

Nom :

Prénom:

# HTML et CSS

---

Nous allons nous intéresser à un acteur fondamental du développement web, le couple HTML+CSS (Hyper Text Markup Language et Cascading Style Sheets).

Dans un premier temps, nous allons exclusivement nous intéresser au HTML.

## HTML

---

Voici la définition que nous en donne Wikipedia :

L'*Hypertext Markup Language*, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias, dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade).

Pour l'instant, nous allons retenir deux éléments de cette définition «**conçu pour représenter les pages web**» et «**un langage de balisage**».

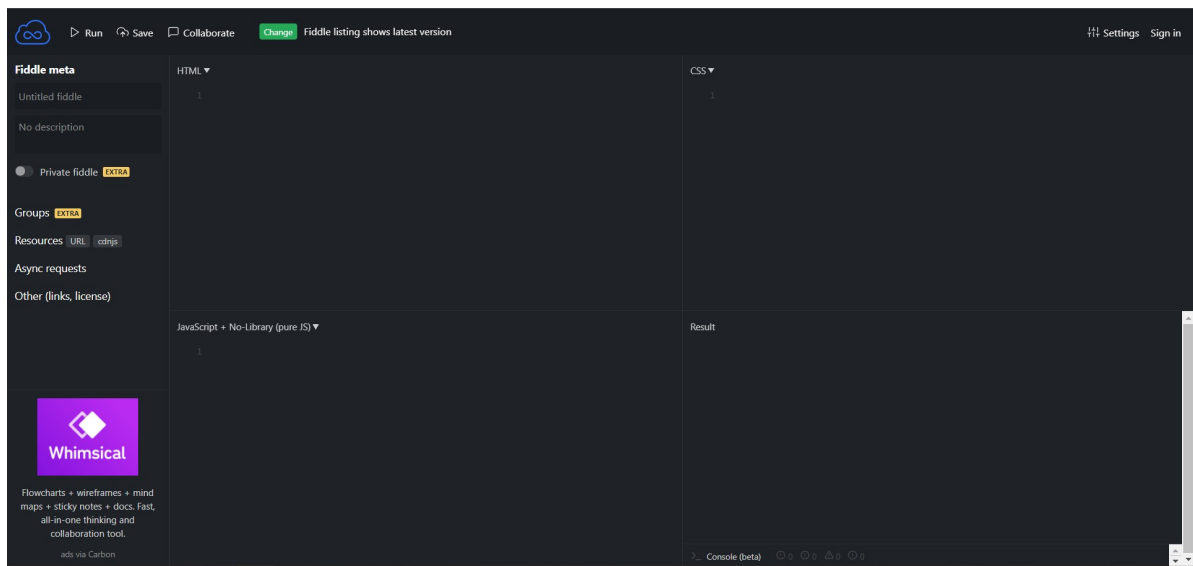
Grâce au HTML vous allez pouvoir, dans votre navigateur (Firefox, Chrome, Opera,...), afficher du texte, afficher des images, proposer des hyperliens (liens vers d'autres pages web), afficher des formulaires et même maintenant afficher des vidéos (grâce à la dernière version du HTML, l'HTML5).

HTML n'est pas un langage de programmation (comme le Python par exemple), ici, pas question de conditions, de boucles....c'est un **langage de description**.

Pour aborder le HTML, nous allons, dans un premier temps utiliser le site jsfiddle.net.

Après avoir lancé votre navigateur web, tapez <http://jsfiddle.net/> dans la barre d'adresse.

Vous devriez voir apparaître ceci :



Nous allons pour l'instant uniquement utiliser la fenêtre « HTML » et la fenêtre « Result ».

1. Ecrire le code suivant dans la partie HTML :

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

2. Dire ce qui s'affiche dans la fenêtre.

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, vous devriez forcément remarquer 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 pas 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 de la question question 1 nous respectons bien cette règle « d'imbrication » des balises avec la balise

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). Vous trouverez une liste non exhaustive des balises et de leur sémantique plus bas.**

**Vous devez aussi savoir qu'il existe des balises qui sont à la fois ouvrantes et fermantes () : un exemple, la balise permettant de sauter une ligne, la balise**

**Il est possible d'ajouter des éléments à une balise ouvrante, on parle d'attribut. Une balise peut contenir plusieurs attributs :**

```
<ma_balise attribut_1= "valeur_1" attribut_2="valeur_2">
```

**Il existe beaucoup d'attributs différents, nous allons nous contenter de 2 exemples avec l'attribut id (id pour identifiant) et class.**

**3. Ecrire le code HTML suivant :**

```
<h1>Ceci est un titre</h1>
<h2 class="titre_1">Ceci est un sous titre</h2>
<p id="para_1">Ceci est un <strong>paragraphe</strong>. Avez-vous bien
compris ?</p>
```

**4. Dire ce qui s'affiche dans la fenêtre.**

**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).**

## CSS

---

**Dans JSFIDDLE, il est possible d'écrire du CSS dans la fenêtre en haut à droite.**

**5. Ecrire le code HTML suivant dans la partie HTML :**

```
<h1>Ceci est un titre</h1>
<h2>Ceci est un sous titre</h2>
<p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

et le code CSS suivant dans la partie CSS:

```
h1
{
  text-align: center;
  background-color: red;
}
h2
{
  font-family: Verdana;
  font-style: italic;
  color: green;
}
```

6. Dire ce qui s'affiche dans la fenêtre.

Dans l'exemple de la question 5, 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 »).....

7. Ecrire le code HTML suivant dans la partie HTML:

```
<h1>Ceci est un titre</h1>
<h2>Ceci est un sous titre</h2>
<p id="para_1">Ceci est un <strong>paragraphe</strong>. Avez-vous bien
compris ?</p>
```

et le code CSS suivant dans la partie CSS:

```
#para_1
{
  font-style: italic;
  color: green;
}
```

8. Dire ce qui s'affiche dans la fenêtre. Que remarquez vous ?

Il est donc possible de cibler un paragraphe et pas un autre en utilisant l'id du paragraphe (en CSS l'id se traduisant par le signe #).

Il est aussi possible d'utiliser l'attribut class à la place de l'id. Dans le CSS on utilisera le point . à la place du #.

L'unique différence entre id et class est que l'id ne cible qu'un seul élément alors que la class permet de cibler plusieurs éléments d'une même page.

Ainsi, si nous avons eu un 3e paragraphe, nous aurions pu avoir :

```
<p class="para_1">Voici un 3e paragraphe</p>
```

Mais nous n'aurions pas pu avoir :

```
<p id="para_1"> Voici un 3e paragraphe </p>
```

car l'id para\_1 a déjà été utilisé pour le 1er paragraphe.

JSFIDDLE est un très bel outil, mais il ne peut pas être utilisé pour la réalisation d'une vraie page web.

Nous allons créer 2 fichiers : un fichier qui contiendra du HTML (index.html) et un fichier qui contiendra du CSS (style.css).

9. À l'aide d'un éditeur de texte, créer un nouveau fichier.
10. Sauvegarder le fichier en précisant son nom, "index.html".
11. Écrire le code suivant dans l'éditeur de texte (sans oublier de sauvegarder quand vous avez terminé) :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <h1>Hello world! Ceci est un titre</h1>
    <p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris
?</p>
  </body>
</html>
```

12. Tester le code à l'aide d'un navigateur web (Firefox ou Chrome) en "double-cliquant" sur le fichier index.html.

Dans le code de la question 11, vous reconnaissez le code se trouvant entre les balises :

```
<body>
.....
</body>
```

Tout votre code HTML devra se trouver entre ces 2 balises.

Le reste des balises devraient vous être inconnues. Passons-les en revue :

La première ligne :

```
<!doctype html>
```

permet d'indiquer au navigateur que nous utiliserons la dernière version du HTML.

La balise est obligatoire, l'attribut lang="fr" permet d'indiquer au navigateur que nous utiliserons le français pour écrire notre page.

Les balises ... délimitent ce que l'on appelle l'en-tête. L'en-tête contient, dans notre exemple, 2 balises : la balise qui permet de définir l'encodage des caractères (plus d'informations en première NSI) et la balise