

Premiers pas en HTML

Le but de cette série d'exercices est de présenter la première technologie amenant à la création de sites Internet.

Sommaire

1	Présentation / Vocabulaire	2
2	Une page de base HTML :	3
3	Ajouter du style décoratif	3
3.1	L'attribut STYLE	3
3.2	CSS : la décoration réutilisable.....	4
3.3	CLASSES dans les feuilles de style	5
4	Lien HyperText : balise <a>	5
5	Ajouter une image GIF	6
6	Notion de SITE ADAPTATIF (<i>Responsiv Design</i>).....	7
7	Aller plus loin en CSS	8
8	Publier son site.....	8

1 Présentation / Vocabulaire

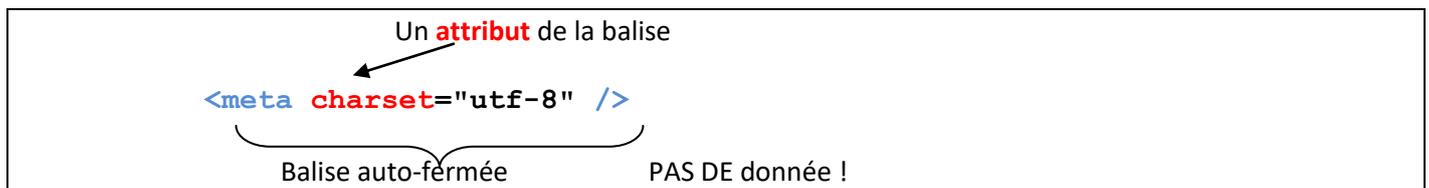
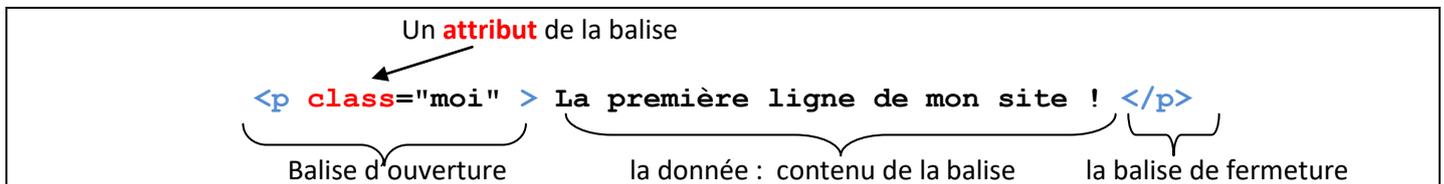
HTML : Hyper-Text Markup Language

C'est un « langage » interprété par un Navigateur Web (Internet Explorer, Edge, Safari, Mozilla, Chrome, Opera...).

Chaque <balise> sert à repérer (à marquer = *markup*) une position ou un contenu (texte, image, ...).

Les balises normalisées sont connues du navigateur et il sait comment traduire pour constituer l'affichage de la page web.

Vocabulaire d'une balise HTML :



CSS : Cascading Style Sheet

Ce sont des « feuilles de style » pour ajouter des options de décoration aux balises (couleur, taille, position, effet ...). On peut même intégrer à une feuille un style conditionnel en fonction d'une règle @media.

JavaScript (Ne pas confondre avec le langage Java)

Langage de programmation interprété par le navigateur.

HTML produit des pages statiques, figées.

Le JavaScript permet de modifier dynamiquement une page en fonction d'évènements liés au clavier, à la souris, à l'horloge système, ...

Les technologies complétant JavaScript sont par exemple Ajax, WebSocket, {Fetch}, SSE, IndexedDB, Worker, JSON, XML...

Et JQuery ? Ce module a rendu de grands services en ajoutant des fonctionnalités qui manquaient à JavaScript. Maintenant, les progrès de JavaScript rendent l'utilisation de JQuery inutile.

Autres technologies non étudiées dans cet exercice : **PHP, ASP, NodeJS** :

Langages de programmation script interprétés côté serveur (Serveur web par exemple).

Le script produit un affichage de type HTML, XML, JSON, ou Texte à destination du navigateur.

Un script exécuté côté serveur reste invisible pour le navigateur (protection intellectuelle, sécurité des informations).

Pour votre formation individuelle, nous vous recommandons les sites suivants :

<https://www.w3schools.com/html>

<https://www.alsacreations.com/tutoriels/>

Un bon site pour le CSS : <http://www.css-faciles.com/>

Pour créer des pages web, nous vous suggérons d'utiliser le logiciel gratuit **NotePad++**

Pour tester vos pages web, nous utiliserons le navigateur **Mozilla Firefox** qui est d'une bonne compatibilité html5.

Le problème de la compatibilité des navigateurs est abordé ici :

<https://developers.google.com/search/docs/advanced/guidelines/browser-compatibility?hl=fr>

2 Une page de base HTML :

a) Créez un fichier **index.html** contenant le code HTML proposé ici :

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

  <body>

    <h1> Les Technologies du Web </h1>
    <p> La premiere ligne de mon site ! </p>
    <p> La deuxieme ligne de mon site ! </p>

  </body>
</html>
```

b) Testez ce code localement sur votre ordinateur en ouvrant ce fichier avec un navigateur internet.

Explication : Tant qu'on utilise seulement du HTML et du JavaScript, il n'y a pas besoin d'un serveur Web pour tester nos pages. Le navigateur sait parfaitement les lire sous forme de fichier local.

c) Remplacez la balise <h1> par <h2> ou <h3> et observez le résultat.

Explications sur la structure d'une page HTML :

- La balise <!DOCTYPE html> indique que la page est en HTML version 5.
- La balise <html> englobe toute le contenu HTML. C'est la balise RACINE.
- La balise <head> contient des informations de configuration de la page. Ces informations sont contenues dans ce qu'on appelle l'en-tête (*header*) du protocole http (c'est le protocole des pages Web).
- La balise <body> contient le texte qui sera affiché par la page web.
- Chaque balise ouvert doit être fermée ensuite : <body> </body>
- Les balises <h1><h2> ... font penser à des balises de « décoration » mais en fait il s'agit de balises de « structuration » de la page, un moyen de repérer les titres pour les robots des moteurs de recherche.

3 Ajouter du style décoratif

3.1 L'attribut STYLE

Il y a une quantité de réglages pour améliorer la présentation.

On appelle ça les **propriétés CSS**. (On expliquera tout à l'heure cette histoire de CSS)

Vous en avez un aperçu ici : <http://www.css-faciles.com/proprietes-css-liste-alphabetique.php>

Il est possible d'intégrer ces « propriétés » dans une balise grâce à l'**attribut style** :

Exemple : Modifiez la balise <p> « premiere ligne » et testez le résultat :

```
<p style='color:green; font-size:2em;'> La premiere ligne de mon site ! </p>
```

Explications : Après l'attribut « style= » on place entre guillemets une liste de réglages CSS séparés par des points-virgules.

3.2 CSS : la décoration réutilisable

Si nous devons réutiliser le style CSS, il est préférable d'utiliser une **feuille de style** séparée. L'idée est de créer des groupes de réglages et d'indiquer à quelles balises ils s'appliquent.

Une bonne habitude à prendre : séparer la décoration (le CSS) du HTML.

Il y aura donc 1 fichier HTML et 1 fichier CSS.

Pour dire au HTML qu'il doit utiliser le CSS, il faut ajouter la ligne suivante dans le bloc **<head>** :

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

On indique ici qu'une feuille de style nommée « deco.css » doit être utilisée dans la page HTML.

TRAVAIL :

1. Ajoutez la balise <link> ci-dessus dans la balise <head> de votre feuille HTML.
2. Créer ce nouveau fichier : **deco.css** qui contiendra :

```
body {  
    background-color: lightgrey;  
}  
  
h1, h2, h3 {  
    color: blue;  
    text-align: center;  
    text-shadow: 6px 6px 6px black;  
}
```

Si vous testez à nouveau votre page html, vous verrez les effets du CSS !

Explications : On a créé 2 réglages : un pour la balise <body>
et un pour les balises <h1><h2><h3>

Pour cibler un type de balise, il suffit d'indiquer son nom.

Si on souhaite appliquer des réglages sous d'autres critères que le nom de la balise, il faut utiliser des **sélecteurs CSS** plus précis :

```
#toto { } Pour les balise qui ont un attribut id='toto'  
.toto { } Pour les balises qui ont un attribut class='toto'
```

Enchaînement de sélecteurs :

```
h1, h2 { } Pour les balises <h1> et <h2>  
p div { } Pour les balises <div> qui sont contenues dans une balise <p>
```

Le site https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Selectors donne les possibilités des sélecteurs.

TRAVAIL :

1. Ajoutez l'attribut **class='moi'** aux 2 balises <p>
2. Ajoutez à votre fichier CSS le réglage suivant :

```
.moi {  
    font-style : italic;  
    text-align: center;  
    color:green;  
    font-size:1.5em;  
}
```

3. Testez !

3.3 CLASSES dans les feuilles de style

Avec le navigateur Mozilla, testez la page **classes.html** fournie.

Vous notez la présence de balises <div> Ces balises permettent de structurer la page et peuvent être considérées comme des « boîtes » que l'on pourra placer à notre guise sur l'écran... A condition de leur associer une décoration via le CSS.

Par exemple :

```
<div class="t t1">Pomme</div>
```

indique que cette <div> utilisera les réglages « t » ET « t1 » dans le fichier CSS.

TRAVAIL :

- Pour que ça fonctionne, il vous reste à relier cette page à son CSS : fichier **classes.css**.
- Modifiez le CSS pour que le cadre « cerise » soit sous le cadre « poire ».
- Créer 2 autres cadres pour obtenir ce résultat :

Les images seront ajoutées après !!



A retenir : Ici, on cumule les réglages grâce à l'attribut « class= ». C'est la puissance du CSS : « Feuilles de style en cascade ». En règle générale, on utilise massivement les *class* pour ajouter la décoration au HTML.

4 Lien HyperText : balise <a>

Le lien HyperText est un outil essentiel du HTML (comme l'indique le HT du HTML). C'est la possibilité de naviguer, dans un document ou vers d'autres documents, par simple clic sur une balise. Par exemple ceci est un lien hypertexte : <http://www.qwant.com> dans ce document PDF.

En HTML, le lien hypertexte est créé avec **la balise <a>**

1. Ajoutez dans votre page **index.html** :

```
<a href="http://www.qwant.com" > Site de recherche </a>
```

2. En cliquant sur le lien, vous êtes redirigés vers le site de QWant. On peut modifier le comportement du lien : en cliquant, on ouvre un autre onglet du navigateur. Notre page reste ouverte. Pour tester cela, dans la balise <a>, ajoutez **l'attribut** `target="_blank"` et vérifiez le nouveau fonctionnement. En mettant `target="blank"` on obtient un autre comportement : testez !
3. Re-HyperText :
Sur **index.html**, créez un lien vers **classes.html**
Sur **classes.html**, créez un lien vers **index.html**.

5 Ajouter une image GIF

Les fichiers images ont plusieurs formats, le plus connu étant le JPEG, utilisé par les appareils photo. Il existe aussi le format GIF : image animée.

Le navigateur sait gérer l'animation des GIF.

Pour ajouter une image, la balise est : `` : c'est une balise auto-fermante.

Exemple :

```

```

L'attribut « src » permet d'indiquer le nom du fichier à utiliser.

Comment placer une image ?

Par exemple pour mettre l'image dans le cadre « pomme » de notre TP, on complète la balise de classe 't t1' comme suit :

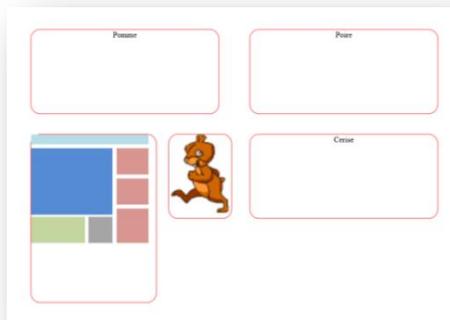
```
<div class="t t1">  </div>
```

Une image peut se mettre :

- Dans une `<div>` ou un lien `<a>`, mélangé avec du texte
- Avec du CSS :
 - En arrière-plan d'une balise (div, input, button, ...)
 - En arrière-plan global (de la balise `<body>`)

TRAVAIL :

1. Dans **classes.html**, insérez les 2 images GIF fournies dans 2 nouveaux cadres `<div>` créée précédemment. Le site devrait ressembler à ceci :



2. Ajoutez le réglage de la taille de l'image en ajoutant la **class** image aux balises `` :

```
.image {
    position:absolute;
    top:0%;
    left:0%;
    width:100%;
    height:auto ;
}
```

Ici, on indique que l'image occupera 100% de la largeur du `<div>` dans lequel elle sera placée.

Image en fond de page ? <https://www.alsacreations.com/tuto/lire/1390-arriere-plans-css3-background.html>
<https://www.alsacreations.com/astuce/lire/1216-arriere-plan-background-extensible.html>

6 Notion de SITE ADAPTATIF (*Responsiv Design*)

L'exercice précédent nous permet d'introduire une notion « philosophique » des créateurs de sites web : les Sites adaptatifs (*Responsiv Design*)

A cause de la grande variété de tailles d'écran, un site web doit être adaptatif, c'est-à-dire adapter sa présentation à la taille de l'écran.

L'image GIF affiche une démonstration de ce que doit faire une page web en fonction de la taille de l'écran.



Il existe de nombreux outils de développement de sites web gérant le *Responsiv Design* (WordPress, Joomla !, Drupal, Bootstrap, Foundation, Gumby, ...)

Tous s'appuient sur la technologie **@media** du CSS.

Voici un exemple :

```
@media screen and (max-width: 450px) {  
    .masquer {  
        display : none;  
    }  
}
```

Cette règle (@media) indique que si l'affichage du navigateur (screen) reste en dessous (max-width) de 450 pixels, ce qui est entre les accolades rouges sera valide : Ici on ajoute une classe « masquer » qui supprime l'affichage de l'objet.

TRAVAIL :

1. Ajoutez la règle ci-dessus dans le fichier **classes.css**.
2. Ajoutez la **classe** « masquer » à une des balises <div> du fichier **classes.html**
Vérifiez que si la taille de la fenêtre diminue, le <div> correspondant disparaît.
3. Ajouter dans la balise <head> la ligne suivante :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0"
```

Explications :

Note à propos du « **viewport** » : Sur les écrans des smartphones, le navigateur fait automatiquement un zoom arrière pour afficher la page en entier. Pour que le *Responsiv Design* fonctionne correctement, il faut indiquer au navigateur de ne pas le faire en

>

NB : Ce n'est qu'une seule des possibilités de la règle @media. On pourrait imaginer des règles pour réorganiser la position et la taille des blocs en fonction de la largeur de la fenêtre. D'autres l'ont fait à notre place : L'outil gratuit le plus facile à prendre en main pour le *Responsiv Design* est sans doute **Bootstrap** (utilisé et développé pour et par Facebook). Il existe de nombreux tutoriels à ce sujet.

7 Aller plus loin en CSS ...

Voir les démos sur le site W3Schools :

Déplacement ponctuel statique :

2D Transform : https://www.w3schools.com/css/css3_2dtransforms.asp

3D Transform : https://www.w3schools.com/css/css3_3dtransforms.asp

Déplacement animé (avec ou sans @keyframe) :

Transitions : https://www.w3schools.com/css/css3_transitions.asp

Animations : https://www.w3schools.com/css/css3_animations.asp

Exemple d'application : Ajout d'une TRANSITION sur un déplacement :

Modifiez le CSS en ajoutant les lignes écrites en rouge :

```
.t {  
    position:absolute;  
    border: 1px solid red;  
    border-radius:20px;  
    text-align:center;  
    transition: transform 1s;  
}  
  
.t:hover {  
    background:blue;  
    transform: rotateY(180deg);  
}
```

Et testez le résultat ...

Explication : le *rotateY* ne se fait pas instantanément quand on place (*hover*) la souris sur la <div> car le CSS indique qu'une *transition* de type *transform* doit se faire en 1 seconde.

Pour des transitions plus complexes, on utilisera les animations @KEYFRAME.

8 Publier son site

Pour l'instant, les pages créées sont locales.

Pour les rendre accessibles au monde entier (Internet), il faut les copier sur un serveur Web (Apache, IIS, ...) accessible par Internet.

Nous avons un tel serveur au lycée.

Sur le réseau du labo, vous pouvez y accéder avec l'adresse **IP locale 10.69.88.1**

NB : A l'extérieur, vous devez utiliser l'adresse URI <http://lycvauv.myfw.us>

Pour copier un fichier sur ce serveur, il faut utiliser le protocole de transfert de fichiers FTP. Le logiciel **Filezilla** est un CLIENT FTP très bien pour cela.

Vous devez disposer d'un compte utilisateur : il faut le demander au professeur.

Sur un site Internet, le premier fichier affiché doit s'appeler index.html ou index.php ou index.asp.

Il faudra donc modifier le nom du fichier exemple3.html en index.html avant de le copier sur le serveur.

Ensuite, votre travail sera accessible avec l'adresse :

Au labo :

http://10.69.88.1/le_nom_utilisateur_donne_par_le_prof

A l'extérieur :

http://lycvauv.myfw.us/le_nom_utilisateur_donne_par_le_prof