de nombreuses balises qui nous permettent d'organiser le texte de notre page. Ces balises donnent des indications comme « Ceci est un paragraphe », « Ceci est un titre », etc.
- Les paragraphes sont définis par la balise `<p>` `</p>` et les sauts de ligne par la balise `<br />`.
- Il existe six niveaux de titre, de `<h1>` `</h1>` à `<h6>` `</h6>`, à utiliser selon l'importance du titre.
- On peut mettre en valeur certains mots avec les balises `<strong>`, `<em>` et `<mark>`.
- Pour créer des listes, on doit utiliser la balise `<ul>` (liste à puces, non ordonnée) ou `<ol>` (liste ordonnée). À l'intérieur, on insère les éléments avec une balise `<li>` pour chaque item.

## Pratiquez

Instructions

- Ajoutez un titre de niveau 1
- A la suite du titre, ajoutez un paragraphe indiquant «Voici mes animaux préférés : »
- Ajoutez une liste ordonnée à 3 puces au moins de vos animaux préférés
- Mettez en valeur (emphase forte) le premier animal de la liste



# Créer des liens

## Un lien vers un autre site

`<a href="http://www.viaformation.com">Via formation</a>`

## Un lien vers une autre page de son site

### Deux pages situées dans un même dossier

Dans un dossier test, nous allons créer deux fichiers correspondant à deux pages HTML différentes. Je vous propose de les appeler page1.html et page2.html.

#### page1.html

Code : HTML

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

#### page2.html

Code : HTML

`<h1>Bienvenue sur la page 2 !</h1>`

### Deux pages situées dans des dossiers différents

Imaginons que page2.html se trouve dans un sous-dossier appelé contenu, comme à la figure suivante.

#### page1.html

Code : HTML

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

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

#### page2.html

Code : HTML

`<h1>Bienvenue sur la page 2 !</h1>`

`<p>Retour sur <a href="../page1.html">la page 1</a></p>`

## Un lien vers une ancre

Une ancre est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues.

#### **Code : HTML**

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

#### **Code : HTML**

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

#### **Un lien qui affiche une infobulle au survol**

##### **Code : HTML**

```
<p>Bonjour. Souhaitez-vous visiter le <a href="http://www.viaformation.fr" title="Réservé aux apprenants">Site de Via formation</a> ?</p>
```

#### **Un lien qui ouvre une nouvelle fenêtre**

##### **Code : HTML**

```
<p>Bonjour. Souhaitez-vous visiter le <a href="http://www.viaformation.fr" target="_blank">Site de Via formation</a> ?<br />
```

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

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

##### **Code : HTML**

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

#### **Un lien pour télécharger un fichier**

##### **Code : HTML**

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

#### **En résumé**

- Les liens permettent de changer de page et sont, par défaut, écrits en bleu et soulignés.
- Pour insérer un lien, on utilise la balise `<a>` avec l'attribut `href` pour indiquer l'adresse de la page cible. Exemple : `<a href="http://www.viaformation.fr">`.
- On peut faire un lien vers une autre page de son site simplement en écrivant le nom du fichier : `<a href="page2.html">`.
- Les liens permettent aussi d'amener vers d'autres endroits sur la même page. Il faut créer une ancre avec l'attribut `id` pour « marquer » un endroit dans la page, puis faire un lien vers l'ancre comme ceci : `<a href="#ancre">`.

# Les images

## Les différents formats d'images

Toutes les images diffusées sur Internet ont un point commun : elles sont compressées. Cela veut dire que l'ordinateur fait des calculs pour qu'elles soient moins lourdes et donc plus rapides à charger.

### Le JPEG (Joint Photographic Expert Group)

Les images JPEG sont enregistrées avec l'extension .jpg ou .jpeg.

### Le PNG (Portable Network Graphics)

Le PNG existe en deux versions, en fonction du nombre de couleurs que doit comporter l'image :

PNG 8 bits : 256 couleurs ;

PNG 24 bits : 16 millions de couleurs (autant qu'une image JPEG).

### Le GIF

Le GIF conserve un certain avantage que le PNG n'a pas : il peut être animé.

## Il existe un format adapté à chaque image

- Si on résume, voici quel format adopter en fonction de l'image que vous avez :
- Une photo : utilisez un JPEG.
- N'importe quel graphique avec peu de couleurs (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.
- N'importe quel graphique avec beaucoup de couleurs : utilisez un PNG 24 bits.
- Une image animée : utilisez un GIF animé.

## Insertion d'une image

`<img />` est la balise qui va nous permettre d'insérer une image.

**Code : HTML**

```
<p>
```

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

```

```

```
</p>
```

- `src` : il permet d'indiquer où se trouve l'image que l'on veut insérer.
- `alt` : cela signifie « texte alternatif ».

## Ajouter une infobulle

### Code : HTML

<p>

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



</p>

### En résumé

- Il existe plusieurs formats d'images adaptées au Web :
  - JPEG : pour les photos ;
  - PNG : pour toutes les autres illustrations ;
  - GIF : similaire au PNG, plus limité en nombre de couleurs mais qui peut être animé.
- On insère une image avec la balise<img />. Elle doit obligatoirement comporter au moins ces deux attributs : src(nom de l'image) et alt (courte description de l'image).

# Partie 2 - Les joies de la mise en forme avec CSS

## Mettre en place le CSS

### En résumé

- CSS est un autre langage qui vient compléter le HTML. Son rôle est de mettre en forme votre page web.
- Il faut être vigilant sur la compatibilité des navigateurs avec certaines fonctionnalités récentes de CSS3. Quand un navigateur ne connaît pas une instruction de mise en forme, il l'ignore simplement.
- On peut écrire le code CSS à plusieurs endroits différents, le plus conseillé étant de créer un fichier séparé portant l'extension.css (exemple : style.css).
- En CSS, on sélectionne quelles portions de la page HTML on veut modifier et on change leur présentation avec des propriétés CSS :

### Code : CSS

balise1

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

- Il existe de nombreuses façons de sélectionner la portion de la page que l'on veut mettre en forme. Par exemple, on peut viser:
  - toutes les balises d'un même type, en écrivant simplement leur nom (h1 par exemple) ;
  - certaines balises spécifiques, auxquelles on a donné des noms à l'aide des attributs class ou id (**.nomclasse** ou **#nomid**) ;
  - uniquement les balises qui se trouvent à l'intérieur d'autres balises (h3 em).
  - etc.

# Formatage du texte

## En résumé

- On modifie la taille du texte avec la propriété CSS **font-size**. On peut indiquer la taille en pixels (16px), en « em » (1.3em), en pourcentage (110%), etc.
- On change la police du texte avec **font-family**. Attention, seules quelques polices sont connues par tous les ordinateurs. Vous pouvez cependant utiliser une police personnalisée avec la directive **@font-face** : cela forcera les navigateurs à télécharger la police de votre choix.
- De nombreuses propriétés de mise en forme du texte existent : **font-style** pour l'italique, **font-weight** pour la mise en gras, **text-decoration** pour le soulignement, etc.
- Le texte peut être aligné avec **text-align**.
- On peut faire en sorte qu'une image soit habillée (« entourée ») de texte avec **float**.

# La couleur et le fond

## En résumé

- On change la couleur du texte avec la propriété **color**, la couleur de fond avec **background-color**.
- On peut indiquer une couleur en écrivant son nom en anglais (black, par exemple), sous forme hexadécimale (#FFC8D3) ou en notation RGB (rgb(250,25,118)).
- On peut ajouter une image de fond avec **background-image**. On peut choisir de fixer l'image de fond, de l'afficher en mosaïque ou non, et même de la positionner où on veut sur la page.
- On peut rendre une portion de la page transparente avec la propriété **opacity** ou avec la notation RGBA (identique à la notation RGB, avec une quatrième valeur indiquant le niveau de transparence).

# Les bordures et les ombres

## En résumé

- On peut appliquer une bordure à un élément avec la propriété **border**. Il faut indiquer la largeur de la bordure, sa couleur et son type (trait continu, pointillés...).
- On peut arrondir les bordures avec **border-radius**.
- On peut ajouter une ombre aux blocs de texte avec **box-shadow**. On doit indiquer le décalage vertical et horizontal de l'ombre, son niveau d'adoucissement et sa couleur.
- Le texte peut lui aussi avoir une ombre avec **text-shadow**.



# Création d'apparences dynamiques

## En résumé

- En CSS, on peut modifier l'apparence de certaines sections dynamiquement, après le chargement de la page, lorsque certains évènements se produisent. On utilise pour cela les pseudo-formats.
- Le pseudo-format **:hover** permet de changer l'apparence au survol (par exemple : **a:hover** pour modifier l'apparence des liens lorsque la souris pointe dessus).
- Le pseudo-format **:active** modifie l'apparence des liens au moment du clic, **:visited** lorsqu'un lien a déjà été visité. Le pseudo-format **:focus** permet de modifier l'apparence d'un élément sélectionné.

# Partie 3 : Mise en page du site

## Structurer sa page

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Patrick - Le Site Web</title>
  </head>
  <body>
    <header>
      <h1>Patrick</h1>
      <h2>Carnets de randonnées</h2>
    </header>
    <nav>
      <ul>
        <li><a href="#">Accueil</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">CV</a></li>
      </ul>
    </nav>
    <section>
      <aside>
        <h1>À propos de l'auteur</h1>
        <p>C'est moi, Patrick !</p>
      </aside>
      <article>
        <h1>Je suis un randonneur</h1>
        <p>Bla bla bla bla (texte de l'article)</p>
      </article>
    </section>
    <footer>
      <p>Copyright Patrick - Tous droits réservés<br />
      <a href="#">Me contacter !</a></p>
    </footer>
  </body>
</html>
```

## En résumé

- Plusieurs balises ont été introduites avec HTML5 pour délimiter les différentes zones qui constituent la page web :
  - **<header>** : en-tête ;
  - **<footer>** : pied de page ;
  - **<nav>** : liens principaux de navigation ;
  - **<section>** : section de page ;
  - **<aside>** : informations complémentaires ;
  - **<article>** : article indépendant.
- Ces balises peuvent être imbriquées les unes dans les autres. Ainsi, une section peut avoir son propre en-tête. Ces balises ne s'occupent pas de la mise en page. Elles servent seulement à indiquer à l'ordinateur le sens du texte qu'elles contiennent. On pourrait très bien placer l'en-tête en bas de la page si on le souhaite.
- Le code JavaScript HTML5shiv permet de faire en sorte que ces balises soient reconnues pour les versions d'Internet Explorer antérieures à IE9.

# Le modèle des boîtes

## En résumé

- On distingue deux principaux types de balises en HTML:
  - Le type block (`<p>`, `<h1>`...) : ces balises créent un retour à la ligne et occupent par défaut toute la largeur disponible. Elles se suivent de haut en bas.
  - Le type inline (`<a>`, `<strong>`...) : ces balises délimitent du texte au milieu d'une ligne. Elles se suivent de gauche à droite.
- On peut modifier la taille d'une balise de type block avec les propriétés CSS **width** (largeur) et **height** (hauteur). On peut définir des minima et maxima autorisés pour la largeur et la hauteur : **min-width**, **max-width**, **min-height**, **max-height**.
- Les éléments de la page disposent chacun de marges intérieures (**padding**) et extérieures (**margin**).
- S'il y a trop de texte à l'intérieur d'un bloc de dimensions fixes, il y a un risque de débordement. Dans ce cas, il peut être judicieux de rajouter des barres de défilement avec la propriété **overflow** ou de forcer la césure avec **word-wrap**.

# Le positionnement en CSS

## En résumé

- La mise en page d'un site web s'effectue en CSS. Plusieurs techniques sont à notre disposition.
- Le positionnement flottant (avec la propriété **float**) est l'un des plus utilisés à l'heure actuelle. Il permet par exemple de placer un menu à gauche ou à droite de la page. Néanmoins, cette propriété n'a pas été initialement conçue pour cela et il est préférable, si possible, d'éviter cette technique.
- Le positionnement *inline-block* consiste à affecter un type *inline-block* à nos éléments grâce à la propriété **display**. Ils se comporteront comme des inlines (placement de gauche à droite) mais pourront être redimensionnés comme des blocs (avec **width** et **height**). Cette technique est à préférer au positionnement flottant.
- Le positionnement absolu permet de placer un élément où l'on souhaite sur la page, au pixel près.
- Le positionnement fixe est identique au positionnement absolu mais l'élément restera toujours visible même si on descend plus bas dans la page.
- Le positionnement relatif permet de décaler un bloc par rapport à sa position normale.
- Un élément A positionné en absolu à l'intérieur d'un autre élément B (lui-même positionné en absolu, fixe ou relatif) se positionnera par rapport à l'élément B, et non par rapport au coin en haut à gauche de la page.

# Partie 4 : Fonctionnalités évoluées

## Les tableaux

C'est un peu déroutant mais il est conseillé d'écrire les balises dans l'ordre suivant :

1. <thead>
2. <tfoot>
3. <tbody>

### Code : HTML

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

## En résumé

- Un tableau s'insère avec la balise `<table>` et se définit ligne par ligne avec `<tr>`.
- Chaque ligne comporte des cellules `<td>` (cellules normales) ou `<th>` (cellules d'en-tête).
- Le titre du tableau se définit avec `<caption>`.
- On peut ajouter une bordure aux cellules du tableau avec `border`. Pour fusionner les bordures, on utilise la propriété CSS `border-collapse`.
- Un tableau peut être divisé en trois sections : `<thead>` (en-tête), `<tbody>` (corps) et `<tfoot>` (bas du tableau). L'utilisation de ces balises n'est pas obligatoire.
- On peut fusionner des cellules horizontalement avec l'attribut `colspan` ou verticalement avec `rowspan`. Il faut indiquer combien de cellules doivent être fusionnées.

# Les formulaires

## Créer un formulaire

Code : HTML

```
<p>Texte avant le formulaire</p>
<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>
<p>Texte après le formulaire</p>
```



## En résumé

- Un formulaire est une zone interactive de la page, dans laquelle vos visiteurs peuvent saisir des informations.
- On délimite un formulaire avec la balise `<form>` à laquelle il faut ajouter deux attributs : *method* (mode d'envoi des données) et *action* (page vers laquelle le visiteur sera redirigé après envoi du formulaire et qui traitera les informations).
- Une grande partie des éléments du formulaire peut s'insérer avec la balise `<input />`. La valeur de son attribut *type* permet d'indiquer quel type de champ doit être inséré :
  - *text* : zone de texte ;
  - *password* : zone de texte pour mot de passe ;
  - *tel* : numéro de téléphone ;
  - *checkbox* : case à cocher ;
  - etc.
- La balise `<label>` permet d'écrire un libellé. On l'associe à un champ de formulaire avec l'attribut *for*, qui doit avoir la même valeur que l'*id* du champ de formulaire.
- On peut rendre un champ obligatoire avec l'attribut *required*, faire en sorte qu'il soit sélectionné par défaut avec *autofocus*, donner une indication dans le champ avec *placeholder*...
- Pour récupérer ce que les visiteurs ont saisi, le langage HTML ne suffit pas. Il faut utiliser un langage « serveur » comme PHP... Si vous voulez aller plus loin, il va donc falloir apprendre un nouveau langage !

# La vidéo et l'audio

## En résumé

- Insérer de la musique ou de la vidéo n'était pas possible autrefois en HTML. Il fallait recourir à un plugin comme Flash. Depuis HTML5, les balises `<audio>` et `<video>` ont été introduites et permettent de jouer de la musique et des vidéos sans plugin.
- Il existe plusieurs formats audio et vidéo. Il faut notamment connaître :
  - pour l'audio : MP3 et Ogg Vorbis ;
  - pour la vidéo : H.264, Ogg Theora et WebM.
- Aucun format n'est reconnu par l'ensemble des navigateurs : il faut proposer différentes versions de sa musique ou de sa vidéo pour satisfaire tous les navigateurs.
- Il faut ajouter l'attribut *controls* aux balises `<audio>` et `<video>` pour permettre au visiteur de lancer ou d'arrêter le média.
- Ces balises ne sont pas conçues pour empêcher le téléchargement de la musique et de la vidéo. Vous ne pouvez pas protéger votre média contre la copie.

# Mise en page adaptative 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.

## Mise en place des media queries

Il existe de nombreuses règles permettant de construire des media queries. Je vous présente ici les principales :

- **color** : gestion de la couleur (en bits/pixel).
- **height** : hauteur de la zone d'affichage (fenêtre). On peut rajouter le préfixe **min-** ou **max-**.
- **width** : largeur de la zone d'affichage (fenêtre). On peut rajouter le préfixe **min-** ou **max-**.
- **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 ;
  - **print** : impression ;
  - **tv** : télévision ;
  - **projection** : projecteur ;
  - **all** : tous les types d'écran.

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)

Les media queries sont surtout utilisées pour adapter le design du site aux différentes largeurs d'écran.

Faisons un test tout simple : nous allons changer la couleur et la taille du texte si la fenêtre fait plus ou moins de 1024 pixels de large.

```
/* Paragraphes en bleu par défaut */
```

```
p
{
  color: blue;
}
```

```
/* Nouvelles règles si la fenêtre fait au plus 1024px de large */
```

```
@media screen and (max-width: 1024px)
```

```
{
  p
  {
    color: red;
    background-color: black;
    font-size: 1.2em;
  }
}
```

## Media queries et navigateurs mobiles

Le viewport : c'est la largeur de la fenêtre du navigateur sur le mobile.

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" />
```

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" />
```

## En résumé

- Les media queries permettent de charger des styles CSS différents en fonction de certains paramètres.
- Les paramètres autorisés par les media queries sont nombreux : nombre de couleurs, résolution de l'écran, orientation... En pratique, on s'en sert surtout pour modifier l'apparence du site en fonction des différentes résolutions d'écran.
- On crée une media query avec la directive **@media** suivie du type d'écran et d'une ou plusieurs conditions (comme la largeur maximale d'écran). Le style CSS qui suit sera activé uniquement si les conditions sont remplies.
- Les navigateurs mobiles simulent une largeur d'écran : on appelle cela le *viewport*.
- On peut cibler les smartphones grâce à une règle basée sur le nombre réel de pixels affichés à l'écran : **max-device-width**.