Structure de base d'un fichier HTML5

Le squelette minimal d'un fichier HTML5 se présente ainsi :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>titre page</title>
  </head>
  <body>
    <!-- Contenu de la page -->
  </body>
</html>
```

Une balise ouvrante est un nom entre les caractères < et >. Une balise fermante contient un / entre le < et le nom de la balise : <html> est une balise ouvrante, </body> est une balise fermante.

Certaines balises peuvent contenir du texte ou d'autres balises. Dans ce cas, les balises doivent être fermées dans sens contraires d'ouverture : <body> est contenu dans <html>, il faut écrire </body> avant </html>

- <!DOCTYPE html> indique au navigateur que le document utilise la dernière version d'HTML (HTML5).
- <html lang="fr"> la balise racine qui englobe tout le contenu. L'attribut lang="fr" précise que le contenu est en français.
- <head> contient les informations sur la page (métadonnées) qui ne sont pas affichées directement, comme l'encodage et le titre.
- <meta charset="UTF-8"> définit le jeu de caractères utilisé (UTF-8), indispensable pour afficher correctement les accents et caractères spéciaux.
- <title> spécifie le titre de la page qui s'affiche dans l'onglet du navigateur.
- <body> contient tout le contenu visible (texte, images, liens, etc.) de la page.

Les balises essentielles et leur utilité

Voici un aperçu des principales balises HTML et de leur rôle :

- <**section**> utilisée pour définir une section thématique d'un document (exemple : chapitre, partie d'un article).
- <**article**> représente un contenu autonome (comme un article de blog ou une nouvelle) pouvant être diffusé indépendamment. Une section peut être composée d'articles.
- <**header**> regroupe le contenu d'en-tête d'une section ou d'une page (titre, logo, menu principal). S'applique à body, section, article
- <**footer**> sert d'élément de pied de page pour ajouter des informations complémentaires comme le copyright, des sources, etc.. S'applique à body, section, article
- <**h1**>, <**h2**>, … , <**h6**> sont pour désigner les titres. <h1> est le plus gros titre, <h2> est son sous-titre, <h3> est le sous-titre de <h2>, etc.
- <**p**> balise pour un paragraphe de texte.
- <**pre**> affiche le texte en gardant la mise en forme (espaces, retours à la ligne), souvent utilisé pour du code.

- <**div**> élément de division ou conteneur générique. Il n'a pas de signification sémantique mais aide à structurer la page.
- <**aside**> contenu complémentaire ou en marge du contenu principal (exemple : barre latérale, publicités).
- <**nav**> contient les liens de navigation permettant de passer d'une section à une autre du site.
- <**a**> balise de lien hypertexte. On y ajoute l'attribut href pour définir la destination du lien. *Exemple :* <a **href**="https://exemple.com">Visitez Exemple
- <**img**> insère une image. On doit préciser l'attribut src (source de l'image) et idéalement l'attribut alt pour décrire l'image. *Exemple :* <img **src**="image.jpg" alt="Description de l'image">
  img est une balise vide : elle ne contient pas de texte, il ne faut pas la fermer
- <**strong**> met en évidence du texte (généralement affiché en gras) pour indiquer son importance.
- <**ul**> déclare une liste non ordonnée (à puces). Ul contient des li.
  <**ol**> déclare une liste ordonnée (numérotée). Ol contient des li.
    - <**li**> Représente un élément de liste à l'intérieur d'une <ul> ou <ol>.
- <**dl**> définit une liste de définitions ou de descriptions. Il contient une suite de dt/dd
    - <**dt**> Spécifie le terme à définir dans une liste de définitions.
    - <**dd**> Apporte la description ou définition du terme précédemment défini avec <**dt**>.
- <**br**> insère un saut de ligne à l'intérieur d'un paragraphe sans créer de nouveau bloc de texte.
  br est une balise vide : elle ne contient pas de texte, il ne faut pas la fermer

Certaines balises ont des attributs nécessaires (comme href pour <a> ou src pour <img>). Les attributs sont mis dans la balise ouvrante. Certains attributs sont communs à quasiment toutes les balises :

- id qui permet de nommer de façon unique une balise (id pour identifiant). Cela sert pour le css (et javascript). Ex : <p id="reponse">
- class qui permet d'associer une ou des classes css à la balise. Ex : <p class="question">

Il est conseillé d'utiliser autant que possible des balises sémantiques (comme <header>, <footer>, <article>, <section>, <nav>, et <aside>) pour clarifier la structure de vos contenus.

Les bases du CSS

CSS signifie *Cascading Style Sheets*. Il s'agit d'un langage qui permet de définir la présentation (couleurs, marges, polices, etc.) des éléments écrits en HTML. Son fonctionnement repose sur trois concepts :

- **Sélecteur** : il cible l'élément HTML auquel on souhaite appliquer du style.
- **Propriétés** : ce sont les caractéristiques que l'on veut modifier (par exemple, la couleur du texte, la taille de la police, etc.).
- **Valeurs** : elles définissent la valeur attribuée à une propriété (par exemple, red pour une couleur, 16px pour une taille).

La **syntaxe** de base s'écrit ainsi :

```
css
sélecteur {
  propriété: valeur;
  propriété: valeur;
}
```

Exemple :

```
p {
  color: blue;
  font-size: 16px;
}
```

Pour appliquer une feuille de style externe à votre document HTML, insérez la balise <link> dans l'élément <head> du fichier HTML :

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
```

Cette balise indique au navigateur de charger et d'appliquer les styles définis dans le fichier styles.css (c'est le nom du fichier qui contient le css). Le nom du fichier peut varier, par contre le reste est obligatoire (nom de la balise link et rel).

Les unités de mesures en CSS peuvent être :

- **px (pixels)**, une unité fixe, idéale pour les tailles précises. *Exemple :* width: 200px;
- **em**, unité relative à la taille de la police de l'élément parent. *Exemple :* font-size: 1.2em;
- **rem**, unité relative à la taille de la police définie sur l'élément racine (html). *Exemple :* padding: 2rem;
- **% (pourcentage)** relative à la dimension de l'élément parent. *Exemple :* width: 50%;

Les sélecteurs simples

Voici quelques sélecteurs fondamentaux pour cibler vos éléments :

- **Sélecteur par élément** : cible tous les éléments html d'un même type. *Exemple : « p »* pour désigner tous les paragraphes.
- **Sélecteur par classe** : cible les éléments html ayant une classe spécifique. *Exemple :* « .bouton » pour désigner toutes les balises html qui ont dans la balise ouvrante l'attribut class="bouton"
- **Sélecteur par id** : cible un élément unique identifié par son id (précédé d'un #). *Exemple :* « #unique »
- **Sélecteur descendant (élément dans élément)** : cible un élément contenu dans un autre. *Exemple : « nav a » pour désigner* tous les <a> à l'intérieur d'un <nav>
- **Sélecteur universel** ; le symbole * s'applique à **tous** les éléments html.
- **Combinaison de sélecteurs** : vous pouvez combiner plusieurs sélecteurs pour un ciblage plus précis. *Exemple :* « section .important » pour désigner toutes les sections de classe « important ».
- element**:hover** : permet de faire un style particulier quand la souris survole l'élément. *Exemple :* « p:hover » permet de changer le style des paragraphes lorsque la souris est placée au-dessus d'eux.

Propriétés CSS courantes

- **color** Définit la couleur du texte.

```
p {   color: #333333; }
```

- **background** Permet de définir un fond en utilisant soit une couleur, soit une image. *Exemples :*

```
/* Fond avec une couleur */
body {
  background: #f0f0f0;
}
/* Fond avec une image */
.banniere {
  background: url('banniere.jpg');
}
```

- **border** définit la bordure d'un élément (largeur, style, couleur).

```
div {   border: 1px solid black; }
```

- **border-radius** arrondit les coins de la bordure.

```
div {   border-radius: 5px; }
```

- **box-shadow** ajoute une ombre autour d'un élément.

```
div {
  box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);
}
```

- **padding** définit l'espace à l'intérieur de l'élément, entre son contenu et sa bordure.
- **margin** définit l'espace extérieur autour d'un élément.
  Note : la valeur « auto » pour les marges gauches et droites permet de centrer l'élément s'il est moins large que là où il est placé.

```
div {
  padding: 5px;
  margin: 10px auto;
}
```

- **font-weight** détermine l'épaisseur de la police (par exemple, normal ou bold).
- **font-size** définit la taille de la police.
- **font-family** spécifie la police de caractères à utiliser.
- **text-align** alignement du texte (ex. : left, center, right).

```
section.important {
     font-weight: bold;
     font-size: 1.2em;
     font-family: Arial, sans-serif;
     text-align: center;
}
```

Propriétés pour la mise en page

- **width et height**
  width définit la largeur d'un élément. height permet de définir la hauteur d'un élément. Elles peuvent être spécifiées en unités fixes (comme px) ou en unités relatives (comme em, %). Ces propriétés permettent de contrôler l'espace horizontal et vertical occupé par un élément.
  La valeur « fit-content » permet d'adapter la largeur ou la hauteur à la valeur nécessaire.

```
img {
  width: 300px; /* largeur fixe */
}
```

- **float**
  La propriété float permet de faire sortir un élément du flux normal du document. Lorsqu'un élément est flotté, le contenu environnant (généralement du texte ou des images) s'enroule autour de celui-ci. Valeurs courantes :
  left : L'élément flotte à gauche.
  right : L'élément flotte à droite.
  l'élément « flottant » doit être placé dans le html avant les éléments qui viendront se mettre à sa gauche ou sa droite.
  avec float, il est conseillé d'utiliser une marge du côté opposé

```
img {
  float: left;
  margin-right: 10px; /* espace « opposé » au float */
}
```

- **position**
  La propriété position permet de définir le mode de positionnement d'un élément dans la page. Selon la valeur utilisée, l'élément peut rester dans le flux normal ou être retiré pour être positionné de manière absolue. Valeurs possibles :
  - static : Valeur par défaut. L'élément suit le flux normal du document.
  - relative : L'élément reste dans le flux, mais ses coordonnées peuvent être ajustées par rapport à sa position d'origine à l'aide de top, right, bottom, et left.
  - absolute : L'élément est retiré du flux et positionné par rapport à son ancêtre le plus proche qui a une position autre que static (ou par rapport à la fenêtre si aucun ancêtre n'est positionné).
  - fixed : Positionne l'élément par rapport à la fenêtre du navigateur. Il reste fixe lors du défilement de la page.

  Avec les valeurs relative, absolute ou fixed, la position de l'élément est à ajuster avec **top, right, bottom, et left**.

```
aside {
  position: absolute;
  top: 20px;          /* à 20 pixel du haut du document */
  left: 30px;         /* à 30 pixel de la gauche du document */
}
```