

II. Interactivité dans une page web, côté client

Il y a quelques années, le Web était statique et tous les utilisateurs voyaient la même page web. Toutefois, de nos jours, pratiquement toutes les pages web sont dynamiques et intègrent d'autres composants que le simple code HTML5.

1) Interactions grâce à CSS

Une manière très simple d'interagir avec un utilisateur consiste à le faire grâce aux fichiers CSS (Cascading Style Sheet). Cette interaction, grâce à l'existence des classes et des pseudo-classes en CSS, permet de faire réagir n'importe quel élément HTML à des actions d'un utilisateur.

Rappel : une classe en CSS est définie grâce à un point. Par exemple, la classe appelée couleur s'écrit :

```
.couleur {  
    color: red;  
    background-color: aliceblue;  
}
```

Pour l'appeler, on utilisera une balise `<h1 class=couleur> le titre </h1>`.

Les pseudo-classes permettent de modifier les classes suite à des interactions. recherchez quelques pseudo-classes CSS qui semblent intéressantes pour nous et notez la syntaxe ci-dessous.

.....
.....



— À faire vous-même 1 —

Dans l'exercice 1, à télécharger sur bouillotvincent.github.io, on souhaite faire un petit recueil de poèmes interactifs.

On dispose de trois images de poèmes de Baudelaire que l'on veut mettre côte à côte.

Lorsque l'on passe notre curseur sur un de ces poèmes, on souhaite qu'il se **transform(e)** à l'aide de **CSS** (aidez-vous de la page de Mozilla) :

- en s'agrandissant d'un facteur 5 ;
- en se translatant sur l'axe des X de 40% ;
- en se translatant sur l'axe des Y de 20%.

On utilisera la classe **zoom** et la pseudo-classe **hover** appliquée à **zoom**.



— À faire vous-même 2 —

On veut maintenant jouer un peu avec le texte.

1) On dispose d'un poème de Verlaine qui est bien trop long. On ne souhaite donc afficher que les 100 premiers pixels verticaux du poème.

Pour cela, on va modifier la classe `hide` en ajoutant un attribut appelé **overflow**.

2) Sommes-nous bêtes! On ne peut plus lire le poème. On veut donc utiliser la pseudo-classe **active** appliquée à la classe `hide` pour afficher le poème tant que l'utilisateur garde le bouton gauche de la souris appuyé.

Comme dans l'exemple projeté au tableau, on souhaite conserver l'encadrement de telle manière à ce que celui-ci s'adapte à la nouvelle taille du texte. Pour cela on va changer **max-height**. Recherchez sur Mozilla cette propriété CSS pour savoir comment adapter la taille verticale !

Conclusion : Les CSS permettent de faire de superbes effets visuels mais demandent un investissement très important. C'est en fait un métier plutôt bien payé : 52 691\$ en moyenne aux USA.

2) Interactions grâce à HTML

Même si, en apparence, cela n'est pas le plus sexy 2.0, HTML permet d'interagir avec l'utilisateur grâce à toute une ribambelle d'inputs dans des formulaires.

Les formulaires sont en fait les éléments les plus importants des sites internet car ils permettent à l'utilisateur de transmettre des informations au serveur.

Un formulaire devra être délimité par une balise **form** (même si ce n'est pas une obligation) :

```
<form>
```

```
.....
```

```
</form>
```

Trois types principaux de balises sont utilisés pour construire un formulaire : `select`, `input` et `label`.

- `label` possède pour attribut `for` permettant de contrôler un `input`.
- `input` possède un attribut `type` qui lui permet de jouer des rôles très différents.
- `select` permet de créer des listes déroulantes et utilise la balise `option` pour créer les différents items de la liste déroulante.



— À faire vous-même 3 —

À partir du dossier Exo2 et à l'aide de <https://developer.mozilla.org/fr/docs/Web/HTML/Element>, explorer les différentes possibilités de formulaire et créez un clone de la page proposée dans l'exercice 2.

Si vous avez fini en avance, créez un CSS permettant de mettre en forme votre formulaire de création de Pokémon !

Conclusion : dans cette partie, nous avons créé un superbe formulaire. Résumez en un mot ce que fait ce formulaire pour l'instant : _____

Il va donc falloir qu'un serveur récupère notre formulaire ou qu'un programme informatique (Javascript...) en fasse quelque chose !

3) Interactions grâce à JavaScript (une introduction)

Un programmeur informatique ne peut aujourd'hui pas se passer de Javascript. Toutefois, en Première, le but 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.

Un peu d'histoire : 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 langages, dont Java, pour mettre au point JavaScript.



— À faire vous-même 4 —

Dans le dossier Exo3, vous avez trois fichiers : index.html, style.css et script.js.

Ouvrez le fichier index.html.

Y'a-t-il une ligne nouvelle ? D'après vous, que fait cette ligne ? _____



— À faire vous-même 5 —

Dans le dossier Exo3, à l'aide de Brackets, modifiez le fichier **script.js** et saisissez le code ci-dessous :

```
alert("Le JavaScript fonctionne !")
```

Rechargez la page index.html. Que remarquez-vous ?

Rappel : Le but n'est pas d'apprendre à programmer en JavaScript : nous nous contenterons pour le moment de cette simple instruction "alert" qui crée un popup. Evidemment, JavaScript permet de faire bien plus de choses : variable, condition, boucle, fonction... Si vous voulez en apprendre plus sur la programmation en JavaScript, je vous invite à consulter le site de Mozilla!

Dans l'exemple ci-dessus, l'instruction "alert" est exécutée dès l'ouverture de la page web, il est tout à fait possible de lier l'exécution de certaines instructions JavaScript à l'action d'un utilisateur (par exemple un clic sur un bouton).



— À faire vous-même 5 —

Dans le code HTML, ajoutez un bouton dans le corps à l'aide de l'instruction :

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

Modifiez le code JavaScript comme suit :

```
function maFonction() {  
    alert("Le JavaScript fonctionne !")  
}
```

A quel moment le code JavaScript est-il exécuté ? _____

Décrivez avec vos propres mots ce qu'il se passe lorsque l'on ouvre le site internet **puis** que l'on appuie sur le bouton.
