

Activités : Aperçu des langages Web

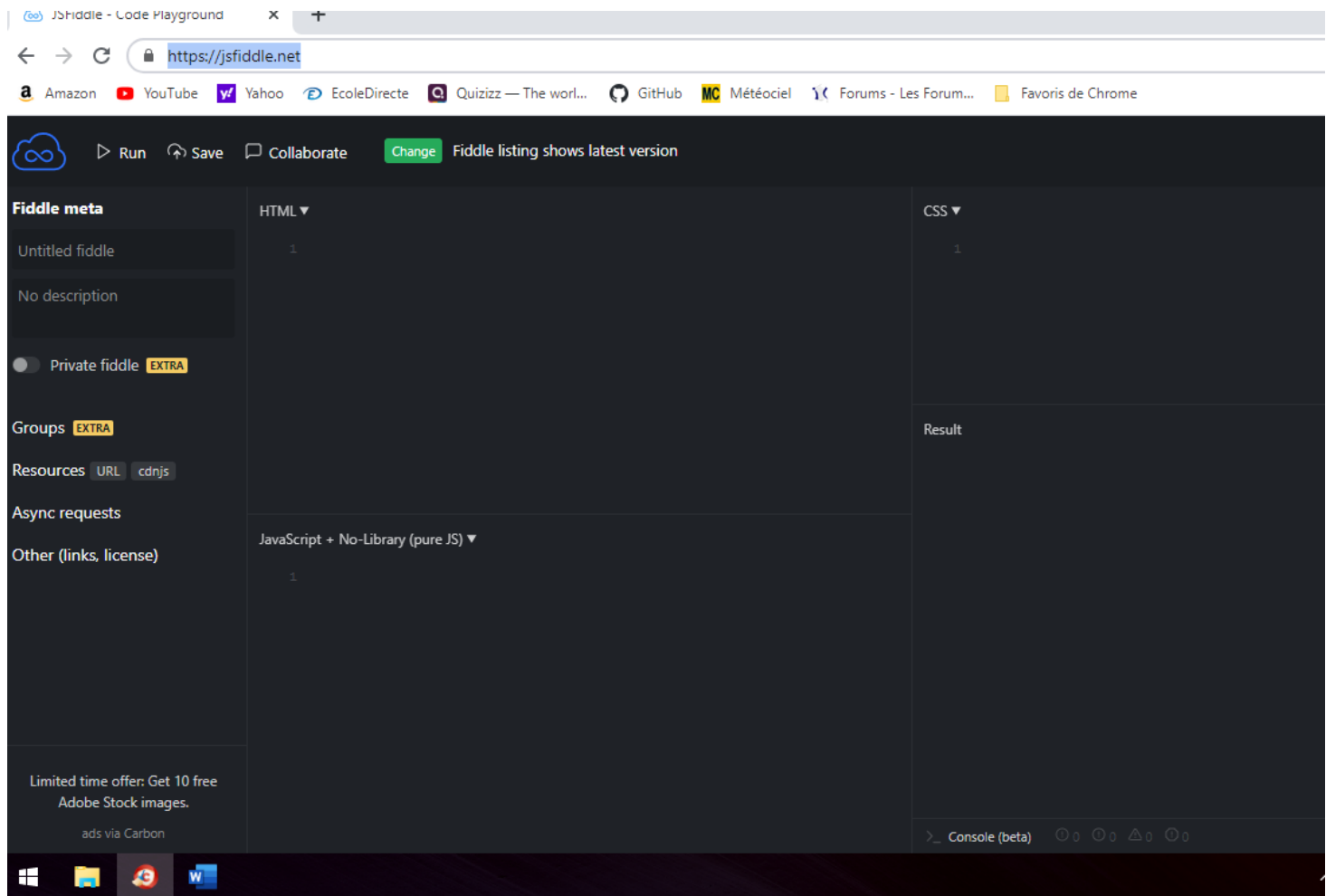
Introduction : Le site jsfiddle.net

Pour ces activités, on utilisera le site en ligne *jsfiddle* à cette adresse : <https://jsfiddle.net/>. Il permet de combiner les langages HTML, CSS et Javascript sur une même page avec un rendu accessible en cliquant sur la touche « **Run** ».

Il permet également de sauvegarder le travail grâce à la touche « **Save** » : le programme sera ainsi téléchargé et réutilisable par la suite.

Important : cette version ne permet pas de créer un site en lui-même. Il faudra pour cela utiliser un éditeur de texte comme *Notepad++* (installé sur les PC) ou *Sublime Text 3* (à installer).

Le site en image :



Activité 1 : Aperçu d'HTLM et CSS

1/ **Ecrire** le code suivant puis l'exécuter :

```
<h1>Hello World! Ceci est un titre</h1>
<p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

Résultat attendu :

Hello World! Ceci est un titre

Ceci est un **paragraphe**. Avez-vous bien compris ?

Quelques explications :

Comme déjà évoqué ci-dessus, en HTML tout est une histoire de **balises** que l'on ouvre et que l'on ferme. Une balise ouvrante est de la forme <nom_de_la_balise>, les balises fermantes sont de la forme </nom_de_la_balise>.

En observant attentivement le code, on remarque **que toute balise ouverte doit être refermée** à un moment ou un autre. La balise ouvrante et la balise fermante peuvent être sur la même ligne ou pas, cela n'a aucune espèce d'importance, la seule question à se poser ici est : ai-je bien refermé toutes les balises que j'ai ouvertes ?

Enfin pour terminer avec les généralités sur les balises, il est important de savoir qu'une structure du type :

```
<balise1>
<balise2>
</balise1>
</balise2>
```

est **interdite**, la balise2 a été ouverte après la balise1, elle devra donc être refermée avant la balise1.

En revanche, l'enchaînement suivant est correct :

```
<balise1>
<balise2>
</balise2>
</balise1>
```

Notez que dans notre exemple, nous respectons bien cette règle « **d'imbrication** » des balises avec la balise <p> et la balise .

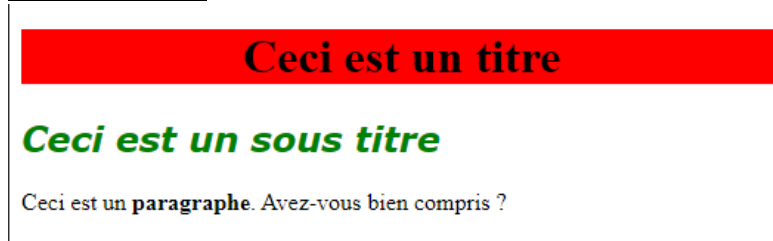
Il est important de comprendre que chaque balise a une signification qu'il faut bien respecter (on parle de la sémantique des balises). Par exemple le texte situé entre la balise ouvrante et fermante **<h1> est obligatoirement un titre important** (il existe des balises <h2>, <h3>.....qui sont aussi des titres, mais des titres moins importants (sous-titre)). La **balise <p>** permet de définir des paragraphes, enfin, la **balise ** permet de **mettre en évidence un élément important**.

2/ Le **HTML** n'a pas été conçu pour gérer la mise en page (c'est possible, mais c'est une mauvaise pratique). Le **HTML s'occupe uniquement du contenu et de la sémantique**, pour tout ce qui concerne **la mise en page et l'aspect « décoratif »** (on parle du « style » de la page), on utilisera le **CSS** (Cascading Style Sheets).

Compléter le code précédent (dans la partie CSS) pour obtenir ceci :

HTML ▼	CSS ▼
1 <h1>Ceci est un titre</h1>	1 h1
2 <h2>Ceci est un sous titre</h2>	2 {
3 <p>Ceci est un paragraphe. Avez-vous bien compris ?</p>	3 text-align: center;
	4 background-color: red;
	5 }
	6 h2
	7 {
	8 font-family: Verdana;
	9 font-style: italic;
	10 color: green;
	11 }
	12 }

Résultat attendu :



Quelques explications :

Dans l'exemple précédent, les propriétés « *text-align* » et « *background-color* » seront appliquées au contenu de toutes les balises de type **h1** (avec respectivement les valeurs « *center* » et « *red* »).
Même raisonnement pour les balises de type **h2**.

3/ Dans *jsfiddle.net*, le lien est automatiquement fait entre la partie HTML et CSS. De même, le site sait que le code entré est du HTML et que l'encodage par défaut est **utf-8**.

Il faut en réalité préciser tout cela lors de la conception d'un site qui nécessite un éditeur de texte.

Voici un exemple plus complet à reporter sur *jsfiddle.net* :

HTML ▼	CSS ▼
1 <!doctype html>	1 h1
2 <html lang="fr">	2 {
3 <head>	3 text-align: center;
4 <meta charset="utf-8">	4 background-color: red;
5 <title>Voici mon site</title>	5 }
6 <link rel="stylesheet" href="style.css">	6 p
7 </head>	7 {
8 <body>	8 font-family: Verdana;
9 <h1>Hello World! Ceci est un titre</h1>	9 font-style: italic;
10 <p>Ceci est un paragraphe. Avez-vous bien compris ?	10 color: green;
11 </p>	11 }
12 </body>	12 }
13 </html>	

Résultat attendu :



Quelques explications (partie HTML) :

Il y a deux parties :

- Celle délimitée par les balises <head> / </head>
- Celle par les balises <body> / </body>.

Dans la première partie, on indique qu'il s'agit d'**HTML** (et sa version éventuellement), de la **langue utilisée** (français ici), du **type d'encodage** (utf-8), du **titre du site** et du **fichier CSS** joint (style.css ici).

La deuxième partie regroupe le contenu du site et un lien vers un éventuel fichier en javascript.

Appeler le professeur (en cas de problèmes)

Activité 2 : Aperçu de Javascript

Important : pour aborder sereinement Javascript, il faut impérativement de bonnes bases de HTML et de CSS.

Recopier le programme suivant :

Partie HTML :

```
HTML ▼
1  <!DOCTYPE html>
2  ▼ <html>
3  ▼  <head>
4      <meta charset="utf-8">
5  ▼  <title>un peu d'action</title>
6      <link href="style.css" rel="stylesheet" type="text/css" />
7  </head>
8  ▼ <body>
9      <script src="script.js"></script>
10 ▼  <p><h2>Une page web extrêmement dynamique</h2></p>
11 ▼  <div>
12 ▼      <label>Changez la couleur d'arrière-plan:</label>
13 ▼      <button type="button" onclick="choix('yellow');">jaune</button>
14 ▼      <button type="button" onclick="choix('green');">vert</button>
15
16 ▼      <button type="button" onclick="choix('purple');">violet</button>
17  </div>
18 ▼  <div>
19 ▼      <p>En JavaScript, le nom de la couleur choisie est :</p>
20      <p id="resultat"></p>
21  </div>
22  </body>
23  </html>
```

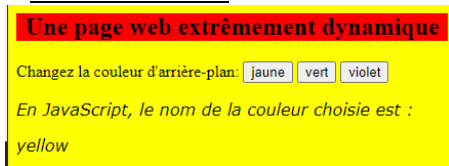
Partie CSS :

```
CSS ▼  
1  h2  
2  {  
3    text-align: center;  
4    background-color: red;  
5  }  
6  p  
7  {  
8    font-family: Verdana;  
9    font-style: italic;  
10   color: black;  
11  }
```

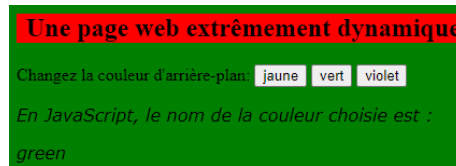
Partie Javascript :

```
JavaScript + No-Library (pure JS) ▼  
1  function choix(color){  
2    document.body.style.background = color;  
3    document.getElementById("resultat").innerHTML=color;  
4  }
```

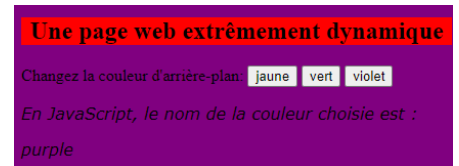
Résultat attendu :



(Clic sur le bouton « jaune »)



(Clic sur le bouton « vert »)



(Clic sur le bouton « violet »)

Quelques explications

Dans la partie HTML (ligne 9), ceci permet d'intégrer le fichier en javascript (*script.js* par défaut).

Dans la partie Javascript :

- Au sein du bouton déclaré par la balise `<button>`, l'attribut `onclick` reçoit le nom d'une fonction déclarée à l'intérieur du fichier `script.js`, ici la fonction `choix()`.
- Cette fonction nous permet de modifier à la fois l'aspect esthétique de la page (changement de la couleur de background) mais aussi le contenu de cette page, en faisant afficher le nom de la couleur.

A noter : une version « Notepad++ » de ce mini-site est disponible en ligne.

Travail à faire :

Créer un petit site à l'aide d'un éditeur de texte au choix (à télécharger si besoin) comportant :

- au moins deux pages,
- des hyperliens, des images,
- de la mise en page avec CSS,
- des interactions avec Javascript (type événements).