

# BTS SN-EC

## Partie 1 : HTML et CSS

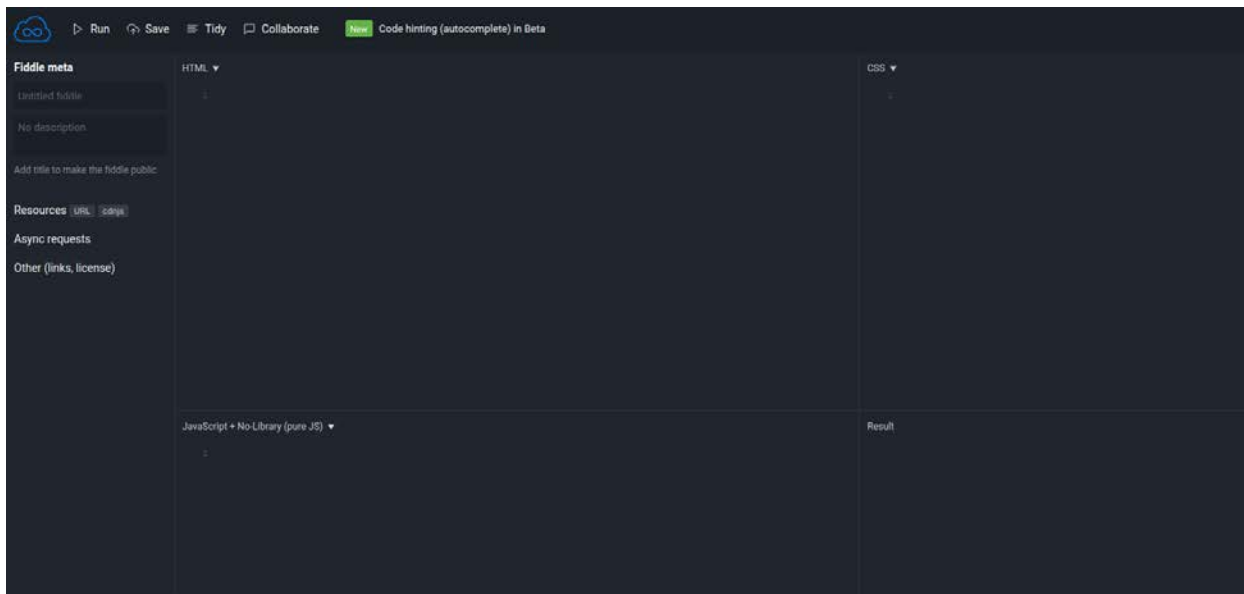
Grâce au HTML vous allez pouvoir, dans votre navigateur (Firefox, Chrome, Opera,...), afficher du texte, afficher des images, proposer des hyperliens (liens vers d'autres pages web), afficher des formulaires et même maintenant afficher des vidéos (grâce à la dernière version du HTML, l'HTML5).

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 ».

# À faire vous-même 1

Écrivez le code HTML suivant :

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

Qu'est-ce qui s'affiche dans la fenêtre ?

Comme déjà évoqué ci-dessus, en HTML tout est une histoire de balise 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 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 <strong>.

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 <strong> permet de mettre en évidence un élément important.

Vous devez aussi savoir qu'il existe des balises qui sont à la fois ouvrantes et fermantes (<balise/>) : un exemple, la balise permettant de sauter une ligne, la balise <br/> ou en HTML 5 <br>

On noterait :

```
<p>Première ligne <br> Deuxième ligne </p>
```

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

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

Il existe beaucoup d'attributs différents, nous allons nous contenter de 2 exemples avec les attributs id (id pour identifiant) et class. Nous verrons l'intérêt de ces attributs dans l'activité suivante.

## À faire vous-même 2

Écrivez 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>
```

Qu'est-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).

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

## À faire vous-même 3

Écrivez le code HTML suivant :

```
<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>
```

Écrivez le code CSS suivant (dans la fenêtre CSS !):

```
h1
{
  text-align: center;
  background-color: red;
```

```

}
h2
{
  font-family: Verdana;
  font-style: italic;
  color: green;
}

```

Qu'est-ce qui s'affiche dans la fenêtre ?

Dans l'exemple du "À faire vous-même 3", 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 »).....

## À faire vous-même 4

Écrivez le code HTML suivant :

```

<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>

```

Écrivez le code CSS suivant :

```

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

```

Qu'est-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'attribut "class" permet de donner la même *classe* à plusieurs éléments d'une page.
- L'attribut "id" désigne un unique élément *identifié*.

Si nous avions 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 le nom 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'un vrai site internet (ou d'une vraie application web).**

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

## À faire vous-même 5

À l'aide d'un éditeur de texte (libre office, word...), créer un nouveau fichier.

Sauvegardez-le en précisant son nom, par exemple "index.html".

Écrivez le code suivant dans votre 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>
```

Testez votre code à l'aide d'un navigateur web (Chrome) en "double-cliquant" sur le fichier index.html

Dans l'exemple du "À faire vous-même 5", vous reconnaissez le code se trouvant entre les balises <body> :

```
<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, le fameux HTML5.
- La balise <html> est obligatoire, l'attribut lang="fr" permet d'indiquer au navigateur que nous utiliserons le français pour écrire notre page.
- Les balises <head>...</head> délimitent ce que l'on appelle l'en-tête. L'en-tête contient, dans notre exemple, 2 balises : la balise <meta charset="utf-8"> qui permet de définir l'encodage des caractères (utf-8) et la balise <title> qui définit le titre de la page (attention ce titre ne s'affiche pas dans le navigateur, ne pas confondre avec la balise <h1>).

## À faire vous-même 6

Toujours à l'aide d'un éditeur de texte, vous allez créer un fichier qui va contenir le CSS de notre page (par exemple style.css). Complétez ce fichier à l'aide du code suivant :

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

Pour l'instant notre CSS ne sera pas appliqué à notre page, pour ce faire, il faut modifier notre code HTML en ajoutant une ligne qui va permettre d'associer notre code CSS à notre page.

Modifiez le code HTML avec la ligne suivante `<link rel="stylesheet" href="style.css">` :

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

Testez votre code à l'aide d'un navigateur web en "double-cliquant" sur le fichier index.html

Dans l'exemple que nous venons de voir, les fichiers "index.html" et "style.css" se trouvent dans le même dossier. Il est souvent utile de placer les fichiers CSS dans un dossier "CSS". Il faudra alors modifier le code HTML en conséquence :

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

Pour terminer, voici quelques balises très utilisées :

La balise a

```
<a href="mon_autre_page.html">Cliquez ici pour vous rendre sur mon autre
page</a>
```

La balise a permet de créer des liens hypertextes, ce sont ces liens hypertextes qui vous permettent de “voyager” entre les pages d’un site ou entre les sites. Les liens hypertextes sont par défaut soulignés et de couleur bleue (modifiable grâce au CSS). La balise a possède un attribut href qui a pour valeur le chemin du fichier que l’on cherche à atteindre ou l’adresse du site cible (exemple : <a href="http://www.google.fr">Cliquez ici pour vous rendre sur google.fr<a>). Entre la balise ouvrante et fermante, on trouve le texte qui s’affichera à l’écran (c’est ce texte qui est souligné et de couleur bleue). La balise peut sans problème se trouver en plein milieu d’un paragraphe.

## La balise image

Comme vous devez déjà vous en douter, la balise image sert à insérer des.....images :

```

```

la balise img est à la fois ouvrante et fermante comme la balise br. Elle possède 2 attributs :

- src qui doit être égal au nom du fichier image (ou au chemin si le fichier image se trouve dans un autre dossier).
- alt qui doit être égal à une description de votre image (cet attribut est utilisé notamment par les systèmes de description des pages web utilisées par les non-voyants), il faut donc systématiquement renseigner cet attribut.

## Les balises form, input et button

Les formulaires sont des éléments importants des sites internet, ils permettent à l’utilisateur de transmettre des informations. Un formulaire devra être délimité par une balise form (même si ce n’est pas une obligation) :

```
<form>
...
</form>
```

Il existe différentes balises permettant de construire un formulaire, notamment la balise input. Cette balise possède un attribut type qui lui permet de jouer des rôles très différents.

La balise button nous sera aussi d’une grande utilité.

## À faire vous-même 7

Créez un fichier html contenant le code suivant :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <form>
```

```
<p>voici un champ de texte : <input type="text"/></p>
<p>voici une checkbox <input type="checkbox"/></p>
<button>Cliquez ici !</button>
</form>
</body>
</html>
```

Testez votre code à l'aide d'un navigateur web en "double-cliquant" sur le fichier html que vous venez de créer.

## Les balises div et span

Ces 2 balises sont très utilisées (surtout la balise div). Pourtant, il faudrait, autant que possible, les éviter, pourquoi ?

Parce qu'elles n'ont aucune signification particulière, ce sont des balises dites "génériques".

À quoi servent-elles alors ?

À organiser la page, à regrouper plusieurs balises dans une même entité.

Pourquoi 2 balises génériques ?

Parce que div est une balise de type "block" et que span est une balise de type "inline" !

Sans vouloir trop entrer dans les détails, il faut bien comprendre que l'ordre des balises dans le code HTML a une grande importance. Les balises sont affichées les unes après les autres, on parle du flux normal de la page.

C'est ici qu'entrent en jeu les balises de type "block" et les balises de type "inline".

- Les contenus des balises de type "block" (p, div, h1, ...) se placent sur la page web les uns en dessous des autres.
- Les contenus des balises de type "inline" (strong, img, span, a) se placent sur la page web les uns à côté des autres.

Cela revient à dire qu'une balise de type "block" prend toute la largeur de la page alors qu'une balise de type "inline" prend juste la largeur qui lui est nécessaire.

## À faire vous-même 8

Créez un fichier html contenant le code suivant :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
```



```
<div>div est une balise de type "block"</div>
<p>la balise p est une autre balise de type block</p>
<span>En revanche, span est une balise de type "inline"</span>
<a href="www.google.fr">Et voici une autre balise de type "inline"</a>
<h1>h1 est bien une balise de type "block"</h1>
<span>la malheureuse balise span est "obligée" de se placer en
dessous</span>
</body>
</html>
```

Testez ce code

# Partie 2 : Introduction à Javascript

## 1 Introduction

Nous avons déjà pu nous familiariser avec le couple HTML-CSS, en fait, le couple est plutôt un trio, car aujourd'hui un développeur web ne peut pas faire l'impasse sur le JavaScript .

Notre but ici n'est pas d'apprendre un nouveau langage de programmation, mais juste d'étudier quelques exemples d'utilisation du JavaScript, notamment dans le cas des interactions entre un utilisateur et une page web.

Avant d'entrer dans le vif du sujet, un petit rappel historique : JavaScript a été créé en dix jours par Brendan Eich en 1995. Malgré son nom, JavaScript n'a rien à voir avec le langage Java, même si Brendan Eich affirme s'être inspiré de nombreux langage, dont Java, pour mettre au point JavaScript. Après des débuts chaotiques, il est, comme déjà dit plus haut, devenu incontournable dans le développement web.

## 2 Intégrer le JavaScript

Pour être exécuté, le JavaScript doit être intégré à la page HTML. Il existe différentes méthodes pour intégrer le JavaScript, nous allons en étudier une : le code JavaScript est écrit dans un fichier avec l'extension .js (par exemple script.js). Il faut ensuite ajouter une ligne dans le code HTML afin de lier ce dernier au fichier *script.js* :

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

L'attribut *src* de la balise *script* doit être égal au chemin vers le fichier qui contient le JavaScript.

Voici un exemple de fichier HTML associé avec un fichier *script.js* :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    ...
  </body>
  <script src="script.js"></script>
</html>
```

Comme vous pouvez le constater ci-dessus la balise *script* est placée après la balise *body* fermante (il est aussi possible de placer la balise *script* dans la partie *head*, mais ce choix peut entraîner certains problèmes si on ne prend pas certaines précautions, on préférera donc placer la balise *script* après la *body*).

## 3 le JavaScript pour répondre à des événements

### 3.1 Notion d'événements

Il est possible d'associer des événements aux différentes balises HTML, ces événements sont très souvent des actions réalisées pour l'utilisateur de la page :

- l'événement *onclick* correspond à un clic de souris (bouton gauche) sur un élément HTML donné
- l'événement *onmouseover* correspond au survol d'un élément HTML par le curseur de la souris

Il existe beaucoup d'autres événements, dans le cadre de ce cours nous utiliserons uniquement les deux cités ci-dessus.

### 3.2 Réponse à un événement

Un clic de souris vient d'être effectué sur un élément HTML, que va-t-il se passer ?

Dans la plupart des cas rien...si le développeur n'a pas prévu de gérer cet événement, il ne se passera rien.

Comment gérer un événement ?

Il suffit d'ajouter un attribut à la balise HTML concernée, prenons un exemple :

```
<button onclick="maFonction()">Cliquer ici</button>
```

Nous avons ici une balise *button* qui possède un attribut *onclick*. Cet attribut *onclick* est égal à *maFonction()*. *maFonction* est une fonction défini dans le fichier JavaScript qui est associé à la page HTML (par exemple le fichier *script.js*). En cas de clic sur le bouton défini ci-dessus, la fonction JavaScript *maFonction* sera exécutée.

### 3.3 Les fonctions en JavaScript

Comment écrire une fonction en JavaScript ?

Il suffit d'utiliser le mot clé *function* suivi du nom de la fonction :

```
function Mafonction(){  
    ....  
}
```

Les limites de la fonction sont définies par l'accolade ouvrante et l'accolade fermante.

## 4 Modifier un élément HTML avec le JavaScript

En cas de clic sur le bouton défini ci-dessus, la fonction JavaScript *maFonction* sera exécutée, mais qu'est-il possible de faire avec le JavaScript ?

Aujourd'hui, presque tout : il est possible de coder des applications extrêmement complexes en JavaScript. Nous allons rester modestes, nous allons uniquement modifier un élément HTML.

Pour modifier un élément HTML il est tout d'abord nécessaire de le sélectionner : c'est le rôle des *querySelector*.

## À faire vous-même 9

- dans votre répertoire de travail, créez un fichier *index.html* et un fichier *script.js*
- copiez-collez le code suivant dans le fichier *index.html* qui vient d'être créé

Considérons le code HTML suivant :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p id="monPara">Voici une page web qui ne fait pas grand chose</p>
    <button onclick="maFonction()">Cliquer ici</button>
  </body>
  <script src="script.js"></script>
</html>
```

et le code JavaScript suivant (fichier *script.js*) :

```
function maFonction() {
  let maBalise = document.querySelector("#monPara");
}
```

Le mot clé *let* permet de définir une variable, dans l'exemple ci-dessus, la variable a pour nom *maBalise* (on aurait évidemment pu choisir un autre nom). Cette variable *maBalise* va nous permettre de manipuler l'élément HTML qui a pour *id monPara* (c'est-à-dire la balise *p* présente dans le code HTML ci-dessus).

- copiez-collez le code suivant dans le fichier *script.js* qui vient d'être créé

Maintenant que nous avons "accès" à une balise HTML depuis le code JavaScript, il est possible de modifier le contenu de cette balise ou encore le style associé à cette balise comme bon nous semble :

```
function maFonction() {
  let maBalise = document.querySelector("#monPara");
  maBalise.innerHTML="coucou"
}
```

dans le cas ci-dessus nous remplacerons le texte présent dans la balise *p* ("Voici une page web qui ne fait pas grand chose") par "coucou".

- copiez-collez le code suivant dans le fichier *script.js* qui vient d'être créé

Autre exemple :

```
function maFonction() {  
  let maBalise = document.querySelector("#monPara");  
  maBalise.style.color="red";  
}
```

dans le cas ci-dessus, le texte du paragraphe sera écrit en rouge.

Il est aussi possible de modifier les classes CSS associées à une balise HTML, prenons un exemple :

## À faire vous-même 10

- dans votre répertoire de travail, créez un fichier *index.html*, un fichier *style.css* et un fichier *script.js*
- copiez-collez le code suivant dans le fichier *index.html* qui vient d'être créé

soit le code HTML :

```
<!doctype html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8">  
    <title>Le trio</title>  
    <link rel="stylesheet" href="style.css">  
  </head>  
  <body>  
    <h1>Le trio : HTML, CSS et JavaScript</h1>  
    <p id="monPara">Voici une page web qui ne fait pas grand-chose</p>  
    <button onclick="foncRouge()">Rouge</button>  
  </body>  
  <script src="script.js"></script>  
</html>
```

soit le code CSS (fichier "style.css") :

- copiez-collez le code suivant dans le fichier *style.css* qui vient d'être créé

```
h1{  
  text-align: center;  
}  
.rou {  
  color: red;
```

```
}
```

soit le code JavaScript (fichier "script.js") :

- copiez-collez le code suivant dans le fichier *script.js* qui vient d'être créé

```
function foncRouge() {  
  let maBalise = document.querySelector("#monPara");  
  maBalise.classList.add("rou");  
}
```

Résumons ce qui peut se passer avec la page HTML ci-dessus :

- un utilisateur clique sur le bouton *Rouge*, comme un événement *onclick* est associé à ce bouton, la fonction JavaScript *foncRouge* est donc exécutée
- la variable *maBalise* correspond à la balise *p* du code HTML (*let maBalise = document.querySelector("#monPara");*)
- on associe ensuite cette balise *p* à la classe CSS *rouge* : le texte contenu dans la balise *p* devient donc rouge (*maBalise.classList.add("rou");*)

## À faire vous-même 11

- dans votre répertoire de travail, créez un fichier *index.html*, un fichier *style.css* et un fichier *script.js*
- copiez-collez le code suivant dans le fichier *index.html* qui vient d'être créé

```
<!doctype html>  
<html lang="fr">  
  <head>  
    <meta charset="utf-8">  
    <title>Le trio</title>  
    <link rel="stylesheet" href="style.css">  
  </head>  
  <body>  
    <h1>Le trio : HTML, CSS et JavaScript</h1>  
    <p id="monPara">Voici une page web qui ne fait pas grand-chose</p>  
    <button onclick="foncRouge()">Rouge</button>  
    <button onclick="foncVert()">Vert</button>  
  </body>  
  <script src="script.js"></script>  
</html>
```

- copiez-collez le code suivant dans le fichier *style.css* qui vient d'être créé

```
h1{  
  text-align: center;  
}  
.rouge {
```

```
color:red;
font-size:20px;
}
.vert {
color:green;
font-size:30px;
}
```

- copiez-collez le code suivant dans le fichier *script.js* qui vient d'être créé

```
function foncRouge() {
  let balp= document.querySelector("#monPara")
  balp.classList.remove("vert");
  balp.classList.add("rouge");
}
function foncVert() {
  let balp= document.querySelector("#monPara")
  balp.classList.remove("rouge");
  balp.classList.add("vert");
}
```

Après avoir testé l'exemple ci-dessus en cliquant sur fichier *index.html*, apportez les modifications nécessaires pour que les boutons *rouge* et *vert* permettent de modifier la couleur du titre *h1*

## À faire vous-même 12

Créez la page Web la page web suivante :



si l'utilisateur clique sur le bouton *Changer*, le texte *COUCOU !* devient rouge, si on appuie de nouveau sur le bouton *Changer*, le bouton redevient noir (si on appuie de nouveau sur le bouton, *COUCOU !* redevient rouge...)

## À faire vous-même 13

Soit la page HTML suivante :

`index.html`

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div></div>
  </body>
  <script src="script.js"></script>
</html>
```

style.css

```
div {
  width : 200px;
  height : 100px;
  background-color : black;
}
```

Modifiez le code ci-dessus pour que le survole de la *div* avec le curseur de la souris modifie la couleur de fond de la div (*background-color*)