



Epitech clicker (2023)

Bienvenue dans ce cours passionnant sur la création d'un jeu inspiré de l'emblématique Cookie Clicker en utilisant les langages web HTML, CSS et JavaScript ! Si vous avez déjà été captivé par l'engouement de cliquer frénétiquement sur des cookies virtuels pour accumuler des richesses virtuelles, alors vous êtes au bon endroit. Au cours de cette aventure, nous explorerons les fondements du développement web pour concevoir un jeu interactif et addictif, où chaque clic compte et chaque amélioration apporte une satisfaction croissante.

Préparez-vous à explorer les principes de base du développement web tout en laissant libre cours à votre créativité pour concevoir un jeu qui, à la manière de Cookie Clicker, vous laissera accroché et désireux d'en apprendre davantage. Alors, mettez-vous à l'aise, préparez-vous à coder et à cliquer, car l'aventure commence ici !

But du jeu

Le but du jeu d'aujourd'hui sera de récréer un epitech clicker, un jeu vidéo officiel epitech, sur la base du cookie clicker en html/css/js.

Voici un exemple de ce à quoi le jeu final pourrait ressembler :



nous vous laissons libre cours à votre imagination pour le design du site, mais les features suivantes doivent être dans votre jeu final :

- Lorsque l'on clique sur le logo d'Epitech, vous gagnez $1 + (1 \times \text{le nombre de workshops})$ xp.
- Lorsque vous atteignez 10 xp, ces 10 xp sont convertis en 1 crédit
- Lorsque vous obtenez 100 crédits vous gagnez 1 année et vos crédits, vos xp, le prix des workshops ainsi que le nombre de workshop que vous avez seront réinitialisé à leurs valeurs de départ
- Lorsque vous atteignez 5 années, vous afficherez un écran de victoire
- Vous pouvez acheter des workshops, leurs prix de départ sera de 3xp puis multipliera son prix par 2.5 à chaque achat.
- Vous pouvez finir une année plus tôt en payant 60 crédit dans le shop
- 1 crédit = 10 xp dans le shop
- A la fin de chaque année votre GPA sera calculer grace au nombre restant de crédits et d'xp et il sera sur 4, à vous de faire le calcul pour trouver comment faire tout ça !

Ormis cela, vous pourrez rajouter autant de détails ou de fonctionnalités que vous le souhaitez, n'oubliez pas que ce site sera le votre, alors amusez vous et soyez créatif !

Si vous n'avez jamais développer un site web ou que vous souhaitez améliorer vos bases, je vous laisse des introductions et des tips sur les langages que vous utiliserez aujourd'hui ci dessous.

Introduction au html

Qu'est-ce que le HTML ?

HTML, acronyme de HyperText Markup Language, est le langage standard utilisé pour créer et concevoir des pages web. Il s'agit d'un langage de balisage qui structure le contenu d'une page web en utilisant des balises. En d'autres termes, il sera le squelette de votre site web, tout les éléments qui existeront sur la page seront d'abord des éléments html avant d'être modifier par le css et le js.

Structure de base d'un document HTML

Un document HTML est généralement composé de deux sections principales : la tête (`<head>`) et le corps (`<body>`). Voici une structure de base d'un document HTML :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Je suis un titre</title>
  </head>
  <body>
    <h1>Apprenons à coder en HTML avec Epitech</h1>
    <p>Commencons par l'Epitech clicker</p>
  </body>
</html>
```

- `<!DOCTYPE html>` déclare la version HTML du document. C'est un indicateur que nous utilisons le langage HTML5 sur le document.
- `<html>` est la balise racine qui englobe tout le contenu de la page. C'est ici que nous écrirons en html.
- `<head>` contient les informations sur le document telles que le titre (`<title>`) de la page, les liens vers vos fichiers CSS, les scripts JavaScript, etc.

- `<body>` contient le contenu principal de la page, tel que les paragraphes, les images, les liens, etc.

Balises et Éléments HTML

Les balises HTML sont utilisées pour définir des éléments sur une page web. Chaque balise a une ouverture (`<balise>`) et une fermeture (`</balise>`). Par exemple, `<p>` est la balise pour les paragraphes, et le texte à l'intérieur est l'élément de cette balise.

```
<p>Ceci est un paragraphe</p>
```

Le nom des balises est très important, car il permet de définir à quoi servira cette balise et donc comment sera affichée cette dernière.

Voici quelques exemples de certaines balises HTML importantes et leurs utilisations :

▼ Titres : `<h1>`, `<h2>`, ..., `<h6>`

Les balises `<h1>` à `<h6>` sont utilisées pour définir les titres de différentes tailles, `<h1>` étant le plus grand et `<h6>` le plus petit.

```
<h1>Titre de Niveau 1</h1>
<h2>Titre de Niveau 2</h2>
<!-- ... -->
<h6>Titre de Niveau 6</h6>
```

▼ Paragraphes : `<p>`

La balise `<p>` est utilisée pour définir des paragraphes de texte. C'est généralement dans cette balise que nous écrivons la plupart de nos textes.

```
<p>Ceci est un paragraphe</p>
```

▼ Images : ``

La balise `` est utilisée pour insérer des images. Elle est généralement utilisée avec l'attribut `src` qui va définir quelle image afficher (je vous explique comment fonctionne les attributs un peu plus tard).

```

```

▼ Conteneur : `<div>`

La balise `<div>` est une balise de conteneur qui est souvent utilisée pour grouper d'autres éléments et appliquer des modifications sur ce qu'il y a à l'intérieur via le css ou le js.

```
<div>
  <p>Contenu à l'intérieur du div</p>
  
</div>
```

▼ Boutons : `<button>`

La balise `<button>` est utilisée pour créer un bouton interactif sur une page web. Il est en général relié à une ou plusieurs fonctions en js pour réaliser divers actions.

```
<button>Cliquez-moi</button>
```

Attributs HTML

Les balises peuvent également avoir des attributs qui fournissent des informations supplémentaires sur l'élément. Les attributs sont spécifiés dans la balise d'ouverture. Par exemple, l'attribut `src` dans la balise `` spécifie la source de l'image. Ainsi l'image qui sera affiché dans cette balise sera l'image `"image.jpg"`.

```

```

Voici donc attributs utile et leurs définitions :

▼ id : `id`

L'attribut `id` est utilisé pour attribuer un identifiant unique à un élément HTML. Cet identifiant peut être utilisé pour le cibler spécifiquement avec du CSS ou du JavaScript

```
<p id="mon_paragraphe">Ceci est un paragraphe.</p>
```

▼ Classes : `class`

L'attribut `class` est utilisé pour attribuer une ou plusieurs classes à un élément HTML. Les classes sont souvent utilisées pour appliquer des styles CSS spécifiques. Contrairement à l'id, plusieurs balises peuvent être de la même classe.

```
<p class="texte_important">Ceci est un texte important.</p>
```

▼ Source (pour les images) : `src`

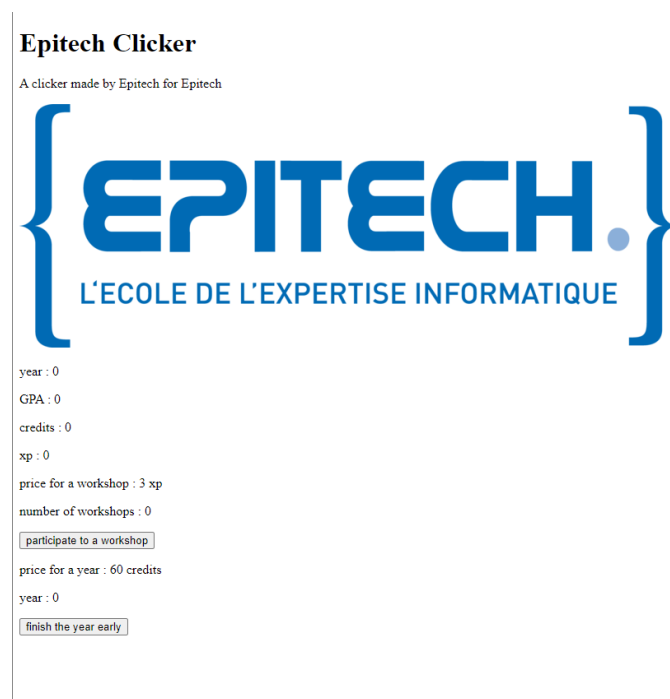
L'attribut `src` est utilisé pour spécifier la source d'une image à afficher sur la page. On utilisera son chemin pour indiquer quel image afficher.

```

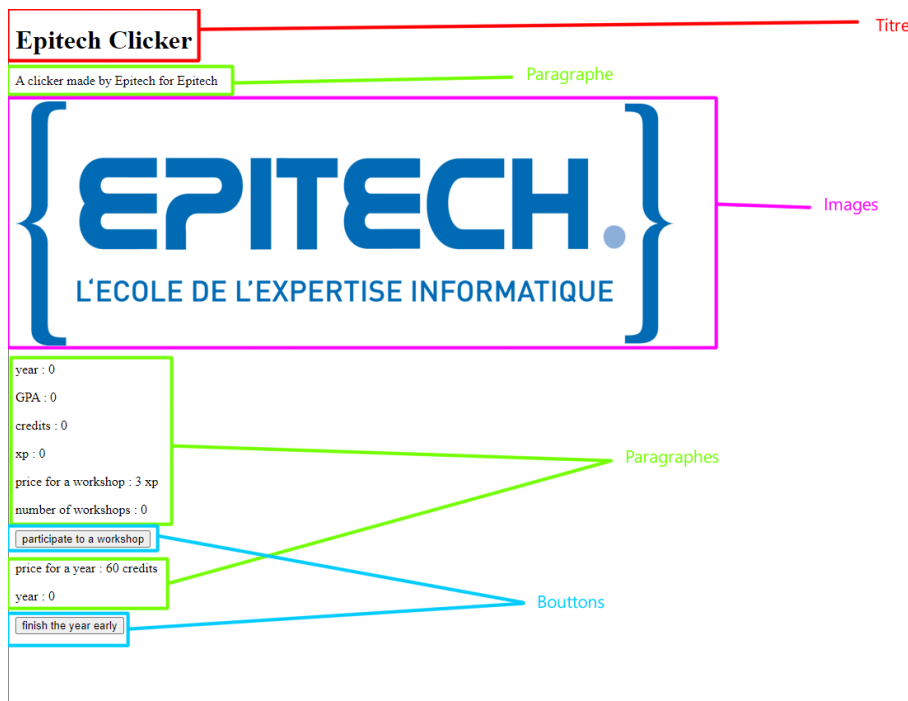
```

Exemple

Voici à quoi ressemble l'epitech clicker fait par Epitech !



Et voici les balises que nous avons utilisé :



Si vous avez besoin de plus d'informations sur l'utilisation de balises html, ou de leurs attributs, vous pouvez suivre ce lien vers la documentation html :

<https://developer.mozilla.org/fr/docs/Web/HTML>

Introduction au CSS

Qu'est-ce que le CSS ?

CSS, ou Cascading Style Sheets, est un langage utilisé pour modifier la présentation visuelle d'une page Web écrite en HTML. Il permet de contrôler la mise en page, la couleur, la police, la taille et d'autres aspects esthétiques d'une page Web.

Comment intégrer le CSS dans une page HTML

Il existe plusieurs façons d'intégrer le CSS dans une page HTML. L'une des méthodes les plus courantes consiste à utiliser une balise `<style>` dans l'en-tête HTML. Voici un exemple :

```
<!DOCTYPE html>
<html>
<head>
  <title>Je suis un titre</title>
  <style>
    body {
```

```

        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
        color: #333;
    }

    h1 {
        color: #0066cc;
    }

    p {
        margin-bottom: 20px;
    }
</style>
</head>
<body>
    <h1>Apprenons à coder en HTML avec Epitech</h1>
    <p>Commençons par l'Epitech clicker</p>
</body>
</html>

```

Une autre manière de l'intégrer consiste à créer un fichier css pour ensuite l'intégrer grâce aux attributs `href` et `rel` dans un élément `<link>`. La balise `<link>` en HTML est utilisée pour lier une page HTML à d'autres ressources externes ou à des fichiers CSS.

Par exemple pour ce fichier css nommé `fichier.css` :

```

body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    color: #333;
}

h1 {
    color: #0066cc;
}

p {

```



```
margin-bottom: 20px;  
}
```

Nous pouvons l'ajouter dans notre css comme ceci :

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Je suis un titre</title>  
  <link rel="stylesheet" href="fichier.css">  
</head>  
<body>  
  <h1 id="numéro1">Apprenons à coder en HTML avec Epitech</h1>  
  <p class="bonjour">Commençons par l'Epitech clicker</p>  
</body>  
</html>
```

Les sélecteurs et les propriétés CSS

- **Sélecteurs** : Les sélecteurs ciblent les éléments HTML que vous souhaitez styliser. Par exemple, `body`, `h1`, et `p` dans l'exemple ci-dessus. Elle peuvent également utiliser des id ou des classes, les id sont représentés par un `#` suivi du nom de l'id et les classes sont représentées par un `.` suivi du nom de la classe.
- **Propriétés** : Les propriétés définissent le style des éléments sélectionnés. Par exemple, `color`, `font-family`, et `background-color` dans l'exemple ci-dessus.

Exemples de propriétés CSS courantes

1. Texte :

- `color` : Couleur du texte. Les valeurs entrées peuvent être hexadécimales ou le nom de la couleur souhaité.
- `font-family` : Police de caractères. Les valeurs entrées seront la police du texte modifiée.
- `font-size` : Taille de la police.

2. Mise en page :

- `margin` : Marge autour de l'élément.

- `padding` : Remplissage à l'intérieur de l'élément.
- `border` : Bordure de l'élément.
- `width` & `height` : La largeur et la longueur de l'éléments.

3. Couleurs :

- `background-color` : Couleur de fond.
- `border-color` : Couleur de la bordure.
- `border-radius` : Rayon des coins de la bordure.
- `box-shadow` : Définis l'ombre de l'élément, sa position et sa couleur.

4. Positionnement :

- `position` : Définit le type de positionnement (relative, absolute, fixed, etc.).
- `top`, `bottom`, `left`, `right` : Positionnement spécifique.
- `text-align` : Défini le positionnement du texte dans l'élément.

5. Animations :

- `transition` : durée d'un mouvement/animation

Remarques

- N'oubliez pas d'utiliser des sélecteurs spécifiques pour cibler précisément les éléments que vous souhaitez styliser.
- N'oubliez pas non plus d'utiliser des `div` pour modifier plusieurs éléments à la fois
- Si vous avez besoin de plus d'informations sur les propriétés du css, js vous laisse ce site qui vous en informera plus sur le sujet :
<https://developer.mozilla.org/fr/docs/Web/CSS>

Introduction au JavaScript

Qu'est-ce que JavaScript ?

JavaScript est un langage de programmation qui permet d'ajouter des fonctionnalités interactives aux pages Web. Il est principalement utilisé pour manipuler le contenu HTML, réagir aux événements utilisateur, et communiquer avec les serveurs pour obtenir ou envoyer des données sans recharger la page.

Intégration du JavaScript dans une page HTML

Vous pouvez intégrer du JavaScript dans une page HTML de différentes manières, mais la méthode la plus courante est d'utiliser la balise `<script>` dans l'en-tête ou à la fin du corps HTML. Voici un exemple :

```
<!DOCTYPE html>
<html>
<head>
  <title>Ma Page JavaScript</title>
</head>
<body>

  <h1 id="mon_titre">c'est un titre banal</h1>

  <script>
    document.getElementById("mon_titre").innerHTML = "je
  </script>

</body>
</html>
```

Vous pouvez également lier votre fichier js avec l'attribut `src` dans l'élément `<script>` puis en ajoutant le chemin de votre fichier js.

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

Concepts de base en JavaScript

▼ Variables :

Pour déclarer des variables, on utilisera les déclarateurs `var`, `let` et `const`

- `var` : a une portée de fonction, peut être redéclaré, est sujette au hoisting (soulevée au sommet de la portée).
- `let` : a une portée de bloc, peut être redéclaré, est soumise au hoisting.
- `const` : a une portée de bloc, ne peut être redéclaré après la première assignation, est soumise au hoisting.

```
var nombre = 42;  
let texte = "Bonjour";  
const PI = 3.14;
```

▼ Fonctions :

Les fonctions sont de bons moyens de réutiliser des bouts de code à plusieurs endroits à la fois.

On les définit avec le déclarateur

`function`

```
function saluer(nom) {  
    alert("Bonjour, " + nom + " !");  
}
```

▼ Manipulation du DOM :

Pour accéder et modifier des balises html, on peut utiliser différentes fonctions :

- `getElementById()` : qui récupérera l'élément en regardant son id

```
document.getElementById("mon_element");
```

- `getElementsByClassName()` : qui récupérera une liste de tous les éléments ayant la classe indiqué

```
document.getElementsByClassName("mes_classes");
```

- `getElementsByTagName()` : récupérera tout les éléments ayant pour nom de balise, le nom indiqué.

```
document.getElementsByTagName("div");
```

▼ Modifier un texte :

Pour modifier une balise html, on modifiera directement ce qu'il se trouve entre la balise d'ouverture et la balise de fermeture. Cette partie s'appelle l'`innerHTML`

```
let titre = document.getElementById("titre_à_modifier");
```

```
titre.innerHTML = "Nouveau titre" //le titre a changer son
```

▼ Modifier une propriété css :

Pour modifier une propriété css, on utilisera l'option `style` pour chercher le css d'un balise, puis on peut chercher sa propriété pour lui assigner une nouvelle valeur.

Par exemple, si je souhaite changer la largeur d'une image :

```
let image_el = document.getElementById("mon_image");  
  
image_el.style.width = "100px"
```

▼ Événements :

Pour réagir à des actions comme les clics ou les événements de la souris, on utilise généralement la fonction `addEventListener` sur un élément du DOM.

La fonction prends en compte 2 arguments :

- L'événement qui activera tout le processus.
- La fonction qui sera activé après l'événement détecté.

Par exemple si je veux afficher dans la console `bonjour je suis un membre d'Epitech` lorsque l'utilisateur clique sur l'élément possédant l'id `boutton_epitech` voici mon code :

```
let bouton_epitech_el = document.getElementById("boutton_  
  
function epitech()  
{  
    console.log("bonjour je suis un membre d'Epitech");  
}  
  
boutton_epitech_el.addEventListener("click", epitech());
```

▼ Conditions et boucles :

N'oubliez pas d'utiliser des boucles et des conditions pour prendre des décisions et répéter des actions.

```
if (condition) {  
    // Code à exécuter si la condition est vraie  
} else {  
    // Code à exécuter si la condition est fausse  
}  
  
for (let i = 0; i < 5; i++) {  
    // Code à répéter  
}
```

Mathématiques

Vous aurez sûrement aussi besoin d'un petit peu de maths pour votre Epitech clicker, voici quelques opérations mathématiques qui pourra vous être utiles pour votre Epitech cliker !

▼ Addition +

L'opérateur d'addition est utilisé pour additionner deux valeurs.

```
let resultat = 5 + 3; // Le résultat est 8
```

▼ Soustraction -

L'opérateur de soustraction est utilisé pour soustraire une valeur d'une autre.

```
let resultat = 7 - 4; // Le résultat est 3
```

▼ Multiplication *

L'opérateur de multiplication est utilisé pour multiplier deux valeurs.

```
let resultat = 6 * 2; // Le résultat est 12
```

▼ Division /

L'opérateur de division est utilisé pour diviser une valeur par une autre.

```
let resultat = 10 / 2; // Le résultat est 5
```

▼ Modulo %

L'opérateur modulo renvoie le reste de la division entière de deux valeurs.

```
let reste = 10 % 3; // Le reste est 1
```

▼ Incrémentation ++ et Décrémentation -

Les opérateurs d'incrément et de décrément ajoutent ou soustraient 1 à une variable, respectivement.

```
let x = 5;  
x++; // Maintenant, x vaut 6 (incrément)  
x--; // Maintenant, x vaut 5 (décrément)
```

▼ Opérations combinées

Vous pouvez combiner plusieurs opérations mathématiques dans une seule expression.

```
let resultat = (4 + 2) * 3; // Le résultat est 18
```

▼ Opérateurs d'affectation

Les opérateurs d'affectation tels que +=, -=, *=, /= permettent de raccourcir l'écriture des opérations.

```
let total = 10;  
total += 5; // Équivalent à total = total + 5; (maintenant, total vaut 15)
```

▼ Arrondir

Pour arrondir un nombre à virgule, on peut utiliser 3 fonction :

- `Math.ceil()` pour arrondir à l'entier supérieur

```
Math.ceil(5.3) //arrondi à 6
```

- `Math.floor()` pour arrondir à l'entier inférieur

```
Math.floor(5.9) //arrondi à 5
```

- `Math.round()` pour arrondir à l'entier le plus proche

```
Math.floor(5.6) //arrondi à 6  
Math.floor(5.4) // arrondi à 5  
Math.floor(5.5) // arrondi à 5
```

Je vous laisse regarder la documentation sur le js si vous avez besoin de plus d'informations sur ce langage : <https://developer.mozilla.org/fr/docs/Web/JavaScript>

Bonus

Vous avez déjà fini votre Epitech clicker ? Faites nous voir tout ça !

Et si jamais vous souhaitez le reprendre à la maison, n'hésitez pas à notifier Epitech technology sur les réseaux ! Vous pouvez également nous envoyer votre jeux dans le discord du coding club !

Pour aller plus loin...

J'imagine que vous souhaitez aller plus loin, laissez moi vous donner quelques pistes pour vous améliorer. Pourquoi pas ...

▼ Ajouter une icone et un titre à la page

Chaque page web a une icone et un titre associé, pourquoi pas en ajouter un vous aussi ?



Pour ajouter un titre, utilisez la balise `<title>` dans la balise `<head>` et inscrivez le titre que vous voulez !

```
<head>  
  <title>Epitech Clicker 2023</title>  
</head>
```


Ensuite, munissez vous de votre meilleur logo et ajouter le avec la balise `<link>` et les attributs `rel` ainsi que `href`.

```
<head>
  <title>Epitech Clicker 2023</title>
  <link rel="icon" href="img/icon.png">
</head>
```

▼ Ajouter des challenges

Pourquoi pas ajouter des challenges et des récompenses à votre jeux, comme un highscore ou un challenge basé sur le temps ?

▼ Augmenter la difficulté

Pourquoi pas non plus augmenter la difficulté chaque année qui passe ? Vous pouvez par exemple divisé les gains par 2 chaque années, ou augmenter les prix des workshops tout les ans !