

Partie A - Cours HTML



1 – Initialisation

Ouvrir l'éditeur de texte

Dans un nouveau fichier saisir les lignes suivantes :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>Titre de la page</title>
  <meta http-equiv="content-type" content="text/html; charset=utf-8">
</head>
<body>
  Mon site Woueb
</body>
</html>
```

Enregistrer le fichier sous le nom : test1.html

Ouvrir Firefox

Aller chercher le fichier et visualiser le résultat

2 – Les balises

Elles permettent de définir des paramètres <.> texte entre paramètre </.>

Ouvrir l'éditeur de texte et rouvrir le fichier test1.html

Compléter le texte entre les balises body et fin body afin de tester les différents codes html

```
...
<body>
  <!-- Voici un premier commentaire sur une ligne. ->
  <!-- La balise h1 pour le titre de niveau 1 -->
  <h1> Titre niveau 1 de ma page </h1>
  <h2> Titre niveau 2 de ma page </h2>
  <p>Voici un premier paragraphe.</p>
  <p>Voici un second paragraphe.</p>
  <b>Ce texte s'affichera en gras.</b>
  <i><u>Ce texte sera souligné et en italique.</u></i>
  <font color="#ff0000">Ce texte sera en rouge.</font>
  <font face="verdana,sans-serif">Texte en verdana ou en sans-serif </font>
  <font size="5">Ce texte sera en taille 5.</font>
  <ul>
    <li>liste a puce 1</li>
    <li>liste a puce 2</li>
  </ul>
  <ol>
    <li>classement 1</li>
    <li>classement 2</li>
  </ol>

  <!-- exemple de tableau -->
  <p>Et maintenant un tableau :</p>
  <table>
    <tr id="ligne1">
```

```

        <th>titre de colonne 1</th><th>titre de colonne 2</th>
    </tr>
    <tr>
        <td>34</td><td>78</td>
    </tr>
    <tr>
        <td>47</td><td>aaa</td>
    </tr>
    <tr>
        <td>00</td><td>bbb</td>
    </tr>
</table>
</body>

```

code couleur :
 #ff0000 => rouge
 #00ff00 => vert
 #0000ff => bleu

...

Site pour trouver le code correspondant à une couleur : <http://www.code-couleur.com/>

Enregistrer le fichier sous le nom : test2.html

Ouvrir Firefox
 Aller chercher le fichier test2.html et visualiser le résultat

3 – Les liens et images

Ouvrir l'éditeur de texte et rouvrir le fichier test1.html

La troisième intègre le nom de l'image sur laquelle est associé le lien vers la page distante
 La dernière intègre une infobulle qui donne des indications sur le lien

```

...
<body>
    <a href="./marvel.png">lien vers l'image </a>
    <a href="./marvel.png" title="Marvel">lien vers l'image</a>
</body>

```

Enregistrer le fichier sous le nom : test3.html

Ouvrir Firefox
 Aller chercher le fichier test3.html et visualiser le résultat

4 – Les Framset

Afin de pouvoir casser la page pour la rendre moins monotone, nous allons créer des cadres pour découper la page .

Ouvrir l'éditeur de texte ainsi qu' un nouveau fichier vierge et saisir ces premiers lignes de code

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
    <title>Titre de la page</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
</head>
...

```

```
</html>
```

Le code rajouter entre les balises `</head>` et `</html>` permet de découper la page en 2 rectangles horizontaux (2 lignes)

```
<frameset rows="100, *">
  <frame>
  <frame>
</frameset>
```

Ce code permet de découper la page en 2 rectangles verticaux (2 colonnes)

```
<frameset cols="20 %, *">
  <frame>
  <frame>
</frameset>
```

Ce code permet de découper la page en 2 rectangles horizontaux puis de découper le second rectangle en 2 colonnes et d'appeler des pages dans ces différentes zones.

```
<frameset rows="100, *">
  <frame name="cadreBandeau" src="test1.html">
  <frameset cols="30 %, *">
    <frame name="cadreTexte" src="test1.html">
    <frame name="cadreImage" src="test3.html">
  </frameset>
</frameset>
```

Enregistrer le fichier sous le nom : Index.html

Ouvrir Firefox
Aller chercher le fichier Index.html et visualiser le résultat

En plus
Rouvrir le fichier test3.html
à la place de ligne

```
<a href="./marvel.png" title="Marvel">lien vers l'image</a>
```

remplacer par :

```
<a href="./marvel.png" target="cadreImage">lien vers l'image </a>
```

Sauvegarder le fichier test3.html

Ouvrir Firefox

Aller chercher le fichier Index.html et visualiser le résultat en cliquant sur le lien

5 – Exercices

Modifier les fichiers HTML pour répondre aux besoins suivant de votre « client »

Dans le cadreBandeau (Test1.html), le titre est « Jeu des héros de Marvel » en rouge sur le fond de l'image fond_étoile.jpg

Dans le cadreTexte (test2.html) une liste à puce avec :

- un lien qui renvoie l'image familleMarvel.jpg dans le cadreImage
- un lien qui renvoie l'image marvel.png dans le cadreImage

Partie B – Les Styles CSS

1 - Introduction des styles

Ouvrir l'éditeur de texte ainsi qu' un nouveau fichier vierge et saisir les codes suivants :

```
body {
    font color: blue;
    background-image: url("../fond_etoile.jpg");
}

h1 {
    font color: #00CCF;
    text-align:center;
    background-color: green;
}

p {
    font color: #ff0000;
    text-align:left;
}
```

Enregistrer le fichier sous styles.css

Puis Ouvrir dans l'éditeur de texte les fichiers Test1.html, test2.html et test3.html

Dans chacun des fichiers rajouter la ligne suivante entre les balise <head> et </head>

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

Enregistrer les 3 fichiers html

Ouvrir Firefox

Aller chercher le fichier Index.html et visualiser le résultat

2 – Exercices

Comment faire avec des feuilles de style pour que chaque page test1.html, test2.html et test3.html aient un background de leur body de couleur ou d'une image différente ?

Partie C – JavaScript

1 – Modification du fichier html

Cette partie propose un code plus compliqué puisque il introduit du code Javascript
Soyez vigilant-e-s aux majuscules et minuscules ainsi qu'à la fermeture des parenthèses et accolades.

Ouvrir le fichier test2.html

Modifier la ligne d'appel à la feuille de style css

```
<link rel="stylesheet" type="text/css" href="style2.css">
```

En dessous des listes à puce, entre les balises <body> et </body>, nous allons insérer 3 blocs pour créer

- des listes déroulantes :

```
<!-- Listes déroulantes -->
<form id="listes" name="listes">
  <div id="liste_d"> Liste</div>
</form>
```

- Ainsi que 2 boutons pour tester les réponses

```
<!-- Bouton -->
<div id="boutons">
  <input type="submit" name="boutonReset" id="boutonReset" Value="Reset"
onclick="fonctionReset()"/>
  <input type="submit" name="boutonValider" id="boutonValider" Value="Valider"
onclick="fonctionValider()"/>
</div>
```

- et la référence du fichier de code javascript

```
<!-- Code JS -->
<script src="code.js"></script>
```

Sauvegarder le fichier test1.html

2 – création d'une seconde feuille de style

Dans un fichier style2.css sauvegarder la mise en forme suivante.

```
body {
    font color: yellow;
    font-size:13px;
    background:#CCCCCC;
}
.OK {
    background:#0066CC;
}
.NOK {
    background:#FF0000;
}
```

3 – Création du fichier JavaScript

Ouvrir avec l'éditeur de texte un nouveau fichier vierge qui sera sauvegardé sous le nom de « code.js »

On va d'abord créer un tableau dans lequel on va enregistrer les noms civils des héros marvel

```
tableauNom= new Array();
tableauNom[1]= 'Bruce Wayne';
tableauNom[2]= 'Wade Wilson';
tableauNom[3]= 'Peter Parker';
tableauNom[4]= 'Barry Allen';
tableauNom[5]= 'Clark Kent';
tableauNom[6]= 'Tony Stark';
```

puis on va générer le code javascript qui permet d'afficher les listes déroulante dans la page html

Pour afficher 1 seule liste déroulante il faut passer le code suivant à

```
liste1 = '<span id="numéro1">a</span> -'+
        '<select name="liste1">'+
        '<option value="">Choisir...</option>'+
        '<option value="'+tableauNom[4]+'">'+tableauNom[4]+'</option>'+
        '<option value="'+tableauNom[1]+'">'+tableauNom[1]+'</option>'+
        '<option value="'+tableauNom[5]+'">'+tableauNom[5]+'</option>'+
        '<option value="'+tableauNom[6]+'">'+tableauNom[6]+'</option>'+
        '<option value="'+tableauNom[2]+'">'+tableauNom[2]+'</option>'+
        '<option value="'+tableauNom[3]+'">'+tableauNom[3]+'</option>'+
        '</select><br>';
document.getElementById("liste_d").innerHTML = liste1;
```

Comme cela est long et fastidieux puisqu'il faut répéter 6 fois, le code est « mutualisé » de la façon suivante :

```
listeChoix = '<option value="">Choisir...</option>'+
            '<option value="'+tableauNom[4]+'">'+tableauNom[4]+'</option>'+
            '<option value="'+tableauNom[1]+'">'+tableauNom[1]+'</option>'+
            '<option value="'+tableauNom[5]+'">'+tableauNom[5]+'</option>'+
            '<option value="'+tableauNom[6]+'">'+tableauNom[6]+'</option>'+
            '<option value="'+tableauNom[2]+'">'+tableauNom[2]+'</option>'+
            '<option value="'+tableauNom[3]+'">'+tableauNom[3]+'</option>'+
            '</select><br>';

liste1 = '<span id="numéro1">a</span> -'+
        '<select name="liste1">'+
        listeChoix;

liste2 = '<span id="numéro2">b</span> -'+
        '<select name="liste2">'+
        listeChoix;

liste3 = '<span id="numéro3">c</span> -'+
        '<select name="liste3">'+
        listeChoix;

liste4 = '<span id="numéro4">d</span> -'+
        '<select name="liste4">'+
        listeChoix;

liste5 = '<span id="numéro5">e</span> -'+
        '<select name="liste5">'+
        listeChoix;

liste6 = '<span id="numéro6">f</span> -'+
        '<select name="liste6">'+
        listeChoix;

// La fonction document.getElementById permet de transférer à la page html le code pour générer les listes
document.getElementById("liste_d").innerHTML = liste1+liste2+liste3+liste4+liste5+liste6;
```

Il convient enfin de programmer le code pour définir le comportements des 2 boutons valider et reset
Le bouton valider ;

```
// fonction Valider
function fonctionValider(){
    // boucle sur les listes
    for (var i=1; i<=6; i++)
    {
```

```

// récupération de l'index, puis de la valeur choisie
var indexChoisi = document.forms["listes"].elements["liste"+i].selectedIndex;
var NomChoisi = document.forms["listes"].elements["liste"+i].options[indexChoisi].value;
// test si la liste i affiche le bon Nom
if (NomChoisi == tableauNom[i]){
    // changement de style de l'element numéro i
    // document.getElementById("numéro"+i).style.backgroundColor = '#0066CC';
    document.getElementById("numéro"+i).className = 'OK';
} else {
    // document.getElementById("numéro"+i).style.backgroundColor = '#FF0000';
    document.getElementById("numéro"+i).className = 'NOK';
}
}
}

```

le bouton reset :

```

// fonction Reset
function fonctionReset(){
    // boucle sur les listes
    for (var i=1; i<=6; i++)
    {
        // Affichage premiere valeur
        document.forms["listes"].elements["liste"+i].selectedIndex = 0;
        //document.getElementById("numéro"+i).style.backgroundColor="#FFFFFF";
        document.getElementById("numéro"+i).className="";
    }
}

```

Ouvrir Firefox et vérifier le comportement du jeu

Partie D - JQuery

Jquery est un fichier contenant une bibliothèque de fonction qui facilite la programmation en Javascript
Pour cela il faut rajouter dans le répertoire des fichiers html le fichier jquery.js et déclarer le <head> du fichier test2.html

```
<script src="jquery.js"></script>
```

Puis dans le <body> de test2.html rajouter l'affichage d'un élément compteur de temps

```
<!-- Temps de jeu -->  
<div id="heure"></div>
```

Dans le fichier code.js, rajouter

- la fonction horloge ici régler sur 60 secondes

```
/* fonction pour le compteur de temps */  
function Horloge() {  
    if(timerIni < 60){  
        timerIni++;  
        $('#heure').text(timerIni);  
    } else {  
        fonctionValider();  
    }  
}
```

les compléments dans les fonctions associer aux boutons enter les {...}

- sur la fonction valider :

```
$('#boutonValider').hide('slow','linear');  
$('#heure').hide('slow','linear');
```

- sur la fonction reset

```
$('#boutonValider').show('slow','linear');  
$('#heure').show('slow','linear');  
timerIni = 0;
```

ainsi que l'initialisation de la fonction horloge avant les définitions des différentes fonctions

```
/* timer */  
timerIni = 0;  
inter = setInterval(Horloge, 1000);
```