

## JS

--

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Table des matières

<i>Objectif Pédagogique</i> .....	<b>3</b>
<i>DOM JS : Présentation</i> .....	<b>4</b>
<i>DOM JS : Sélectionner un élément Html</i> .....	<b>7</b>
<i>DOM JS : Placer un élément Html</i> .....	<b>10</b>
<i>DOM JS : Créer un élément Html</i> .....	<b>13</b>
<i>DOM JS : Modifier les attributs d'un élément Html</i> .....	<b>16</b>
<i>DOM JS : Transformer une liste d'élément Html en tableau</i> .....	<b>17</b>
<i>DOM JS : Obtenir la taille d'un élément Html</i> .....	<b>21</b>
<i>DOM JS : modifier le style d'un élément Html</i> .....	<b>22</b>
<i>DOM JS : Ajouter ou supprimer des classes Css</i> .....	<b>26</b>
<i>DOM JS : Gestion des évènements</i> .....	<b>30</b>

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS

### Objectifs Pédagogiques

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS

### DOM JS : Présentation

**Le Document Object Model (DOM)** : Le DOM est un modèle de document chargé dans le [navigateur](#).

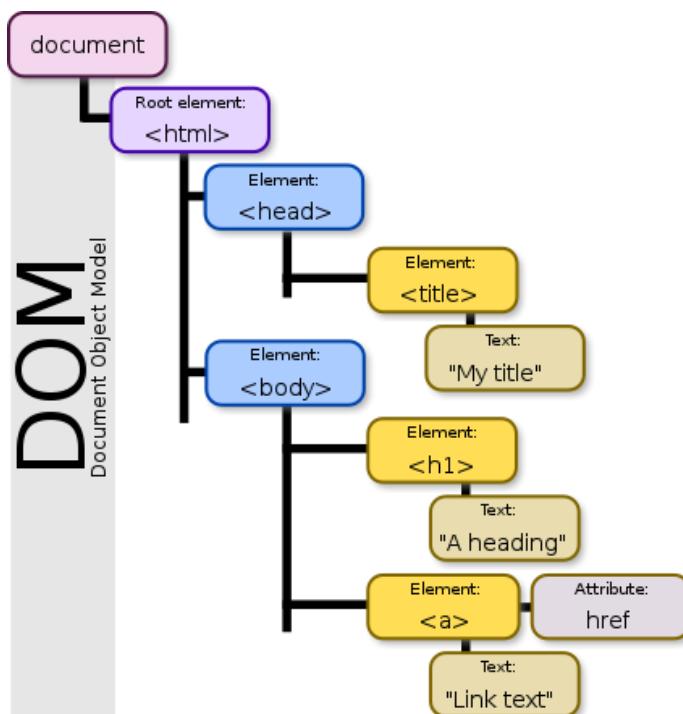
La représentation du document est un arbre nodal. Chaque nœud représente une partie du document (par exemple, un [élément](#), une chaîne de caractères ou un commentaire). (MDN : <https://developer.mozilla.org/fr/docs/Glossary/DOM> )

Le navigateur va lire notre document HTML et organiser les éléments sous forme d'arborescence.

Il en fait une « traduction » pour l'afficher d'une certaine manière.

JS va pouvoir interagir avec ce Document créé par le navigateur

On va pouvoir sélectionner, modifier des éléments, en ajouter, en supprimer...

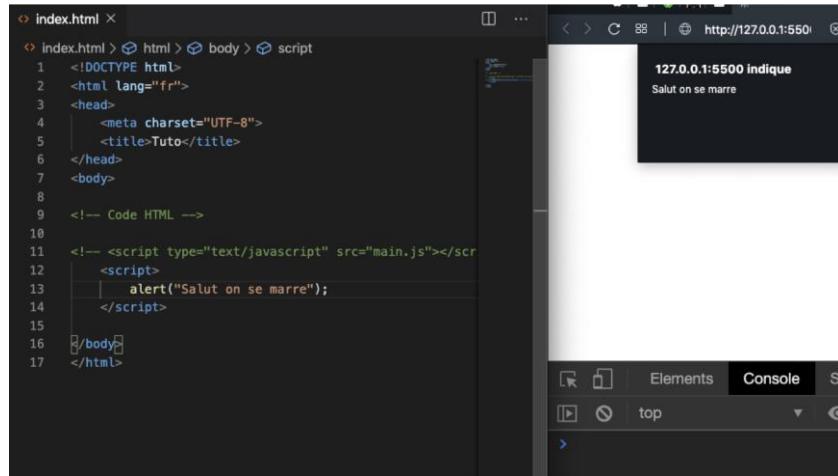


Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## SetUp :

On peut coder du JS directement dans un fichier html entre des balises <script></script> :



The screenshot shows a code editor with the file 'index.html'. The code contains a script block with an alert message:

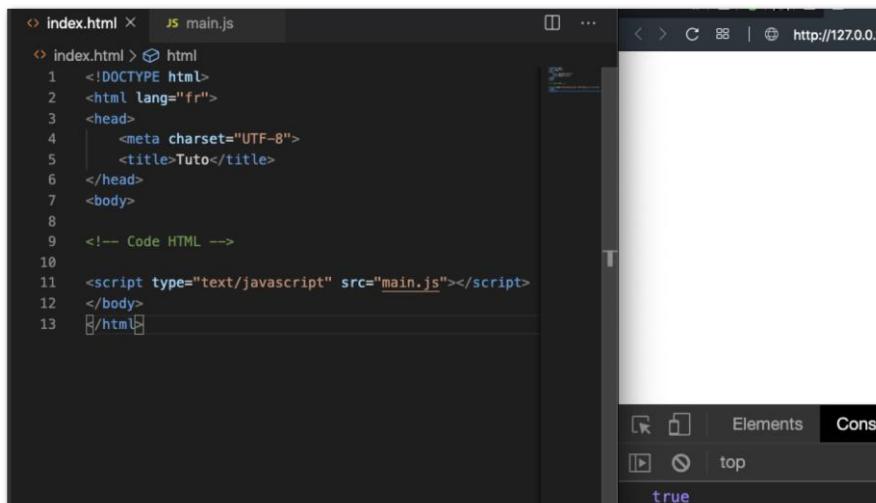
```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
</head>
<body>
    <!-- Code HTML -->
    <!-- <script type="text/javascript" src="main.js"></script>
        |     alert("Salut on se marre");
        | </script>
    15
    16 </body>
    17 </html>

```

To the right, a browser window shows the result: "127.0.0.1:5500 indique Salut on se marre". The developer tools' console tab shows the output of the alert command.

La méthode la plus recommandée sera d'avoir un fichier JS et de le relier au fichier html :



The screenshot shows a code editor with the file 'index.html'. It includes a script tag linking to 'main.js':

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
</head>
<body>
    <!-- Code HTML -->
    <script type="text/javascript" src="main.js"></script>
    </body>
    </html>

```

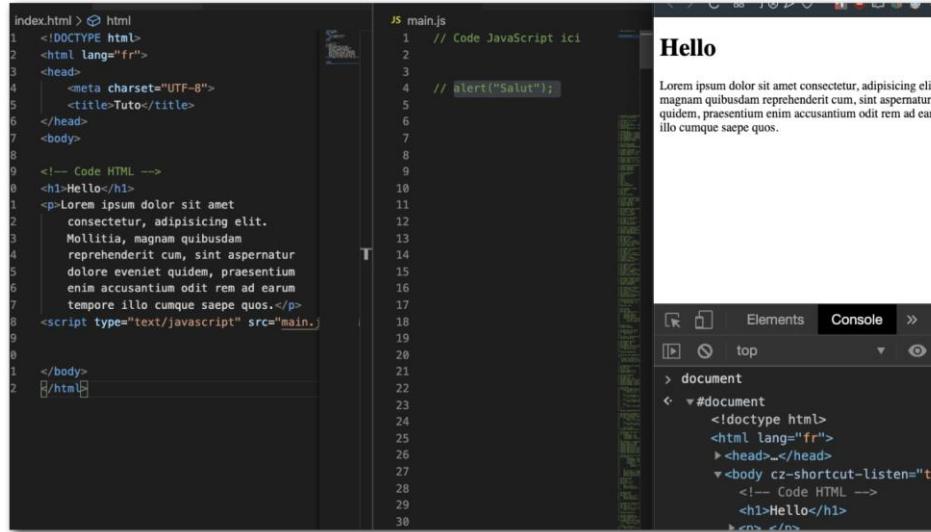
To the right, a browser window shows the result: "true". The developer tools' console tab shows the output of the alert command.

Souvent on relie notre JS à la fin du fichier html  
(pour s'assurer que les scripts soient importés une fois tous les éléments html chargés)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

On peut aussi (plus rarement) écrire du JS directement dans la console du navigateur.



The screenshot shows a browser window with developer tools open. On the left, the DOM tree displays the HTML structure of a page. In the center, the JavaScript console shows a single line of code: `// alert("Salut");`. To the right, the preview pane shows a simple "Hello" page with a "Hello" heading and some placeholder text. The bottom of the screen shows the browser's navigation bar and the developer tools toolbar.

En tapant « document » dans la console on affiche tout le Html lu par le navigateur.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

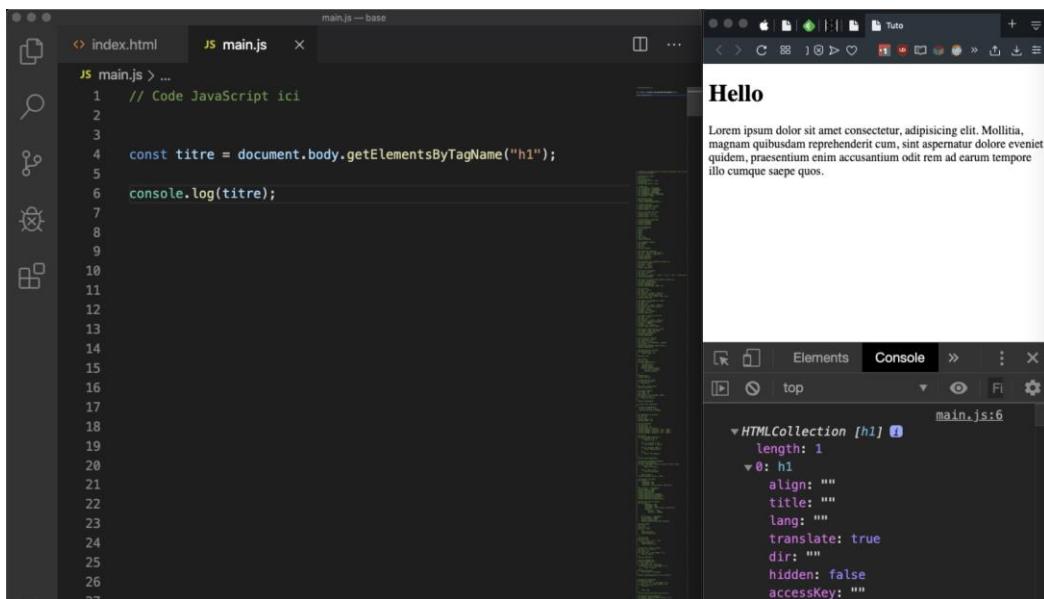
## DOM JS : Sélectionner un élément Html

Une première manière de sélectionner un élément Html (un lien, un titre, une image, ... , n'importe quel élément Html).

**getElementsByName( )**

permet de sélectionner un élément par le nom de sa balise

Côté Html nous avons une structure de page simple avec dans le <body> un titre <h1> et un paragraphe <p>.



The screenshot shows a development setup with two main windows. On the left is a code editor displaying a file named 'main.js' with the following content:

```
// Code JavaScript ici
const titre = document.body.getElementsByTagName("h1");
console.log(titre);
```

On the right is a browser window showing the result of running the script. The title bar says 'Tuto'. The main content area displays the word 'Hello' in bold. Below it, the browser's developer tools are open, specifically the 'Console' tab. It shows the output of the console log command: an object representing an HTMLCollection with one item (index 0), which is an h1 element with the text 'Hello'.

Ici dans une constante titre on se place sur document puis sur le body et sur le body on sélectionne TOUS les titres « h1 » de la page en utilisant getElementsByTagName

On affiche alors la variable titre en console

La console du navigateur nous affiche un tableau de type HTMLCollection, un tableau qui va contenir tous les titres h1 de notre page. On a accès à toutes les propriétés de ces éléments Html (son alignement, la couleur du texte, sa position dans la page, etc...)

On constate également que ce tableau est indexé (à l'indice [0], le premier titre h1 de la page)

Défi : dans la console, trouver les propriétés innerText ou innerHtml.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

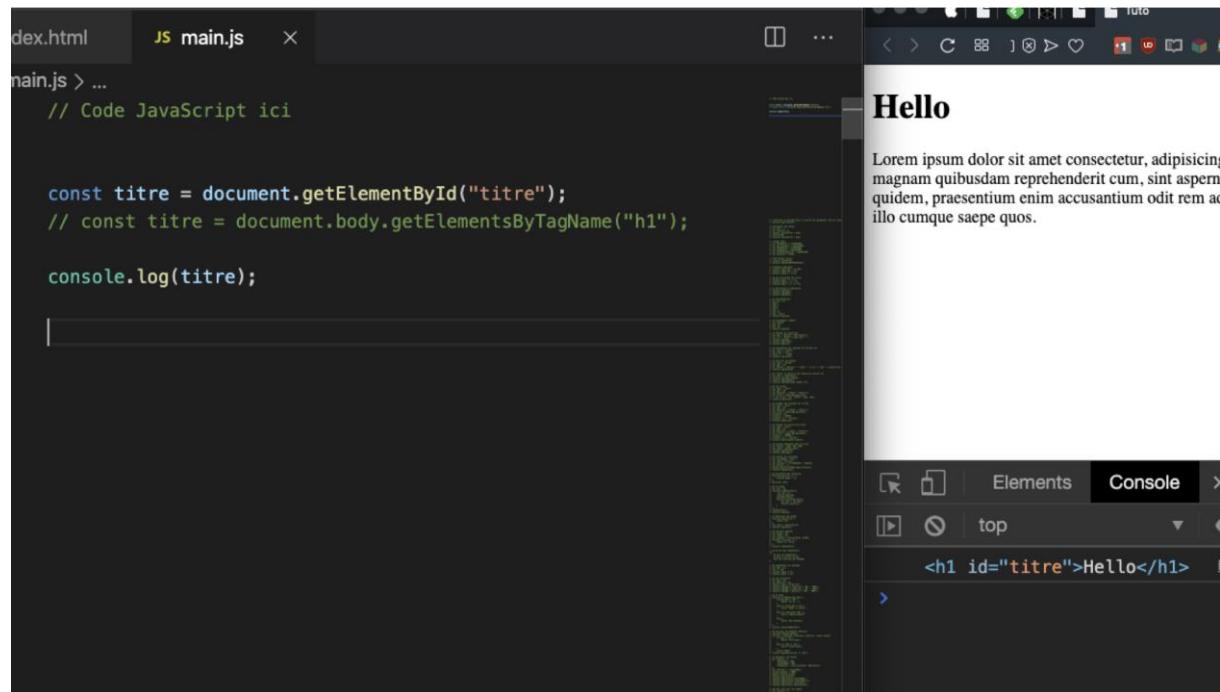
## getElementById()

On peut aussi chercher un élément html par son id, on va utiliser getElementById ()

(récupérer un élément selon le nom de son attribut id)

getElementById () renvoi directement les données elles-mêmes et non pas sous forme de tableau

Il n'y a pas de S à Element dans le nom de la fonction, la console nous affichera le code html d'un seul élément en particulier (\*en html on n'a qu'un seul id avec un certain nom par page)



```

dex.html   JS main.js  ...
main.js > ...
// Code JavaScript ici

const titre = document.getElementById("titre");
// const titre = document.body.getElementsByTagName("h1");

console.log(titre);

```

Ici dans la constante titre on sélectionne l'élément html qui possède l'id (attribut « titre »)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## querySelector() / querySelectorAll()

CSS utilise déjà un système de sélecteur très précis, on peut en profiter en JS  
**querySelector ()** va sélectionner l'élément qu'on lui donne à chercher.  
Dès qu'il le trouve il s'arrête de chercher, il sélectionne le premier élément.

```
const titre = document.querySelector("h1");
```

Ci-dessus on sélectionne le premier titre <h1> de la page.

Nous pouvons utiliser **querySelector** pour sélectionner le premier élément possédant un attribut id avec cette notation #nomId :

```
const qqch = document.querySelector("#monTitre");
```

Sélection de tous les éléments possédant une classe Css nommée « maClasseSpeciale », pour cela on utilise la fonction **querySelectorAll()** :

```
const liens = document.querySelectorAll(".maClasseSpeciale");
```

**querySelectorAll()**

Permet de sélectionner tous les éléments sous forme d'objet [NodeList](#).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

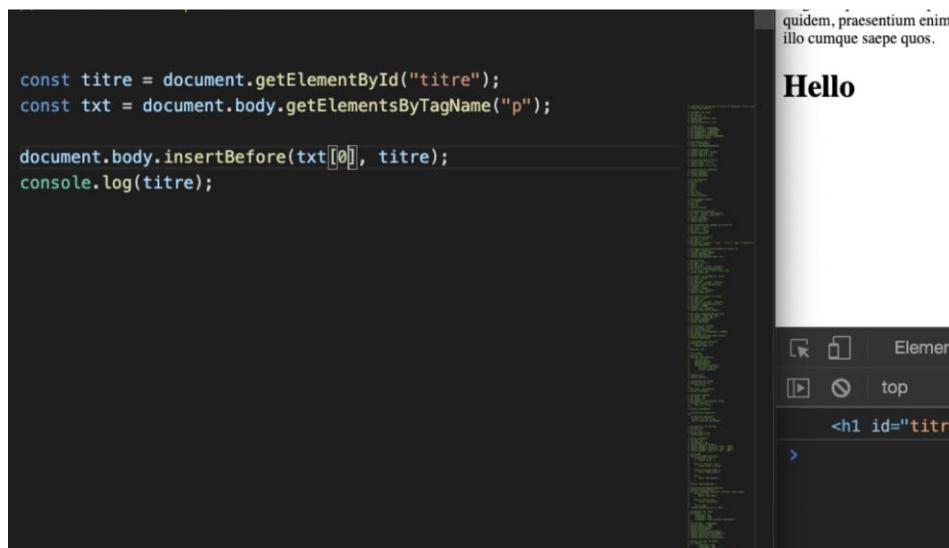
# JS

## DOM JS : Placer un élément Html

Maintenant que l'on sait comment trouver un élément du DOM, on va pouvoir en plus le sélectionner, le placer dans notre page.

### **insertBefore()**

Cette fonction permet de placer un élément AVANT un autre :



```
const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");

document.body.insertBefore(txt[0], titre);
console.log(titre);
```

Ici dans document.body nous allons insérer le 1<sup>er</sup> paragraphe avant le titre.

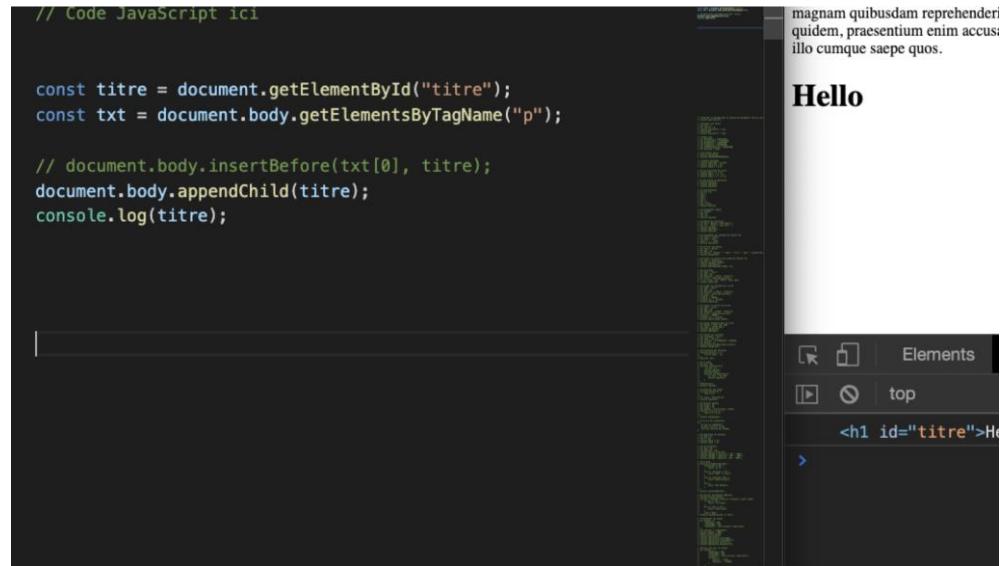
L'utilisation de cette fonction : on lui renseigne quel élément on veut placer (txt[0]) et avant quel éléments on veut l'insérer (titre)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## appendChild()

Via cette fonction il nous sera possible d'insérer un élément à la suite d'un autre.



```
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");

// document.body.insertBefore(txt[0], titre);
document.body.appendChild(titre);
console.log(titre);
```

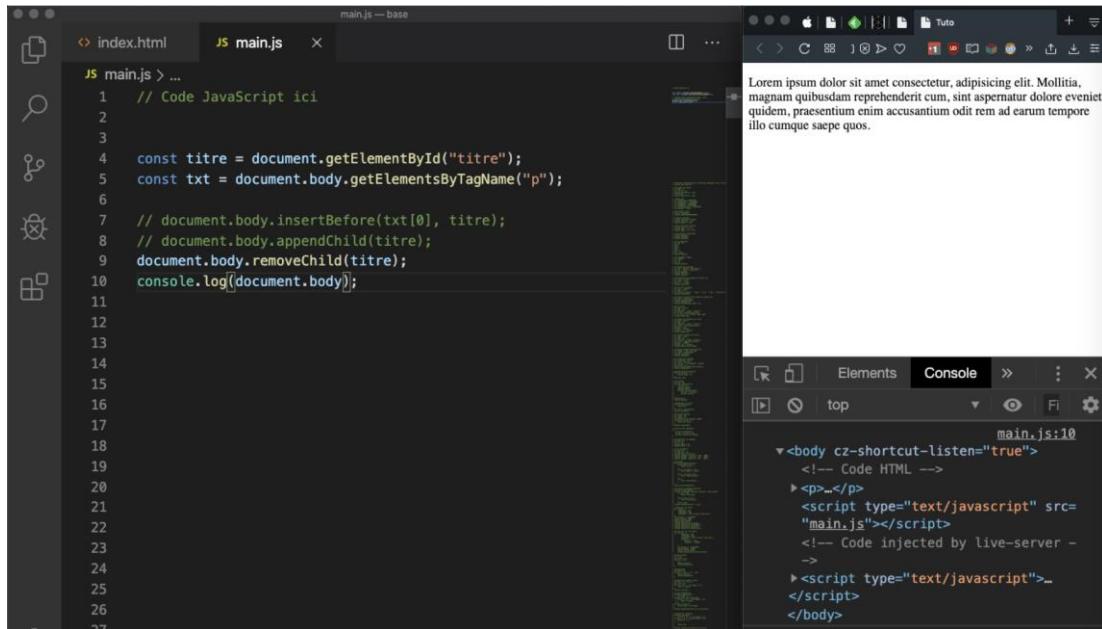
Ici en utilisant cette fonction sur document.body je vais placer le titre à la fin de la balise <body>

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## removeChild()

Avec cette fonction nous allons pouvoir supprimer un élément Html



The screenshot shows a code editor with a file named `main.js` containing the following JavaScript code:

```

1 // Code JavaScript ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.body.getElementsByTagName("p");
6
7 // document.body.insertBefore(txt[0], titre);
8 // document.body.appendChild(titre);
9 document.body.removeChild(titre);
10 console.log(document.body);
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

```

To the right, a browser window displays the resulting HTML structure. The `Console` tab of the developer tools shows the output of the `console.log` statement, which prints the current state of the `document.body`.

Ici on supprime le titre de notre <body>

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

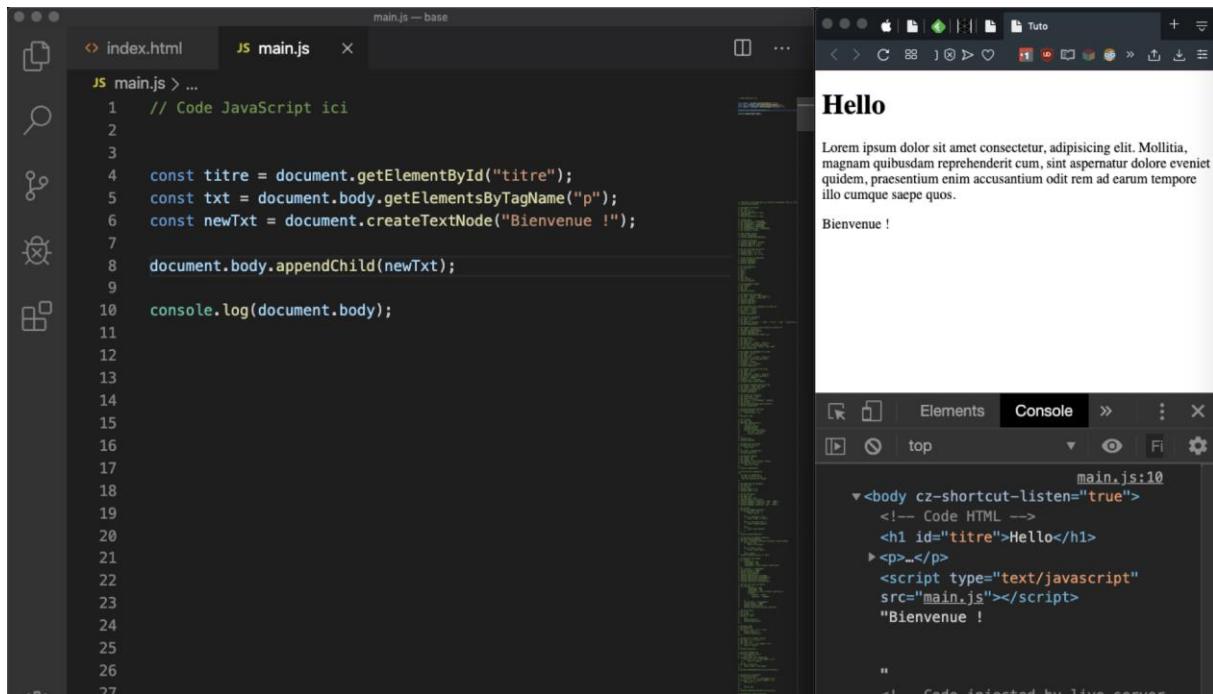
## DOM JS : Créer un élément Html

En JS il est tout à fait possible de créer des éléments Html dynamiquement. Il y a deux étapes à prendre en compte :

La création d'un élément Html puis son placement dans la page Html.

### createTextNode()

Une première fonction qui va nous permettre de créer du texte.



The screenshot shows a browser window with developer tools open. On the left, the code editor displays a JavaScript file named 'main.js' with the following content:

```

main.js — base
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.body.getElementsByName("p");
6 const newTxt = document.createTextNode("Bienvenue !");
7
8 document.body.appendChild(newTxt);
9
10 console.log(document.body);
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

```

On the right, the browser window shows the resulting HTML output:

Hello

Bienvenue !

The browser's developer tools also show the DOM structure and the injected JavaScript code:

```

main.js:10
<body cz-shortcut-listen="true">
  <!-- Code HTML -->
  <h1 id="titre">Hello</h1>
  <p></p>
  <script type="text/javascript" src="main.js"></script>
  "Bienvenue !
"
```

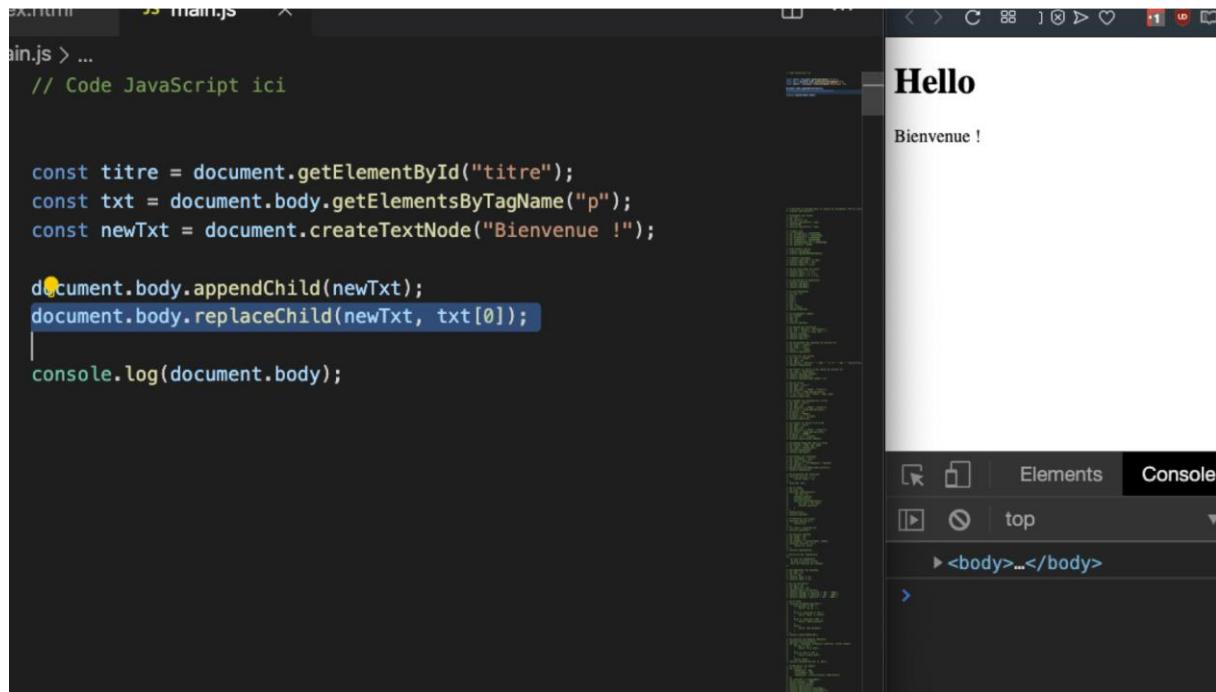
L6 on crée un nouvel élément de texte dans la variable newTxt, L8 on n'oublie pas de placer l'élément dans la page ici dans le <body>.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## replaceChild()

Cette fonction va permettre de remplacer un élément Html par un autre.



The screenshot shows a browser window with developer tools open. On the left, the code in `main.js` is displayed:

```

main.js > ...
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");
const newTxt = document.createTextNode("Bienvenue !");

document.body.appendChild(newTxt);
document.body.replaceChild(newTxt, txt[0]);

console.log(document.body);

```

The `document.body.replaceChild(newTxt, txt[0]);` line is highlighted with a blue box. On the right, the browser window shows the result: "Hello" above "Bienvenue !" and the browser's developer tools console shows the output of `><body>...</body>`.

Ici dans le body notre variable newTxt remplacera le 1<sup>er</sup> paragraphe de la page (txt[0])

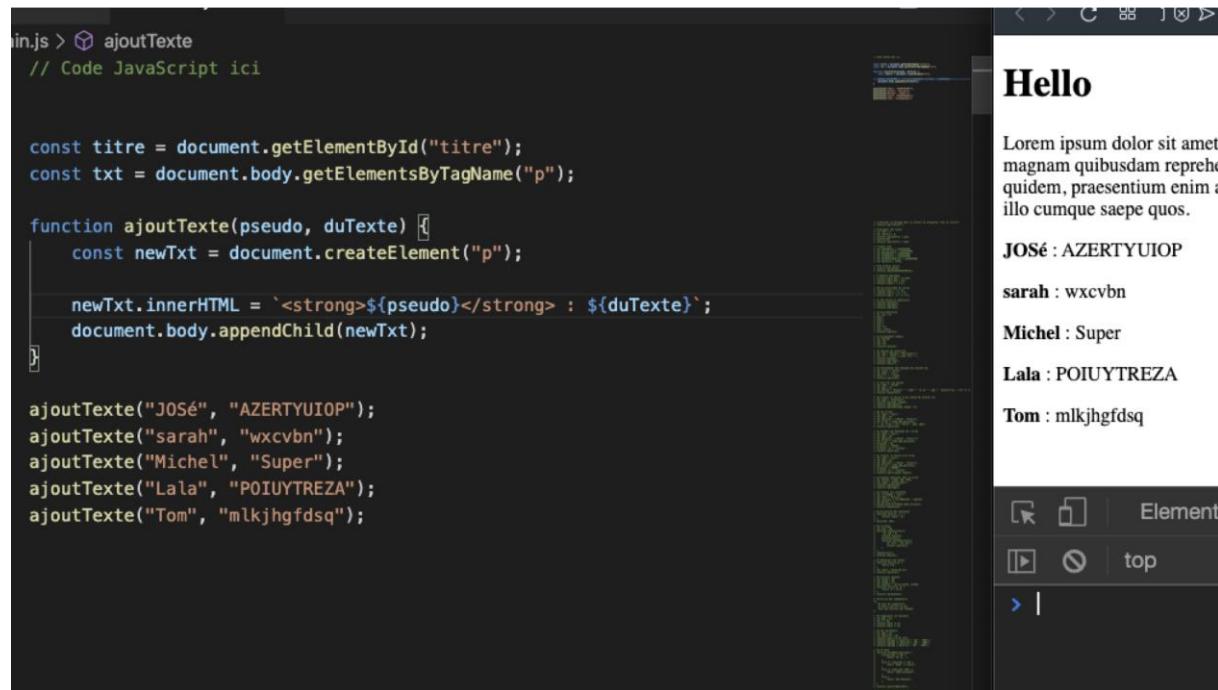
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## createElement()

Cette fonction permet de créer n'importe quel élément Html (un titre, une image, une div etc...)

Bonus : avec la propriété innerHTML que possède tout élément Html, nous pourrons modifier ce dernier.



```

in.js > ajoutTexte
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");

function ajoutTexte(pseudo, duTexte) {
    const newTxt = document.createElement("p");

    newTxt.innerHTML = `<strong>${pseudo}</strong> : ${duTexte}`;
    document.body.appendChild(newTxt);
}

ajoutTexte("JOSé", "AZERTYUIOP");
ajoutTexte("sarah", "wxcvbn");
ajoutTexte("Michel", "Super");
ajoutTexte("Lala", "POIUYTREZA");
ajoutTexte("Tom", "mlkjhgfdsq");

```

The screenshot shows a browser window with developer tools open. On the left, the JavaScript console displays the code above. On the right, the browser's content area shows the output: "Hello" followed by a list of names and their messages. The developer tools sidebar shows the element tree and the current selection path.

Ici dans une fonction ajoutTexte, dans une constante newTxt on crée un élément Html <p> (un paragraphe), ensuite avec newTxt.innerHTML on modifie directement depuis JS son code HTML

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

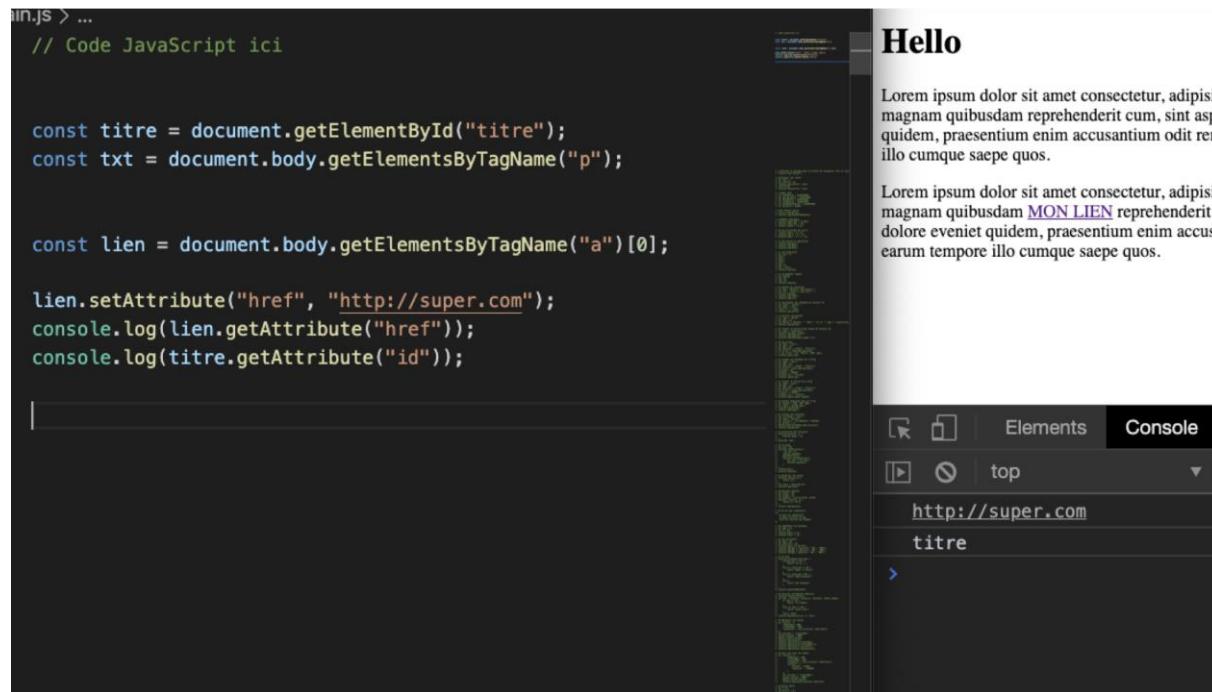
## JS

### DOM JS : Modifier les attributs d'un élément Html

Sur chaque élément HTML on retrouve des attributs, <h1 id="titre">Titre</h1>  
(Les id, les class, le type, le src etc ...)

En JS on peut récupérer les attributs d'un élément avec la fonction **getAttribute()**

Et on peut les modifier avec **setAttribute()**. On indique ce qu'on veut remplacer et par quoi on le remplace.



The screenshot shows the browser's developer tools open. On the left, the 'Elements' tab displays the DOM tree. In the center, there is a preview of the page with the text "Hello". On the right, the 'Console' tab shows the following output:

```

Hello
Lorem ipsum dolor sit amet consectetur, adipisicing magnam quibusdam reprehenderit cum, sint aspernatur quidem, praesentium enim accusantium odit rem illo cumque saepe quos.
Lorem ipsum dolor sit amet consectetur, adipisicing magnam quibusdam MON LIEN reprehenderit cum dolore eveniet quidem, praesentium enim accusantium tempore illo cumque saepe quos.

Elements
top
http://super.com
titre
>

```

The code in the 'Console' tab is as follows:

```

// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");

const lien = document.body.getElementsByTagName("a")[0];

lien.setAttribute("href", "http://super.com");
console.log(lien.getAttribute("href"));
console.log(titre.getAttribute("id"));

```

Ici je récupère le 1<sup>er</sup> lien sur la page avec `getElementsByTagName("a")[0]`, puis on modifie l'attribut href de ce lien avec `setAttribute`, enfin en console on affiche directement la valeur de cet attribut en utilisant `getAttribute`.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

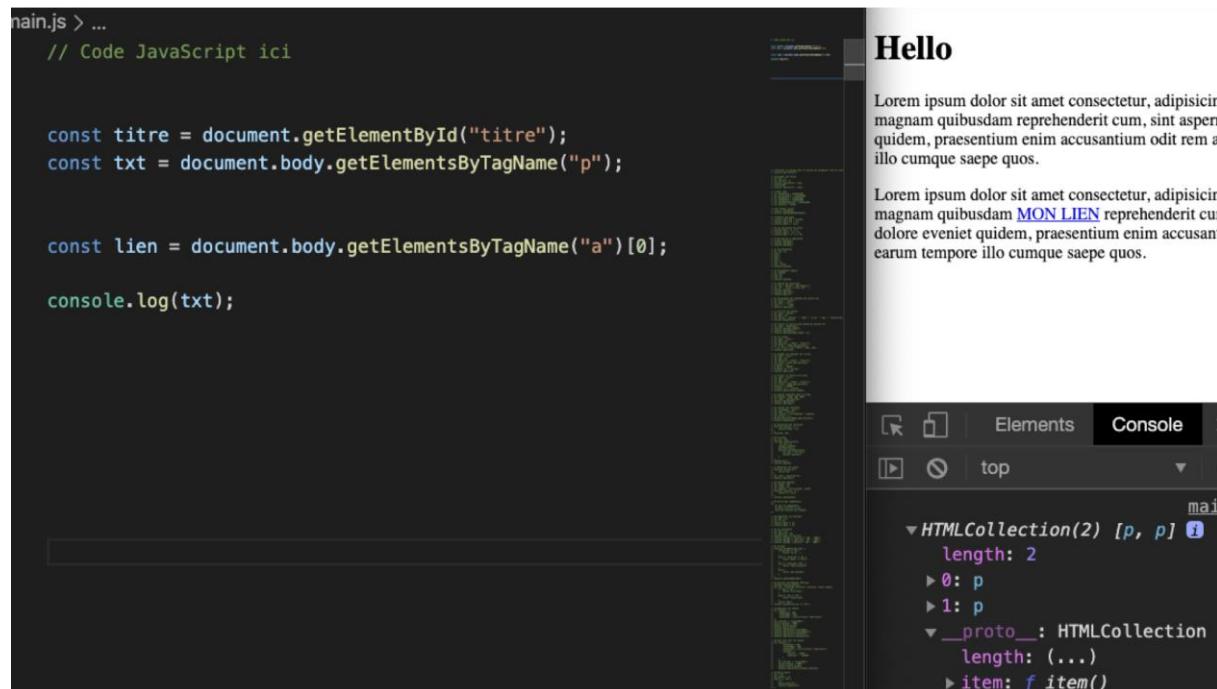
## DOM JS : Transformer une liste d'élément Html en tableau

### Array.from()

Si on veut changer tous les paragraphes d'une page, on a déjà vu Map avec les tableaux ou on lui passe des entrées et il les remplace.

Problème : les éléments HTML ont beaucoup d'attributs mais ils n'ont pas .map()

En JS on a une méthode simple pour transformer quelque chose en tableau la méthode Array.from()



```

main.js > ...
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");

const lien = document.body.getElementsByTagName("a")[0];
console.log(txt);

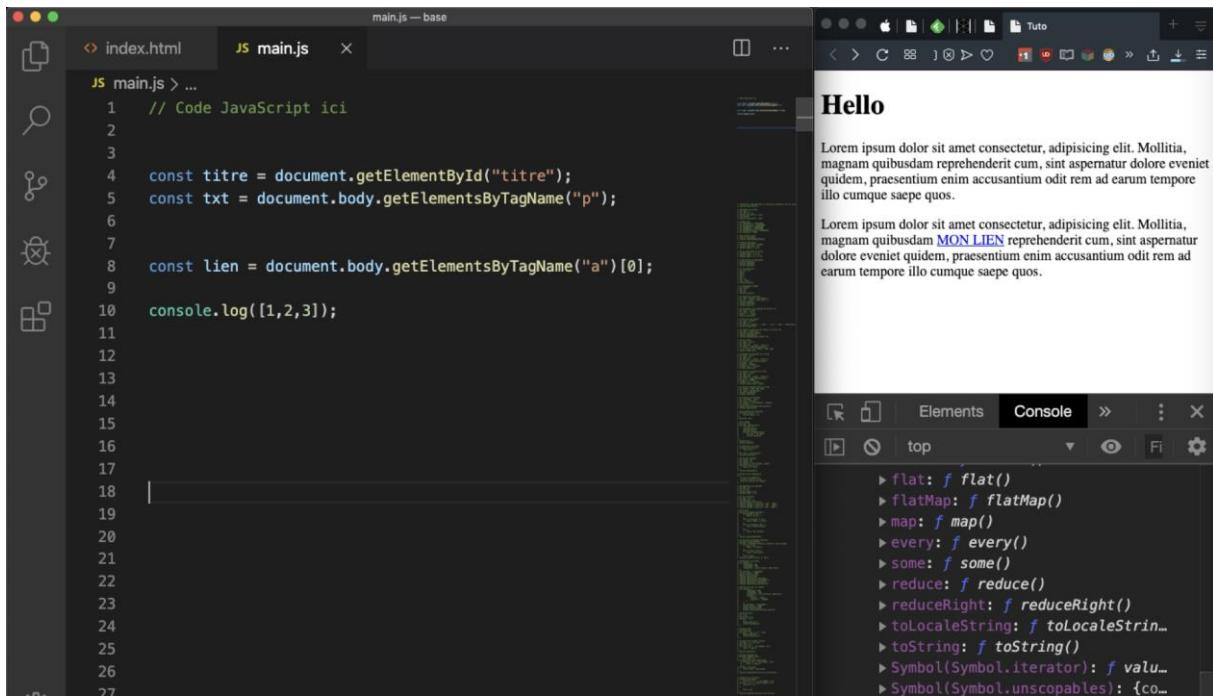
```

The screenshot shows the browser's developer tools. On the left, the 'Elements' tab displays the DOM tree. In the center, the 'Console' tab shows the output of the code execution. The output includes the title element and two paragraph elements. On the right, there is some placeholder text: "Hello" followed by two paragraphs of Latin text. Below the placeholder text, the browser's developer tools show the 'Console' tab with the output of `txt` being an `HTMLCollection(2)` object, which is expanded to show its length (2) and two items (both `p` elements).

Ci-dessus dans une constante txt on récupère tous les paragraphes de la page. La console nous affiche un tableau d'un type particulier HTMLCollection.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

main.js — base
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.body.getElementsByTagName("p");
6
7
8 const lien = document.body.getElementsByTagName("a")[0];
9
10 console.log([1,2,3]);
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
  
```

Hello

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Mollitia, magnam quibusdam reprehenderit cum, sint aspernatur dolore eveniet quidem, praesentium enim accusantium odit rem ad earum tempore illo cumeque saepe quos.

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Mollitia, magnam quibusdam **MON LIEN** reprehenderit cum, sint aspernatur dolore eveniet quidem, praesentium enim accusantium odit rem ad earum tempore illo cumeque saepe quos.

Console

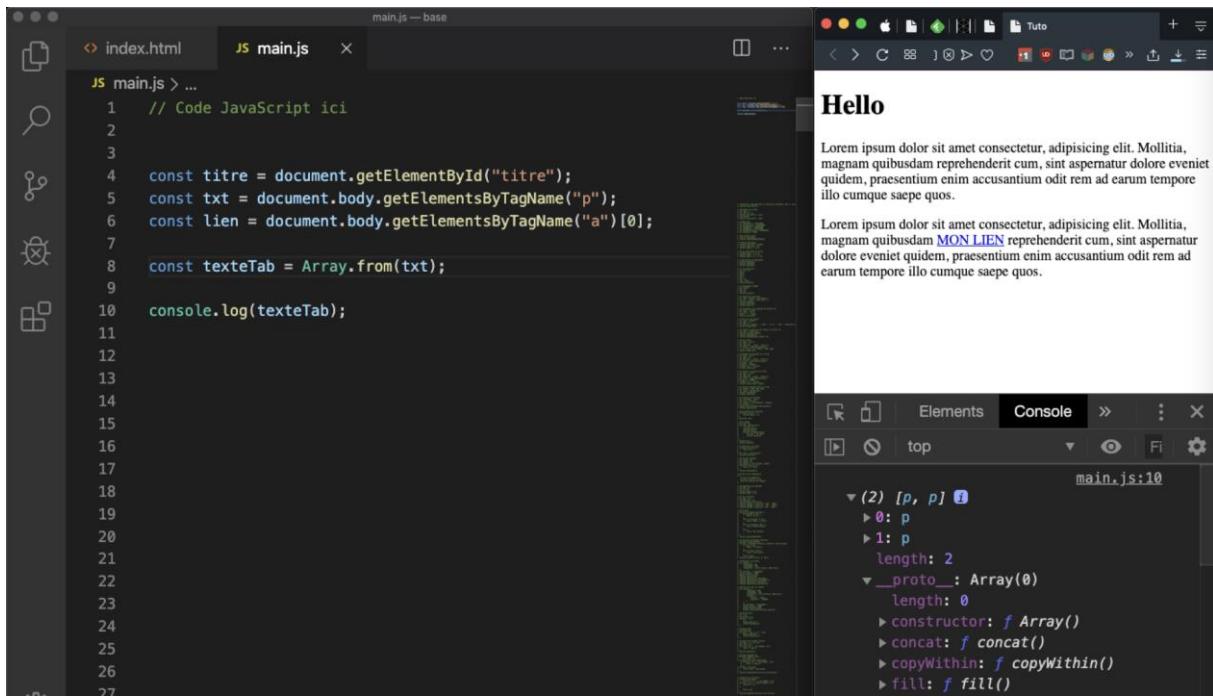
```

▶ flat: f flat()
▶ flatMap: f flatMap()
▶ map: f map()
▶ every: f every()
▶ some: f some()
▶ reduce: f reduce()
▶ reduceRight: f reduceRight()
▶ toLocaleString: f toLocaleString()
▶ toString: f toString()
▶ Symbol(Symbol.iterator): f value...
▶ Symbol(Symbol.unscopables): {co...
  
```

En revanche on a déjà constaté que si j'affiche un tableau en console, on a accès à la méthode map () spécifique aux tableaux.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor window with a file named 'main.js' containing the following code:

```

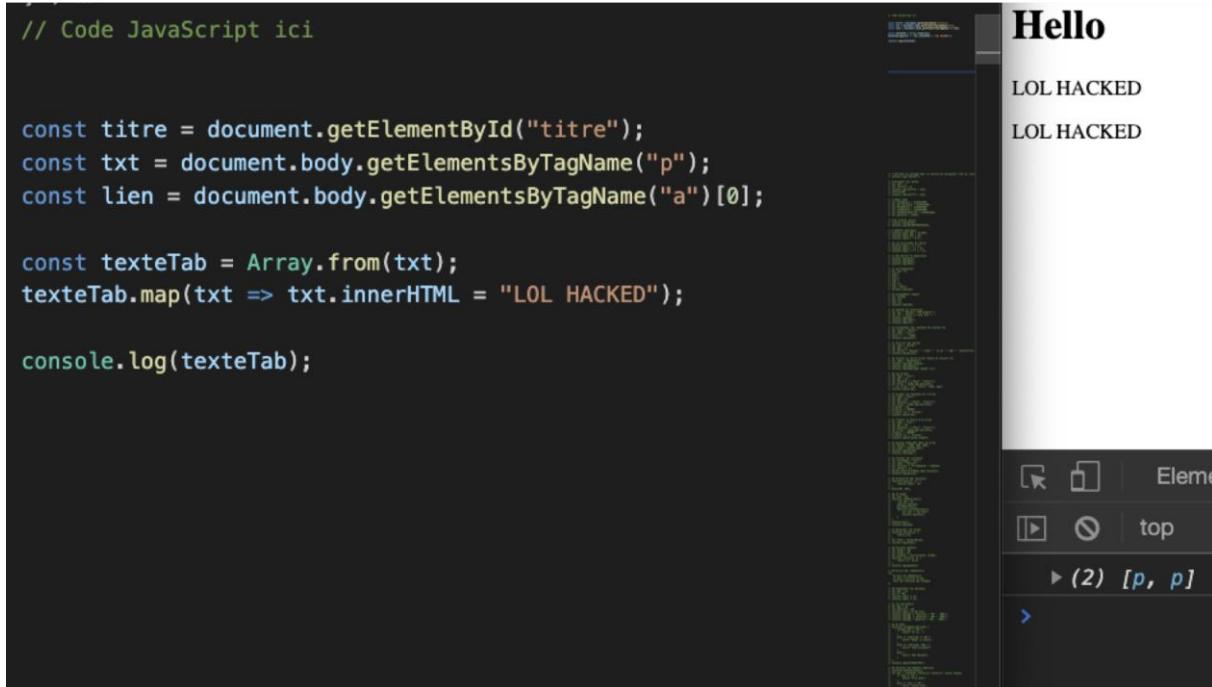
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.body.getElementsByTagName("p");
6 const lien = document.body.getElementsByTagName("a")[0];
7
8 const texteTab = Array.from(txt);
9
10 console.log(texteTab);
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
  
```

To the right, a browser window displays the output of the script. The page content is "Hello" followed by two paragraphs of placeholder text: "Lorem ipsum dolor sit amet consectetur, adipisicing elit. Mollitia, magnam quibusdam reprehenderit cum, sint aspernatur dolore eveniet quidem, praesentium enim accusantium odit rem ad earum tempore illo cumque saepe quo". Below the browser is a developer tools console window showing the variable 'texteTab' as an array of two paragraph elements.

Ici dans une constante texteTab on va stocker sous forme de tableau, tous les paragraphes de la page.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");
const lien = document.body.getElementsByTagName("a")[0];

const texteTab = Array.from(txt);
texteTab.map(txt => txt.innerHTML = "LOL HACKED");

console.log(texteTab);
```

Exemple où l'on transforme tous les paragraphes en tableau pour ensuite pouvoir utiliser la fonction map qui aura pour but de modifier le code HTML de tous les paragraphes en « lol hacked ».

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS

### DOM JS : Obtenir la taille d'un élément Html

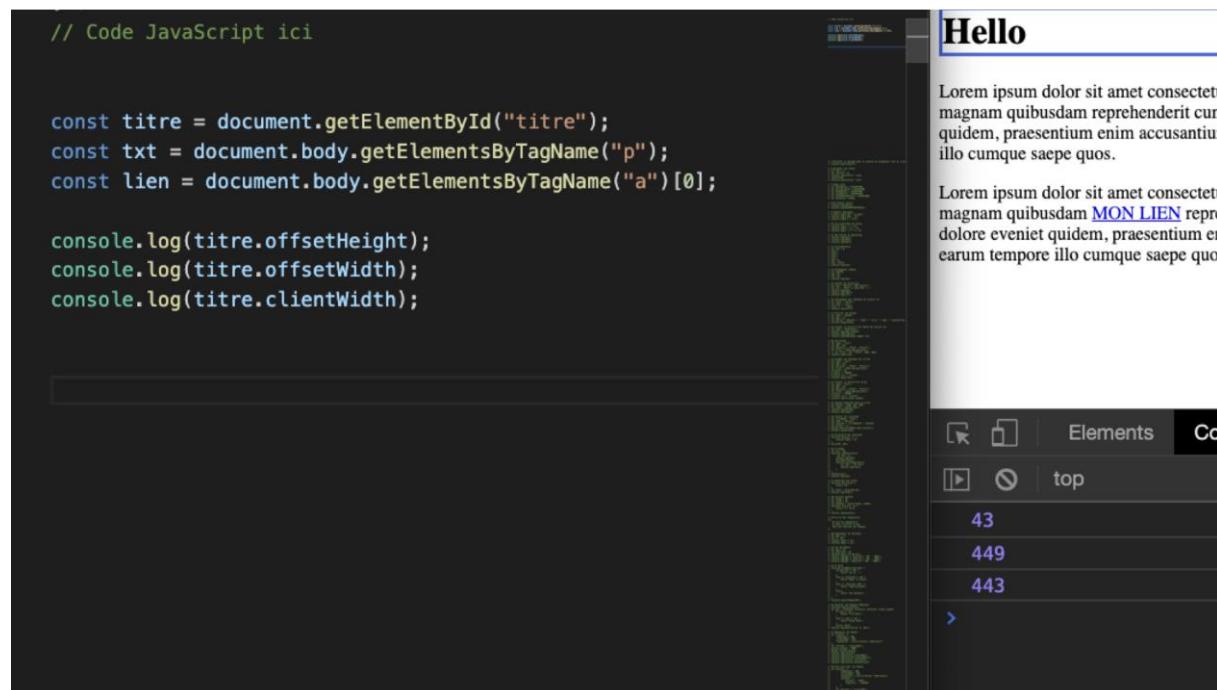
Récupérer la taille en hauteur et en largeur d'un élément peut s'avérer utile pour ensuite le modifier avec précision (le déplacer, le réduire etc...)

- 2 types de calculs pour JS
- Offset : on calcule la taille de l'élément dans sa globalité. (Bordures comprises)

-> offsetHeight et offsetWidth (hauteur et largeur)

- client : on calcule la taille de ce qu'il y a à l'intérieur de l'élément. (ne compte pas les bordures etc...)

-> clientHeight et clientWidth



The screenshot shows a browser's developer tools open to the 'Console' tab. On the left, there is a code editor window containing the following JavaScript code:

```
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");
const lien = document.body.getElementsByTagName("a")[0];

console.log(titre.offsetHeight);
console.log(titre.offsetWidth);
console.log(titre.clientWidth);
```

On the right, the results of the console logs are displayed in a table-like structure:

Result	Value
titre.offsetHeight	43
titre.offsetWidth	449
lien.clientWidth	443

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

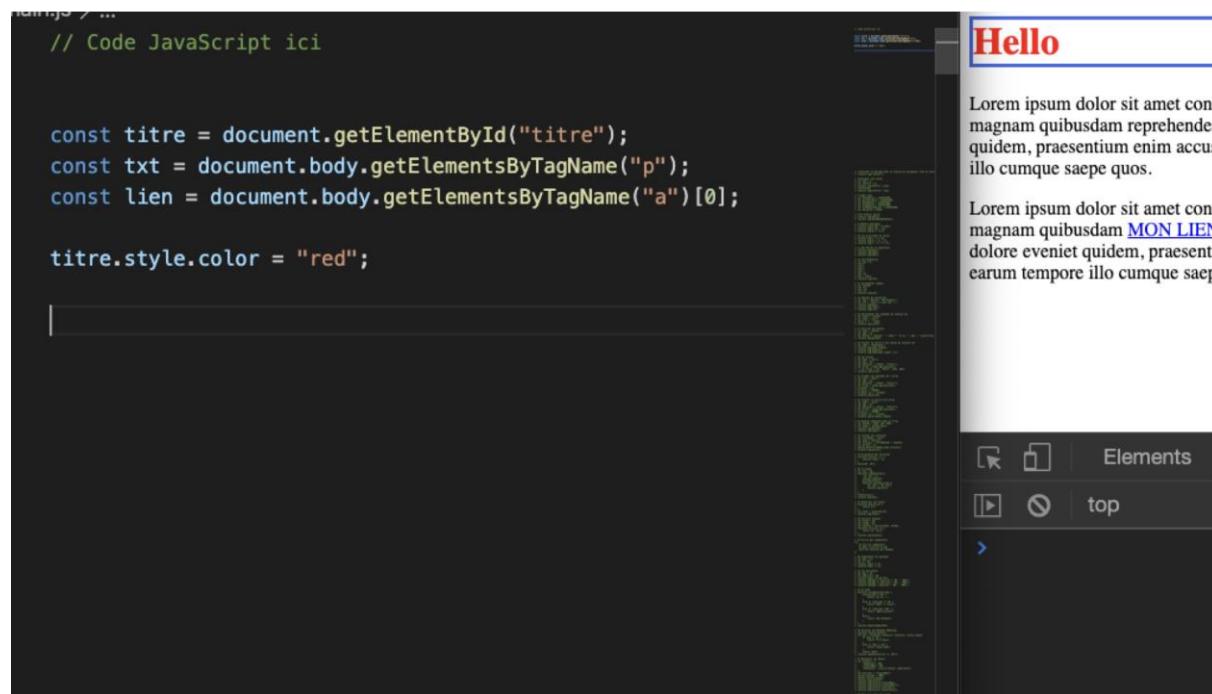
## JS

### DOM JS : modifier le style d'un élément Html

On a déjà vu comment intégrer du code css à l'intérieur de notre code HTML. Avec les balises <style> </style>

Du coup en JS on peut manipuler n'importe quel élément HTML ainsi que sa propriété style au sein de laquelle on retrouvera toutes les propriétés CSS que l'on connaît.

On va pouvoir modifier très simplement du CSS.



```
main.js / ...
// Code JavaScript ici

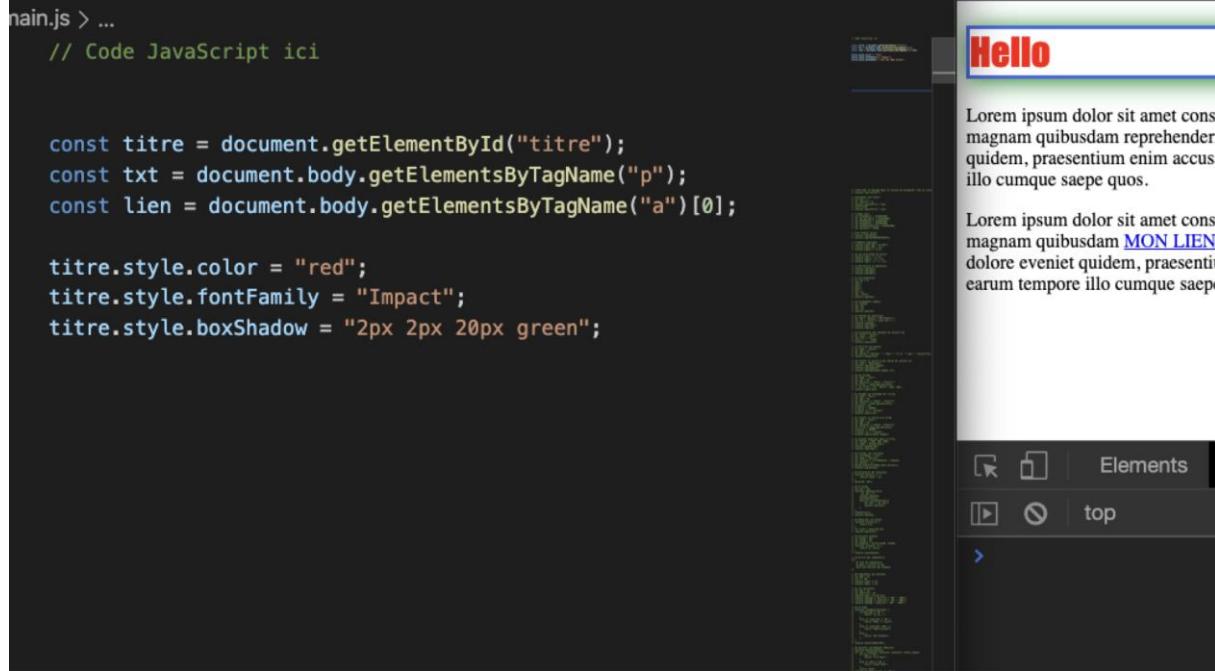
const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");
const lien = document.body.getElementsByTagName("a")[0];

titre.style.color = "red";
```

Ici on accède à titre puis à sa propriété style dans laquelle on retrouve color (la propriété css pour modifier la couleur)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```

main.js > ...
// Code JavaScript ici

const titre = document.getElementById("titre");
const txt = document.body.getElementsByTagName("p");
const lien = document.body.getElementsByTagName("a")[0];

titre.style.color = "red";
titre.style.fontFamily = "Impact";
titre.style.boxShadow = "2px 2px 20px green";

```

Ci-dessus on modifie la couleur du titre, sa police de caractères et sa boxshadow (gestion des ombres)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## DOM JS : Sélectionner des éléments (alternative)

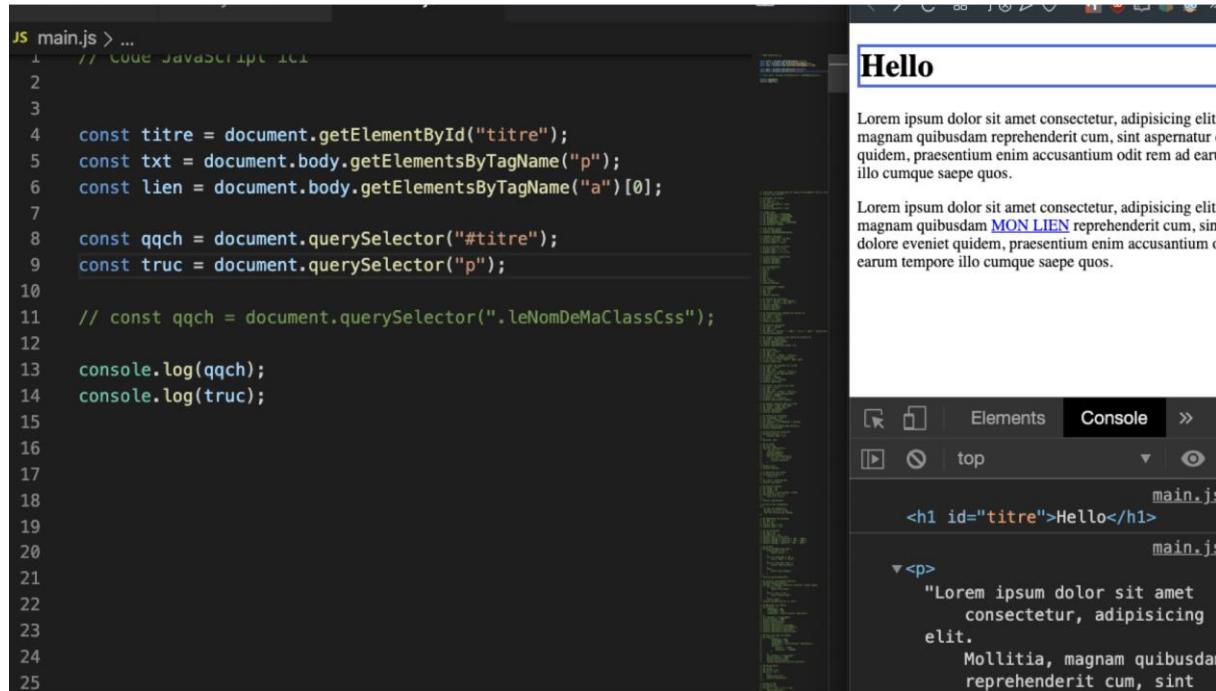
Nous avons déjà vu comment nous pouvons sélectionner des éléments Html via JS, avec les méthodes comme `getElementsByName()`, `getElementById()` etc...

JS propose 2 autres méthodes pour sélectionner un élément, il s'agit des méthodes `querySelector`.

On retrouve 2 types de `querySelector` :

- `querySelector()` va sélectionner l'élément qu'on lui donne à chercher, dès qu'il trouve cet élément il arrête de chercher.
- `querySelectorAll()` va sélectionner tous les éléments qu'on lui donne à chercher et nous les retourne sous une forme d'objet de type `NodeList`.

L'atout des `querySelector` réside dans le fait que nous pouvons utiliser les sélecteurs que nous connaissons déjà en css (`#nomId`), (`.maClasse`)



```

JS main.js > ...
1 // Code JAVASCRIPT ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.body.getElementsByTagName("p");
6 const lien = document.body.getElementsByTagName("a")[0];
7
8 const qqch = document.querySelector("#titre");
9 const truc = document.querySelector("p");
10
11 // const qqch = document.querySelector(".leNomDeMaClassCss");
12
13 console.log(qqch);
14 console.log(truc);
15
16
17
18
19
20
21
22
23
24
25

```

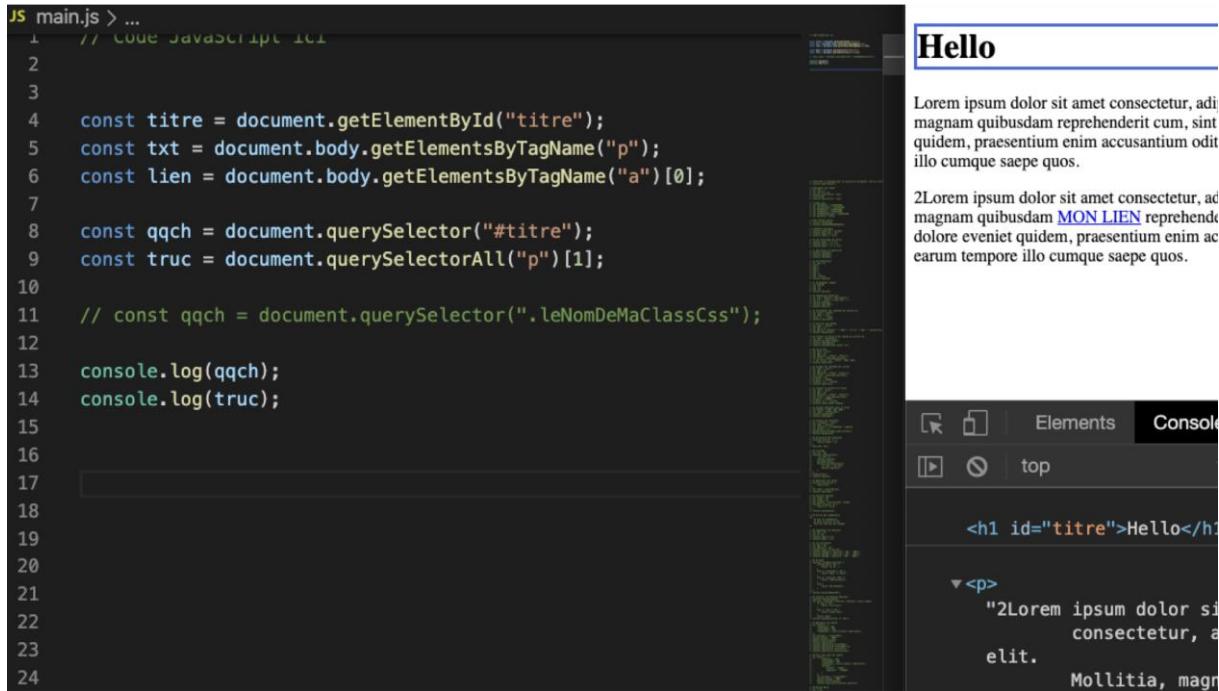
The screenshot shows the browser's developer tools. On the left, the code editor displays the JavaScript code above. On the right, the browser window shows the resulting page with the title "Hello" and some placeholder text. The developer tools' "Console" tab is active, showing the output of the `console.log` statements: the element with id "titre" and the first paragraph element. The "Elements" tab shows the DOM tree with nodes corresponding to the code.

L8 : on sélectionne un élément par son id

L9 : On sélectionne un élément par le nom de sa balise html.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



JS main.js > ...

```

1 // Code JavaScript ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.body.getElementsByTagName("p");
6 const lien = document.body.getElementsByTagName("a")[0];
7
8 const qqch = document.querySelector("#titre");
9 const truc = document.querySelectorAll("p")[1];
10
11 // const qqch = document.querySelector(".leNomDeMaClassCss");
12
13 console.log(qqch);
14 console.log(truc);
15
16
17
18
19
20
21
22
23
24

```

DOM Tree:

- <html>
- <head>
- <body>
  - <h1 id="titre">Hello</h1>
  - <p>2Lorem ipsum dolor si  
consectetur, a  
elit.  
Mollitia, mag

Console:

```

<h1 id="titre">Hello</h1>
<p>2Lorem ipsum dolor si
consectetur, a
elit.
Mollitia, mag

```

L9 : même avec un querySelectorAll on peut quand même sélectionner 1 élément en particulier en utilisant le système d'indice, ici on sélectionne le second paragraphe de la page.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

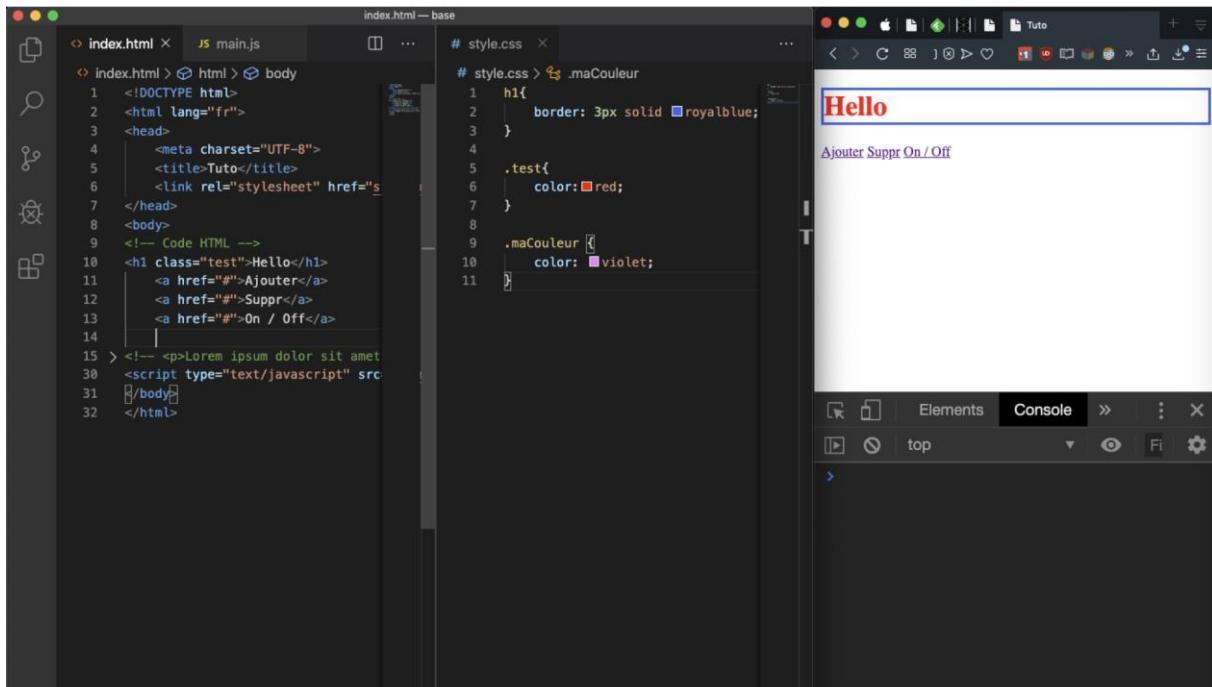
## DOM JS : Ajouter ou supprimer des classes Css

classList nous renvoi la liste de toutes les classes CSS qui sont appliquées sur un élément HTML.

Avec classList.add("leNomdeMaClasse") on ajoute une nouvelle classe

Avec classList.remove("leNomdeMaClasse") on supprime une classe

Avec classList.toggle("leNomdeMaClasse") on active ou non la classe.



The screenshot shows a development environment with three tabs: index.html, main.js, and style.css. The index.html file contains an h1 element with a class of "test". The main.js file contains a script that adds and removes the "maCouleur" class from the h1 element. The style.css file defines styles for ".maCouleur" and ".test". The browser window shows the word "Hello" in red. The developer tools' DOM tab shows the h1 element with a border, and the console tab shows the command "Ajouter Suppr On / Off".

```

index.html
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Tuto</title>
6   <link rel="stylesheet" href="style.css" type="text/css" />
7 </head>
8 <body>
9 <!-- Code HTML -->
10 <h1 class="test">Hello</h1>
11 <a href="#">Ajouter</a>
12 <a href="#">Suppr</a>
13 <a href="#">On / Off</a>
14 <br/>
15 <!-- <p>Lorem ipsum dolor sit amet</p>
30 <script type="text/javascript" src="main.js" />
31 </body>
32 </html>
```

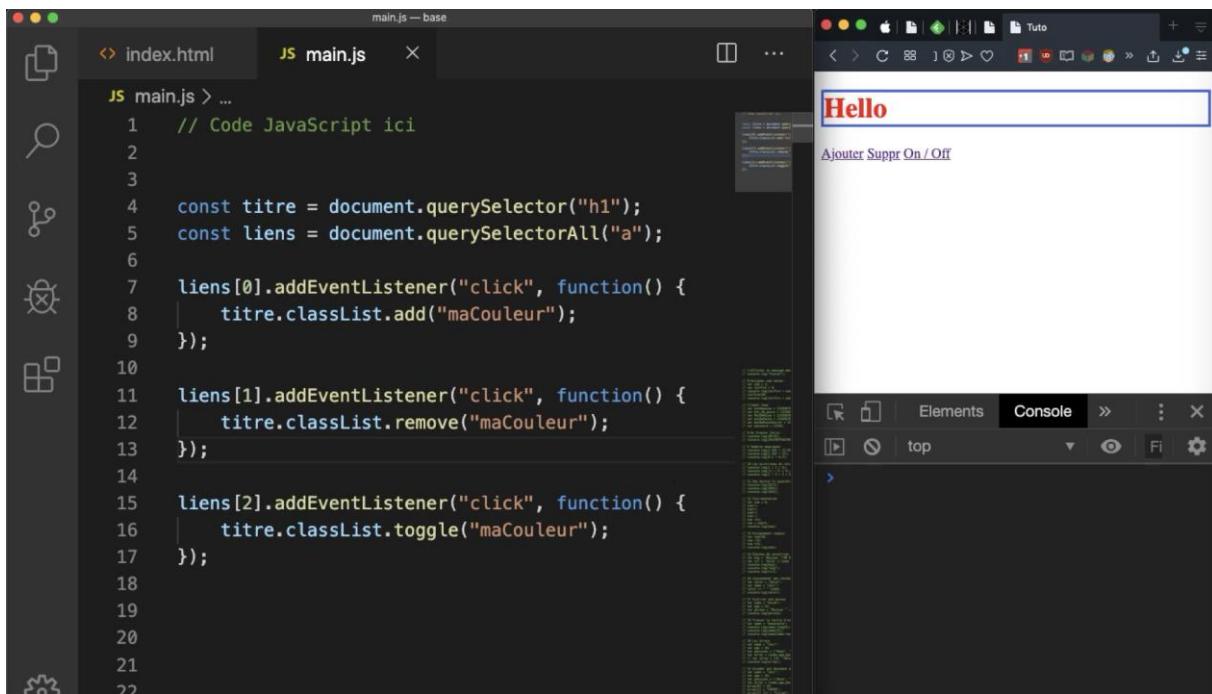
```

style.css
# style.css > .maCouleur
1 h1{
2   border: 3px solid royalblue;
3 }
4
5 .test{
6   color: red;
7 }
8
9 .maCouleur {
10   color: violet;
11 }
```

Ici on a préparé du code css avec une classe maCouleur qui se chargera de modifier la couleur d'un texte en violet.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following JavaScript code:

```

main.js — base
JS main.js × ...
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const titre = document.querySelector("h1");
5 const liens = document.querySelectorAll("a");
6
7 liens[0].addEventListener("click", function() {
8   titre.classList.add("maCouleur");
9 });
10
11 liens[1].addEventListener("click", function() {
12   titre.classList.remove("maCouleur");
13 });
14
15 liens[2].addEventListener("click", function() {
16   titre.classList.toggle("maCouleur");
17 });
18
19
20
21
22

```

The browser window shows a red "Hello" button. The developer tools console is open at the bottom, showing no output.

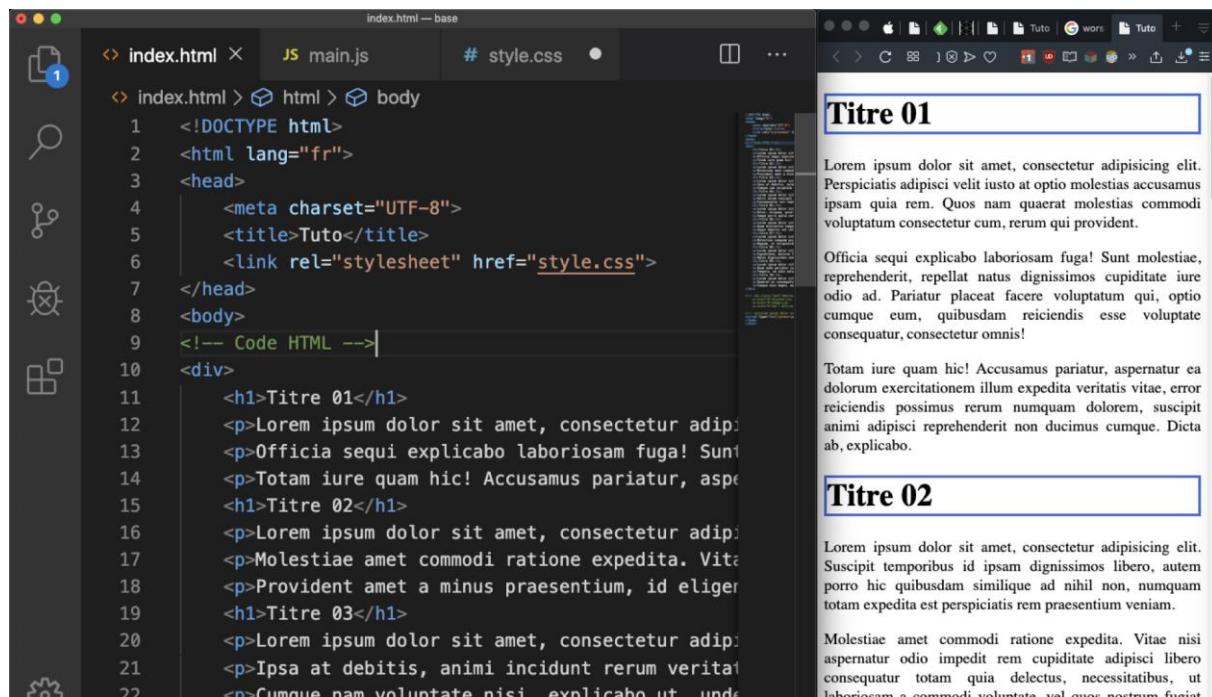
Ici sur les liens on va placer un addEventListener permettant de réagir au click sur nos liens. Lorsque le click sur un lien est détecté, on ajoute la classe maCouleur (L8), sur le second lien on supprimera la classe maCouleur, sur le 3<sup>ème</sup> lien, on active ou désactive la classe maCouleur.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Exercice : Recréer un système d'ancre Html mais en JS

- Coder une fonction qui permet d'adapter le scroll de la page pour aller de titres en titres.
- offsetTop qui est une propriété d'un élément qui nous dit à quelle distance il est dans notre page par rapport au haut de la page. (en px)
- L'objet window de JS qui permet d'accéder à tous les éléments de notre page (Tout le DOM)
- La méthode ScrollTo(x, y) on lui passe des coordonnées X Y (On peut lui passer des options (behaviour smooth par exemple))



The screenshot shows a code editor with three tabs: index.html, main.js, and style.css. The index.html tab contains the following code:

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <!-- Code HTML -->
    <div>
        <h1>Titre 01</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipisci...</p>
        <p>Officia sequi explicabo laboriosam fuga! Sunt...</p>
        <p>Totam iure quam hic! Accusamus pariatur, asper...</p>
        <h1>Titre 02</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipisci...</p>
        <p>Molestiae amet commodi ratione expedita. Vit...</p>
        <p>Provident amet a minus praesentium, id eliger...</p>
        <h1>Titre 03</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipisci...</p>
        <p>Ipsa at debitis, animi incidentum rerum verit...</p>
        <p>Cumque nam voluntate nisi explicabo ut unde...</p>
    </div>

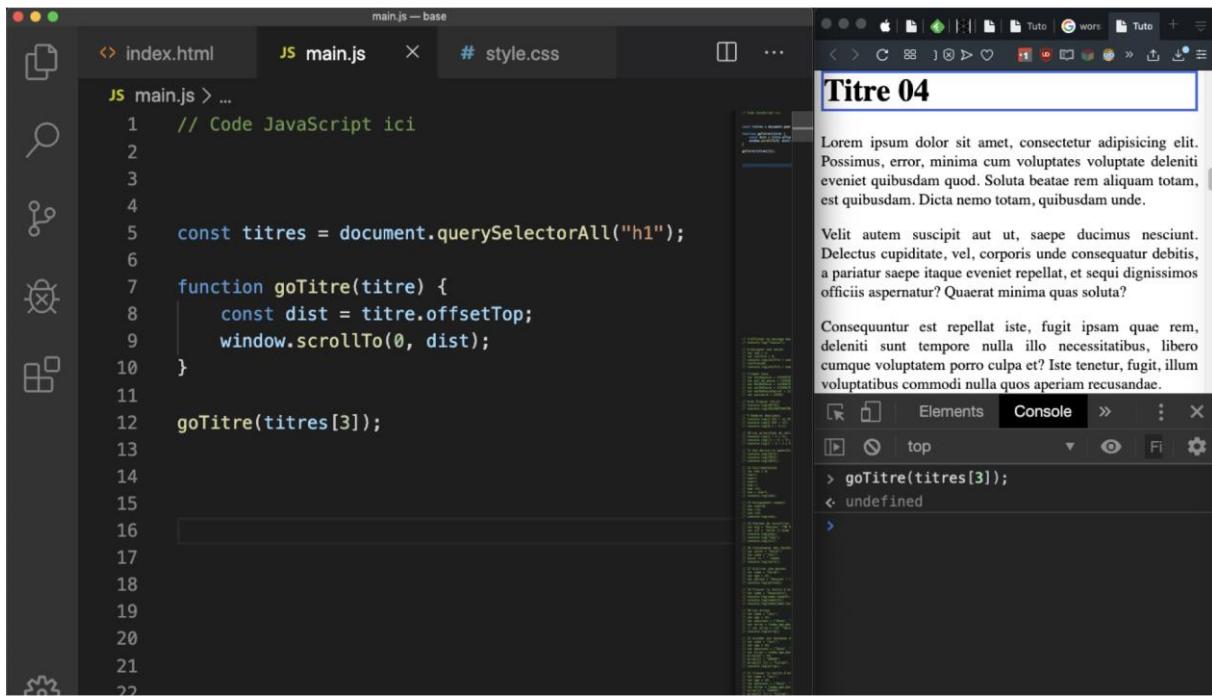
```

The browser preview shows two sections. The first section has a title "Titre 01" and contains three paragraphs of placeholder text. The second section has a title "Titre 02" and also contains three paragraphs of placeholder text.

La page Html avec 1 titre et 3 paragraphes, on répète ce schéma-là plusieurs fois.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The image shows a code editor window with a file named 'main.js' open. The code contains JavaScript code to scroll to a specific element. Below the code editor is a screenshot of a web browser displaying a page with a large heading 'Titre 04'. The browser's developer tools console is open, showing the command 'goTitre(titres[3]);' being run and its result as undefined.

```

main.js — base
JS main.js X # style.css ...
JS main.js > ...
1 // Code JavaScript ici
2
3
4
5 const titres = document.querySelectorAll("h1");
6
7 function goTitre(titre) {
8     const dist = titre.offsetTop;
9     window.scrollTo(0, dist);
10 }
11
12 goTitre(titres[3]);
13
14
15
16
17
18
19
20
21
22

```

On va devoir sélectionner tous nos titres, on va les stocker dans une constante titres avec un querySelectorAll (il stocke tout dans une sorte d'objet (nodeList))

On fait une fonction pour adapter le scroll en fonction de titres[qqch]

On va utiliser offsetTop pour connaître la distance en PX de l'élément par rapport au haut de la page.

On va stocker cette valeur-là dans une variable distance.

Et on lui dit de scroller jusqu'à distance pour ça on utilise window et scrollTo (0, distance)

On scroll de 0 en horizontal et de distance en vertical.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

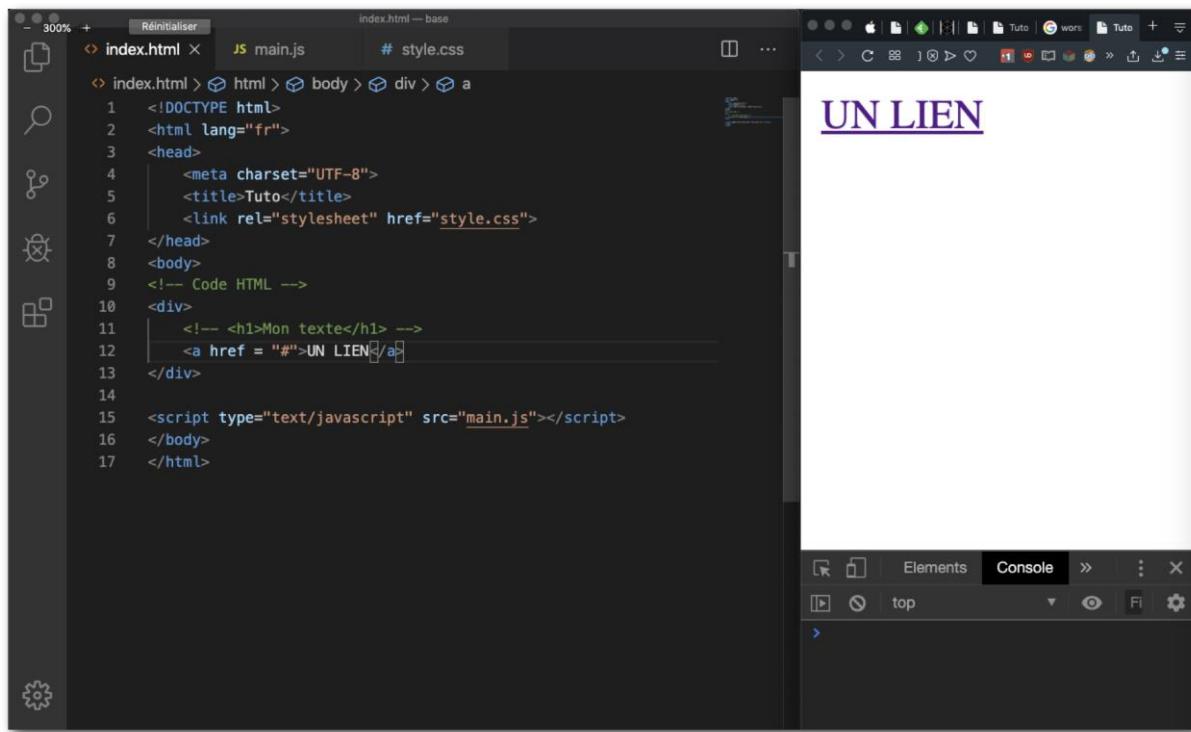
## DOM JS : Gestion des évènements

### **addEventListener()**

Cette fonction va nous permettre en JS de réagir à plusieurs types d'évènements, par exemple lorsque qu'un utilisateur clique sur un bouton.

[https://www.w3schools.com/jsref/dom\\_obj\\_event.asp](https://www.w3schools.com/jsref/dom_obj_event.asp)

(Liste de tous les évènements auxquels nous pouvons réagir en JS).



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays an HTML file named index.html with the following content:

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <!-- Code HTML -->
    <div>
        <!-- <h1>Mon texte</h1> -->
        <a href = "#">UN LIEN</a>
    </div>
<script type="text/javascript" src="main.js"></script>
</body>
</html>

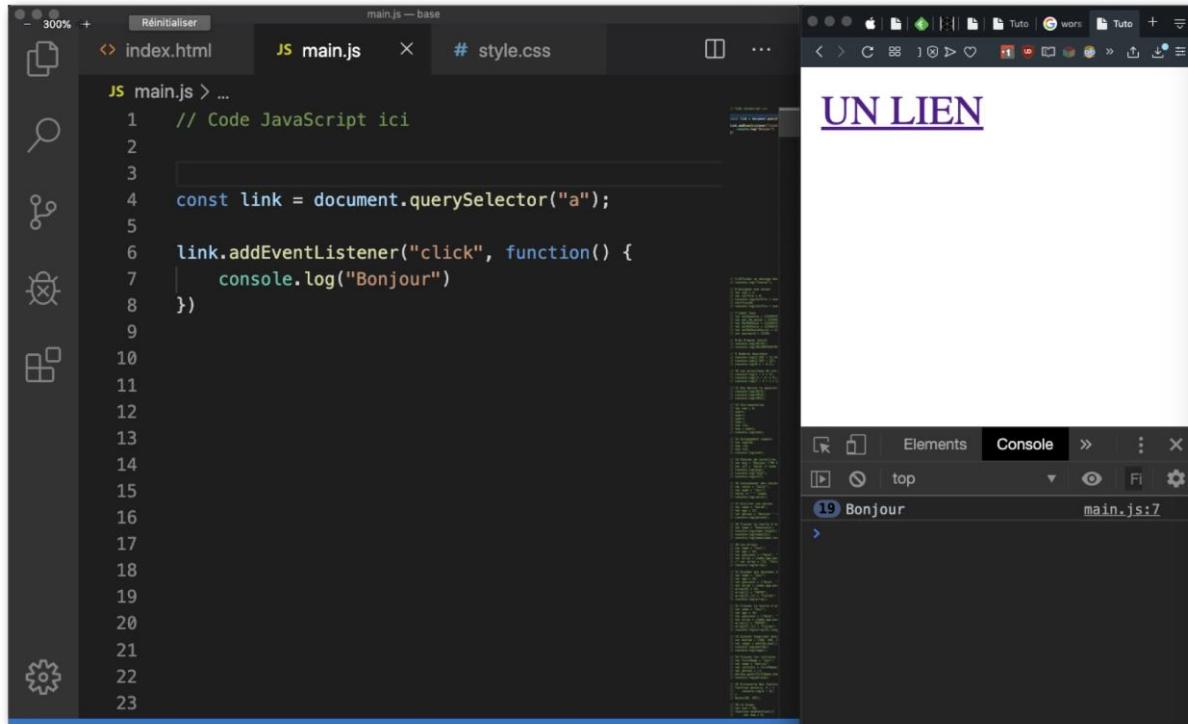
```

The browser window on the right shows the text "UN LIEN" underlined, indicating it is a hyperlink.

Une page Html comprenant une div qui contient un lien

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

JS main.js > ...
1 // Code JavaScript ici
2
3
4 const link = document.querySelector("a");
5
6 link.addEventListener("click", function() {
7     console.log("Bonjour")
8 })
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

UN LIEN

On va récupérer notre lien dans 1 const avec querySelector a

On ajoute un eventListener (il va écouter la page et dès que qqch se produit ici un clic sur l'élément qu'on a choisi : ici le lien "a")

Alors il va déclencher une fonction addEventListener() qui prend 2 paramètres :

Quel event je veux écouter

Ce qu'il doit déclencher

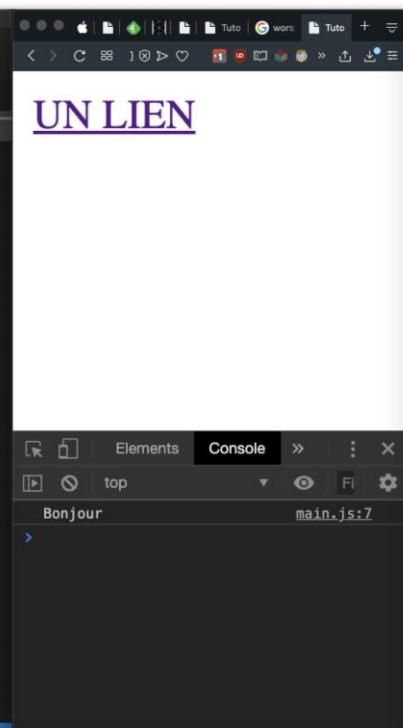
Là c'est pas super utile mais imaginez les possibilités au clic on ouvre une popup, change un style, lancer une animation, faire un compteur, etc...

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## removeEventListener()

Si l'on peut Ajouter des Eventlistener pour écouter un évènement particulier dans notre page, on peut tout aussi bien supprimer l'écoute de cet évènement.



The screenshot shows a browser window with developer tools open. On the left, the code editor displays a JavaScript file named 'main.js' with the following content:

```

1 // Code JavaScript ici
2
3
4 const link = document.querySelector("a");
5
6 function monClic() {
7     console.log("Bonjour");
8     link.removeEventListener("click", monClic);
9 }
10
11 link.addEventListener("click", monClic);
12
13
14
15
16
17
18
19
20
21
22
23

```

On the right, the browser window shows a link with the text "UN LIEN". Below it, the developer tools' console tab is active, showing the output of the console.log statement: "Bonjour" at main.js:7.

Dans ma fonction monClic je rajoute un removeEvenListener() qui prend 2 paramètre :

Quel Event on veut supprimer ici c'est le click mais ça peut être une touche de clavier ou autre chose

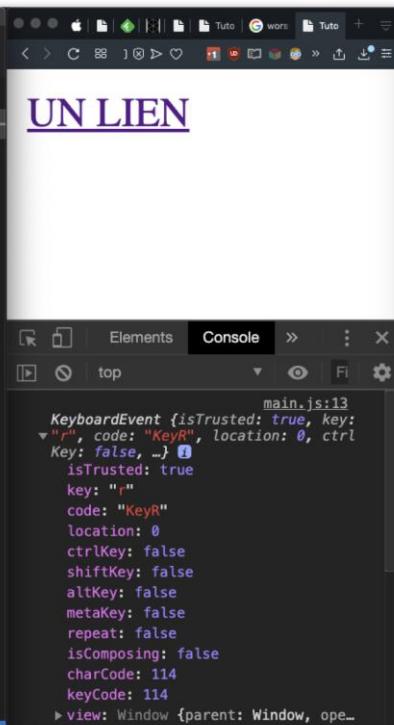
Quelle action on remove ici la fonction monClic, de fait quand je clique sur le lien on affiche un message en console, puis on cesse d'écouter le click.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## L'évènement Keypress

L'évènement Keypress nous permet de détecter ce que l'utilisateur va taper au clavier



```

main.js — base
index.html JS main.js # style.css ...
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const link = document.querySelector("a");
5
6 function monClic() {
7     console.log("Bonjour");
8     link.removeEventListener("click", monClic);
9 }
10 link.addEventListener("click", monClic);
11
12 addEventListener("keypress", function(event) {
13     console.log(event);
14 })
15
16
17
18
19
20
21
22
23

```

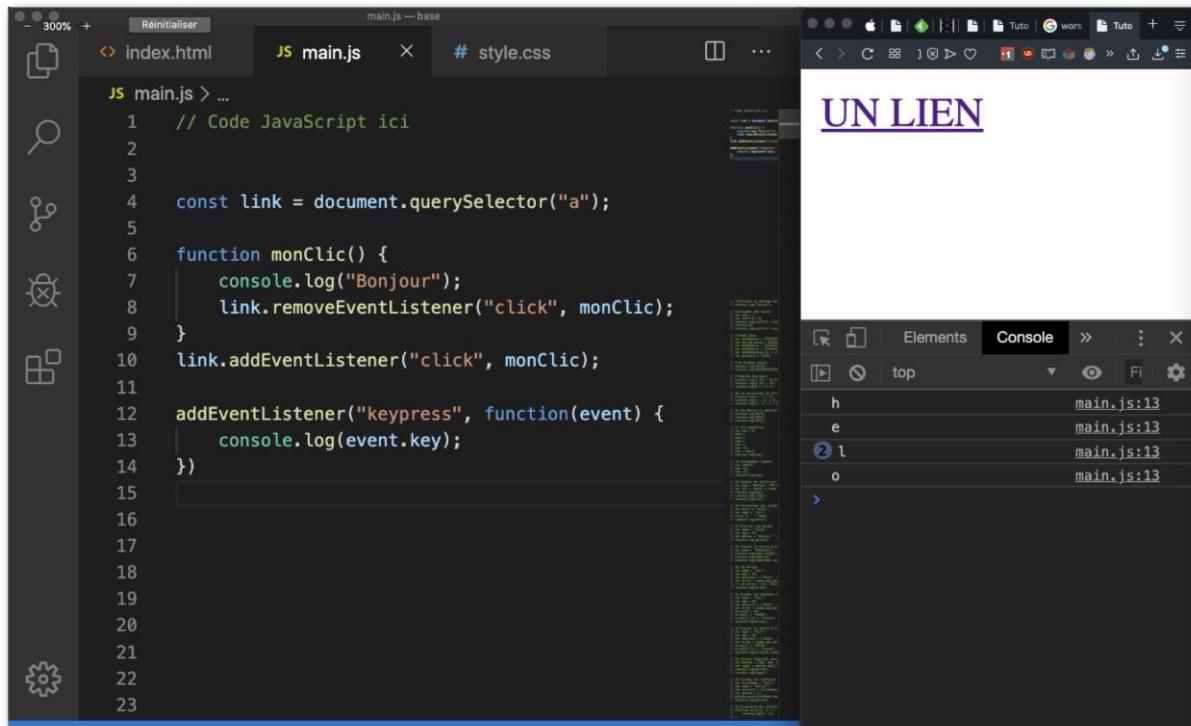
L12 un addEventListener surveille keypress et déclenche une fonction, cette fonction elle a une variable event en paramètre (c'est dans cette variable qu'on va stocker l'évènement auquel on réagit)

Ensuite dans la fonction on affiche en console la variable event.

Quand on utilise le clavier sur notre page, la console affiche un objet de type KeyBoardEvent qui va contenir toutes les propriétés de cet évènement.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a browser window with developer tools open. On the left, the code editor displays a JavaScript file named 'main.js' with the following content:

```

JS main.js > ...
1 // Code JavaScript ici
2
3
4 const link = document.querySelector("a");
5
6 function monClic() {
7     console.log("Bonjour");
8     link.removeEventListener("click", monClic);
9 }
10 link.addEventListener("click", monClic);
11
12 addEventListener("keypress", function(event) {
13     console.log(event.key);
14 })
15
16
17
18
19
20
21
22
23

```

On the right, the browser window shows the text "UN LIEN". Below the browser window, the developer tools' console tab is active, displaying the following log entries:

- h main.js:13
- e main.js:13
- l main.js:13
- o main.js:13

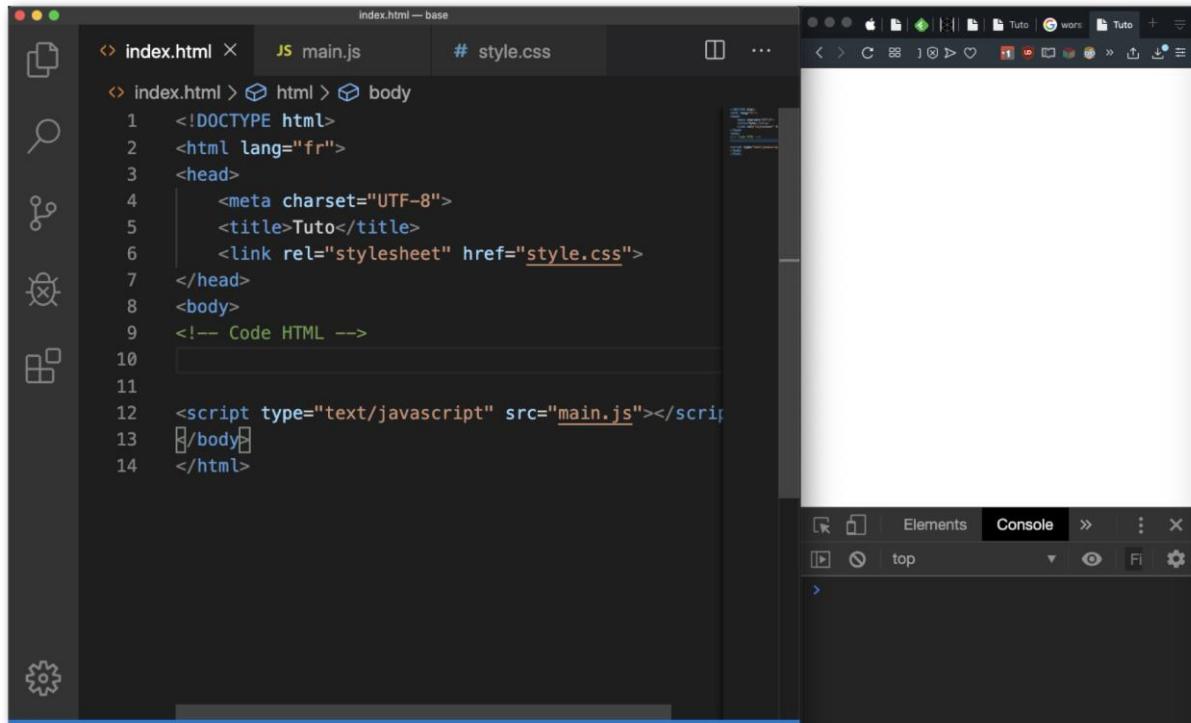
On console log event.key qui est la propriété dans laquelle on retrouve les caractères que tape l'utilisateur.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Exercice : Afficher une image à l'endroit du clic

Nous allons coder un script pour afficher une image à l'endroit où l'on clique.



```

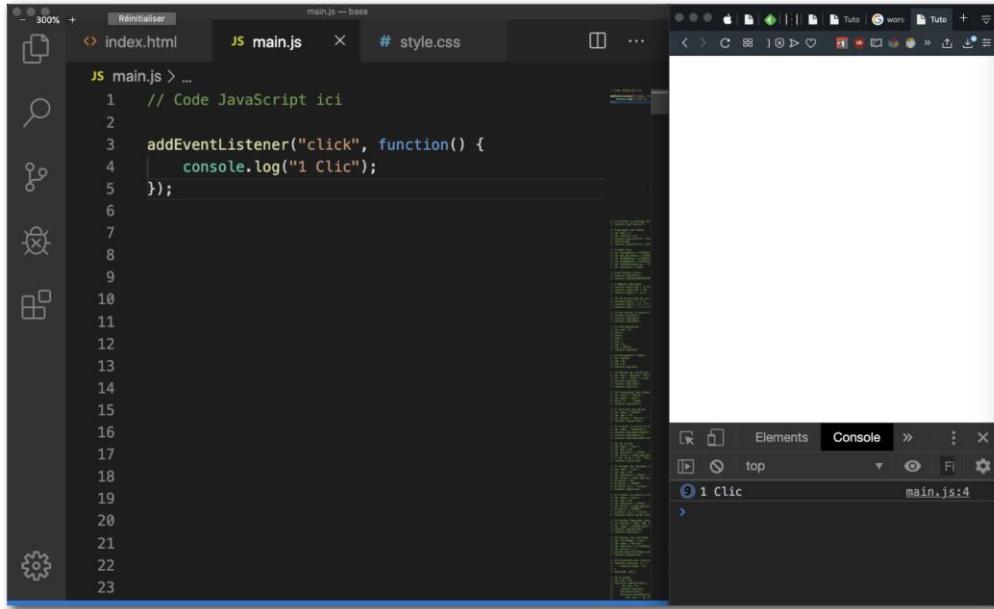
index.html — base
index.html X JS main.js # style.css
index.html > html > body
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <title>Tuto</title>
6      <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9  <!-- Code HTML -->
10
11
12  <script type="text/javascript" src="main.js"></script>
13  </body>
14  </html>

```

Le html très simple (rien dans le <body>) on s'en sert juste pour link le JS, on va tout faire dans le JS

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

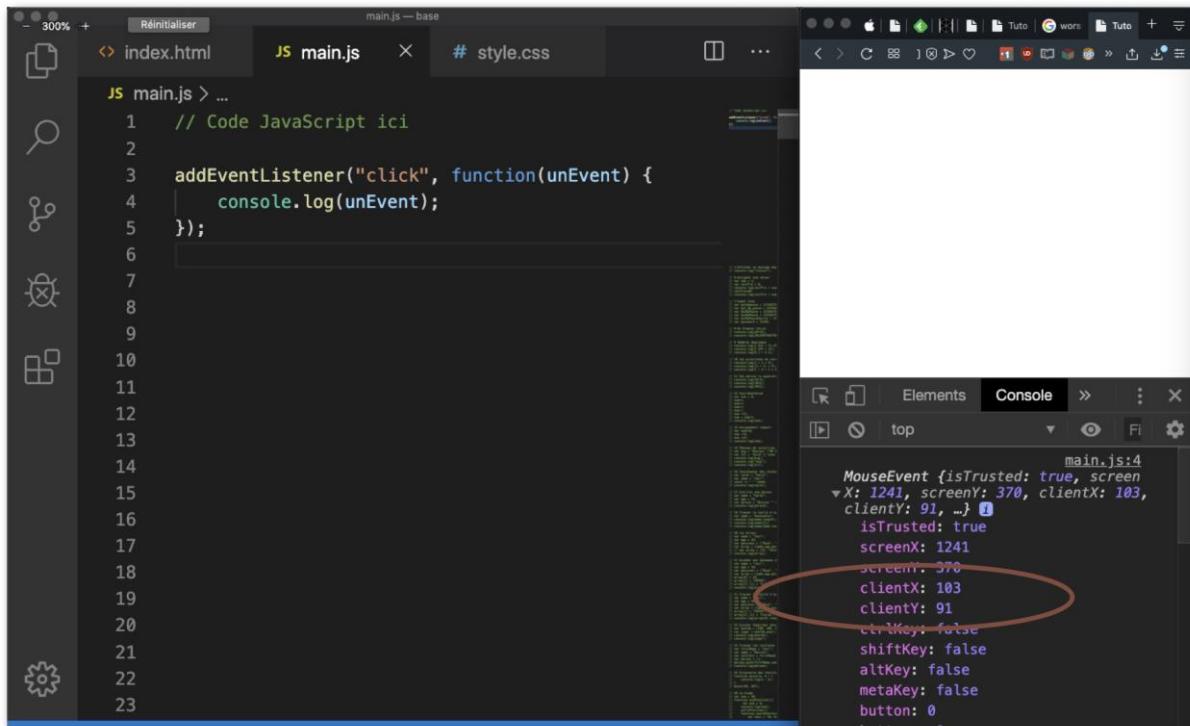
main.js > ...
1 // Code JavaScript ici
2
3 addEventListener("click", function() {
4     console.log("1 Clic");
5 });
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

Premier réflexe : un test pour vérifier qu'on arrive bien à détecter le clic.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

main.js — base
JS main.js X # style.css ...
JS main.js > ...
1 // Code JavaScript ici
2
3 addEventListener("click", function(unEvent) {
4     console.log(unEvent);
5 });
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

main.js:4

```

MouseEvent {isTrusted: true, screenX: 370, clientX: 103,
clientY: 91, ...}
  isTrusted: true
  screenX: 1241
  screenY: 370
  clientX: 103
  clientY: 91
  ctrlKey: false
  shiftKey: false
  altKey: false
  metaKey: false
  button: 0

```

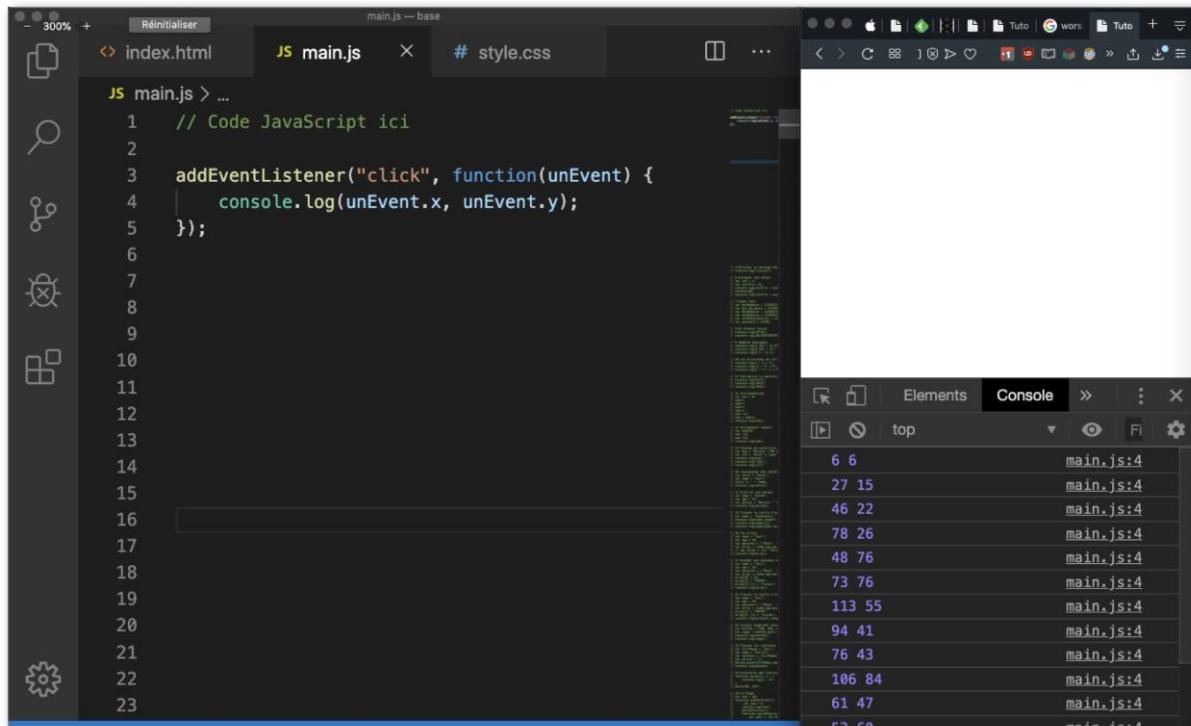
Ensuite on affiche en console l'évènement (unEvent), pour voir si des propriétés de cet objet peuvent nous être utiles.

Pour retrouver les coordonnées exactes de notre souris au moment de l'évènement donc du clic.

clientX et client Y (ou juste X et Y)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The image shows a code editor window and a browser window side-by-side. The code editor displays the following JavaScript code:

```

JS main.js > ...
1 // Code JavaScript ici
2
3 addEventListener("click", function(unEvent) {
4     console.log(unEvent.x, unEvent.y);
5 });
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

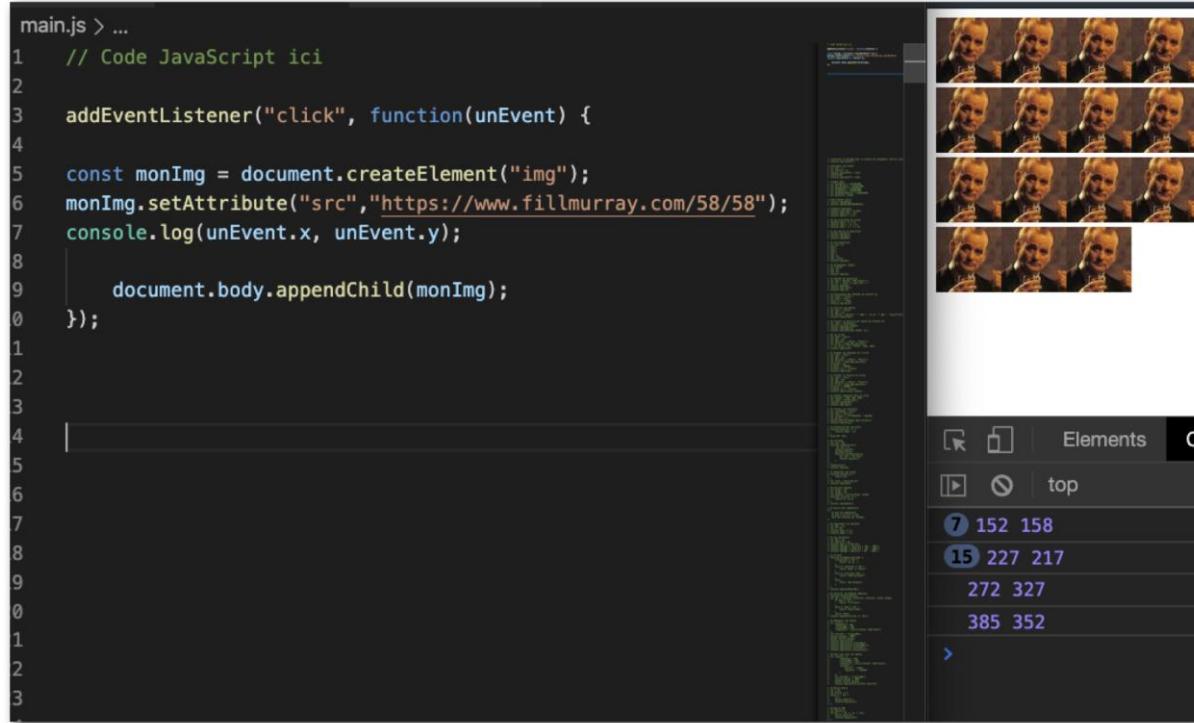
The browser window shows the output of the console.log statements in the developer tools' Console tab:

- top
- 6 6 main.js:4
- 27 15 main.js:4
- 46 22 main.js:4
- 78 26 main.js:4
- 48 76 main.js:4
- 73 76 main.js:4
- 113 55 main.js:4
- 94 41 main.js:4
- 76 43 main.js:4
- 106 84 main.js:4
- 61 47 main.js:4
- 52 68 main.js:4

Nous parvenons à afficher en console les coordonnées X et Y de l'endroit où l'on clique

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```
main.js > ...
1 // Code JavaScript ici
2
3 addEventListener("click", function(unEvent) {
4
5 const monImg = document.createElement("img");
6 monImg.setAttribute("src", "https://www.fillmurray.com/58/58");
7 console.log(unEvent.x, unEvent.y);
8
9 document.body.appendChild(monImg);
0});
```

Nous allons nous servir d'une API pour créer les images, ici j'utilise fillmurray.com (ne fonctionne plus aujourd'hui)

d'autres Api dans ce style (<https://lorem ipsum.io/fr/21-of-the-best-placeholder-image-generators/>)

On va créer un élément image (L5), puis modifier son attribut src (L6) avec l'adresse de l'API

On affiche notre image appenChild(monImg) pour l'ajouter à la fin de <body>

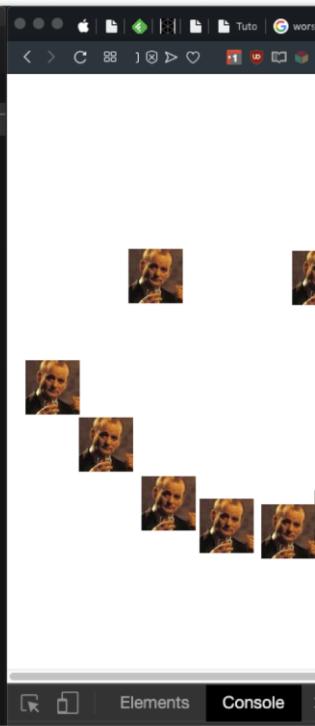
Ok ça affiche les images quand on clique mais pas comme on voulait.

L'image s'ajoute à la fin du document

On doit dire en CSS l'endroit où il doit se positionner.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

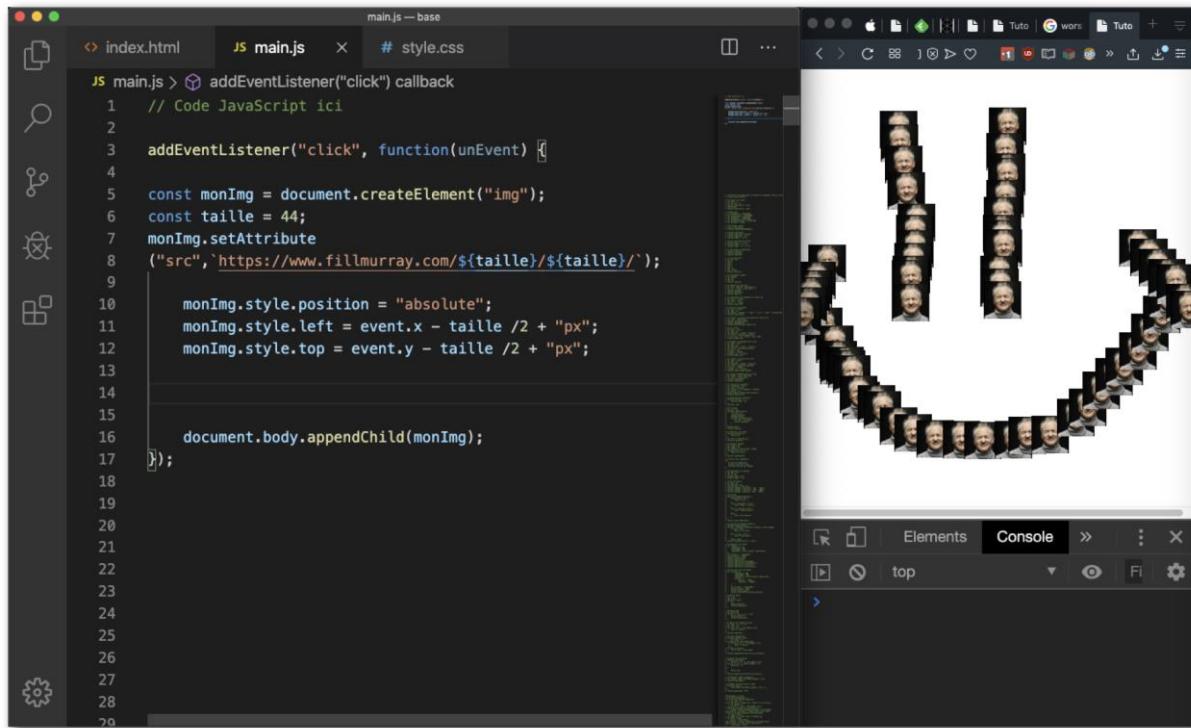
main.js — base
index.html JS main.js X # style.css
JS main.js > addEventListener("click") callback
1 // Code JavaScript ici
2
3 addEventListener("click", function(unEvent) {
4
5 const monImg = document.createElement("img");
6 monImg.setAttribute("src", "https://www.fillmurray.com/58/58");
7 // console.log(unEvent.x, unEvent.y);
8
9 monImg.style.position = "absolute";
10 monImg.style.left = event.x + "px";
11 monImg.style.top = event.y + "px";
12
13
14 document.body.appendChild(monImg);
15
16 });
17
18
19
20
21
22
23
24
25
26

```

Petite modification, L9 on modifie la position de l'image en absolute (pour qu'elle puisse se placer où l'on veut dans la page), puis L10 et 11 on modifie la position de l'image avec les propriétés left et top de css.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



The screenshot shows a browser developer tools interface. On the left, the 'Console' tab is active, displaying the following JavaScript code:

```

main.js — base
JS main.js × # style.css
JS main.js > addEventListener("click") callback
1 // Code JavaScript ici
2
3 addEventListener("click", function(unEvent) {
4
5 const monImg = document.createElement("img");
6 const taille = 44;
7 monImg.setAttribute
8 ("src", "https://www.fillmurray.com/${taille}/${taille}/");
9
10 monImg.style.position = "absolute";
11 monImg.style.left = event.x - taille / 2 + "px";
12 monImg.style.top = event.y - taille / 2 + "px";
13
14
15
16 document.body.appendChild(monImg);
17 });
18
19
20
21
22
23
24
25
26
27
28
29

```

On the right, the 'Elements' tab is active, showing a visual representation of the rendered HTML. It consists of a grid of small, identical portrait images of a man, arranged in a curved, stepped pattern that radiates from a central point.

Dernier correctif pour que l'image s'insère pile à l'endroit où l'on clique de manière centré, nous prenons en compte la taille de l'image (L6), L11 et 12 on soustrait les coordonnées de l'endroit cliqué avec la taille de l'image.

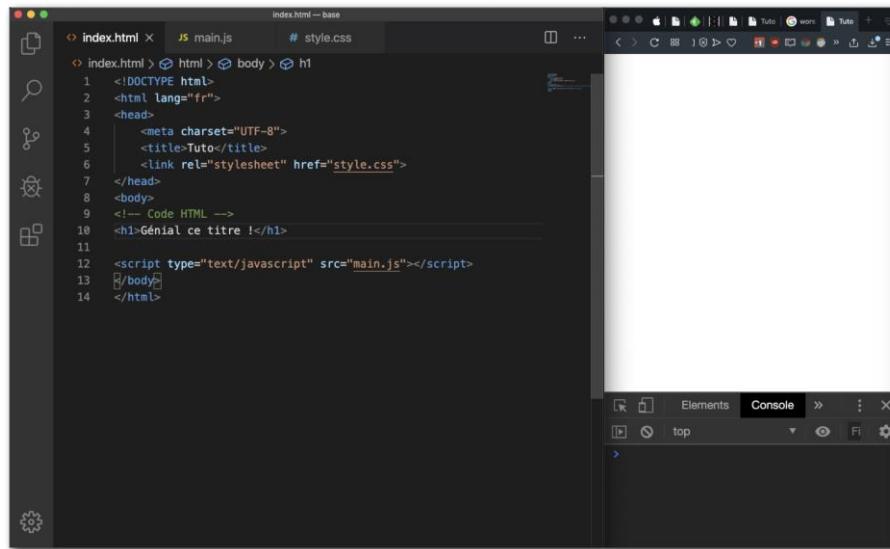
On clique et l'image s'affiche pile au centre de là où l'on a cliqué.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Mouseout

Cet évènement permet de réagir quand la souris de l'utilisateur quitte la page.

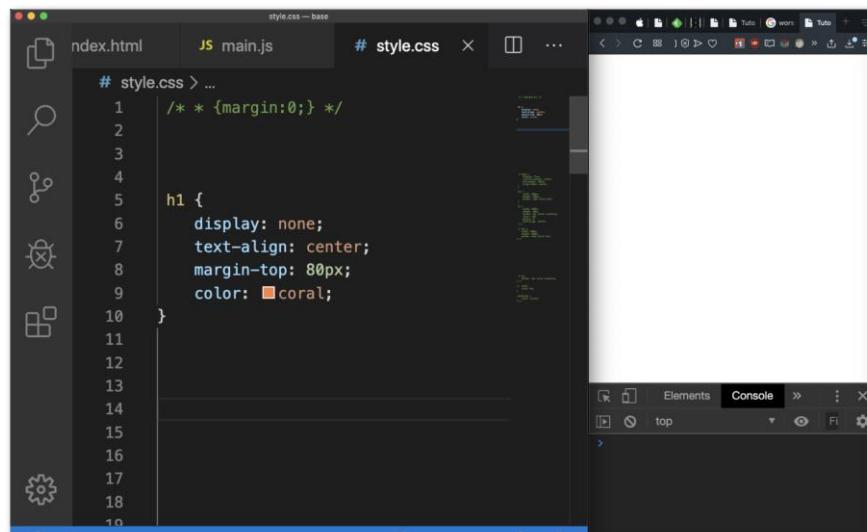


```

index.html
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <!-- Code HTML -->
    <h1>Génial ce titre !</h1>
<script type="text/javascript" src="main.js"></script>
</body>
</html>

```

Page Html : Simplement un titre dans le <body>



```

style.css - base
# style.css > ...
h1 {
    display: none;
    text-align: center;
    margin-top: 80px;
    color: coral;
}

```

Page CSS, le display du titre est à none (le titre n'est pas affiché)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

JS

```
const unTruc = document.querySelector("h1");

unTruc.addEventListener("mouseout", function() {
    unTruc.style.display = "block";
})
```

On va récupérer notre h1 via query selector

On ajoute un eventListener sur l'entièreté de notre page.

On lui dit de surveiller le mouse out (quand la souris est dehors)

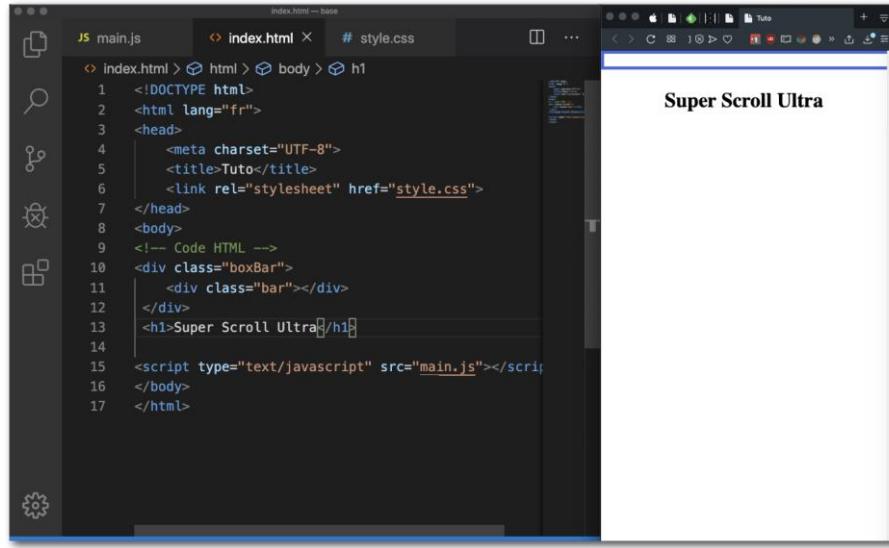
On lui passe une fonction qui va modifier le css et changer le type de display à block (donc visible) pour prendre toute la largeur dispo.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Scroll

Cet évènement surveille le scroll dans la page.



```
index.html --- Branche
JS main.js    index.html    # style.css
```

```

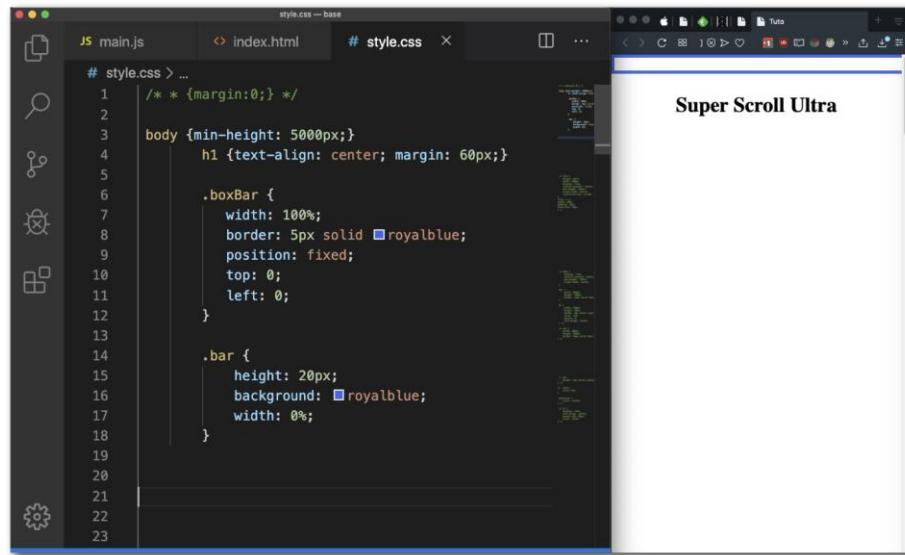
<Index.html > <html> <body> <h1>
1   <!DOCTYPE html>
2   <html lang="fr">
3     <head>
4       <meta charset="UTF-8">
5       <title>Tuto</title>
6       <link rel="stylesheet" href="style.css">
7     </head>
8     <body>
9       <!-- Code HTML -->
10      <div class="boxBar">
11        <div class="bar"></div>
12      </div>
13      <h1>Super Scroll Ultra</h1>
14
15      <script type="text/javascript" src="main.js"></script>
16    </body>
17  </html>
```

Page Html : 1 div boxBar qui représente notre cadre

À l'intérieur une Div bar qui est vide (le blanc à l'intérieur 20 px)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

JS main.js      index.html      # style.css
# style.css > ...
1  /* * {margin:0;} */
2
3  body {min-height: 5000px;}
4      h1 {text-align: center; margin: 60px;}
5
6  .boxBar {
7      width: 100%;
8      border: 5px solid royalblue;
9      position: fixed;
10     top: 0;
11     left: 0;
12 }
13
14 .bar {
15     height: 20px;
16     background: royalblue;
17     width: 0%;
18 }
19
20
21
22
23

```

Le CSS : Le body j'ai mis une hauteur minimale (pour pouvoir scroller)

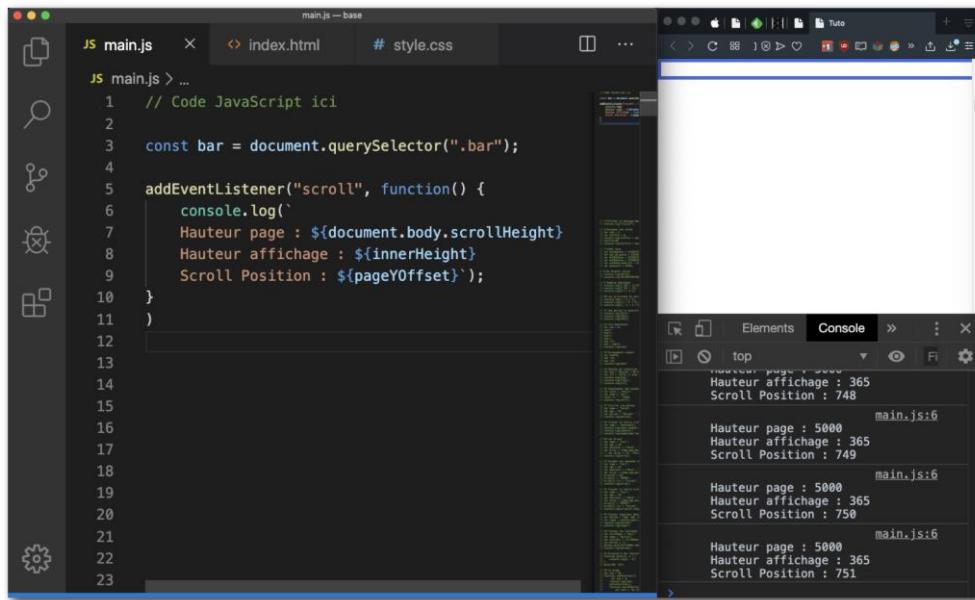
La boxbar va faire la largeur de toute la page, une position fixée en haut (même quand je scroll)

La bar (à l'intérieur) hauteur 20 px un fond de la même couleur et qui pour l'infant à une largeur à 0%

En JS on va animer cette largeur

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a browser developer tools window with two panes. The left pane displays a snippet of JavaScript code in a file named 'main.js'. The code is as follows:

```

JS main.js X index.html # style.css
JS main.js > ...
1 // Code JavaScript ici
2
3 const bar = document.querySelector(".bar");
4
5 addEventListener("scroll", function() {
6   console.log(`Hauteur page : ${document.body.scrollHeight}
7   Hauteur affichage : ${innerHeight}
8   Scroll Position : ${pageYOffset}`);
9
10 }
11 )
12
13
14
15
16
17
18
19
20
21
22
23

```

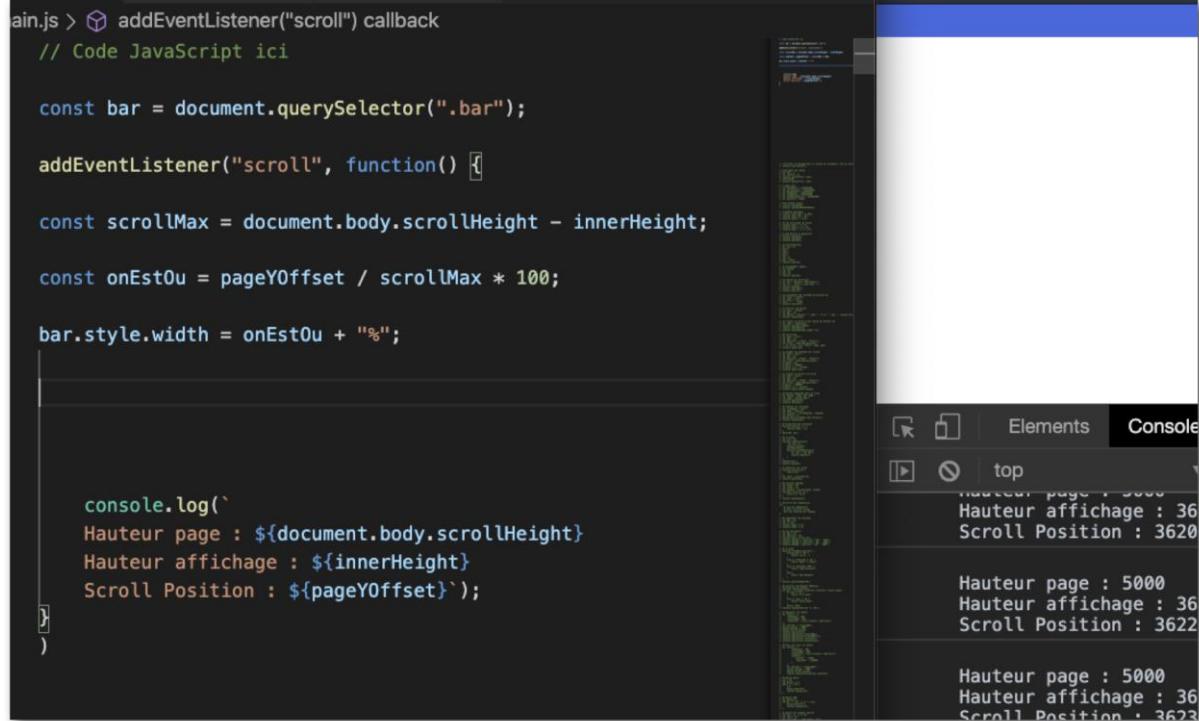
The right pane shows the browser's console output. It logs three sets of values for each scroll event:

- Hauteur page : 365  
Hauteur affichage : 365  
Scroll Position : 748
- Hauteur page : 5000  
Hauteur affichage : 365  
Scroll Position : 749
- Hauteur page : 5000  
Hauteur affichage : 365  
Scroll Position : 750
- Hauteur page : 5000  
Hauteur affichage : 365  
Scroll Position : 751

Ici en surveillant le scroll on va afficher en console la hauteur de la page, la hauteur de l'affichage de notre page (on a tous des tailles d'écrans ≠) et la position Y du scroll

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```

main.js > ⚡ addEventListener("scroll") callback
// Code JavaScript ici

const bar = document.querySelector(".bar");

addEventListener("scroll", function() {
  const scrollMax = document.body.scrollHeight - innerHeight;
  const onEstOu = pageYOffset / scrollMax * 100;

  bar.style.width = onEstOu + "%";

  console.log(`Hauteur page : ${document.body.scrollHeight}
  Hauteur affichage : ${innerHeight}
  Scroll Position : ${pageYOffset}`);
})

```

The screenshot shows a browser window with developer tools open. The console tab is active, displaying three log entries. The first entry shows the initial state: Hauteur page : 5000, Hauteur affichage : 36, Scroll Position : 3620. The second entry shows the state after scrolling: Hauteur page : 5000, Hauteur affichage : 36, Scroll Position : 3622. The third entry shows the state after further scrolling: Hauteur page : 5000, Hauteur affichage : 36, Scroll Position : 3623.

On va récupérer notre bar (le blanc à l'intérieur)

On ajoute un event listener sur l'ensemble de la page et on va écouter le scroll

Il nous faut plusieurs choses pour savoir le pourcentage que l'on doit afficher. Console log pour vérifier

La taille entière de notre page

La taille de l'affichage innerHeight Récupère la hauteur (en pixels) de la partie visible de la fenêtre de navigation en incluant, si elle est affichée, la barre de défilement horizontale.

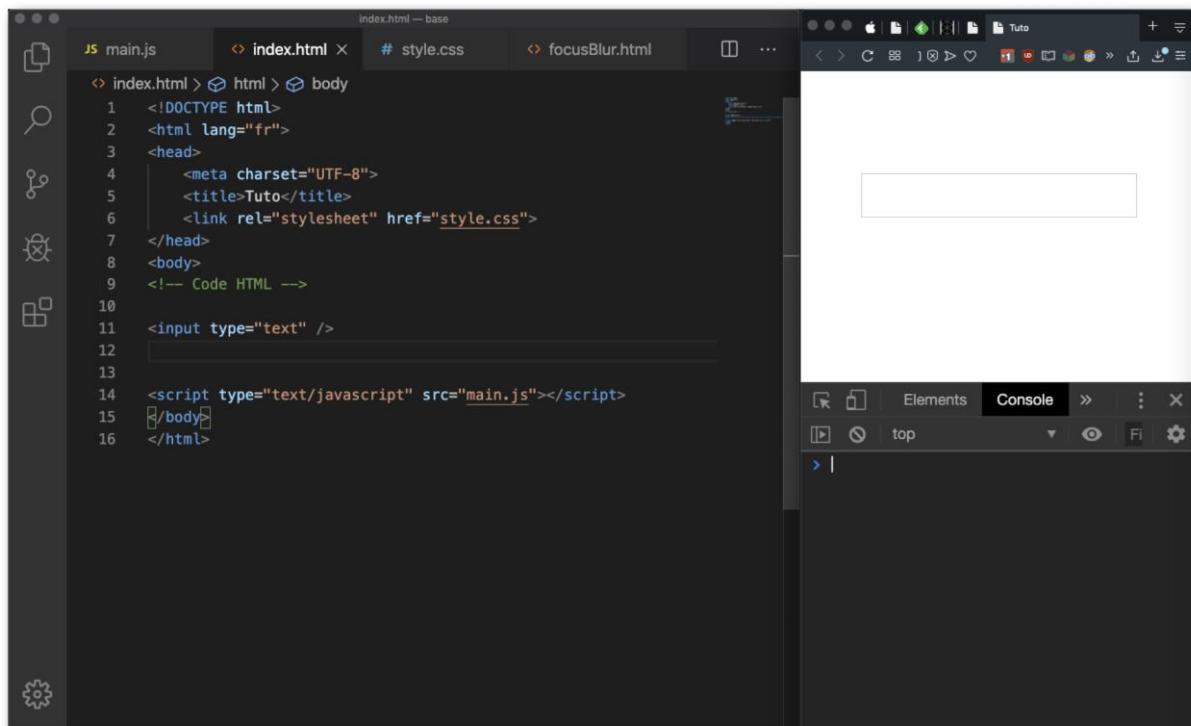
La position du scroll vertical pageYOffset ( $\neq$  pageXOffset)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Évènement Focus et Blur

Un petit exercice, sur notre page nous aurons un input, lorsque l'on click dans l'input nous ferons en sorte que sa couleur change et quand on click en dehors de l'input la couleur redevient normale.



The screenshot shows a code editor with three tabs: `index.html`, `main.js`, and `style.css`. The `index.html` tab contains the following HTML code:

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <!-- Code HTML -->
    <input type="text" />
</body>
</html>

```

The `main.js` tab contains the following JavaScript code:

```

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

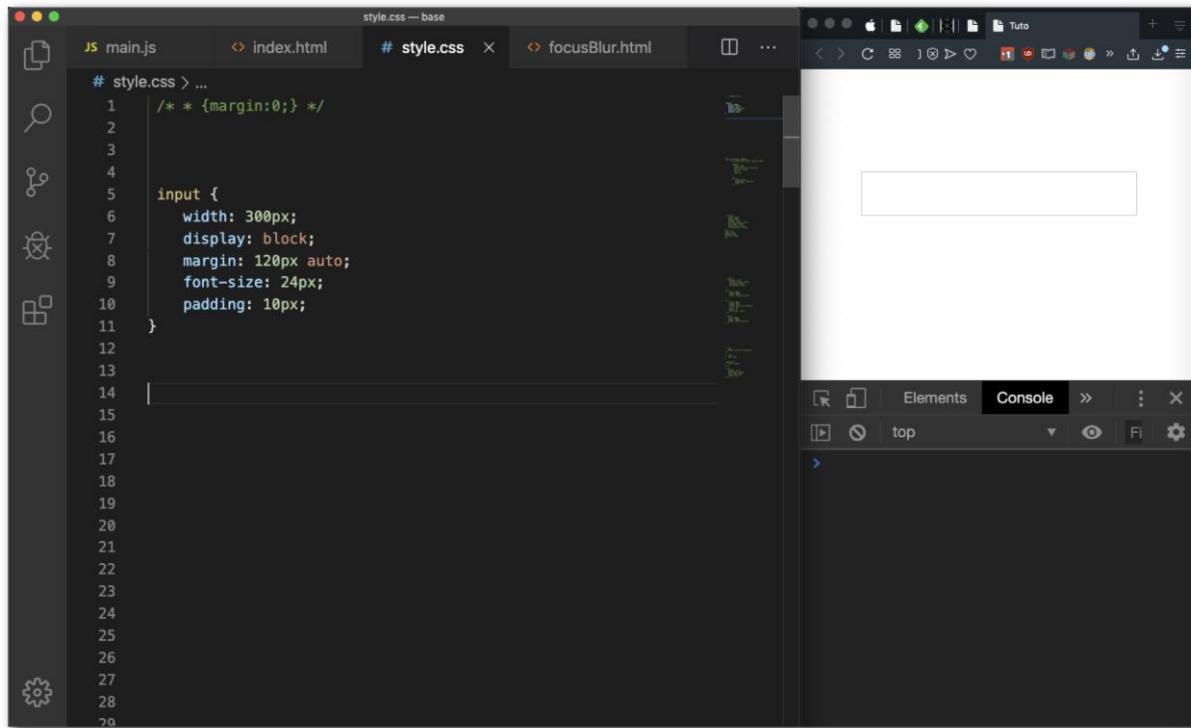
```

To the right, a browser window displays a simple text input field. The browser's developer tools are open, showing the `Elements` and `Console` tabs.

HTML : simplement un input

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor with a CSS file named style.css. The code contains a single rule for an input element:

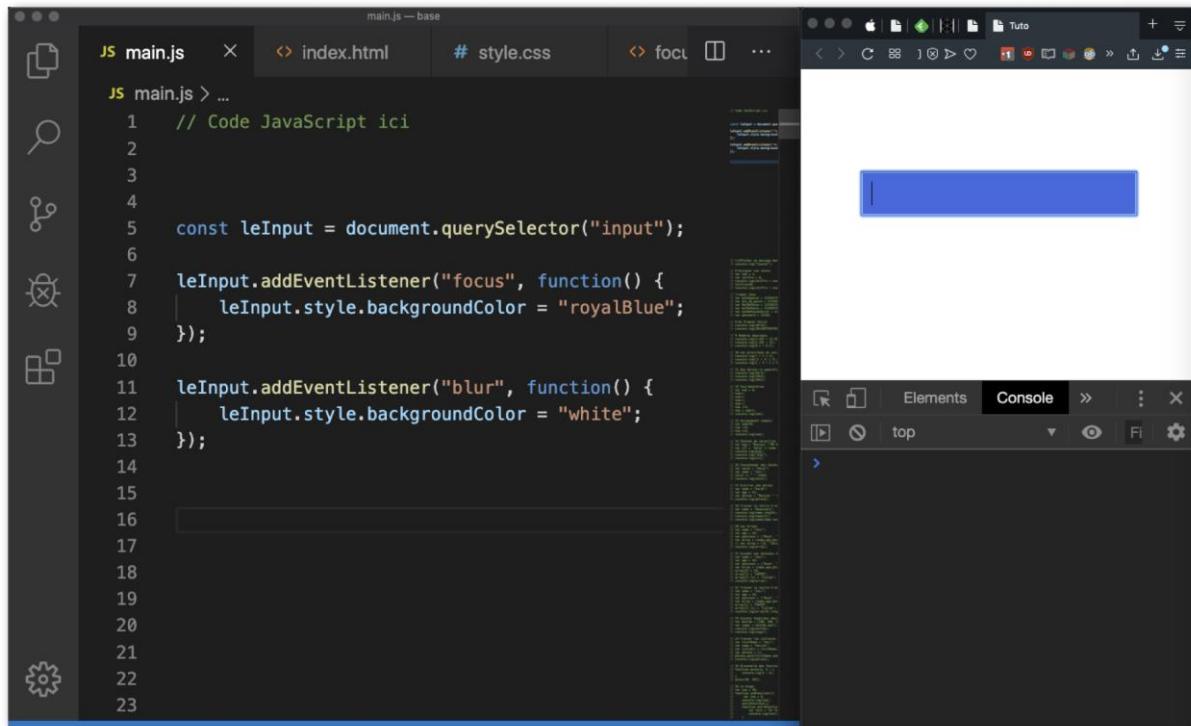
```
# style.css > ...
1  /* * {margin:0;} */
2
3
4
5  input {
6    width: 300px;
7    display: block;
8    margin: 120px auto;
9    font-size: 24px;
10   padding: 10px;
11 }
```

To the right, a browser window displays a simple input field centered on the page. Below the browser is a developer tools console window showing the command "top".

Un léger CSS pour centrer mon input.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor with a file named `main.js` open. The code contains JavaScript code that adds event listeners to an input element. When the input gains focus, its background color changes to royal blue. When it loses focus (blur), its background color changes back to white. The browser's developer tools are also visible, showing the DOM structure and the console tab.

```

main.js — base
JS main.js  X  index.html  # style.css  <| focus  ... 

JS main.js > ...
1 // Code JavaScript ici
2
3
4
5 const leInput = document.querySelector("input");
6
7 leInput.addEventListener("focus", function() {
8   leInput.style.backgroundColor = "royalBlue";
9 });
10
11 leInput.addEventListener("blur", function() {
12   leInput.style.backgroundColor = "white";
13 });
14
15
16
17
18
19
20
21
22
23

```

Il nous faut notre élément input qu'on va stocker dans une variable `leInput`.

Un `addEventListner` qui surveille le focus (quand la souris passe sur un élément précis)

Et qui execute une fonction qui va motif la couleur de fond.

Là le style est appliqué mais il ne se désactive pas !!! Mais heureusement on a un attribut qui fait l'inverse (défocuse).

Sur le input on remet un `addEventListner` mais on surveille le Blur et il exécute une fonction qui remet son BackGround à white. (Notre page par défaut est blanche).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS

### Évènement load

Un nouvel event, load

Permet de faire une action uniquement après qu'un élément est chargé.

On met le load sur un élément en particulier

ex : images trop lourdes sur un site on veut pouvoir utiliser la page avant que les images soit totalement chargées.

On met le load sur l'ensemble de la page pour exécuter notre script une fois que toute la page soit chargée.

```
<meta charset="UTF-8">
<title>Tuto</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<!-- Code HTML -->



















<script type="text/javascript" src="main.js"></script>
</body>
</html>
```

HTML : dans notre page on va vouloir afficher plusieurs images de grande taille.

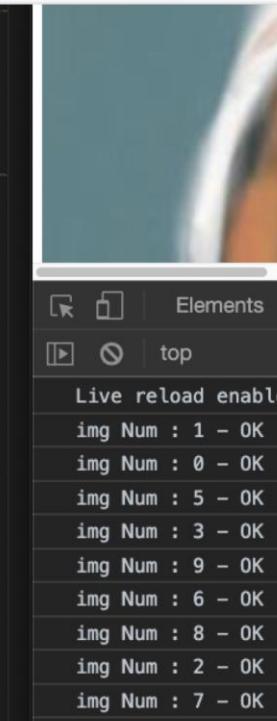
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS

```
const images = document.querySelectorAll("img");

const tabImg = Array.from(images);

tabImg.map((image,i) => image.addEventListener("load",
function(){
    console.log(`img Num : ${i} - OK`);
}))
```



On va voir dans quel ordre sont chargées les img

On met toutes nos img dans une variable avec querySelectorAll ça nous renvoi un nodeList

On le transforme en array pour pouvoir utiliser map dessus. Array.from(images)

On va utiliser map pour ajouter un addeventListner load sur chaque image.

Si je mets un 2nd paramètre à map avec, automatiquement ça va être l'index

Le addEventListener va surveiller load et exécuter une fonction qui dès que l'image est chargée il va m'afficher dans quel ordre les images sont chargées.

On peut constater que les images ne se chargent pas dans l'ordre.

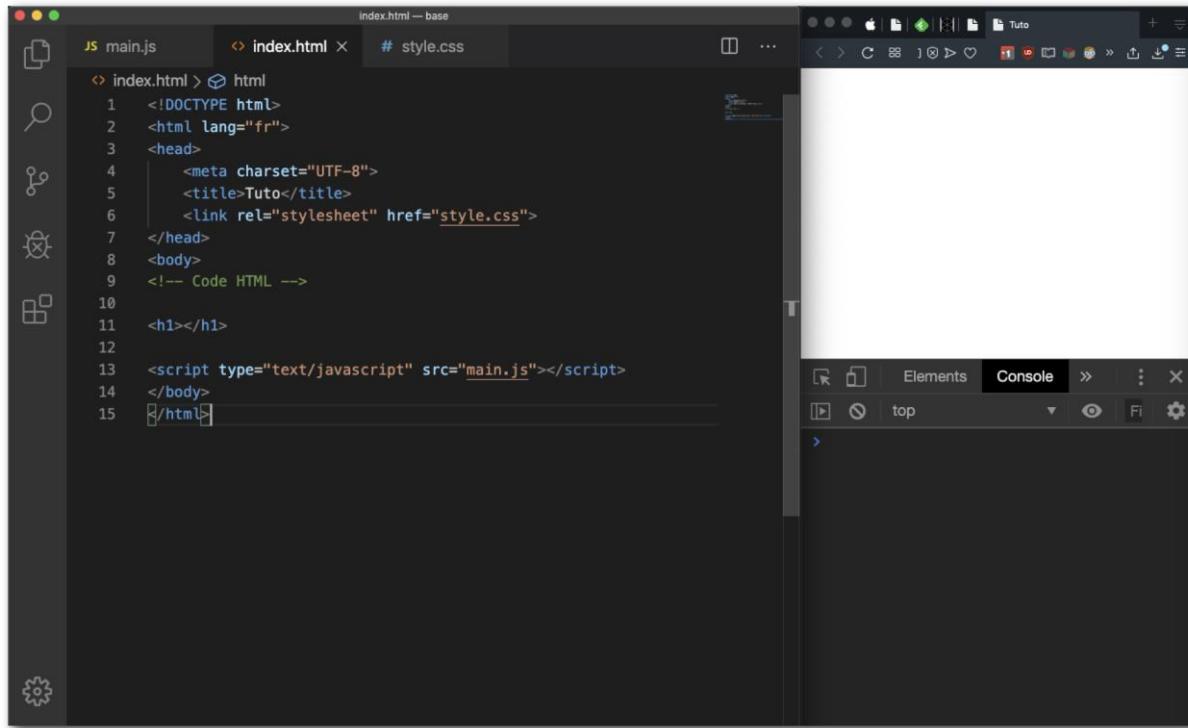
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## setTimeout()

Comment exécuter un script après un temps donnée.

Dans cet exemple nous allons créer une page dans laquelle un titre apparaît après un certain temps.



```

index.html — base
JS main.js index.html # style.css
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <title>Tuto</title>
6      <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9  <!-- Code HTML -->
10 <h1></h1>
11
12 <script type="text/javascript" src="main.js"></script>
13 </body>
14 </html>

```

HTML : Juste un titre h1 vide

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

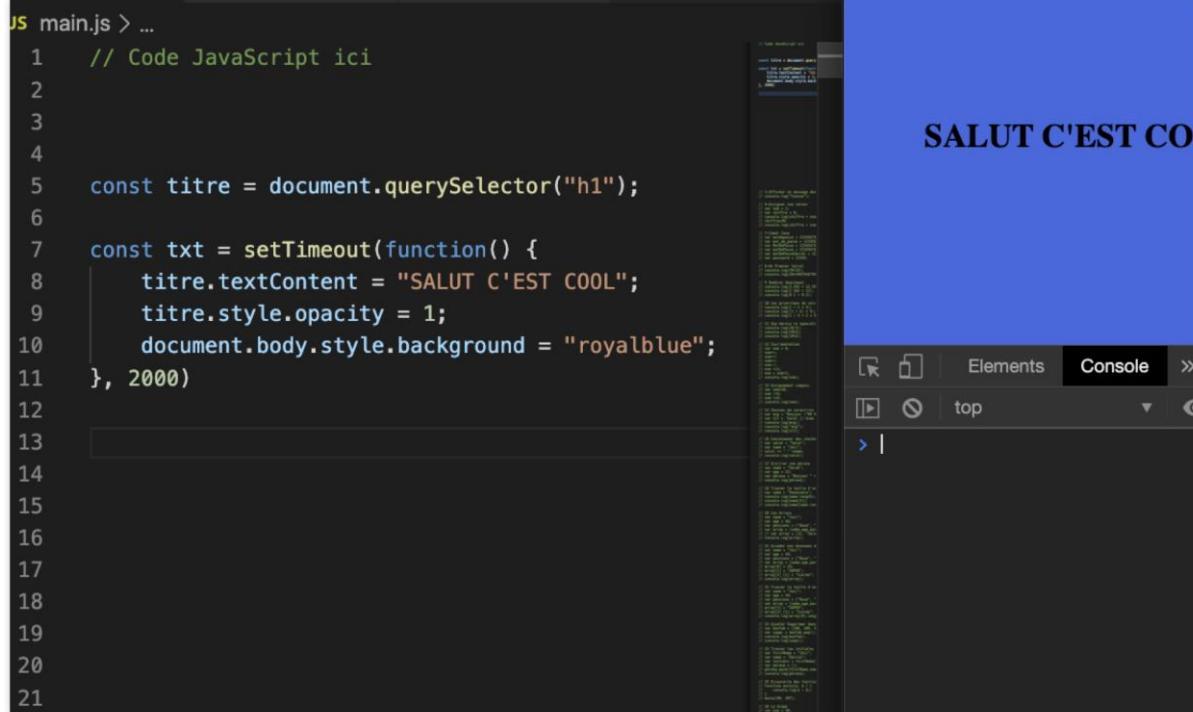
JS

```
style.css > ...
1
2
3
4
5
6 * {
7   transition: 1s;
8 }
9 h1 {
10   text-align: center;
11   margin-top: 120px;
12   opacity: 0;
13 }
14
15
16
17
18
19
20
21
22
23
24
```

CSS : sur tous les éléments, une transition d'une seconde et sur le titre, une opacité à 0.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```
JS main.js > ...
1 // Code JavaScript ici
2
3
4
5 const titre = document.querySelector("h1");
6
7 const txt = setTimeout(function() {
8     titre.textContent = "SALUT C'EST COOL";
9     titre.style.opacity = 1;
10    document.body.style.backgroundColor = "royalblue";
11 }, 2000)
12
13
14
15
16
17
18
19
20
21
```

On va sélectionner notre balise h1 pour pouvoir écrire du texte dedans.

1 constante txt dans laquelle on va appeler la fonction setTimeout qui prend 2 paramètres

Une Fonction et un temps 1000 = 1 sec

La fonction qu'on veut qu'il exécute.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

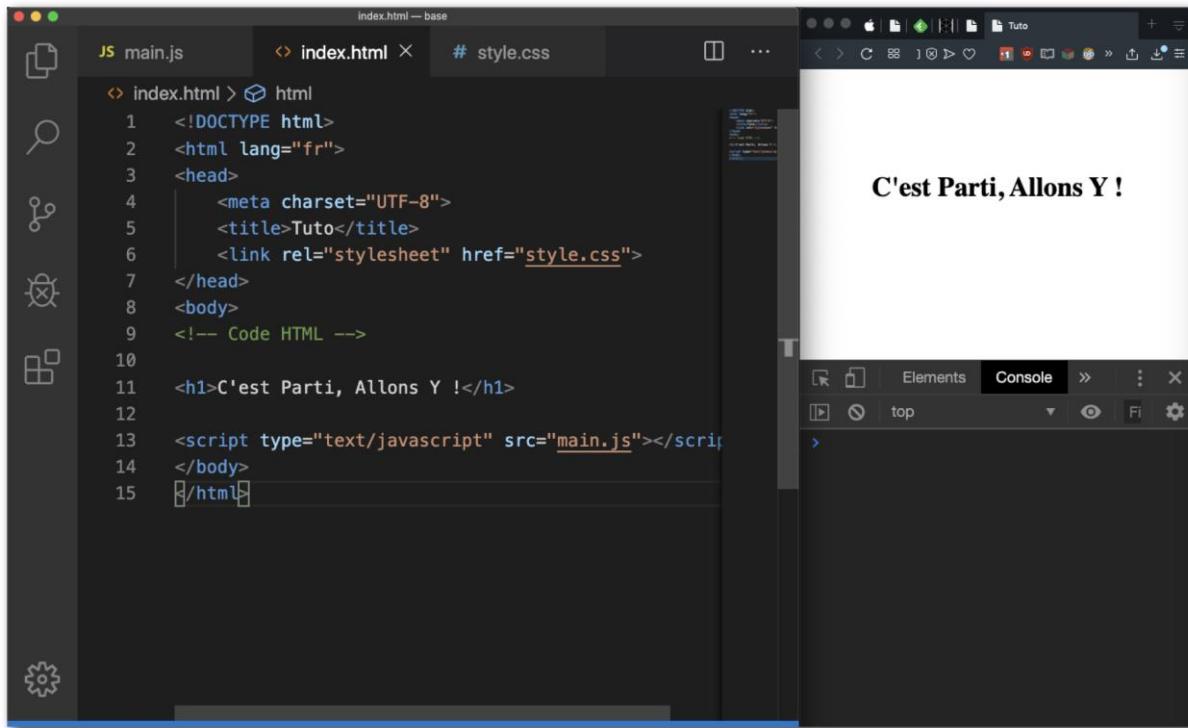
# JS

## setInterval()

Cette méthode va nous permettre d'exécuter un code toutes les X secondes.

Nous avons aussi à disposition la méthode clearInterval pour stopper l'exécution du code.

Dans cet exemple nous allons créer une page web qui affiche un décompte puis affiche un titre.



The screenshot shows a code editor with two files open: `index.html` and `main.js`. The `index.html` file contains:

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <!-- Code HTML -->
    <h1>C'est Parti, Allons Y !</h1>
<script type="text/javascript" src="main.js"></script>
</body>
</html>

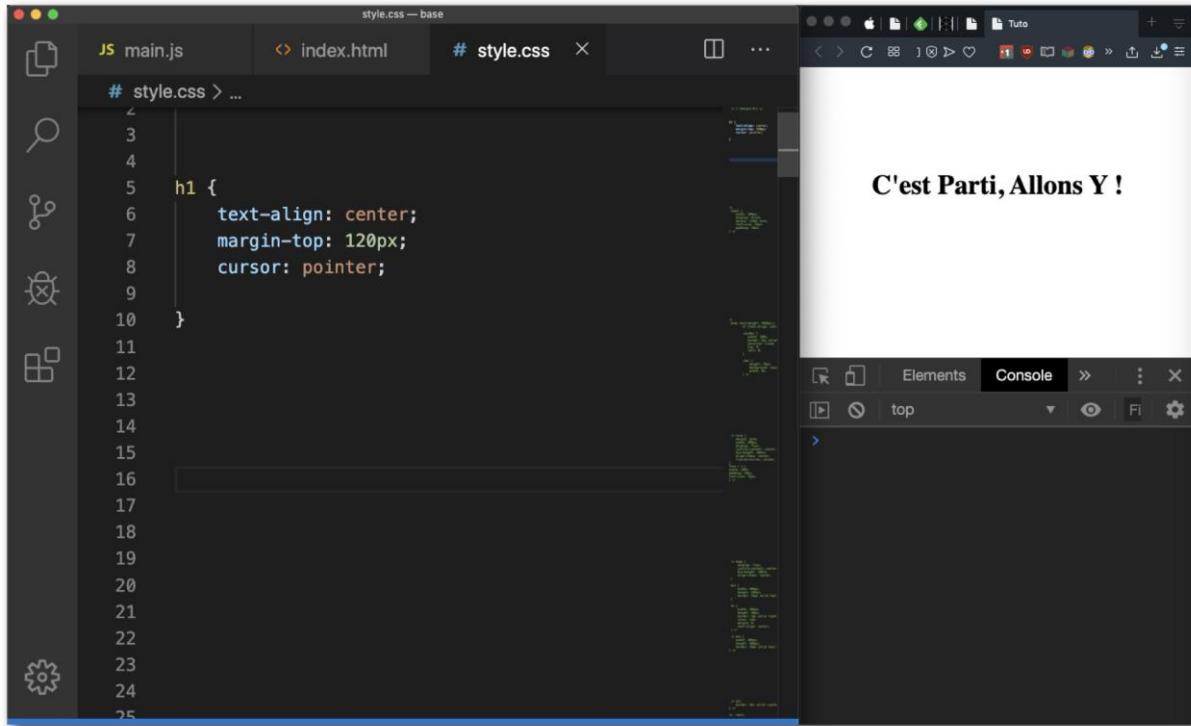
```

The browser window shows the title "C'est Parti, Allons Y !" in large black font. The developer tools console is open, showing no errors.

Html : juste un titre h1

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

style.css — base
JS main.js index.html # style.css ...
# style.css > ...
3
4
5   h1 {
6     text-align: center;
7     margin-top: 120px;
8     cursor: pointer;
9
10 }
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```

The screenshot shows a code editor with three tabs: 'main.js', 'index.html', and '# style.css'. The '# style.css' tab is active, displaying the following CSS code:

```

# style.css > ...
3
4
5   h1 {
6     text-align: center;
7     margin-top: 120px;
8     cursor: pointer;
9
10 }
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

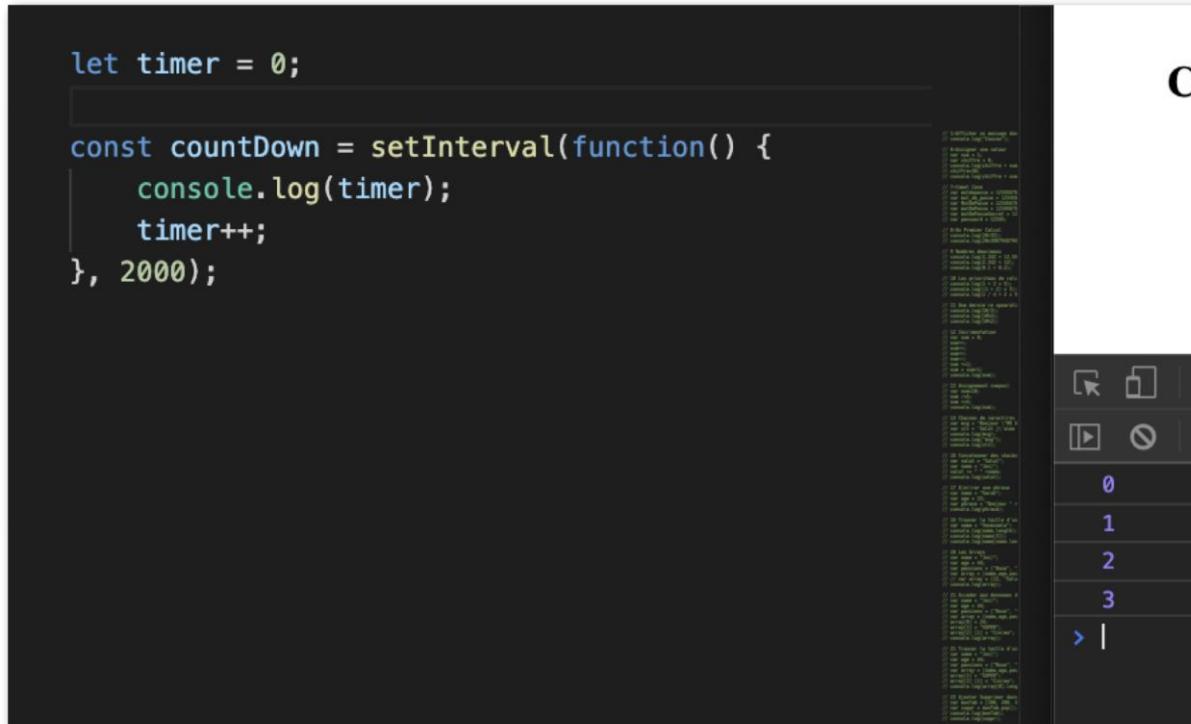
```

To the right of the code editor is a browser window showing the result. The page displays a single line of text: "C'est Parti, Allons Y !". The browser's developer tools console tab is open, showing no errors.

CSS : pour centrer le titre.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```

let timer = 0;

const countDown = setInterval(function() {
    console.log(timer);
    timer++;
}, 2000);

```

The screenshot shows a code editor with a dark theme. On the left is the code provided above. On the right is a terminal window showing the output of the script: '0', '1', '2', and '3'. There are also some icons and a scroll bar visible.

On va stocker notre set interval dans 1 Const countDown on lui passe 1 fonction et des secondes  
On check si ça fonctionne avec un console.log

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

JS

The screenshot shows a browser developer tools console window with the title bar "main.js — base". The console tab is selected, displaying the following log entries:

```
2  
1  
0  
-1  
-2  
-3  
-4  
-5  
-6  
-7
```

The console also shows the file path "main.js:16" repeated for each entry.

Ça fonctionne mais on n'a pas fini , je veux modifier mon titre du coup je la stocke dans une const avec un querySelector

Sur ce titre j'ajoute un addEventListener qui va surveiller le clic et exécuter une fonction qui va faire :

Si le timer > 0 alors le titre contiendra le timer (3, 2, 1)

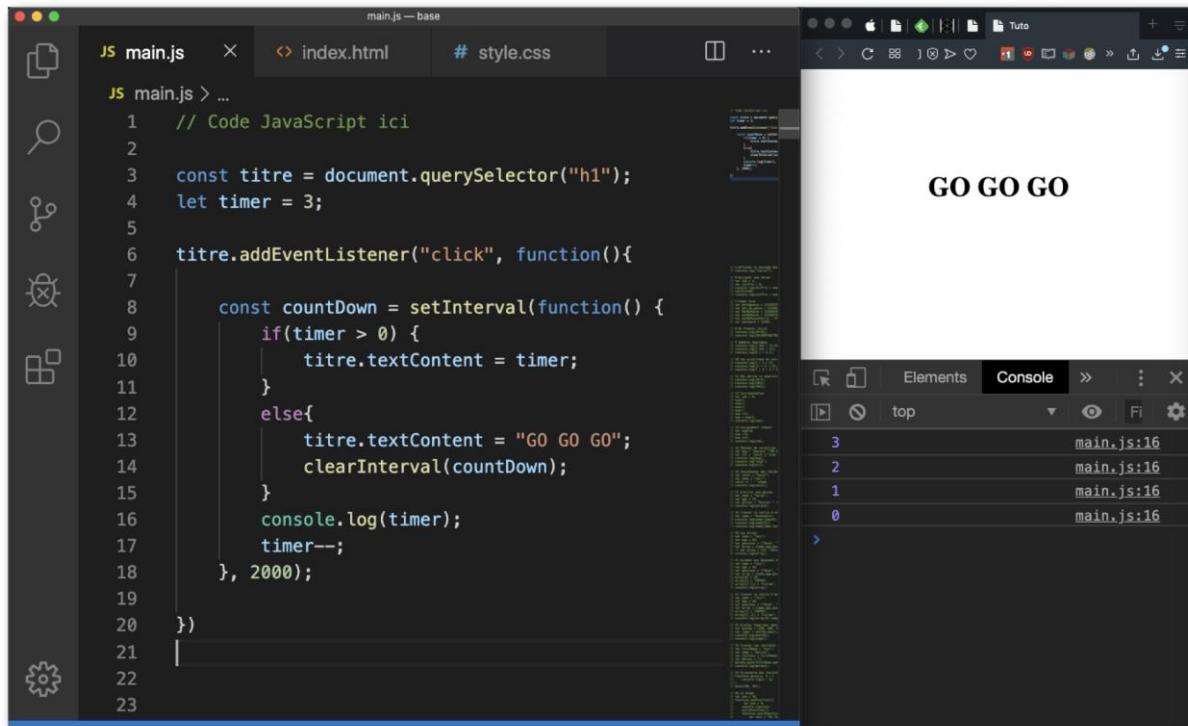
Sinon le titre contiendra le message « GO »

Je fais i - - pour avoir un effet de décompte.

PB : notre boucle tourne indéfiniment

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		<i>Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.</i>
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor on the left and a browser window on the right. The code editor has tabs for 'main.js', 'index.html', and '# style.css'. The 'main.js' tab contains the following code:

```

JS main.js > ...
1 // Code JavaScript ici
2
3 const titre = document.querySelector("h1");
4 let timer = 3;
5
6 titre.addEventListener("click", function(){
7
8     const countDown = setInterval(function() {
9         if(timer > 0) {
10             titre.textContent = timer;
11         }
12         else{
13             titre.textContent = "GO GO GO";
14             clearInterval(countDown);
15         }
16         console.log(timer);
17         timer--;
18     }, 2000);
19
20 })
21
22
23

```

The browser window shows the word 'GO GO GO' in large letters. The developer tools' console tab is open, showing the following log entries:

- 3 main.js:16
- 2 main.js:16
- 1 main.js:16
- 0 main.js:16

une fois arrivé à GO qu'il arrête de faire tourner l'intervalle  
On fait un clearInterval de countdown dans le else.

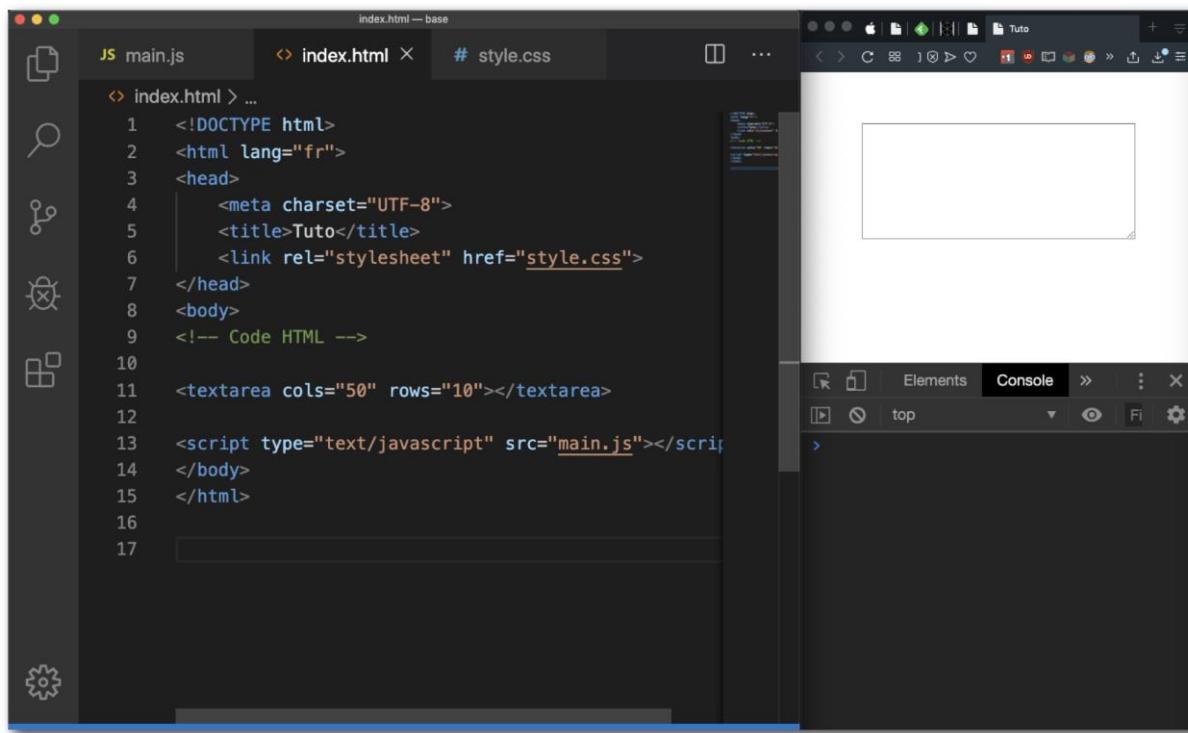
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Array.includes(), Array.join()

Dans cet exemple nous allons voir comment récupérer le texte que tape l'utilisateur et nous n'en afficherons que les consonnes.

Nous utiliserons principalement 2 méthodes liés aux objets de type tableau, à savoir array.includes() et array.join()



```

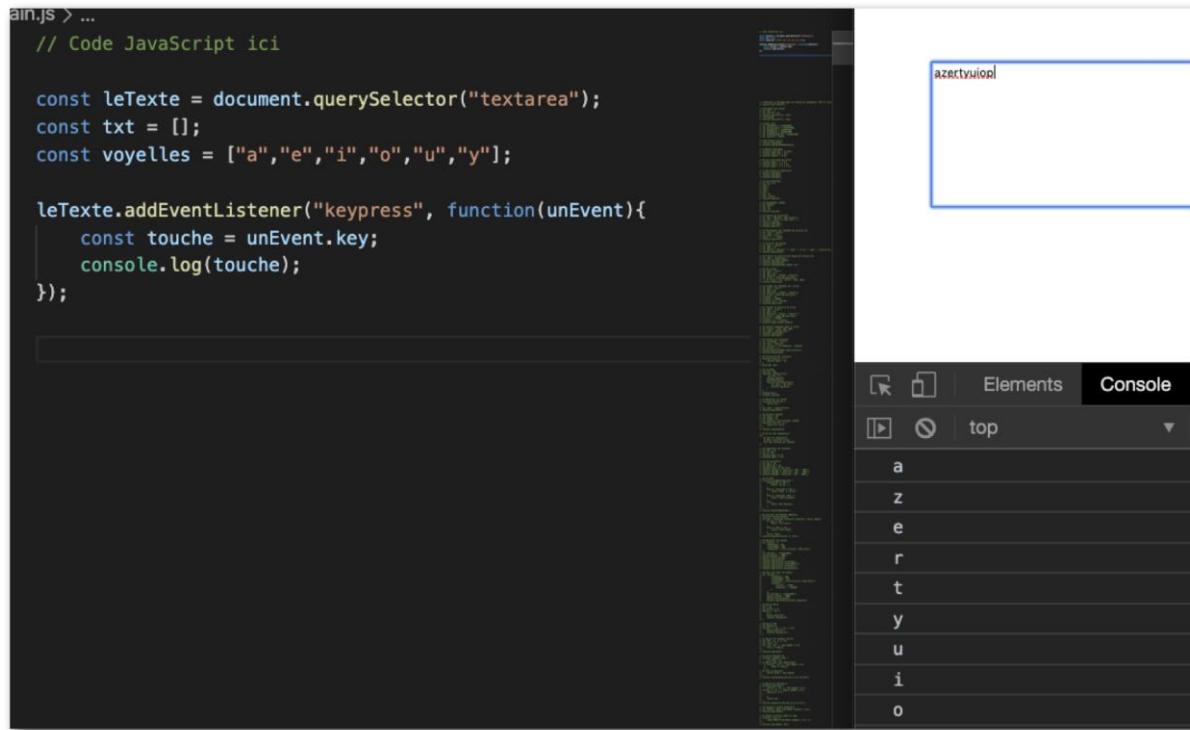
index.html — base
JS main.js    index.html ×  # style.css
index.html > ...
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <title>Tuto</title>
6      <link rel="stylesheet" href="style.css">
7  </head>
8  <body>
9  <!-- Code HTML -->
10 <textarea cols="50" rows="10"></textarea>
11 <script type="text/javascript" src="main.js"></script>
12 </body>
13 </html>
14
15
16
17

```

Html : Juste un textarea

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```
ain.js > ...
// Code JavaScript ici

const leTexte = document.querySelector("textarea");
const txt = [];
const voyelles = ["a","e","i","o","u","y"];

leTexte.addEventListener("keypress", function(unEvent){
    const touche = unEvent.key;
    console.log(touche);
});
```

On va récupérer notre textarea pour voir si l'utilisateur tape bien dedans. (querySelector)

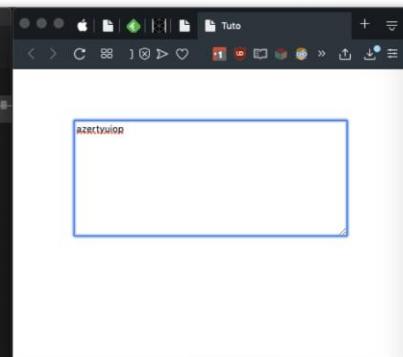
1 const qui va stocker le texte sans les voyelles sous forme de array

On va savoir quelles sont nos voyelles sous forme de tableau

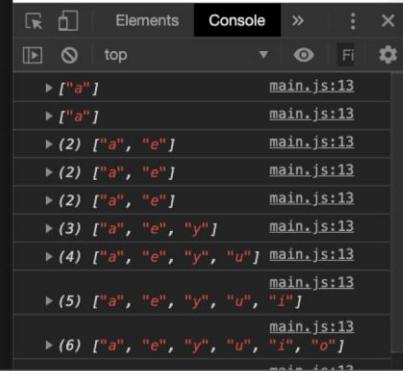
Ajouter un AddEventListener pour surveiller Keypress (quand on tape sur le clavier) et exécuter 1 fonction qui va récupérer les lettres que je tape. (.key)  
on console log pour vérifier.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

## JS



```
main.js — base
JS main.js < index.html # style.css
JS main.js > ↗ leTexte.addEventListener("keypress") callback
1 // Code JavaScript ici
2
3 const leTexte = document.querySelector("textarea");
4 const txt = [];
5 const voyelles = ["a","e","i","o","u","y"];
6
7 leTexte.addEventListener("keypress", function(unEvent){
8     const touche = unEvent.key;
9
10    if(voyelles.includes(touche)) {
11        txt.push(touche);
12    }
13    console.log(txt);
14});
```



```
Elements Console
top
▶ ["a"]
▶ ["a"]
▶ (2) ["a", "e"]
▶ (2) ["a", "e"]
▶ (2) ["a", "e"]
▶ (3) ["a", "e", "y"]
▶ (4) ["a", "e", "y", "u"]
▶ (5) ["a", "e", "y", "u", "i"]
▶ (6) ["a", "e", "y", "u", "i", "o"]
```

Maintenant on va indiquer que si la touche est ≠ des voyelles qu'on a dans le tableau.

Alors on peut la mettre dans txt

Si touche est inclus dans voyelle alors on fait un push de la touche dans txt

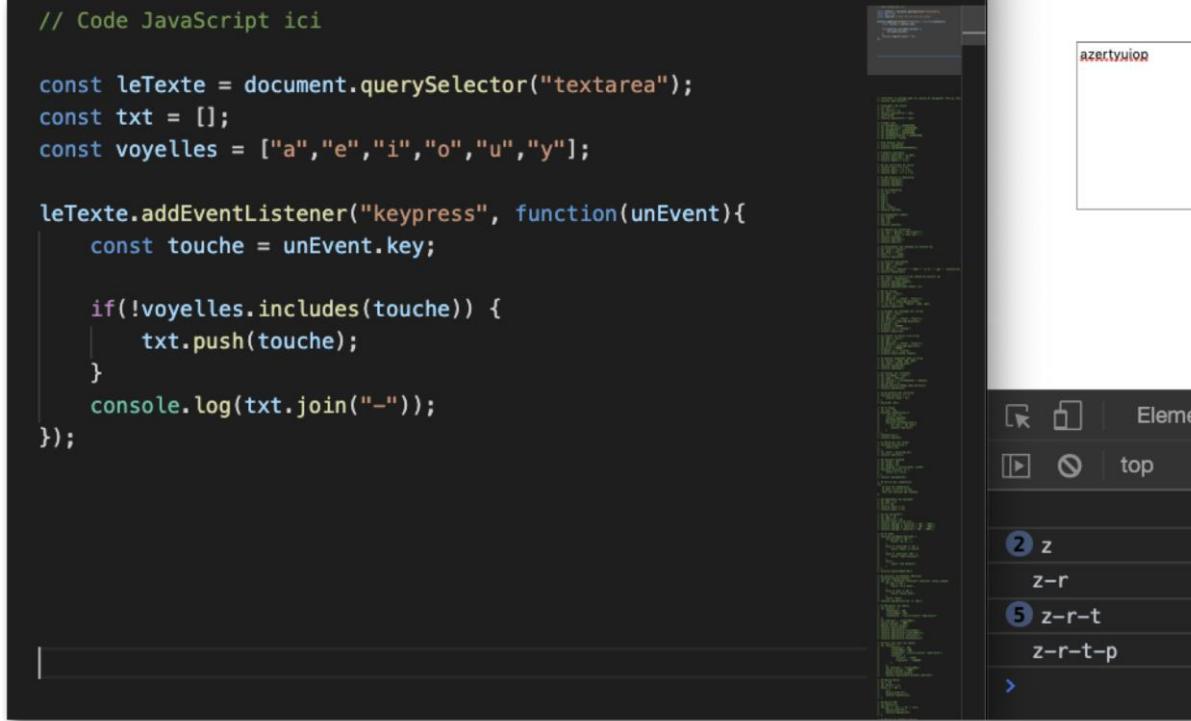
Je console log pour vérifier.

PB En fait on a fait l'inverse de ce qu'on voulait : la console n'affiche que les voyelles.

Si la touche est dans le tableau on la rajoute dans notre texte sans voyelle.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```
// Code JavaScript ici

const leTexte = document.querySelector("textarea");
const txt = [];
const voyelles = ["a","e","i","o","u","y"];

leTexte.addEventListener("keypress", function(unEvent){
    const touche = unEvent.key;

    if(!voyelles.includes(touche)) {
        txt.push(touche);
    }
    console.log(txt.join("-"));
});
```

Dans le if j'inverse ma condition avec le « ! »

Avec les array, on a une fonction dispo le join qui permet de joindre les cellules du tableau.  
ici on met un tiret entre chaque lettre que je tape.

Il prend tous les éléments du array et il les colle ensemble en rajoutant ce qu'on lui donne ici le -

Si je laisse vide ça fait bien une chaîne de caractères.

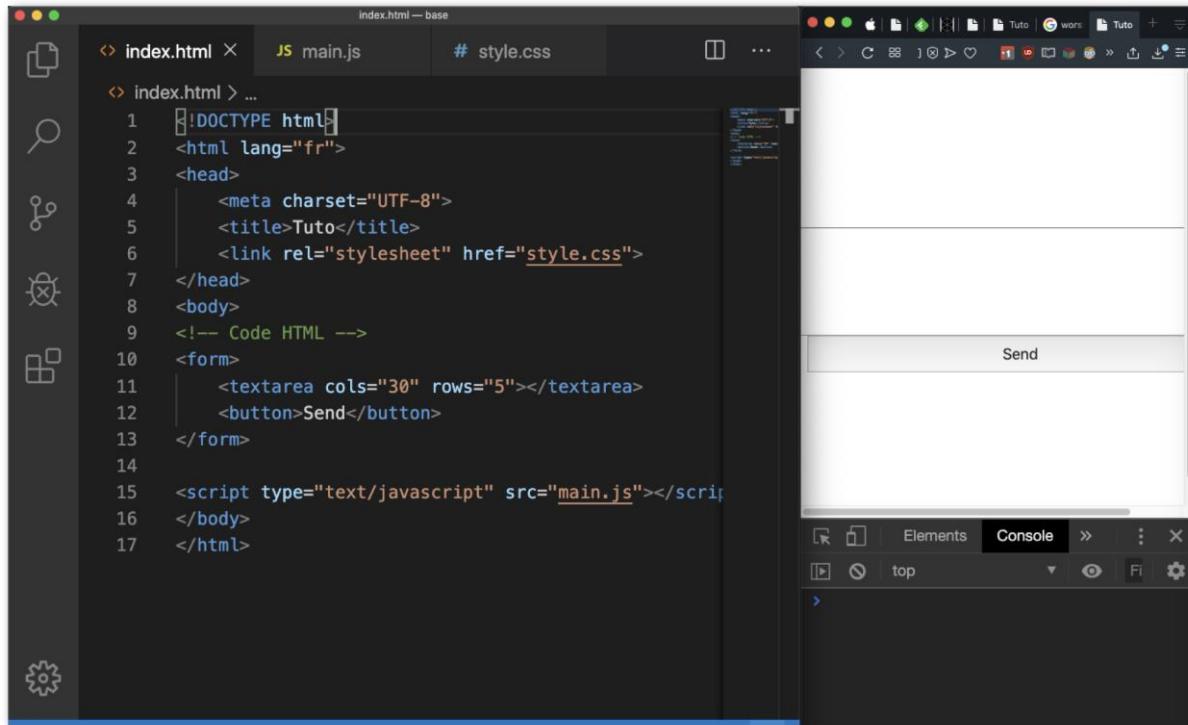
Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

## Exercice : Désactiver un bouton selon la longueur d'un texte.

Dans cet exercice nous allons mettre en place une page web contenant une zone de texte ainsi qu'un bouton.

Le but sera de désactiver le bouton si l'utilisateur tape plus de 5 caractères dans la zone de texte.



```

index.html — base
index.html × JS main.js # style.css

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <!-- Code HTML -->
    <form>
        <textarea cols="30" rows="5"></textarea>
        <button>Send</button>
    </form>
</body>
</html>

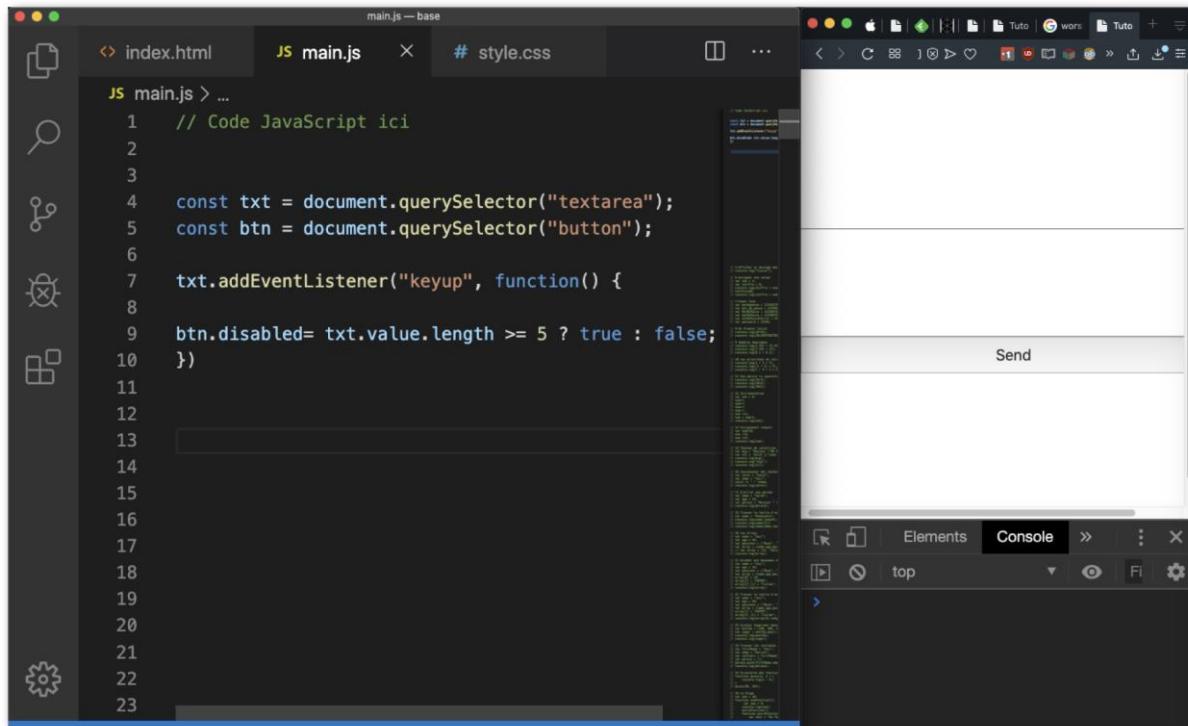
<script type="text/javascript" src="main.js"></script>

```

Html : Un formulaire contient un textarea et un bouton

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



```

main.js — base
JS main.js > ...
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const txt = document.querySelector("textarea");
5 const btn = document.querySelector("button");
6
7 txt.addEventListener("keyup", function() {
8
9 btn.disabled= txt.value.length >= 5 ? true : false;
10})
11
12
13
14
15
16
17
18
19
20
21
22
23

```

The screenshot shows a code editor with a file named 'main.js'. The code is as follows:

```

const txt = document.querySelector("textarea");
const btn = document.querySelector("button");

txt.addEventListener("keyup", function() {
  btn.disabled= txt.value.length >= 5 ? true : false;
})

```

To the right of the code editor is a browser window displaying a simple form with a text area and a button. The button has a 'Send' label. Below the browser window is a developer tools console tab labeled 'Console'.

On récupère nos 2 éléments html dans des variables txt et btn

Ensuite sur txt on place un addEventListener qui réagit à l'évènement keyup et déclenche une fonction.

Dans cette fonction on va pouvoir utiliser une condition qui renverra true ou false à la propriété disabled du bouton.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS

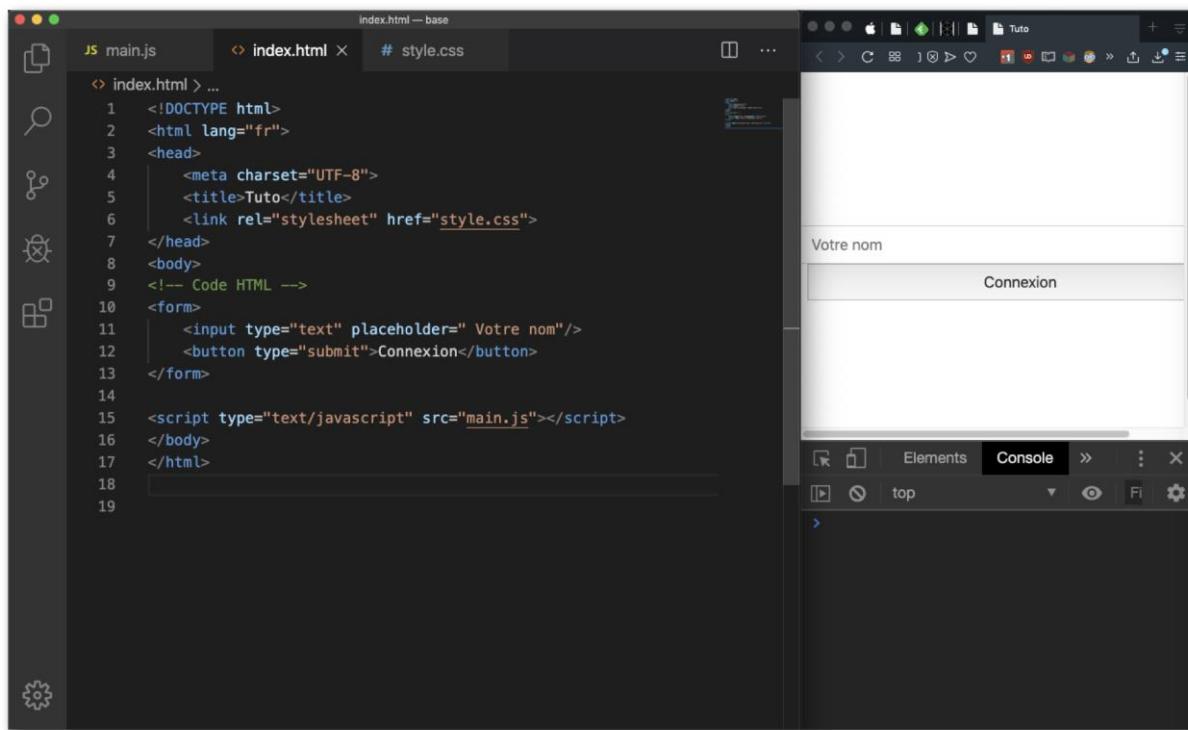
## DOM JS : PreventDefault

preventDefault() est une méthode liée à la gestion des évènements, pour simplifier, cette méthode permet d'éviter certains comportement de base des navigateurs.

Dans l'exemple ci-après nous allons créer une page dans laquelle nous aurons un formulaire avec un input et un bouton.

Le comportement par défaut sur les navigateurs consiste lorsque l'on soumet un formulaire à réinitialiser les champs de ce formulaire.

Nous allons faire en sorte d'éviter ce comportement du navigateur.



The screenshot shows a code editor with three tabs: 'main.js', 'index.html', and '# style.css'. The 'index.html' tab contains the following HTML code:

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Tuto</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<!-- Code HTML -->
<form>
    <input type="text" placeholder=" Votre nom"/>
    <button type="submit">Connexion</button>
</form>
<script type="text/javascript" src="main.js"></script>
</body>
</html>

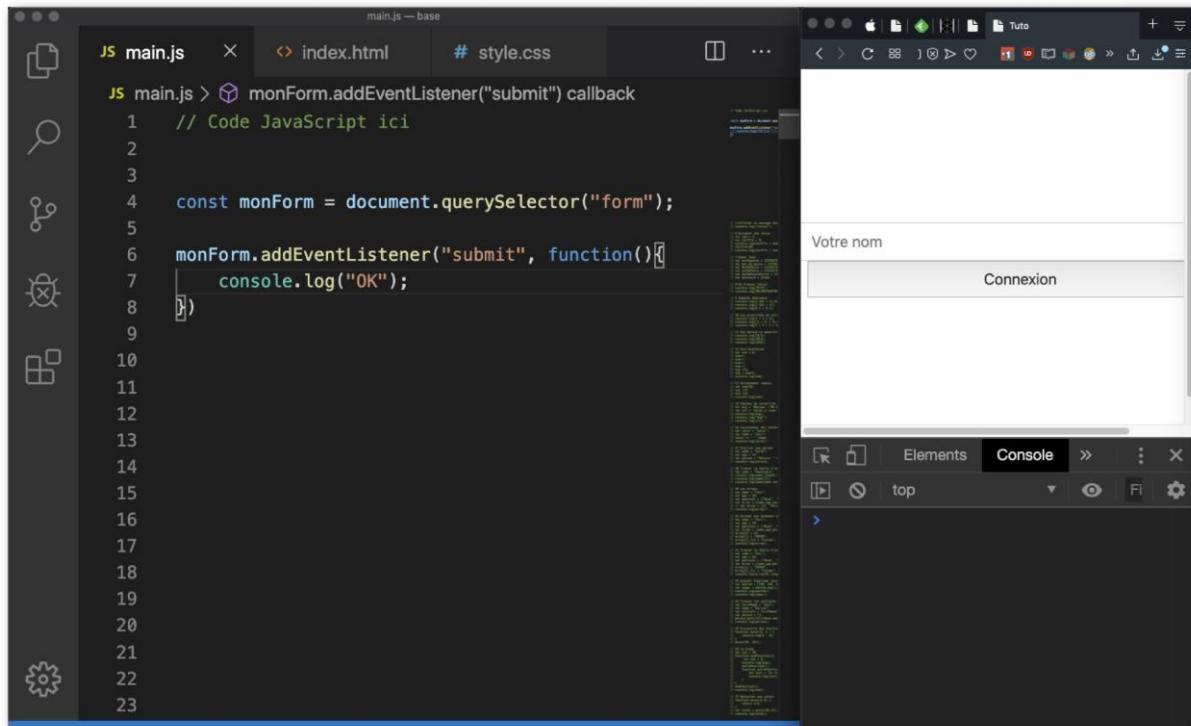
```

The browser window shows a simple form with a text input field labeled 'Votre nom' and a submit button labeled 'Connexion'.

Html : un formulaire contenant un input et un bouton.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following JavaScript code:

```

main.js — base
JS main.js X > index.html # style.css ...
JS main.js > monForm.addEventListener("submit") callback
1 // Code JavaScript ici
2
3
4 const monForm = document.querySelector("form");
5
6 monForm.addEventListener("submit", function(){
7   console.log("OK");
8 })
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

The browser window shows a simple form with two input fields: "Votre nom" and "Connexion". The developer tools' console tab is open at the bottom, showing the message "OK" printed to the console.

On stock le formulaire dans une constante.

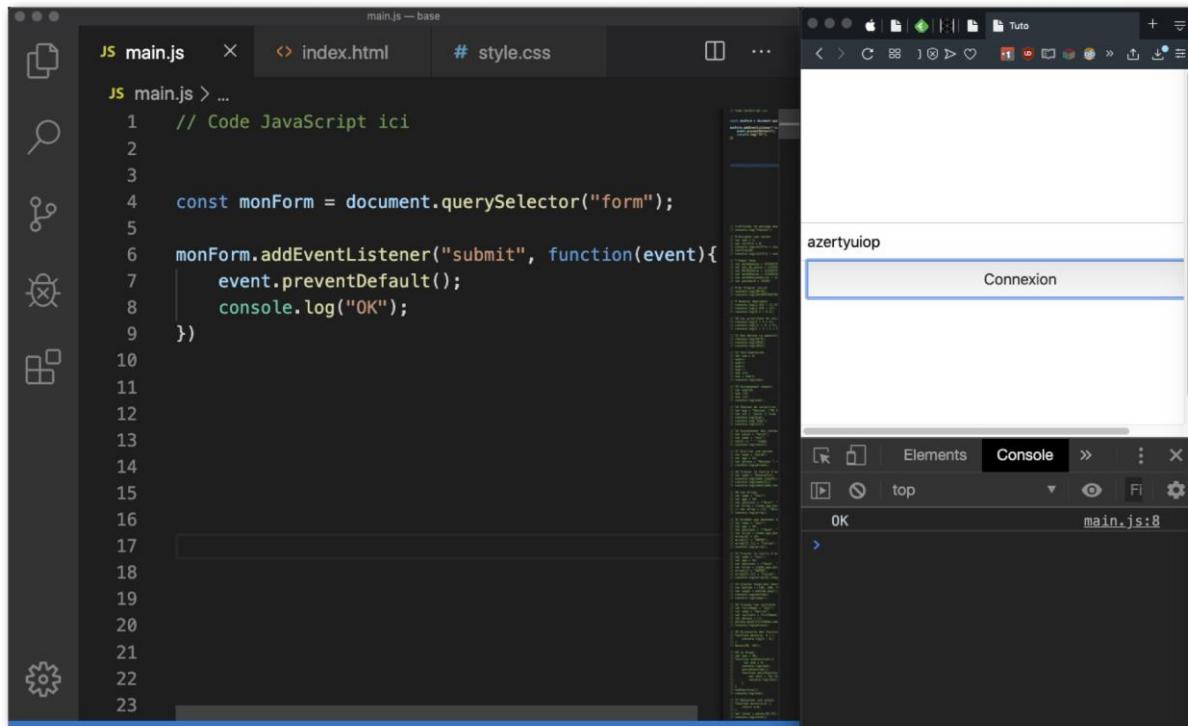
Sur mon form j'ajoute un eventListener qui surveille submit et qui exécute une fonction  
Qui me console log un message pour check.

PB : l'affichage en console log se fait mais de manière très brève.

Du coup on constate que par défaut notre navigateur recharge la page quand on soumet le formulaire.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The image shows a code editor window with a tab for 'main.js' containing the following JavaScript code:

```

main.js — base
JS main.js X index.html # style.css ...
JS main.js > ...
1 // Code JavaScript ici
2
3
4 const monForm = document.querySelector("form");
5
6 monForm.addEventListener("submit", function(event){
7     event.preventDefault();
8     console.log("OK");
9 })
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```

Below the code editor is a screenshot of a web browser displaying a login form. The browser's developer tools are open, showing the 'Console' tab which contains the message 'OK'.

Pour empêcher une action par défaut fait par le navigateur

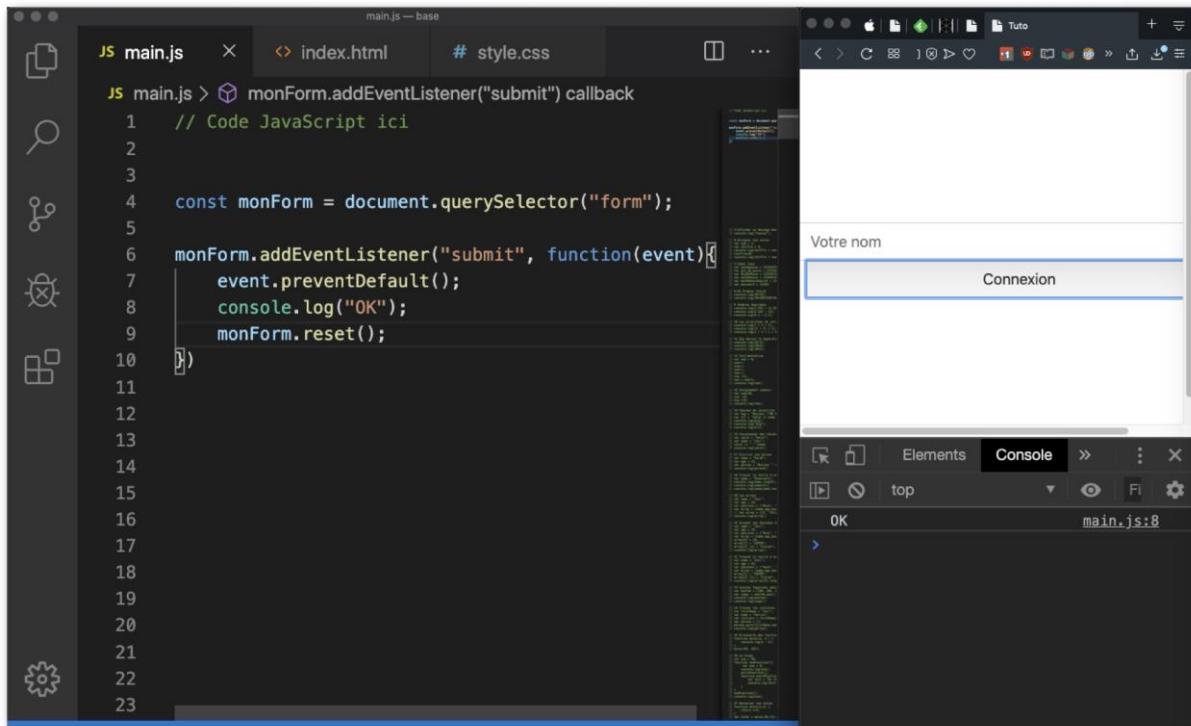
Dans fonction on lui passe une variable event sur laquelle on va appeler une fonction de JS prévue pour ça :

preventDefault (on va se prévenir du comportement par défaut du navigateur)

Cela fonctionne la page n'est pas entièrement rechargée mais du coup le formulaire n'est pas réinitialisé.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

# JS



The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following JavaScript code:

```

main.js -- base
JS main.js  X  index.html  # style.css  ...
JS main.js >  monForm.addEventListener("submit") callback
1   // Code JavaScript ici
2
3
4   const monForm = document.querySelector("form");
5
6   monForm.addEventListener("submit", function(event){
7       event.preventDefault();
8       console.log("OK");
9       monForm.reset();
10  })
11
12
13
14
15
16
17
18
19
20
21
22
23

```

The browser window shows a simple form with a single input field labeled "Votre nom" and a "Connexion" button. The developer tools' console tab is open, showing the message "OK" from the log statement in the script.

On appelle la fonction reset() sur notre formulaire quand on soumet le formulaire

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX		Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		