Fiches de Synthèse : HTML et CSS

HTML (Hypertext Markup Language)

1. Rôle du HTML

Langage de structure des pages web. Il décrit le contenu (titres, paragraphes, images, liens, etc.), pas la mise en forme (c'est le rôle du CSS). Utilise des balises entourées de < >.

2. Structure minimale d'une page HTML

```html
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8">
        <title>Titre de la page</title>
    </head>
    <body>
        <h1>Bienvenue</h1>
        <p>Ceci est un paragraphe.</p>
    </body>
</html>
```

3. Principales balises

Les balises sont le fondement du html, on les trouve partout et on peut facilement les reconnaitre car elles sont toujours encadrées de « <> » et doivent toujours être fermées lorsqu'elles sont ouvertes avec « </> » et leur nom après le slash. On a plein de différentes balises, il y en a qu'on ne va probablement jamais utiliser de notre vie même en tant que développeur web. Mais ils ont au contraire que tout le monde y en utilisé, donc voici une liste des balises les plus utilisées en html.

- ***<h1> à <h6>*** Crée un titre
- ***<p>*** Crée un paragraphe
- ***<a>*** Permet de mettre un lien
- ***<img>*** Permet de mettre une image fonctionne un peu comme « ***<a> »***
- ***
*** Permet de faire une liste non ordonnée
- ***
*** Permet de faire une liste ordonnée
- ***<table><tr><th>*** Crée un tableau
- ***<form>*** Crée un formulaire
- *** *** Dvise la page
- *** / / *** utilisé pour la mise en forme du texte

Voici un exemple d'utilisation de ces balises sauf « ***<a> »*** qui a une section dédiée à la fin :

```html
<h1>Titre</h1>
<h2>Sous Titre</h2>

<p>Ceci est un paraagraphe</p>

<img src="imageTresStylée.png" alt="c'est stylé">

<ul><li>Pomme</li><li>Banane</li></ul>

<ol><li>Element1</li><li>Element2</li></ol>

<table><tr>ligne1<th>Cellule pour le titre</th></tr><tr>Ligne2<td>Cellule normale</td></tr></table>

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

<div>Bloc</div>

<strong>Ce texte en gras</strong>
<em>Ce texte en italique</em>
<span class="rouge">Pemet de selectionner ce bout de texte de lui appliquer une mise en forme speciale</span>
```

4. Attributs utiles

Un attribut est une information supplémentaire placée à l'intérieur d'une balise HTML pour modifier son comportement, son apparence, ou pour lui ajouter des données.

Des attributs utiles sont :

- ***id*** □ Donne un identifiant unique (pour le css ou le javascript)
- ***class*** □ Regroupe des éléments dans une class ou « catégorie » (pour appliquer un style commun par exemple)
- ***src*** □ permet de spécifier la source d'un élément (par exemple une image, une vidéo etc.)
- ***href*** □ Spécifie cette fois l'adresse du lien (par exemple pour aller sur une autre page)
- ***alt*** □ Permet de donner une étiquette a l'image, c'est un petit texte qui s'affiche quand on laisse notre curseur sur l'image)
- ***type*** □ Définit le type d'un élément (un input, un bouton etc.)
- ***value*** □ Définit une valeur affichée ou renvoyée

Un exemple d'utilisation d'attributs :

```html
<img src="logo.png" alt="Logo du site" class="logo">
```

5. Liens internes et externes (balise <a> surtout)

La balise <a> (pour anchor, "ancre") sert à créer un lien cliquable.

Ce lien peut :

- envoyer vers une autre page web (externe ou interne),
- ou vers un endroit précis de la même page

```html
<a href="https://wikipedia.org">Wikipedia</a>

<a href="contact.html">Page de contact</a>

<a href="#bas">Aller en bas</a>
<h2 id="bas">Bas de la page</h2>
```

Ici la première balise a permis d'aller sur un site externe (qui est ici Wikipédia)

La deuxième est un lien interne au site (il renvoie à la page nommée « contact.html » qui fait parti de notre site on peut noter qu'ici ce n'est pas une url mais un nom de fichier)

La troisième balise permet de naviguer dans la même page a un autre endroit (c'est pratique pour les sites qui n'ont qu'une page permet d'aller tout en bas tout en haut etc.)

CSS (Cascading Style Sheets)

1. Rôle du CSS

Langage de mise en forme des pages web. Permet de modifier les couleurs, tailles, polices, marges, positions, etc.

2. Trois façons d'ajouter du CSS

La première :

- **Interne :** dans <style> quand c'est pour des petites modifications rapides (à éviter)
- **Externe :** dans un fichier a part (style.css par exemple) la meilleure méthode, permet de tout bien ranger et repérer
- **En ligne :** La pire méthode consiste à tout faire en une ligne

3. La structure de base

Un fichier CSS est composée de trois règles :

- **Sélecteur :** Indique a quels éléments du fichier html il faut appliquer ce style
- **Propriétés :** c'est le style qu'on veut changer (la couleur, la taille etc.)
- **Valeur :** le réglage de la propriété choisie

Ce qui nous donne :

```
sélecteur {
    propriété: valeur;
}


p {
    color: blue;
    font-size: 18px;
}
```

4. Principales propriétés

Comme on l'a vu les propriétés permettent de changer divers choses donc voici une liste des propriétés les plus utilisées :

- **Couleur :** ***color*** Permet de changer la couleur on peut saisir soit un code couleur le nom de la couleur ou un code couleur rgb()
- **Texte :** On peut changer la taille avec ***font-size***, on peut changer la police avec ***font-family*** et ensuite on peut l'aligner a droite au centre ou à gauche avec ***text-align***
- **Bordures :** On peut régler l'espacement à l'extérieur ou à entre les éléments avec ***margin*** (pour l'extérieur) et ***padding*** (pour l'intérieur)
- **Dimensions** : avec ***width*** ou ***height*** (***width*** pour la largeur et ***height*** pour la hauteur)
- **Positionnement** : avec ***display postion*** ou ***float***

5. Selecterurs

Donc les sélecteurs sont très importants et peuvent être longs a mettre en place c'est pour cela qu'il y a des sélecteurs spéciaux qui peuvent sauver du temps :

Il y a d'abord la base on prend par exemple ***p{}*** qui sélectionnera tous les paragraphes de tout le html ensuite il y a le ***.info*** qui sélectionnera tous les éléments de la classe ***important*** (avec un class = ***« info »***) ***#titre{}*** qui permet de sélectionner les éléments avec ***l'id*** titre donc pas forcement une ***classe*** ensuit il y ***div p{]*** qui permet de sélectionner seulement les éléments p dans des ***div*** (peut marcher avec ***div form{}*** par exemple et plein d'autres)

Ensuite il y a aussi des priorités car imaginons que un élément correspond a deux classes, qu'elle classe va-t-il prendre : C'est assez simple en fait le plus spécifique sera le prioritaire (par exemple ***id > classes > balises***) ensuite le style écrit le plus tard est prioritaire et sinon

on peut forcer la priorité avec ***!important.***