

**Présentent**

## **Cours complet d'HTML**



**Formatrice: Nene SYLLA**

## **I. Introduction a HTML :**

HTML est un langage de balise également appelé langage de description (et non un langage de programmation) qui a été créé en 1991 et qui va nous permettre de décrire l'aspect d'un document, d'y inclure des informations variées (textes, images, sons, animations etc.) et d'établir des relations cohérentes entre ces informations grâce aux liens hypertextes.

### ***Pourquoi apprendre le HTML et le CSS :***

- Les langages HTML et CSS sont incontournables car ils n'ont pas de concurrent.
- Une base pour comprendre son site, le modifier et résoudre les problèmes au besoin.
- Indispensable pour optimiser son référencement.
- Pour comprendre et se faire comprendre des développeurs avec qui vous travaillez.

### ***Les outils nécessaires :***

- Editeur de texte,
- Navigateur

## **II. Les balises et leurs attributs :**

### ***a. Les balises :***

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher.

Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles < et >, comme ceci : **<balise>**

À quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc.

On distingue deux types de balises : les balises en paires et les balises orphelines.

- **Les balises en paires :**

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

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

On distingue une balise ouvrante (<titre>) et une balise fermante (</titre>) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est pas entre ces deux balises... n'est pas un titre.

```
Ceci n'est pas un titre <title>Ceci est un titre</title> Ceci n'est pas un titre
```

- **Les balises orphelines :**

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis. Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ». Une balise orpheline s'écrit comme ceci :

```
<img />
```

**NB :** Notez que le / de fin n'est pas obligatoire. On pourrait écrire seulement <image>. Néanmoins, pour ne pas les confondre avec le premier type de balise, les webmasters recommandent de rajouter ce / (slash) à la fin des balises orphelines.

**b. Les attributs :**

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires.

L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise `<img />` que nous venons de voir. Seule, elle ne sert pas à grand-chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```
<img nom="photo.jpg" />
```

L'ordinateur comprend alors qu'il doit afficher l'image contenue dans le fichier photo.jpg.

Dans le cas d'une balise fonctionnant « par paire », on ne met les attributs que dans la balise ouvrante et pas dans la balise fermante.

### iii. Structure de base d'une page HTML5 :

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

  </body>
</html>
```

**NB :** Vous noterez que les balises s'ouvrent et se ferment dans un ordre précis. Par exemple, la balise **<html>** est la première que l'on ouvre et c'est aussi la dernière que l'on ferme (tout à la fin du code, avec **</html>**). *Les balises doivent être fermées dans le sens inverse de leur ouverture.* Un exemple : ○ **<html><body></body></html>** : **correct**. Une balise qui est ouverte à l'intérieur d'une autre doit aussi être fermée à l'intérieur. ○ **<html><body></html></body>** : **incorrect**, les balises s'entremêlent.

**Explication :**

### **Le doctype :**

```
<!DOCTYPE html>
```

La toute première ligne s'appelle le **doctype**. Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation).

### **La balise <html> :**

```
<html>  
  
</html>
```

C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante **</html>** se trouve tout à la fin du code !

### **L'en-tête <head> et le corps <body> :**

Une page web est constituée de deux parties :

L'en-tête **<head>** : cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc.

Cette section est généralement assez courte. Les informations que contient l'entête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes ! Le corps **<body>** : c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

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

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

Cette balise indique l'encodage utilisé dans votre fichier .html. L'encodage indique la façon dont le fichier est enregistré.

C'est lui qui détermine comment les caractères spéciaux vont s'afficher (accents, idéogrammes chinois et japonais, caractères arabes, etc.). Le titre principal de la page :

```
<title>
```

C'est le titre de votre page, probablement l'élément le plus important ! Toute page doit avoir un titre qui décrit ce qu'elle contient.

Il est conseillé de garder le titre assez court (moins de 100 caractères en général). Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur). Enregistrez votre page web et ouvrez-la dans votre navigateur.

### ***Les commentaires :***

Un commentaire en HTML est un texte qui sert simplement de mémo. Il n'est pas affiché, il n'est pas lu par l'ordinateur, cela ne change rien à l'affichage de la page. A quoi sert le commentaire ?

Cela sert à *vous* et aux personnes qui liront le code source de votre page. Vous pouvez utiliser les commentaires pour laisser des indications sur le fonctionnement de votre page.

### □ **Insérer un commentaire :**

Un commentaire est une balise HTML avec une forme bien spéciale :

```
<!-- Ceci est un commentaire -->
```

Vous pouvez le mettre où vous voulez au sein de votre code source : il n'a aucun impact sur votre page, mais vous pouvez vous en servir pour vous aider à vous repérer dans votre code source (surtout s'il est long).

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>
  <body>
    <!-- Corps de la page -->
  </body>
</html>
```

### **En résumé :**

- On utilise l'éditeur de texte (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 écrirons 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. Il y a de nombreuses balises HTML. Certaines existent depuis la toute première version de HTML, d'autres ont été introduites plus récemment dans HTML5. Nous allons voir successivement :
  - comment rédiger des paragraphes ;
  - comment structurer sa page avec les titres ;
  - comment donner de l'importance à certains mots de son texte ; comment organiser les informations sous forme de listes.



La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise `<p>` pour délimiter les paragraphes.

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

`<p>` signifie « Début du paragraphe » ; `</p>` signifie « Fin du paragraphe ».

**Exemple :**

```
<!DOCTYPE html>
```



```

<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Paragraphe</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <p>Bonjour et bienvenue sur mon site !</p>
  </body>
</html>


```

### **Sauter une ligne :**

En HTML, si vous appuyez sur la touche Entrée, cela ne crée pas une nouvelle ligne comme vous en avez l'habitude. En fait, si vous voulez écrire un deuxième paragraphe, il vous suffit d'utiliser une deuxième balise **<p>**.

Votre code HTML devrait donc être au final rempli de balises de paragraphe !

#### **Exemple :**



```

<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Paragraphe</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <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>

```

```
</body>
</html>
```

### **Les titres :**

Lorsque le contenu de votre page va s'étoffer avec de nombreux paragraphes, il va devenir difficile pour vos visiteurs de se repérer. C'est là que les titres deviennent utiles.

En HTML, on a le droit d'utiliser six niveaux de titres différents. C'est à dire qu'on peut dire « Ceci est un titre très important », « Ceci est un titre un peu moins important », « Ceci est un titre encore moins important », etc. On a donc six balises de titres différentes :

- **<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.

**NB** : ne confondez pas avec la balise **<title>** ! La balise **<title>** affiche le titre de la page dans la barre de titre du navigateur comme nous l'avons vu. Les titres **<h1>** et compagnie, eux, servent à créer des titres qui seront affichés *dans* la page web.

### **Exemple :**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Niveau de titre</title>
  </head>
```

```
<body>
  <!-- Corps de la page -->
  <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 :**

Au sein de vos paragraphes, certains mots sont parfois plus importants que d'autres et vous aimeriez les faire ressortir. HTML vous propose différents moyens de mettre en valeur le texte de votre page.

#### **□ Mettre un peu en valeur :**

Pour mettre un peu en valeur votre texte, vous devez utiliser la balise **<em> </em>**. Son utilisation est très simple : encadrez les mots à mettre en valeur avec ces balises

**Exemple :**



et c'est bon !

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Emphase</title>
```

```

</head>
<body>
  <!-- Corps de la page -->
  <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>

```

**NB :** Utiliser la balise `<em>` a pour conséquence de mettre le texte en italique. En fait, c'est le navigateur qui choisit comment afficher les mots. On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique.

### □ **Mettre bien en valeur :**

Pour mettre un texte bien en valeur, on utilise la balise `<strong>` qui signifie « fort », ou « important » si vous préférez. Elle s'utilise exactement de la même manière que `<em>` :

```

<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Forte emphase</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <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>

```

**NB :** Le texte s'affichera en gras. Là encore, le gras n'est qu'une conséquence. Le navigateur a choisi d'afficher en gras les mots importants pour les faire ressortir davantage.

La balise `<strong>` ne signifie pas « mettre en gras » mais « important ». On pourra décider plus tard, en CSS, d'afficher les mots « importants » d'une autre façon que le gras si on le souhaite.

### **Marquer le texte :**

La balise `<mark>` permet de faire ressortir visuellement une portion de texte. L'extrait n'est pas forcément considéré comme important mais on veut qu'il se distingue bien du reste du texte. Cela peut être utile pour faire ressortir un texte pertinent après une recherche sur votre site par exemple.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Marquage du texte</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <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>
```

### **Les listes :**

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations.

Nous allons découvrir ici deux types de listes : •

- les listes non ordonnées ou listes à puces ;

- les listes ordonnées ou listes numérotées ou encore énumérations.

## **1. Liste non ordonnée :**

Une liste non ordonnée ressemble à ceci :

- Fraises
- Framboises
- Cerises

C'est un système qui nous permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de « premier » ni de « dernier »).

Créer une liste non ordonnée est très simple. Il suffit d'utiliser la balise `<ul>` que l'on referme un peu plus loin avec `</ul>`. Commencez donc à taper ceci :

```
<ul></ul>
```

Et maintenant, voilà ce qu'on va faire : on va écrire chacun des éléments de la liste entre deux balises `<li></li>`. Chacune de ces balises doit se trouver entre `<ul>` et `</ul>`.

**Exemple :**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Liste ordonnee</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <h1>Ma journée</h1>
    <ul>
      <li>Fraises</li>
      <li>Framboises</li>
      <li>Cerises</li>
    </ul>
  </body>
</html>
```

### **Retenez donc ces deux balises :**

**<ul></ul>** délimite toute la liste ;

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

Vous pouvez mettre autant d'éléments que vous voulez dans la liste à puces, vous n'êtes pas limités à trois éléments.

### **2. Liste ordonnée :**

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer **<ul></ul>** par **<ol></ol>**.

À l'intérieur de la liste, on ne change rien : on utilise toujours des balises **<li></li>** pour délimiter les éléments.

L'ordre dans lequel vous placez les éléments de la liste est important. Le premier **<li></li>** sera l'élément n° 1, le second sera le n°2 etc... **Exemple :**

```

<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Liste ordonnee</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <h1>Ma journée</h1>
    <ol>
      <li>Je me lève.</li>
      <li>Je mange et je bois.</li>
      <li>Je retourne me coucher.</li>
    </ol>
  </body>
</html>

```

### 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.

### Créer des liens :



- **Un lien vers un autre site :**

Il est facile de reconnaître les liens sur une page : ils sont écrits d'une façon différente (par défaut, en bleu et soulignés) et un curseur en forme de main apparaît lorsqu'on pointe dessus.

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

Par exemple, le code ci-dessous est un lien qui amène vers youtube, situé à l'adresse <http://www.youtube.com>:

```
<a href="http://www.youtube.com">Youtube</a>
```

- **Un lien vers une autre page de son site :**

### **1. Deux pages situées dans un même dossier :**

Pour commencer, nous allons créer deux fichiers correspondant à deux pages HTML différentes. Les fichiers seront appelés page1.html et page2.html. Nous aurons donc ces deux fichiers sur notre disque dans le même dossier.

Comment faire un lien de la page 1 vers la page 2, sans avoir d'adresse en `http://` ? En fait, c'est facile : si les deux fichiers sont situés dans le même dossier, il suffit d'écrire comme cible du lien le nom du fichier vers lequel on veut amener. Par exemple :

`<a href="page2.html">`. On dit que c'est un lien relatif.

Voici le code que nous allons utiliser dans nos fichiers page1.html et page2.html.

#### **Page1.html :**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
```

```
<title>Lien de page</title>
</head>
<body>
  <!-- Corps de la page -->
  <p>Bonjour. Souhaitez-vous consulter <a href="page2.html">la
page2</a> ?</p>
</body>
</html>
```

### **Page2.html :**

La page 2 (page d'arrivée) affichera simplement un message pour indiquer que l'on est bien arrivé sur la page 2 :



```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Lien de page</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <h1>Bienvenue sur la page 2 !</h1>
  </body>
</html>
```

## **2. Deux pages situées dans des dossiers différents :**

Les choses se corsent un petit peu si les pages sont situées dans des dossiers différents. Idéalement, elles ne devraient pas être trop loin l'une de l'autre (dans un sous-dossier par exemple).

Imaginons que page2.html se trouve dans un sous-dossier appelé contenu :

```
<!DOCTYPE html>
<html>

  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Lien de page</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <p>Bonjour.      Souhaitez-vous consulter
href="contenu/page2.html">la page2</a> ?</p>
  </body>
</html>                                     <a
```

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Ceci est un commentaire -->
    <meta charset="utf-8" />
    <title>Lien de page</title>
  </head>
  <body>
    <!-- Corps de la page -->
    <p>Bonjour.      Souhaitez-vous      consulter
href="contenu/autredossier/page2.html">la page2</a> ?</p>
  </body>
</html>                                     <a
```

## Page1.html :

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

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

## En résumé :

- Les liens permettent de changer de page et sont, par défaut, écrits en bleu

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

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.monsite.com">`.
- On peut faire un lien vers une autre page de son site simplement en écrivant le nom du fichier : `<a href="page2.html">`.

## Les images :

Insérer une image dans une page web ? Il existe différents **formats** d'image que l'on peut utiliser sur des sites web, et on ne doit pas les choisir au hasard. En effet, les images sont parfois volumineuses à télécharger, ce qui ralentit le temps de chargement de la page (beaucoup plus que le texte !).

Pour faire en sorte que vos pages restent lisibles et rapides à télécharger, suivez donc activement mes conseils !

## **Les différents formats d'images :**

Quand vous avez une image « entre les mains », vous avez la possibilité de l'enregistrer dans plusieurs « formats » différents. 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.

Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...). Notre but ici est de faire le tour des différents formats utilisés sur le Web pour que vous les connaissiez et sachiez choisir celui qui convient le mieux à votre image.

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.

### **a) Le JPEG :**

Les images au format JPEG (*Joint Photographic Expert Group*) sont très répandues sur le Web. Ce format est conçu pour réduire le poids des photos (c'est-à-dire la taille du fichier associé), qui peuvent comporter plus de 16 millions de couleurs différentes. La figure suivante est une photo enregistrée au format JPEG.

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

Notez que le JPEG détériore un peu la qualité de l'image, d'une façon généralement imperceptible. C'est ce qui le rend si efficace pour réduire le poids des photos.

Quand il s'agit d'une photo, on ne peut généralement pas détecter la perte de qualité. Par contre, si ce n'est pas une photo, vous risquez de voir l'image un peu « baver ». Dans ce cas, il vaut mieux utiliser le format PNG.

### **b) Le PNG :**

Le format PNG (*Portable Network Graphics*) est le plus récent de tous. Ce format est adapté à la plupart des graphiques. Le PNG a deux gros avantages : il peut être rendu transparent et il n'altère pas la qualité de l'image.

Le PNG a été inventé pour concurrencer un autre format, le GIF, à l'époque où il fallait payer des royalties pour pouvoir utiliser des GIF. Depuis, le PNG a bien évolué et c'est devenu le format le plus puissant pour enregistrer la plupart des images. 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).

### **c) Le GIF :**

C'est un format assez vieux, qui a été néanmoins très utilisé (et qui reste très utilisé par habitude). Aujourd'hui, le PNG est globalement bien meilleur que le GIF : les images sont généralement plus légères et la transparence est de meilleure qualité. Il est recommandé donc d'utiliser le PNG autant que possible.

Le format GIF est limité à 256 couleurs (alors que le PNG peut aller jusqu'à plusieurs millions de couleurs).

Néanmoins, 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.

### ***Les erreurs à éviter :***

#### ***Bannissez les autres formats :***

Les autres formats non cités ici, comme le format BITMAP (\*.bmp) sont à bannir car bien souvent ils ne sont pas compressés, donc trop gros. Ils ne sont pas du tout adaptés au Web. On peut en mettre sur son site mais le chargement sera vraiment *extrêmement* long !

#### ***Choisissez bien le nom de votre image :***

Si vous voulez éviter des problèmes, prenez l'habitude d'enregistrer vos fichiers avec des noms en minuscules, sans espace ni accent, par exemple : mon\_image.png.

Vous pouvez remplacer les espaces par le caractère *Underscore* (« \_ ») comme je l'ai fait ici.

### ***Insérer une image :***

Revenons maintenant au code HTML pour découvrir comment placer des images dans nos pages web !

### ***Insertion d'une image :***

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

C'est une balise de type **orpheline** (comme **<br />**). Cela veut dire qu'on n'a pas besoin de l'écrire en deux exemplaires comme la plupart des autres balises que nous avons vues jusqu'ici. En effet, nous n'avons pas besoin de délimiter une portion de texte, nous voulons juste insérer une image à un endroit précis.

La balise doit être accompagnée de deux attributs obligatoires :

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

Les images doivent se trouver obligatoirement à l'intérieur d'un paragraphe

(`<p></p>`). Voici un exemple d'insertion d'image :

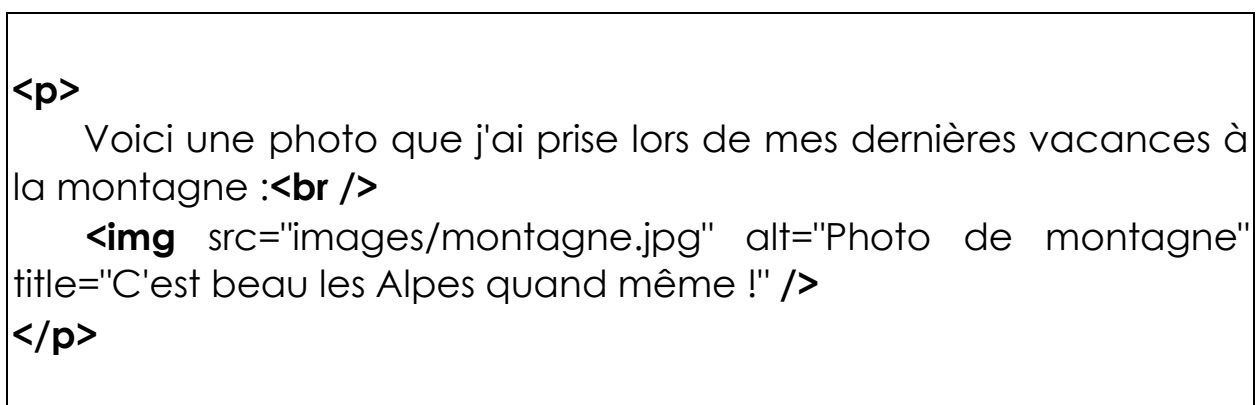


L'insertion d'image est quelque chose de très facile pour peu qu'on sache indiquer où se trouve l'image, comme on avait appris à le faire avec les liens.

La plus grosse « difficulté » (si on peut appeler cela une difficulté) consiste à choisir le bon format d'image. Ici, c'est une photo donc c'est évidemment le format JPEG qu'on utilise.

### ***Ajouter une infobulle :***

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



### ***Miniature cliquable :***

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien sur cette miniature pour que vos visiteurs puissent afficher l'image en taille originale.



Il existe des millions de logiciels permettant de créer des miniatures d'images. Voici le code HTML que je vais utiliser pour cela :

```
<p>
    Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez
    dessus !<br />
    <a href="img/montagne.jpg"></a>
</p>
```

### **Les figures :**

Ce sont des éléments qui viennent enrichir le texte pour compléter les informations de la page, tout ce qui vient *illustrer* le texte est une figure. Les figures peuvent être de différents types :

- images ;
- codes source ; citations ; etc.

### **Création d'une figure:**

En HTML5, on dispose de la balise **<figure>**. Voici comment on pourrait l'utiliser :  
<figure>

```

</figure>
```

Une figure est le plus souvent accompagnée d'une légende. Pour ajouter une légende, utilisez la balise **<figcaption>** à l'intérieur de la balise **<figure>**, comme ceci :

```
<figure>
    
    <figcaption>Le logiciel Bloc-Notes</figcaption>
```

```
</figure>
```

La balise **<figure>** a un rôle avant tout **sémantique**. Cela veut dire qu'elle indique à l'ordinateur que l'image a du sens et qu'elle est importante pour la bonne compréhension du texte. Cela peut permettre à un programme de récupérer toutes les figures du texte et de les référencer dans une table des figures, par exemple. Enfin, sachez qu'une figure peut très bien comporter plusieurs images. Voici un cas où cela se justifie:

```
<figure>
  
  
  
  <figcaption>Logos des différents navigateurs</figcaption>
</figure>
```

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

### Les balises universelles :

Il arrivera parfois que vous ayez besoin d'appliquer une class (ou un id) à certains mots qui, à l'origine, ne sont pas entourés par des balises.

En effet, le problème de class, c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise.

En fait, on a inventé deux balises dites universelles, qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une différence minime (mais significative !) entre ces deux balises :

- **<span> </span>** : c'est une balise de type inline, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises <strong> et <em> sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe.

### Exemple :

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

- **<div> </div>** : c'est une balise de type block, qui entoure un bloc de texte. Les balises <p>, <h1>, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. <div> est une balise fréquemment utilisée dans la construction d'un design.

### Mise en page du site :

#### Structurer sa page :

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).

#### Les balises structurantes de HTML5 :

Il existe de nouvelles balises introduites par HTML5 pour structurer nos pages. Cela ne va pas beaucoup changer l'apparence de notre site, mais il sera bien construit et prêt à être mis en forme ensuite !

- **<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... Vous devrez placer ces informations à l'intérieur de la balise <header> :

```
<header>
```

```
<!-- Placez ici le contenu de l'en-tête de votre page -->  
</header>
```

**NB :** L'en-tête peut contenir tout ce que vous voulez : images, liens, textes... Il peut y avoir plusieurs en-têtes dans votre page. Si celle-ci est découpée en plusieurs sections, chaque section peut en effet avoir son propre <header>.

- **<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.

```
<footer>
```

```
<!-- Placez ici le contenu du pied de page -->
```

```
</footer>
```

- **<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> :

```
<nav>
```

```
  <ul>
```

```
    <li><a href="index.html">Accueil</a></li>
```

```
    <li><a href="forum.html">Forum</a></li>
```

```
<li><a href="contact.html">Contact</a></li>
</ul>
</nav>
```

- **<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.

```
<section>
  <h1>Ma section de page</h1>
  <p>Bla bla bla bla</p>
</section>
```

- **<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). Il peut y avoir plusieurs blocs <aside> dans la page.

```
<aside>
<!-- Placez ici des informations complémentaires -->
</aside>
```

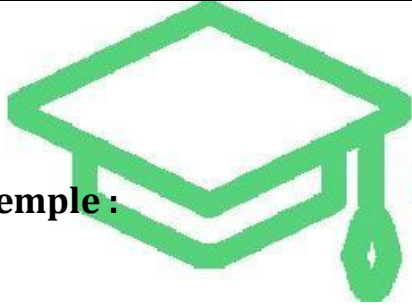
- **<article> : un article indépendant**

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).

```
<article>
```

```
<h1>Mon article</h1>
<p>Bla bla bla bla</p>
</article>
```

**Exemple :**



eMalischool

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mon blog</title>
  </head>
  <body>
    <header>
      <h1>Bien venue sur mon blog</h1>
      <h2>Carnets de voyage</h2>
    </header>
    <nav>
      <ul>
        <li><a href="#">Accueil</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">Contacte</a></li>
      </ul>
    </nav>
    <section>
      <aside>
        <h1>À propos de l'auteur</h1>
        <p>C'est moi, le créateur du blog.</p>
      </aside>
      <article>
```

```

        <h1>Je suis un grand voyageur</h1>
        <p>J'aime partager avec mes lecteurs tous les bon plans
que je pourrai avoir lors de mes voyages à travers le monde!</p>
    </article>
</section>
<footer>
    <p>Copyright voyageur - 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.

## Les tableaux :

Indispensables pour organiser les informations, les tableaux sont un petit peu délicat à construire en HTML. Il va en effet falloir imbriquer de nouvelles balises HTML dans un ordre précis.

Nous allons commencer par construire des tableaux basiques, puis nous les complexifierons au fur et à mesure : fusion de cellules, division en multiples sections... Nous découvrirons aussi les propriétés CSS liées aux tableaux, qui nous permettront de personnaliser leur apparence.

## 1. Un tableau simple :

La première balise à connaître est `<table> </table>`. C'est cette balise qui permet d'indiquer le début et la fin d'un tableau.

Cette balise est de type bloc, il faut donc la placer en dehors d'un paragraphe.

### Exemple :

```
<p>Ceci est un paragraphe avant le tableau.</p>
<table>
<!-- Ici, on écrira le contenu du tableau -->
</table>
<p>Ceci est un paragraphe après le tableau.</p>
```

### Qu'écrit-on à l'intérieur du tableau ?

Pour commencer, voici deux nouvelles balises très importantes :

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

En HTML, un tableau se construit ligne par ligne. Dans chaque ligne (`<tr>`), on indique le contenu des différentes cellules (`<td>`).

### Exemple :

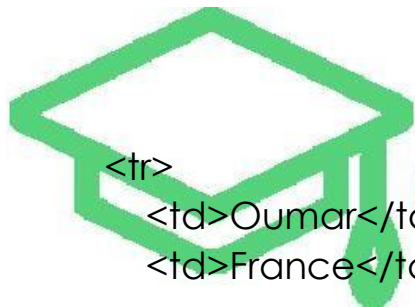
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma table</title>
  </head>
```



```

<body>
  <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>
    <tr>
      <td>Abdoulaye</td>

```



```

    <tr>
      <td>Oumar</td>
      <td>72 ans</td>
      <td>France</td>
      <td>40 Ans</td>
      <td>Mali</td>
    </tr>

```

```

  </tr>
</table>
</body>

```

```

</html>

```

Comme résultat, nous aurons un tableau sans bordures. Il faut comprendre qu'un tableau sans CSS paraît bien vide.

### **La ligne d'en-tête:**

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>` !

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



La ligne d'en-tête est très facile à reconnaître pour deux raisons :

- les cellules sont des <th> au lieu des <td> habituels ;
- c'est la première ligne du tableau

### ***Titre du tableau :***

Normalement, tout tableau doit avoir un titre. Le titre permet de renseigner rapidement le visiteur sur le contenu du tableau.

La balise <caption> se place tout au début du tableau, juste avant l'en-tête. C'est elle qui contient le titre du tableau.

### **Exemple:**

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma table</title>
  </head>
  <body>
    <table>
      <caption>Passagers du vol 377</caption>
      <tr>

```



eMalischool

```

      <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>
</body>

```

```
</html>
```

## 2. Un tableau structuré :

Nous avons appris à construire des petits tableaux simples. Ces petits tableaux suffisent dans la plupart des cas, mais il arrivera que vous ayez besoin de réaliser des tableaux plus... complexes.

Nous allons découvrir deux techniques particulières :

- Pour les gros tableaux, il est possible de les diviser en trois parties :
  - En-tête ;
  - Corps du tableau ;
  - Pied de tableau.
- Pour certains tableaux, il se peut que vous ayez besoin de fusionner des cellules entre elles.

### Diviser un gros tableau :

Si votre tableau est assez gros, vous aurez tout intérêt à le découper en plusieurs parties. Pour cela, il existe des 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>`. Que mettre dans le pied de tableau ? Généralement, si c'est un long tableau, vous y recopiez les cellules d'en-tête. Cela permet de voir, même en bas du tableau, à quoi se rapporte chacune des colonnes.

### Exemple :

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

```
<body>
  <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>
  <tr>
    <td>François</td>
    <td>43 ans</td>
    <td>France</td>
  </tr>
  <tr>
    <td>Martine</td>
    <td>34 ans</td>
    <td>France</td>
  </tr>

```



```
<tr>  
  <td>Jonathan</td>  
  <td>13 ans</td>
```





```

        <td>Australie</td>
    </tr>
    <tr>
        <td>Xu</td>
        <td>19 ans</td>
        <td>Chine</td>
    </tr>
</tbody>
</table>
</body>
</html>

```

**NB :** Il n'est pas obligatoire d'utiliser ces trois balises (**<thead>**, **<tbody>**, **<tfoot>**) dans tous les tableaux. En fait, vous vous en servirez surtout si votre tableau est assez gros et que vous avez besoin de l'organiser plus clairement.

Pour les « petits » tableaux, vous pouvez garder sans problème l'organisation plus simple que nous avons vue au début.

### Les formulaires :

Toute page HTML peut être enrichie de formulaires interactifs, qui invitent vos visiteurs à renseigner des informations : saisir du texte, sélectionner des options, valider avec un bouton... tout est possible !

#### 1. Créer un formulaire :

Lorsqu'il vous prend subitement l'envie d'insérer un formulaire dans votre page HTML, vous devez pour commencer écrire une balise **<form>** **</form>**. C'est la balise principale du formulaire, elle permet d'en indiquer le début et la fin.

#### Exemple :

```

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

```

```
</head>
<body>
  <p>Texte avant le formulaire</p>
  <form>
    <p>Texte à l'intérieur du formulaire</p>
  </form>
  <p>Texte après le formulaire</p>
</body>
</html>
```

On va prendre un exemple pour que les choses soient claires. 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. Eh bien c'est là le problème, ou plutôt les problèmes, que l'on va se poser :

- 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 ?

Pour fournir les réponses 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.

On va donc maintenant compléter la balise **<form>** avec les deux attributs qu'on vient de voir.

Pour method, nous allons mettre la valeur post.

Pour action, nous allons taper le nom d'une page fictive en PHP (traitement.php). C'est cette page qui sera appelée lorsque le visiteur cliquera sur le bouton d'envoi du formulaire.

### Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Formulaire </title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <form method="post" action="traitement.php">
      <p>Texte à l'intérieur du formulaire</p>
    </form>
    <p>Texte après le formulaire</p>
  </body>
</html>
```

## 2. Les zones de saisie basiques :

Nous allons passer en revue les différentes balises HTML permettant de saisir du texte dans un formulaire.

Il faut savoir qu'il y a 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...).

### **a) Zone de texte monoligne :**

La figure suivante montre à quoi ressemble une zone de texte monoligne.

Votre pseudo :  Une zone de texte monoligne

Pour insérer une zone de texte dans une ligne, on va utiliser la balise **<input />**. Pour créer une zone de texte à une ligne, on doit écrire :

```
<input type="text" />
```

Ce n'est pas encore suffisant : il faut donner un nom à votre zone de texte. Ce nom n'apparaît pas sur la page mais il vous sera indispensable par la suite.

Pour donner un nom à un élément de formulaire, on utilise l'attribut name.

### **Syntaxe :**

```
<input type="text" name="pseudo" />
```

### **Exemple:**

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

```

</head>
<body>
  <p>Texte avant le formulaire</p>
  <form method="post" action="traitement.php">
    <p><input type="text" name="pseudo" /></p>
  </form>
</body>
</html>

```

### **b) Les libellés :**

Cette zone de texte est bien jolie mais si votre visiteur tombe dessus, il ne sait pas ce qu'il doit écrire. C'est justement le rôle de la balise `<label>` :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <form method="post" action="traitement.php">
      <p>
        <label>Votre pseudo</label> :
        <input name="pseudo" />
      </p>
    </form>
  </body>
</html>

```


type="text"

Mais cela ne suffit pas. Il faut lier le label à la zone de texte.

Pour ce faire, on doit donner un nom à la zone de texte, non pas avec l'attribut Name mais avec l'attribut id (que l'on peut utiliser sur toutes les balises). Pour lier le label au champ, il faut lui donner un attribut for qui a la même valeur que l'id du champ.

### Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <form method="post" action="traitement.php">
      <p>
        <label for="pseudo">Votre pseudo</label> : <input
type="text" name="pseudo" id="pseudo" />
      </p>
    </form>
  </body>
</html>
```



### 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 :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <form method="post" action="traitement.php">
      <p>
        <label for="pseudo">Votre pseudo :</label>
        <input type="text" name="pseudo" id="pseudo"
placeholder="mon pseudo" size="30"
maxlength="10" />
      </p>
    </form>
  </body>
</html>
```

The logo for eMalischool features a green graduation cap icon on the left, followed by the text 'eMalischool' in a green, sans-serif font.

### **c) Zone de mot de passe :**

Vous pouvez facilement faire en sorte que la zone de texte se comporte comme une « zone de mot de passe », c'est-à-dire une zone où on ne voit pas à l'écran les caractères saisis. Pour créer ce type de zone de saisie, utilisez l'attribut `type="password"`.

Nous complétons notre formulaire. Il demande maintenant au visiteur son pseudo et son mot de passe :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <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>
  </body>
</html>

```

#### **d) Zone de texte multiligne :**

Pour créer une zone de texte multiligne, on change de balise : nous allons utiliser `<textarea>` `</textarea>`.

Comme pour tout autre élément du formulaire, il faut lui donner un nom avec `name` et utiliser un label qui explique de quoi il s'agit.

```

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

```



```
</head>
<body>
  <p>Texte avant le formulaire</p>
  <form method="post" action="traitement.php">
    <p>
      <label for="ameliorer">Comment pensez-vous que je
pourrais améliorer mon site ?</label><br />
      <textarea name="ameliorer" id="ameliorer"></textarea>
    </p>
  </form>
</body>
</html>
```

#### **e) E-mail :**

Vous pouvez demander à saisir une adresse e-mail :

```
<input type="email" />
```

Le champ vous semblera a priori identique mais votre navigateur sait désormais que l'utilisateur doit saisir une adresse e-mail. Il peut afficher une indication si l'adresse n'est pas un e-mail.

#### **f) Une URL :**

Avec le type url, on peut demander à saisir une adresse absolue (commençant généralement par http://) :

```
<input type="url" />
```

Même principe : si le champ ne vous semble pas différent sur votre ordinateur, sachez que celui-ci comprend bel et bien que le visiteur est censé saisir une adresse.

### **g) Numéro de téléphone :**

Ce champ est dédié à la saisie de numéros de téléphone :

```
<input type="tel" />
```

### **h) Nombre:**

Ce champ permet de saisir un nombre entier :

```
<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...).

### **i) Un curseur:**

Le type range permet de sélectionner un nombre avec un curseur (aussi appelé *slider*), comme à la figure suivante :

```
<input type="range" />
```

### **j) Couleur :**

Ce champ permet de saisir une couleur :

```
<input type="color" />
```

### **k) Date :**

Différents types de champs de sélection de date existent :

- date : pour la date (05/08/1985 par exemple) ;
- 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).

```
<input type="date" />
```

### **l) Recherche:**

On peut créer un champ de recherche comme ceci :

```
<input type="search" />
```

Le navigateur décide ensuite comment afficher le champ de recherche. Ainsi, il peut ajouter une petite loupe au champ pour signifier que c'est un champ de recherche et éventuellement mémoriser les dernières recherches effectuées par le visiteur.

### **3. Les éléments d'options :**

HTML vous offre une ribambelle d'éléments d'options à utiliser dans votre formulaire. Ce sont des éléments qui demandent au visiteur de faire un choix parmi une liste de possibilités. Nous allons passer en revue :

- les cases à cocher ; □
- les zones d'options ; □
- les listes déroulantes.

### **a. Les cases à cocher:**

Pour créer une case à cocher, nous allons réutiliser la balise `<input />`, en spécifiant cette fois le type checkbox :

```
<input type="checkbox" name="choix" />
```

### **Exemple:**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <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>
```

```
</body>
</html>
```

N'oubliez pas de donner un nom différent à chaque case à cocher, cela vous permettra d'identifier plus tard lesquelles ont été cochées par le visiteur. Enfin, sachez que vous pouvez faire en sorte qu'une case soit cochée par défaut avec l'attribut checked :

```
<input type="checkbox" name="choix" checked />
```

### ***b. Les zones d'options :***

Les zones d'options vous 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.

#### **Exemple :**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <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>
</body>
</html>

```

### c. Les listes déroulantes :

Les listes déroulantes sont un autre moyen élégant de faire un choix parmi plusieurs possibilités. Le fonctionnement est un peu différent. 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.

**Exemple :**

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>

```

```

<p>Texte avant le formulaire</p>
<form method="post" action="traitement.php">
  <p>
    <label for="pays">Dans quel pays
    habitez-vous ?</label><br />
    <select name="pays" id="pays">
      <option value="franke">France</option>
      <option value="Espagne">Espagne</option>
      <option value="Italie">Italie</option>
      <option value="Royaume-Unis">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>
</body>
</html>

```

Si vous voulez qu'une option soit sélectionnée par défaut, utilisez cette fois l'attribut `selected` :

```

<option value="Canada" selected>Canada</option>

```

Vous pouvez aussi grouper vos options avec la balise `<optgroup>` `</optgroup>`.

**Exemple :**

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <p>Texte avant le formulaire</p>
    <form method="post" action="traitement.php">
      <p>
        <label for="pays">Dans quel
        pays ?</label><br />
        habitez-vous
        <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>
  </body>
</html>

```





#### **d. 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>**.

#### **Exemple :**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Formulaire</title>
  </head>
  <body>
    <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="prénom">Quel est votre prénom ?</label>
```



```

        <input type="text" name="prénom" id="prénom" />
        <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>
    <label for="precisions">Si "Autre", veuillez préciser
    :</label>
    <textarea name="precisions" id="precisions" cols="40"
    rows="4"></textarea>
    </p>
</fieldset>
</form>
</body>
</html>

```

### **e. Sélectionner automatiquement un champ :**

Vous pouvez placer automatiquement le curseur dans l'un des champs de votre formulaire avec l'attribut autofocus. Dès que le visiteur chargera la page, le curseur se placera dans ce champ.

```
<input type="text" name="prénom" id="prénom" autofocus />
```

#### **f. Rendre un champ obligatoire :**

Vous pouvez faire en sorte qu'un champ soit obligatoire en lui donnant l'attribut `required`.

```
<input type="text" name="prénom" id="prénom" required />
```

Le navigateur indiquera alors au visiteur, si le champ est vide au moment de l'envoi, qu'il doit impérativement être rempli.

#### **g. 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.

#### **Exemple :**

```
<input type="submit" value="Envoyer" />
```

### 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.

### **La vidéo et l'audio:**

Depuis l'arrivée de Youtube et Dailymotion, il est devenu courant aujourd'hui de regarder des vidéos sur des sites web. Il faut dire que l'arrivée du haut débit a aidé à démocratiser les vidéos sur le Web.

Cependant, aucune balise HTML ne permettait jusqu'ici de gérer la vidéo. Il fallait à la place utiliser un **plugin**, comme Flash.

Encore aujourd'hui, Flash reste de loin le moyen le plus utilisé pour regarder des vidéos sur Youtube, Dailymotion, Vimeo et ailleurs. Mais utiliser un plugin a de nombreux défauts : on dépend de ceux qui gèrent le plugin (en l'occurrence,

l'entreprise Adobe, qui possède Flash), on ne peut pas toujours contrôler son fonctionnement, il y a parfois des failles de sécurité... Au final, c'est assez lourd. C'est pour cela que deux nouvelles balises standard ont été créées en HTML5 : **<vidéo>** et **<audio>** !

Les formats audio et vidéo :

### **1. Les formats audio :**

Pour diffuser de la musique ou n'importe quel son, il existe de nombreux formats. La plupart d'entre eux sont compressés (comme le sont les images JPEG, PNG et GIF) ce qui permet de réduire leur poids :

- 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.

Aucun navigateur ne gère tous ces formats à la fois. Retenez surtout la compatibilité pour les MP3 et OGG :

Navigateur	MP3	OGG
Internet Explorer	Oui	-
Chrome	Oui	Oui
Firefox	-	Oui
Safari	Oui	-
Opera	-	Oui

### **2. 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.

Ici aussi, il n'y a pas de la compatibilité des codecs vidéo sur les différents navigateurs :

Navigateur	H.264	Ogg Theora	WebM
Internet Explorer	Oui	-	Oui
Chrome	-	Oui	Oui
Firefox	-	Oui	Oui
Safari	Oui	-	-
Opera	Oui	Oui	Oui

Là encore, aucun format ne sort du lot. Il est conseillé de proposer chaque vidéo dans plusieurs formats pour qu'elle soit lisible sur un maximum de navigateurs.

### **Insertion d'un élément audio:**

En théorie, il suffit d'une simple balise pour jouer un son sur notre page :

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

En pratique, c'est un peu plus compliqué que cela.

Si vous testez ce code... vous ne verrez rien ! En effet, le navigateur va seulement télécharger les informations générales sur le fichier (on parle de métadonnées) mais il ne se passera rien de particulier.

Vous pouvez compléter la balise des attributs suivants :

- **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement. Cela peut sembler indispensable, et vous vous demandez peut-être pourquoi cela n'y figure pas par défaut, mais certains sites web préfèrent créer eux-mêmes leurs propres boutons et commander la lecture avec du JavaScript.
- **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.



## Syntaxe :

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

## Insertion d'une video:

Il suffit d'une simple balise **<video>** pour insérer une vidéo dans la page :

```
<video src="intel.webm"></video>
```

Rajoutons quelques attributs (la plupart sont les mêmes que pour la balise **<audio>**) :

- **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 ou d'une image peu représentative de la vidéo, je vous conseille d'en créer une ! 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. Cet attribut peut prendre les valeurs :
  - **auto (par défaut)** : le navigateur décide s'il doit précharger toute la vidéo, uniquement les métadonnées ou rien du tout.



- **metadata** : charge uniquement les métadonnées (durée, dimensions, etc.).
- **none** : pas de préchargement. Utile si vous souhaitez éviter le gaspillage de bande passante sur votre site.

Voici un code un peu plus complet :

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

### 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.

### La liste des balises HTML :

#### **Balises de premier niveau**

Les balises de premier niveau sont les principales balises qui structurent une page HTML. Elles sont indispensables pour réaliser le « code minimal » d'une page web.

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

### **Balises d'en-tête**

Ces balises sont toutes situées dans l'en-tête de la page web, c'est-à-dire entre


Balise	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



### **Balises de structuration du texte**

<head> et </head> :

Balise	Description
<abbr>	Abréviation
<blockquote>	Citation (longue)
<cite>	Citation du titre d'une œuvre ou d'un événement
<q>	Citation (courte)
<sup>	Exposant
<sub>	Indice

<strong>	Mise en valeur forte
<em>	Mise en valeur normale
<mark>	Mise en valeur visuelle
<h1>	Titre de niveau 1
<h2>	Titre de niveau 2
<h3>	Titre de niveau 3
<h4>	Titre de niveau 4
<h5>	Titre de niveau 5
<h6>	Titre de niveau 6
<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
 	Retour à la ligne
<p>	Paragraphe
<hr />	 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

### ***Balises de listes***

Cette section énumère toutes les balises HTML permettant de créer des listes (listes à puces, listes numérotées, listes de définitions...)

Balise	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

### ***Balises de tableau***

Balise	Description
<table>	Tableau
<caption>	Titre du tableau
<tr>	Ligne de tableau

### Balises de formulaire

<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



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

### Balises sectionnantes

Ces balises permettent de construire le squelette de notre site web.

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

### **Balises génériques**

Les balises génériques sont des balises qui n'ont pas de sens sémantique. En effet, toutes les autres balises HTML ont un *sens* : <p> signifie « Paragraphe », <h2> signifie « Sous-titre », etc.

Parfois, on a besoin d'utiliser des balises génériques (aussi appelées balises universelles) car aucune des autres balises ne convient. On utilise le plus souvent des balises génériques pour construire son design.

Il y a deux balises génériques : l'une est inline, l'autre est block.

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

Ces balises ont un intérêt uniquement si vous leur associez un attribut class, id ou style :

- **class** : indique le nom de la classe CSS à utiliser.
- **id** : donne un nom à la balise. Ce nom doit être unique sur toute la page car il permet d'identifier la balise. Vous pouvez vous servir de l'ID pour de nombreuses choses, par exemple pour créer un lien vers une ancre, pour un style CSS de type ID, pour des manipulations en JavaScript, etc.
- **style** : cet attribut vous permet d'indiquer directement le code CSS à appliquer. Vous n'êtes donc pas obligés d'avoir une feuille de style à part,

vous pouvez mettre directement les attributs CSS. Notez qu'il est préférable de ne pas utiliser cet attribut et de passer à la place par une feuille de style externe, car cela rend votre site plus facile à mettre à jour par la suite. Ces trois attributs ne sont pas réservés aux balises génériques : vous pouvez aussi les utiliser sans aucun problème dans la plupart des autres balises.

Powered by Berthec Corporation

