

Introduction

Le premier jour, quand nous avons travaillé tous ensemble, nous avons à la fin utilisé une balise très complexe:

```
<button onclick="document.getElementById('monparagraphe')
.setAttribute('style','color:red');">Rouge</button>
```

Il faut l'interpréter étape par étape pour la comprendre:

Au début et à la fin `<button></button>` ce sont les indicateurs de la balise HTML pour créer un bouton.

Le “Rouge” au milieu c’est le contenu qui va s’afficher sur le bouton dans la page.

Le “onclick=” c’est une caractéristique de ce `<boutton>`. En vocabulaire HTML on dit c’est un “attribute” du “element” bouton.

Cet “attribute” en particulier indique qu’est-ce qu’il faut faire quand l'utilisateur clique sur le bouton.

Son contenu `"document.getElementById('monparagraphe').setAttribute('style','color:red');"` c’est du code Javascript qui sera exécuté quand le bouton est pressonné.

Si vous regardez un peu avant dans le code il y a:

```
<p id="monparagraphe" style="color:blue;">
  Un paragraphe avec un identificateur "id", dont la
  couleur de départ est bleue, mais sera modifiée en
  cliquant sur un des boutons suivants
</p>
```

Cette balise `<p></p>` elle a aussi des “attributes”:

- “id=” pour l’identifier, et
- “style=” pour changer sa couleur.

Si vous regardez le code Javascript du onclick du bouton, ce qu’il fait c’est justement modifier le “style” du `<p>` dont l’”id” est “monparagraphe”

Les fonctions

Regardez cette page HTML:

```
<p id="monpara" style="color:orange">
  Un paragraphe avec "id" et dont sa couleur sera modifié
  par les boutons suivants
</p>

<button onclick="clickBouton();">Rouge</button>
<button onclick="clickBouton();">Vert</button>
<button onclick="clickBouton();">Bleue</button>

<script>
  function clickBouton() {
    document.getElementById('monpara')
      .setAttribute('style','color:red');
  }
</script>
```

Nous avons beaucoup simplifié le code de “onclick” et nous avons mis toute la complexité dans <script>. Il nous a fallu nommer ce code qui est dans <script> avec le “function clickBouton”; “function” est une spécificité de JavaScript, mais “clickBouton” c’est un nom qui moi j’ai inventé. Vous pouvez écrire combien de fonctions vous voulez, à condition qu’elles aient des noms différents.

Ce code que j’ai montré a un petit problème: tous les boutons changent la couleur du paragraphe à rouge!!!

Exercice: Modifiez le code pour avoir 3 “function” au lieu d’une seule, une pour chaque bouton, et que chacune donne une couleur différente au paragraphe, de façon que les boutons marchent comme on s’attend.

Encore une amélioration:

```
<p id="monpara" style="color:orange">
  Un paragraphe avec "id" et dont sa couleur sera modifié par
  les boutons suivants
</p>

<button onclick="clickBouton('red');">Rouge</button>
<button onclick="clickBouton('green');">Vert</button>
<button onclick="clickBouton('blue');">Bleue</button>

<script>
  function clickBouton(couleur) {

document.getElementById('monpara').setAttribute('style','color
:'+couleur);
  }
</script>
```

Maintenant la fonction reçoit un parametre: “couleur”, et le parametre qu’elle reçoit est différent selon le bouton. Nous n’avons besoin que d’une fonction et la page fonctionne comme attendu!

Exercice:

Regardez cette page:

```
<p>Voici l'espace pour un mot: <span id="monmot"></span></p>
<button onclick="clickBouton();">Remplir l'espace</button>
<script>
  function clickBouton(couleur) {
    document.getElementById('monmot').innerHTML = 'voici un
mot';
  }
</script>
```

Basé sur cette page, faites une page avec 3 boutons, “Caillou”, “Ciseaux”, “Papier” et selon le bouton appuyé il remplit ce mot dans un

Méthode de travail

Désormais nous allons à chaque nouveau projet ouvrir un nouveau fichier.

Sur Bloc Notes il faut utiliser Ctrl+N pour “nouveau fichier” et au moment de l’enregistrer rappelez vous de

- Le sauvegarder dans le dossier avec votre nom
- Après Ctrl+S sur Bloc Notes, choisir “Type” “Tous les fichiers”
- Lui donner un nom qui commence par “s01” ou “s02” etc selon la section que vous êtes en train de travailler
- Le nom doit finir par “.html”
- Donc un nom valable est par exemple “s01-gauche-droite.html”

Les documents devront tous avoir une balise <title> avec un titre qui aie :

- le même “S01”, “S02”, etc,
- une description du projet ou exercice,
- et votre nom.

Regardez l’exemple ci dessous, de la section suivante.

```
<title>
  S01 - Fonctions - Gauche et Droite - Gustavo
</title>
```

S01 - Utilisation de fonctions

s01-gauche-droite.html

```
<title>
  S01 - Fonctions - Gauche et Droite - Gustavo
</title>
```

```
<button onclick="choisirCote('gauche');">
Choisir côté gauche
</button>
```

```
<p>
<button onclick="choisirCote('droit');">
Choisir côté droit
</button>
```

```
<p>
<button onclick="choisirCoteAleatoire();">
Choisir côté aléatoire
```

```

</button>

<p>
  Le côté choisi est le :
  <span id="montrer"></span>
</p>

<script>
  function choisirCote ( cote ) {
    document.getElementById("montrer").innerHTML = cote;
  }

  function choisirCoteAleatoire () {
    document.getElementById("montrer").innerHTML =
      coteAleatoire();
  }

  function coteAleatoire () {
    r = Math.random();
    if (r <= 0.5) {
      return "gauche";
    } else {
      return "droit";
    }
  }
</script>

```

À noter sur cet exemple:

- La balise <title>
- Il y a plusieurs fonctions et une des fonctions, la choisirCoteAleatoire() appelle une autre fonction, la coteAleatoire()
- Une fonction qui a plusieurs instructions, pas une seule ligne comme les autres que nous avons fait jusque là.
- Math.random() est une fonction de Javascript qui retourne un nombre aléatoire entre 0 et 1. Pour être plus précis: $0 \leq \text{Math.random()} < 1$, ça veut dire le résultat peut être 0, mais ne sera jamais 1.
- La construction `if (...) { ... } else { ... }` qui permet d'utiliser des conditions pour décider les instructions à exécuter

Exercice: Écrire une page **s01-terre-lune-soleil.html** avec un bouton qui affiche aléatoirement le mot “terre”, “lune” ou “soleil”, avec la même probabilité pour chacun des mots.

S02 Images

Démarrez l'application Camera de Windows. Prenez votre photo et sauvegardez la dans votre dossier avec le nom **moi.jpg**.

Ensuite créez une page comme ça:

s02-images.html

```
<title>
    S02 - Images et miroir - Gustavo
</title>




```

S03 Projet Papier Caillou Ciseaux

Créez une page **s03-papier-caillou-ciseaux.html** avec:

- trois boutons "Papier" "Caillou" "Ciseaux"
- Une ligne: "Mon choix:" et un `` pour afficher notre choix
- Une ligne: "Choix de l'ordinateur" et un `` pour afficher le choix de l'ordinateur

Quand un des boutons est cliqué, il faut afficher le bon choix sur "Mon choix" et afficher un choix aléatoire de l'ordinateur sur la ligne "Choix de l'ordinateur"

Attention: A partir de ce moment nous aurons plusieurs étapes qui seront des améliorations de notre projet S03. Vous devez créer un nouveau fichier pour chaque étape, exemple:

s03-papier-caillou-ciseaux-etape-2.html,
s03-papier-caillou-ciseaux-etape-3.html etc.

Continuer à travailler sur le fichier d'origine **s03-papier-caillou-ciseaux.html** est possible mais je vous conseille fortement de créer un nouveau fichier à chaque fois.

La raison est que comme ça vous avez toujours disponible l'étape précédente comme sauvegarde, au cas où vous vous trompez et perdez quelque chose qui marchait avant. Il est inévitable que ça va se passer à un moment ou à l'autre, tout le monde est déjà passé par là et a regretté de ne pas avoir des sauvegardes!!!

Étape 2 - Partez de votre page précédente **s03-papier-caillou-ciseaux.html** et créez un nouveau fichier **s03-papier-caillou-ciseaux-etape-2.html** pour cette étape. La modification que nous allons faire est que, au lieu de montrer un des mots “papier”, “caillou” ou “ciseaux”, selon le choix, montrer l’image d’une main qui fait papier, caillou ou ciseaux. Pour cela vous avez des fichiers d’image **papier.jpg**, **caillou.jpg** et **ciseaux.jpg** dans vos dossiers. Demandez au prof si vous ne le trouvez pas!

Il faut se souvenir que, comme nous l'avons vu en S02, pour afficher une image on utilise la balise ``.

Une piste qui peut vous aider c’est l’instruction `document.getElementById('id').setAttribute('src','fichier-d-image.jpg')`. Mais attention... selon la voie que vous décidez de prendre vous n’aurez pas besoin de cette instruction. Il ne faut pas le prendre comme une obligation de l’utiliser. Regardez deux méthodes différentes de faire ce travail:

Méthode où on glisse un <code></code> dans un <code></code> existant	Méthode où on modifie le “src” d’un <code></code> existant
<pre> <button onclick="jaiClique('papier');">Papier</button> <button onclick="jaiClique('caillou');">Caillou</button> <button onclick="jaiClique('ciseaux');">Ciseaux</button> <p> Mon choix: </p> <script> function jaiClique(monChoix) { if (monChoix == "caillou") { imgDuChoix = ""; } else if (monChoix == "ciseaux") { imgDuChoix = ""; } else if (monChoix == "papier") { imgDuChoix = ""; } document.getElementById("spanMonChoix") .innerHTML = imgDuChoix; } </script> </pre>	<pre> <button onclick="jaiClique('papier');">Papier</button> <button onclick="jaiClique('caillou');">Caillou</button> <button onclick="jaiClique('ciseaux');">Ciseaux</button> <p> Mon choix: </p> <script> function jaiClique(monChoix) { if (monChoix == "caillou") { nomFichier = "caillou.jpg"; } else if (monChoix == "ciseaux") { nomFichier = "ciseaux.jpg"; } else if (monChoix == "papier") { nomFichier = "papier.jpg"; } document.getElementById("imgMonChoix") .setAttribute("src", nomFichier); } </script> </pre>

Le choix de l’utilisateur et le choix de l’ordinateur doivent tous les deux être affichés comme des images.

Étape 3 - Faites de façon que la “main” de l’ordinateur soit affichée juste à droite de la “main” de l’utilisateur, et inversez l’image de l’ordinateur comme un miroir pour que les mains s’affichent comme dans le vrai jeu qu’on joue. Exemple:



Référez vous à l’exercice S02 pour savoir comment inverser un image en miroir.

Étape 4 - Ajoutez une ligne à la fin de la page, avec “Vainqueur : ” et, après les choix faits, affichez sur cette ligne si c’est égalité ou pas. Ce n’est pas encore le but de vraiment trouver le vainqueur. Pour l’instant on cherche juste à décider si c’est une égalité. Affichez donc soit “égalité”, soit “*il y a un vainqueur mais nous ne savons pas encore qui*”.

Comment allez-vous décider si c’est égalité? Qu’est-ce qu’il faut utiliser dans le code pour faire *si ... c’est égalité ... j’affiche “égalité” .. en cas contraire ... j’affiche “il y a un vainqueur”* ?

Attention, sur Javascript pour tester si deux valeurs sont égales il faut utiliser == deux fois le signal de égal.

Exemple: `if (1+1==2) resultat = "oui, un plus un vaut deux!";`

Étape 5 - Modifiez votre code pour qu'il aie une variable `monChoix` avec le choix de l'utilisateur, et une autre variable `choixOrdi` avec le choix de l'ordinateur. Ces deux variables `monChoix` et `choixOrdi` doivent être établies dès que le choix est fait, et **il faut se baser sur elles pour afficher les mains pour les deux choix et pour vérifier si le résultat est une égalité.**

Regardez l'exemple ci-dessous, que nous avons utilisé en S01. Ce `r` est une variable!

```
function coteAleatoire () {  
  r = Math.random();  
  if (r <= 0.5) {  
    return "gauche";  
  } else {  
    return "droit";  
  }  
}
```

Vous vous souvenez du concept de “paramètre” d’une fonction?
Par exemple “`function choisirCote (cote)`” sur S01?
Un paramètre fonctionne de façon très semblable à une variable.

Étape 6 - Améliorer la page pour que, en plus de réussir à afficher “égalité” quand c’est le cas, elle puisse afficher “Ordinateur” comme vainqueur dans le cas que l'utilisateur choisit “papier” et l'ordinateur choisit “ciseaux”.

Pour l’instant je vous demande juste ce cas **utilisateur:papier et ordinateur:ciseaux**, il n’a pas besoin de traiter les autres cas qui amènent à la victoire de l'ordinateur ou de l'utilisateur.

Regardez cet exemple **pour vous inspirer**. “a” et “b” sont des variables.

```
a = Math.random();  
b = Math.random();  
if (a == b) {  
  resultat = "A est égal à B!";  
} else {  
  if (a < 0.5 && b > 0.5) {  
    resultat = "A est plus petit que ½ et B est plus grand que ½";  
  } else {  
    resultat = "A et B n'entrent pas dans les conditions examinées";  
  }  
}
```

Attention: le symbole `&&` veut dire “et”.

Étape 7 - Améliorer la page pour qu'elle traite correctement tous les cas d'égalité, victoire de l'ordinateur et victoire de l'utilisateur.

Regardez l'exemple pour inspiration:

```
a = Math.random();  
b = Math.random();  
if (a < 0.5 || b < 0.5) {  
    resultat = "Un des deux est plus petit que 0.5";  
} else {  
    resultat = "Au moins un des deux est plus grand que 0.5";  
}
```

Attention: le symbole `||` veut dire "ou".