

Tutorat M1 - Javascript/HTML/CSS

Vincent Casamayou

Novembre 2019

Bonjour à tous et bonjour à toutes, nous allons entrer dans le merveilleux monde du Web, de Javascript et des points virgules. Dans ce tutorat, vous apprendrez à créer une page HTML, et même lui faire faire toute sortes de chose avec du JavaScript.

Si vous tenez jusqu'au bout, vous partirez même avec un Pierre-Feuille-Ciseaux tout frais sur votre navigateur!

Le Point-Virgule (ou Semicolon);

Avant de commencer à coder des choses concrète, on va juste aborder un point (virgule) importants pour coder en JS.



Figure 1: ;

Vous, voyez ce truc là? Ça ressemble à rien, c'est un peu l'enfant illégitime du point et de la virgule et en langue française c'est quand tu finis une phrase..... mais en fait pas trop... ;

Et pourtant ça va être la cause de 90% de vos erreurs en JS;

Pour faire simple, en Javascript (et dans beaucoup d'autres langages de programmation), **vous devez le rajouter à la fin de vos lignes de codes!**;

Un print: un point-virgule;

une variable: un point-virgule;

un appel de fonction: je vous laisse deviner..;

Une boucle:Raté; on verra ça sur le moment.

1 Quelqu'un a parlé de Pierre-Feuille-Ciseaux?

Plutôt que de faire un long PDF à parcourir que vous ne retiendrez probablement pas, pourquoi ne pas passer à quelque chose de pratique.

On va donc, ensemble, faire un petit jeu de Pierre-Feuille-Ciseaux (PFC) où vous affronterez l'ordinateur (et essayerez de faire le meilleur score!).

Première étape, donc, définir l'endroit où va se jouer votre partie, qu'est ce que vous diriez....d'Internet?

Munissez-vous de votre éditeur de code préféré, et créez un fichier "Nomdevotrechoix.html".

2 Il était une fois, un fichier HTML....

2.1 In Your Head

Avant de passer à la création des éléments qui composeront le jeu de PFC, on va regarder ce qui compose une page HTML.

Figure 2: Page HTML type

Ça à l'air un peu barbare comme ça mais vous allez vous y faire très vite.

HTML est un langage en balise, c'est à dire que tout ce vous allez écrire devra se trouver entre deux balises.

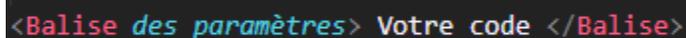


Figure 3: Une balise en HTML

Un petit conseil, pour éviter de vous compliquer la tâche et limiter les erreurs, **si vous ouvrez une balise <.....>, écrivez la fermeture de celle-ci </....> directement après** comme ça

2

vous n'oubliez pas la fermer plus tard.

Globalement, on peut séparer un fichier HTML en deux parties :

- **Le Header (balise <head>)** qui va contenir plein d'informations pour votre navigateur (les métadonnées) et le titre de votre page. Dans l'exemple plus haut, vous pouvez voir les métadonnées classiques pour une page basique (je ne rentre pas dans les détails mais les noms des attributs sont assez parlants)
- **Le Body (balise <body>)** qui va contenir tout ce qui est visible sur votre page Web.

Les deux premières lignes ne sont pas dans le header ou le body, mais elles servent à définir que vous utilisez du HTML et en français.

Exercice 1

Revenez à la page vide que vous avez créé et essayez de créer un header et un body (n'oubliez pas les deux premières lignes. Laissez le body vide pour l'instant.

Vous n'avez pas besoin de toucher aux lignes concernant l'utf-8 (ligne 4 de l'exemple) et l'html-equiv (ligne 6), ajoutez les simplement.

2.2 Bodybuilder, les protéines en moins

Maintenant que le fichier est proprement initialisé, on va pouvoir commencer à remplir notre page Web.

Pour cela passons en revue ce qu'il nous faut pour notre Pierre-Feuille-Ciseau :

- Un titre, ça rendra mieux
- un affichage de votre choix (Pierre, Feuille ou Ciseau)
- un affichage pour le choix adverse
- des boutons pour sélectionner Pierre, Feuille ou Ciseau
- du texte pour indiquer le résultat
- un compteur de score

Cela devrait suffire pour avoir un premier jet. Mais comment afficher ce texte ?

Comme je le disais, HTML est un langage à balise, la réponse est donc simplement entre 2 balises. La balise la plus basique que l'on utilise en HTML, c'est la balise <div> (pour division).

```
<div>
  Blablabla
</div>
```

Figure 4: Une div

Vous pouvez simplement écrire le texte que vous voulez entre cette balise et il s'affichera sur votre page HTML (ne vous souciez pas du style pour le moment, on verra plus tard).

Google sera toujours ton ami

Si vous voulez donner un titre à votre page, vous ne devriez pas utiliser une div, puisque que le style du texte ne changera pas par rapport à un autre. C'est donc l'occasion pour moi de vous parler de site comme W3school ou Developers.mozilla qui vous filerons plein de tips, de notions, et d'outils a utiliser pour vos pages web.

Par exemple pour créer un titre vous pouvez utiliser la balise H1, je vous laisse chercher sur le net comment l'implanter sur votre page.

Exercice 2

Pierre - Feuille - Ciseaux

Test
Test 2

Voici un exemple d'affichage pour ce début de page, il a été réalisé uniquement avec des divs et une balise H1.

Essayez de reproduire cet affichage.

2.3 "Et si j'appuie la?"

Maintenant que nous avons vu comment afficher un texte de base, voyons comment créer des boutons.

Comme d'habitude, Il y a également une balise pour cela!

```
<button type="button">Le Texte sur le bouton</button>
```

Figure 5: Un bouton

Cela va permettre de montrer un point important en HTML.

Quand vous écrivez une balise, vous pouvez lui assigner des attributs. Certains sont communs à toutes les balises (id, class, style...) et d'autres sont spécifiques ("type" pour button par exemple). Quand vous utilisez une balise, je vous invite à aller quels attributs existe pour cette balise, cela peut être utile.

Ici, "type" va permettre de définir quel bouton on souhaite pour notre page.

Il y a de l'ambiance dans ces boîtes jaunes

Pour le jeu, nous avons besoin de 3 boutons, un pour chaque coup. Si on veut organiser notre page, il est préférable de les intégrer au préalable dans une balise <div> ou <p> (pour paragraphe).

En fait, vous pouvez voir les balises comme des boîtes. Il peut donc être important de ranger les choses au même endroit sur la page, dans la même boîte (cela pourra être important vis à vis du CSS que nous verrons plus tard).

Exercice 3

Pierre - Feuille - Ciseaux

Test
Test 2

Faites votre choix.
Score : 0

De la même manière que pour l'exercice 2, voici un exemple d'affichage de la page. Essayez de le reproduire. Les deux lignes après les boutons sont deux div.

Bravo! Vous avez complété la partie HTML, maintenant rendons notre petit jeu fonctionnel!

3 Java Bien, Javascript

Avant de commencer cette partie, il vous faudra répondre à une question importante!

Que met-on dans son code a la fin d'une ligne en Javascript?

Si vous savez répondre à cette question vous pouvez passer à la suite ;)

3.1 Un nouveau langage??!

"Un nouveau langage ! Ça y est! Après avoir galéré à apprendre Python, je vais devoir tout recommencer et apprendre un nouveau truc...."

C'est probablement ce que vous vous dites à l'heure actuelle. Maintenant que vous avez vu Python, vous avez vu beaucoup de structures qui permettent de construire votre code: les variables, les boucles, les fonctions etc.....

Ces structures, vous allez également les retrouver en Javascript et elles fonctionnent globalement de la même façon, seul la manière de les écrire change. Plutôt pratique non? Vous n'avez donc pas besoin de recommencer à apprendre des concepts entièrement mais seulement leur syntaxe et c'est ce qu'on va voir durant l'exercice.

Exercice 4

Exercice très court, placez vous dans le répertoire contenant votre page HTML et créez un fichier en ".js" pour pouvoir commencer à coder en Javascript!

3.2 Le 83

Commençons donc par la base, les variables.

Pour définir une variable vous aurez besoin d'écrire un mot-clé + un nom = votre variable.

```
let un_nom = 0;  
let un_autre_nom = "une string";
```

Figure 6: Définir une variable

En JS, il existe 3 moyens de définir des variables:

- **let**: let est la définition la plus basique d'une variable. Si vous l'utilisez votre variable ne sera connu que dans l'endroit où vous la définissez (attention donc si vous le faites dans une fonction et que vous voulez l'appeler en dehors)
- **var**: var est la définition la plus puissante d'une variable. C'est l'équivalent d'une variable globale en Python. Elle sera connue n'importe où et pourra même être appelée avant sa création.
- **const**: const permet de définir une constante. Vous ne pourrez pas modifier une variable créée avec const. Plutôt pratique en programmation fonctionnelle, ça existe mais mettez-le de côté pour l'instant.

Une fois la variable créée, plus besoin du mot-clé, vous pouvez la modifier simplement comme en Python.

Une variable peut contenir tout un tas de chose en Javascript: une integer, une string, une liste (un array), un dico (un object).

Exercice 5

Pour commencer à coder notre Pierre Feuille Ciseaux, nous allons avoir besoin de 3 variables.

- une liste contenant 3 strings qui correspondent aux possibilités de l'adversaire (donc Pierre, Feuille et Ciseaux)
- un integer qui correspondra à un index de cette liste (pour l'instant, 0)
- un integer qui correspondra au score du joueur (pour l'instant, 0)

3.3 La Pêche aux ID

HTML

Nous avons commencer à coder en Javascript dans un fichier totalement différent du fichier HTML. Il va donc falloir trouver un moyen pour que les deux fichiers puissent se reconnaître et interagir entre eux.

Commencez par retourner dans votre fichier HTML. Dans ce dernier, vous pourrez rajouter un balise qui va vous permettre d'ajouter votre JavaScript aux fichiers. La voici:

```
<script type="text/javascript" src="le_nom_de_votre_script.js"> </script>
```

Figure 7: Balise Script

C'est une balise "script". Le **type** permet d'indiquer le langage utilisé et **src**, le chemin d'accès à votre fichier.

A noter que vous auriez pu écrire directement votre script entre les balises et cela aurait produit le même effet puisque HTML ne fait que lire votre fichier à l'endroit de la balise. Indiquer un fichier permet simplement de rendre le code lisible et de ne pas mélanger un gros script et votre HTML. Ajoutez donc votre script après le body.

Javascript

Retournez maintenant dans votre fichier JS. Si vous voulez utiliser vos éléments HTML (comme les boutons ou le texte) dans votre Javascript, il va également falloir aller les chercher et les stocker dans des variables.

Cela va me permettre d'aborder "un point" (littéralement) très important en Javascript (et d'autres langages également).

Je mets des points, et souvent après mes objets

En Javascript, beaucoup de méthodes (les fonctions) sont contenues dans des Objets, un peu comme une mini librairie Python. Ceux qui nous intéressent sont directement intégré dans le langage, pas besoin de les importer, il faut juste y accéder.

Pour cela il y a un syntaxe bien précise que vous allez utiliser tout le temps:

```
le_nom_de_l'objet.le_nom_de_la_méthode()
```

Par exemple, l'objet "**document**" contient une méthode "**getElementById()**", qui va nous permettre de stocker un élément HTML, par son ID, dans une variable. Comment feriez-vous pour l'appeler?

J'espère que vous avez pu répondre à la question précédente parce que c'est pile selon on a besoin pour continuer. Si vous ne l'avez pas déjà fait, il est possible d'associer des ID à chaque élément HTML en le précisant dans les paramètres d'une balise (retournez voir le schéma en page 2 pour voir où sont situé les paramètres).

```
id = "votre-id"
```

Ces id doivent être uniques et permettre d'identifier l'élément pour interagir avec d'autres. On peut également définir une classe (`class = ""`) pour plusieurs éléments si on veut pouvoir les récupérer en même temps.

Exercice 6

Pour notre jeu, il vous faut stocker 7 éléments (les 3 boutons et les 4 éléments de textes) dans des variables grâce à la méthode `getElementById` dans le fichier Javascript. Bon courage!

3.4 Remonte ta bracket

Dernière grosse partie du Javascript: les structures!

Nous allons maintenant voir comment écrire des boucles et des fonctions.

Fonctions

Pour écrire une fonction (ou méthode), nous avons besoin de 2 choses: un mot-clé et des brackets .

Le mot-clé n'est pas "def" comme en Python mais "function", simple non?

Les brackets sont des éléments assez nouveaux pour vous mais comme le semi-colon, vous allez en mettre souvent.

```
function Mafonction()  
{  
  
};
```

Figure 8: Une fonction

Ce sont ces petits trucs là : {}

Lorsque vous allez écrire votre fonction, vous devrez le faire entre ces brackets. Voyez ça comme une boîte qui va contenir votre fonction. Votre code devra également être indenté comme en Python.

Petit conseil: Écrivez toujours les deux brackets (ouverture et fermeture) pour éviter de l'oublier après et de faire tout planter (par expérience).

Exercice 7

Vous allez maintenant implanter une fonction `AddScore` qui va simplement ajouter +1 au score du joueur, modifier le texte du score dans le HTML et retourner le nouveau score. Pour modifier du texte en HTML il existe plusieurs méthodes qui devront agir sur votre variable contenant le texte, je vous laisse les chercher sur le net.

Boucles

```
if (votre_condition)
{
};

for(let i =0; votre_condition; i++)
{
};
```

Figure 9: les boucles en JS

Voilà les deux boucles if et for en Javascript. On retrouve une syntaxe proche de celles des méthodes, avec les brackets.

La principale différence avec Python réside juste à mettre des parenthèses autour de sa condition. La boucle for possède néanmoins quelques particularités par rapport à python. Elle se compose en trois parties séparées par des ; :

1. Initialisation : `let i = 0;`

La boucle va, avant de se lancer, créer une variable que vous définissez ici, et qui ne sera accessible que dans le cadre de la boucle

2. Condition while: `votre_condition;`

Condition pour que la boucle exécute son code, à la façon d'un while. C'est à dire que la boucle s'arrête à partir du moment où la condition renvoie false

3. Incrémentation `i++`

Ce que va faire la boucle à chaque fois qu'elle va finir un tour. Il existe différentes astuces pour changer des variables en peu de code que voici:

- `ma_variable++` : `ma_variable = ma_variable + 1`
- `ma_variable+=5` : `ma_variable = ma_variable + 5`
- `ma_variable--` : `ma_variable = ma_variable - 1`
- `ma_variable-=4` : `ma_variable = ma_variable - 4`
- Il en existe pour chaque opération: [Lien vers les différents opérateurs d'affectation](#)

Il existe aussi un équivalent à la boucle python `for variable in liste_variables`:

```
for (let variable of liste_variables) {
}
```

3.5 Le gros du travail

Maintenant que nous avons vu la base de Javascript, nous allons enfin pouvoir passer à votre première grosse implémentation !

Pour faire fonctionner le jeu, nous allons avoir besoin de 3 fonctions selon le choix que vous ferez entre Pierre, Feuille et Ciseaux. Ces 3 fonctions seront assez similaires et définirons les règles du Pierre, Feuille, Ciseaux.

Je ne vous donnerais pas la réponse parce qu'il n'y en a pas qu'une seule mais je vais vous donner globalement la logique à suivre

```
function Rock_move()  
{  
    //if Scissors then Win  
  
    //else if Paper then Lose  
  
    //else Draw  
}
```

Figure 10: Algo quand le joueur choisi Pierre

Pour vous aider, votre première ligne permettra de définir l'IA de l'ordinateur, quand vous faites un choix. Vous vous souvenez, nous avons définis une liste comprenant 3 strings avec "Pierre", "Ciseaux", "Papier".

A chaque appel de la fonction l'ordinateur va choisir aléatoirement un élément de la liste et cela représentera son mouvement.

Je vais vous donner la fonction a utiliser vu qu'elle n'est pas intuitive

```
enemy_choice = Math.floor(Math.random() * 3); //Permet d'avoir un nombre entre 0 et 2
```

Figure 11: Un nombre aléatoire

Cette ligne fait appel à deux méthodes de **Math**. **random** pour avoir un nombre aléatoire et **floor** qui va l'arrondir. **enemy_choice** prendra donc un nombre entre 0 et 2 qui pourra correspondre à un index de votre liste de choix.

Pour créer ces 3 fonctions, vous avez déjà tous les outils en main. Rien de plus n'a été utilisé. Uniquement ce que vous avez déjà vu dans ce tutorat.

Bon courage !

3.6 Amélie n'est pas la seule VP Event

Étape finale de la création du jeu, la gestion des événements!

Pour l'instant si vous allez sur le HTML et que vous cliquez sur les boutons, rien ne devrait se passer. C'est normal, nous allons affecter aux boutons, les 3 fonctions que vous avez créés précédemment.

```
function setupListeners()
{
    rock.addEventListener("click", Rock_move);
    paper.addEventListener("click", Paper_move);
    scissors.addEventListener("click", Scissors_move);
}

window.addEventListener("load", setupListeners);
```

Figure 12: Les événements en Web

addEventListener() est une fonction importante en web. Elle permet de donner à un élément HTML stocké dans une variable, une fonction lorsqu'un événement se produit (ici, un click sur l'élément). Elle prend en argument, un événement puis une fonction. Il existe beaucoup d'autres événements que je vous invite à regarder dans la documentation de **addEventListener()**.

Vous pouvez remarquer au passage que l'on ne met pas de parenthèses après le nom des fonctions que vous associez. En effet la méthode **addEventListener()** attend une fonction comme

deuxième argument. Si vous mettiez des parenthèses, vous appelleriez la fonction au moment où vous utiliseriez `addEventListener()`, vous n'ajouteriez alors que le retour de la fonction, ce qui n'est pas ce qu'on désire.

Cela permet de se rendre compte que à chaque fois que vous déclarez une fonction, vous déclarez en réalité une variable qui contient le code de la fonction. Vous appelez le code de ces fonctions en mettant des parenthèses après. C'est ça qu'on appelle la programmation fonctionnelle: le fait que les fonctions soient également des variables, et qu'on peut donc les manipuler comme bon nous semble.

En conséquence, il faut bien faire attention à ne pas déclarer de variables du même nom que vos fonctions, cela écraserait votre fonction !

Pour terminer, la dernière ligne (qui ne doit pas être dans une fonction) permet de charger tous ses événements au chargement ("load") de la fenêtre ("windows") HTML dans votre navigateur.

3.7 Conclusion

Si vous êtes arrivés ici, vous devriez avoir un jeu de Pierre Feuille Ciseaux fonctionnel! Bravo à vous!

Maintenant, n'hésitez à approfondir le concept, coder de nouvelles règles, mettre en place un classement, les possibilités sont infinies et vous apprendront à coder et réfléchir.

Je voudrais vous remercier d'être allé aussi loin et j'espère que ce tutorat aura été aussi sympathique à suivre qu'il a été à écrire.

Vous avez fait vos premiers pas dans le monde du web et de Javascript, mais il vous reste encore plein de choses à découvrir alors si ne savez pas comment faire n'hésitez pas à fouiner sur le net, quelqu'un aura sûrement répondu à votre problème. Sinon si vous avez des questions supplémentaires, vous pourrez les poser à vos chers M2.

Ketchup!

3.8 Pour aller plus loin: Pimp My HTML

Vous aurez remarqué que le jeu marche mais n'est vraiment pas très beau... Il y a une solution toute simple pour cela: le CSS

CSS est le dernier élément de la triade du web (JS, HTML et CSS). C'est un langage très puissant et assez simple d'utilisation qui vous permettra de mettre en page chacune de vos pages HTML. N'hésitez pas à aller regarder, voici un tutoriel assez bien foutu (en anglais):

https://www.w3schools.com/css/css_intro.asp