

Du code qui s'exécute dans le navigateur :

- Sélectionner des **éléments**
- Détecter et gérer les **événements** (click...)
- Changer de **classe** ou style

## Les variables

Pour créer une variable, il faut utiliser un mot clé :

**let** → variable locale

**var** → variable globale

**const** → constante

**let** est à privilégier

```
let x = 42
```

```
let beer = "BMB Rousse"
```

## Les boucles for

Une liste peut être parcourue selon ses indices ou ses valeurs

```
let list = ["Chimay",
            "Duvel", "BMB" ]
```

```
for (let i in list) { ...
i est un numéro : 0, 1, 2...
```

```
for (let x of list) { ...
x est une valeur : Chimay, Duvel...
```

## Intervalles

L'instruction **setInterval** permet de répéter une action indéfiniment avec un délai (en millisecondes)

```
setInterval(action, time)
```

```
setInterval(function() {
  h1.classList.toggle("important")
}, 100)
```

## Sélection d'éléments

Selon un sélecteur CSS :

**querySelector** → récupère un seul élément

**querySelectorAll** → récupère tous les éléments

```
let h1 = document.querySelector("h1")
```

```
<h1>Bières</h1>
<ul>
  <li class="beer">Chimay</li>
  <li class="beer">Duvel</li>
  <li class="beer">BMB</li>
</ul>
```

```
let beers = document.querySelectorAll(".beer")
```

## Gestion de classes CSS

Tout élément possède un attribut « **classList** »

**classList.add** → ajoute une classe

**classList.remove** → supprime une classe

**classList.toggle** → bascule une classe

**classList.contains** → interroge l'existence d'une classe

La classe, définie en CSS, est appliquée immédiatement.

```
h1.classList.add("important")
```

```
.important {
  color : orange;
}
```

## Événements

Un événement peut être détecté et une action peut être réalisée

```
element.addEventListener(event, action)
```

Exemple d'événements : **click**, **dblclick**, **keyup**, **keydown**, **mouseenter**, **mouseleave**...

```
h1.addEventListener("click", function() {
  h1.classList.add("important")
})
```