



L'ECOLE  
REGIONALE DU  
NUMERIQUE



GRANDE  
ÉCOLE DU  
NUMÉRIQUE

Microsoft  
IT Academy

LinkedIn Learning

Networking  
Cisco Academy

ADRAR\o/  
POLE NUMERIQUE



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

## Table des matières

<b>Objectif Pédagogique.....</b>	<b>3</b>
<b>DOM JS : Présentation.....</b>	<b>4</b>
<b>DOM JS : Sélectionner un élément Html.....</b>	<b>7</b>
<b>DOM JS : Placer un élément Html.....</b>	<b>10</b>
<b>DOM JS : Créer un élément Html.....</b>	<b>13</b>
<b>DOM JS : Modifier les attributs d'un élément Html.....</b>	<b>16</b>
<b>DOM JS : Transformer une liste d'élément Html en tableau.....</b>	<b>17</b>
<b>DOM JS : Obtenir la taille d'un élément Html.....</b>	<b>21</b>
<b>DOM JS : modifier le style d'un élément Html.....</b>	<b>22</b>
<b>DOM JS : Ajouter ou supprimer des classes Css.....</b>	<b>24</b>
<b>DOM JS : Gestion des évènements.....</b>	<b>28</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		



L'ECOLE  
REGIONALE DU  
NUMERIQUE



GRANDE  
ÉCOLE DU  
NUMÉRIQUE

Microsoft  
IT Academy

LinkedIn Learning

Networking  
cisco Academy

ADRAR\o/  
POLE NUMERIQUE



JS

## Objectif Pédagogique

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 : Présentation

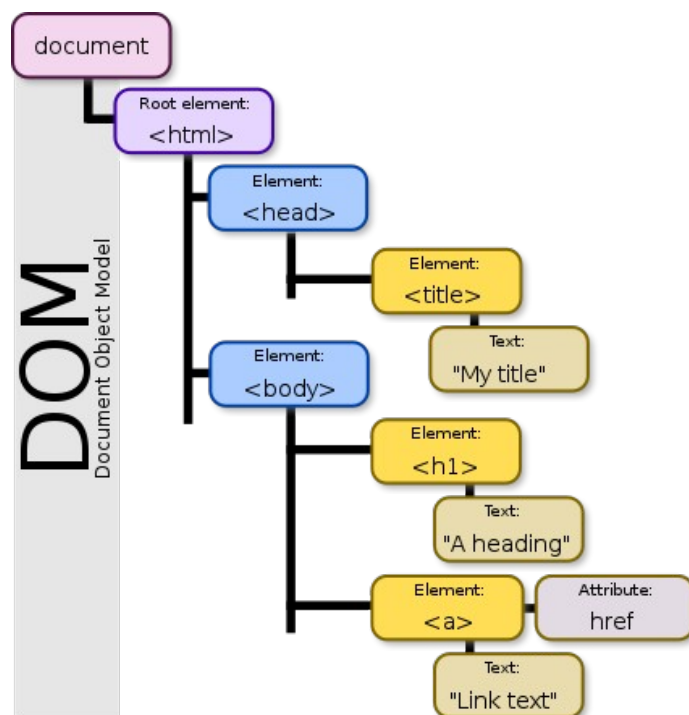
Le **Document Object Model (DOM)** est une [interface de programmation](#) normalisée par le [W3C](#), qui permet à des scripts d'examiner et de modifier le contenu du [navigateur web](#)<sup>1</sup>. Par le DOM, la composition d'un document [HTML](#) ou [XML](#) est représentée sous forme d'un jeu d'objets – lesquels peuvent représenter une fenêtre, une phrase ou un style, par exemple – reliés selon une structure en [arbre](#)<sup>1</sup>. À l'aide du DOM, un script peut modifier le document présent dans le navigateur en ajoutant ou en supprimant des nœuds de l'arbre<sup>1</sup>. (Wikipedia)

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...

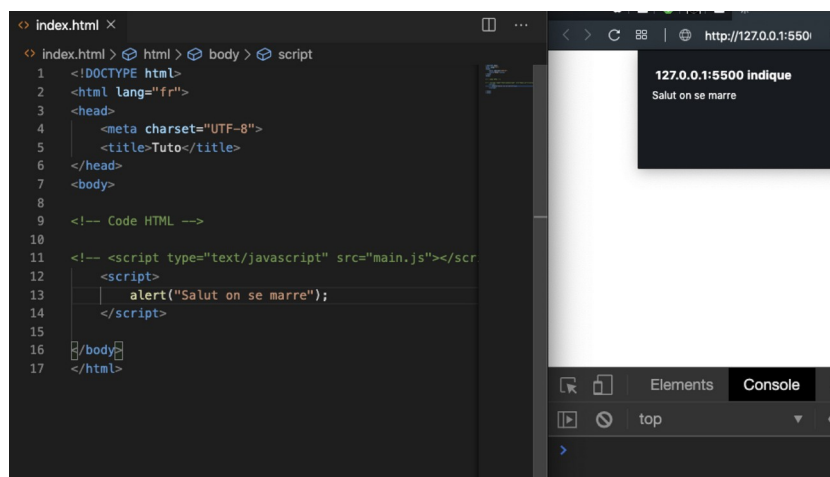


Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
	Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE		
	Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son 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>` :

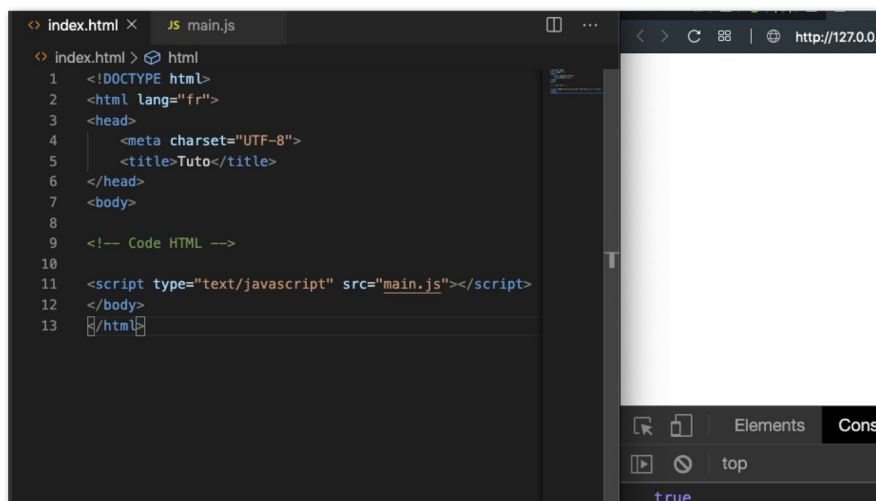


```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Tuto</title>
6 </head>
7 <body>
8
9 <!-- Code HTML -->
10
11 <!-- <script type="text/javascript" src="main.js"></script> -->
12 <script>
13   alert("Salut on se marre");
14 </script>
15
16 </body>
17 </html>

```

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



```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Tuto</title>
6 </head>
7 <body>
8
9 <!-- Code HTML -->
10
11 <script type="text/javascript" src="main.js"></script>
12 </body>
13 </html>

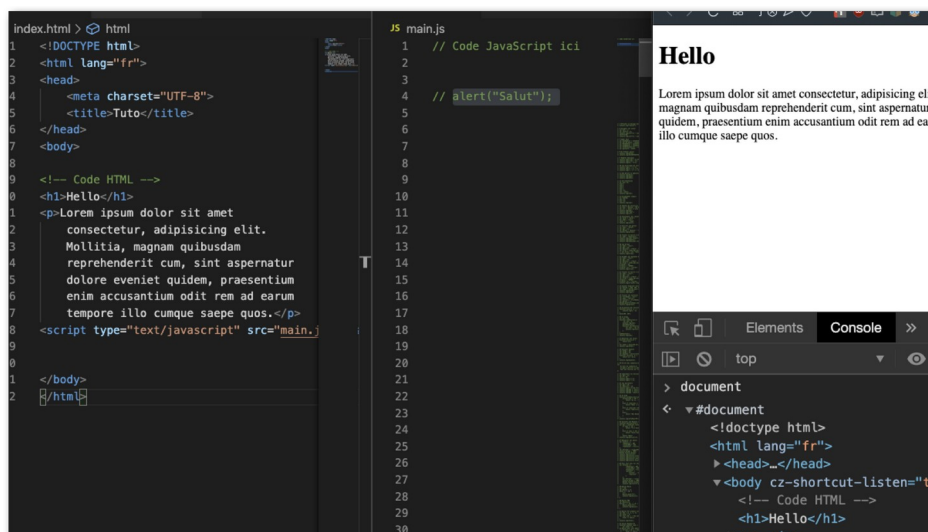
```

Souvent on relie notre JS à la fin du fichier html  
(pour d'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.



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

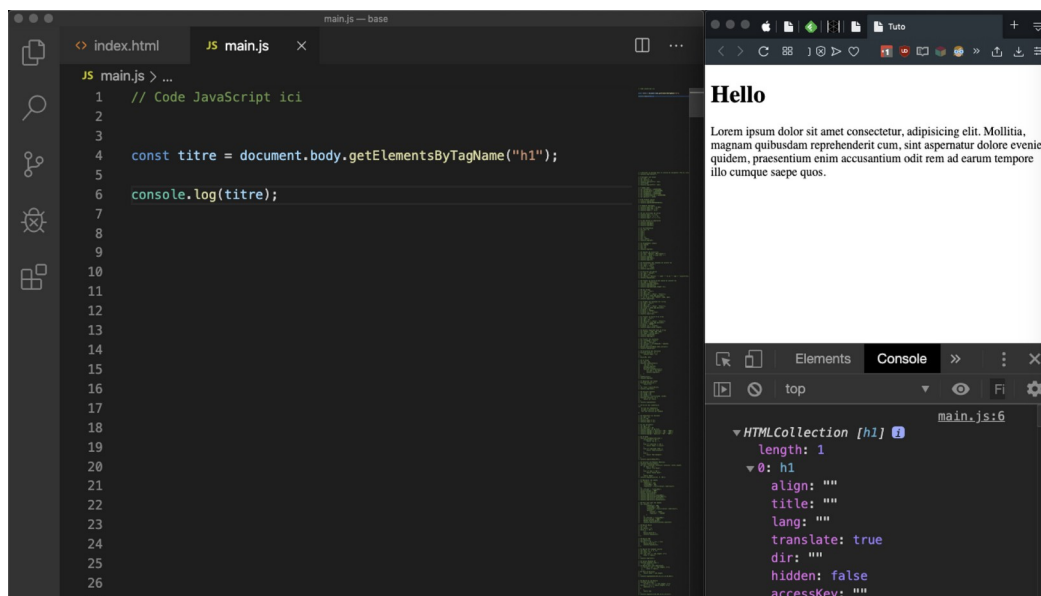
## DOM JS : Sélectionner un élément Html

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>.



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 getElementsByName

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		



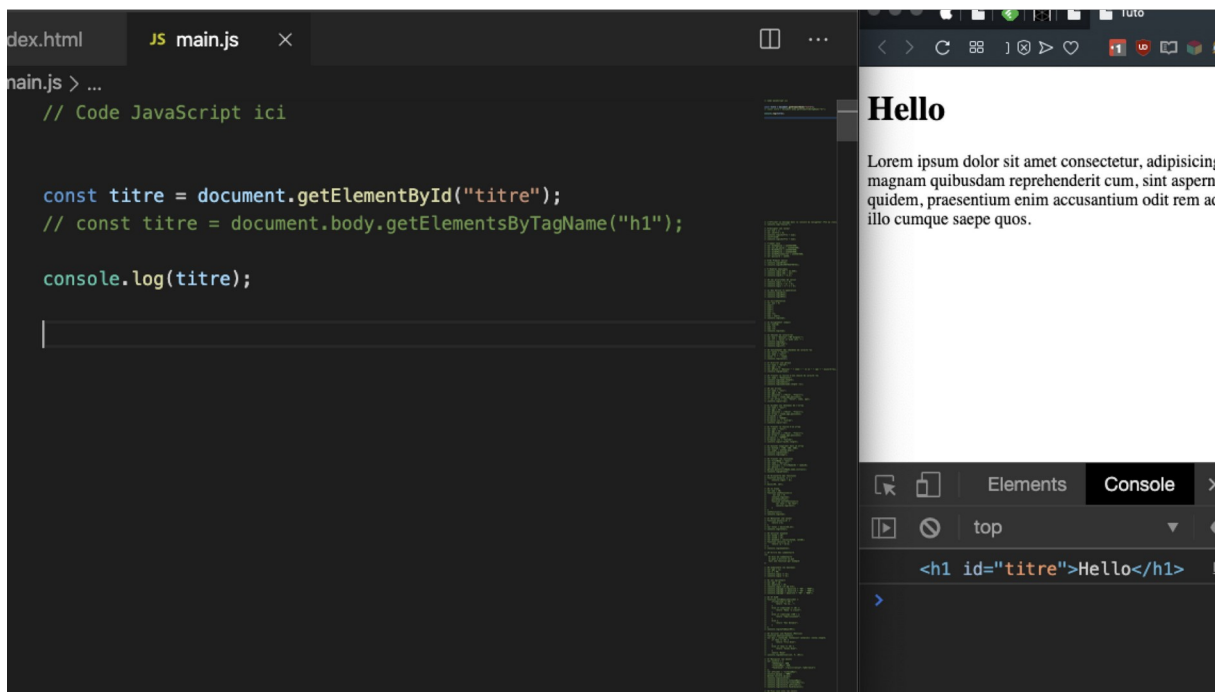
# JS

## 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'ya 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)



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		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		

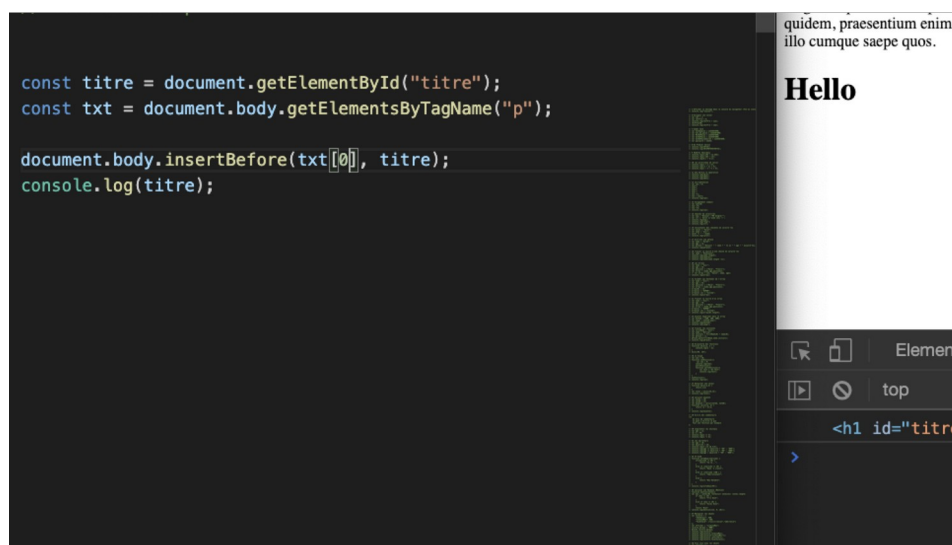
## JS

### DOM JS : 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 :



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		
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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.

# 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);
```

magnam quibusdam reprehenderit quidem, praesentium enim accus illo cumque saepe quos.

**Hello**

Elements

top

<h1 id="titre">Hello

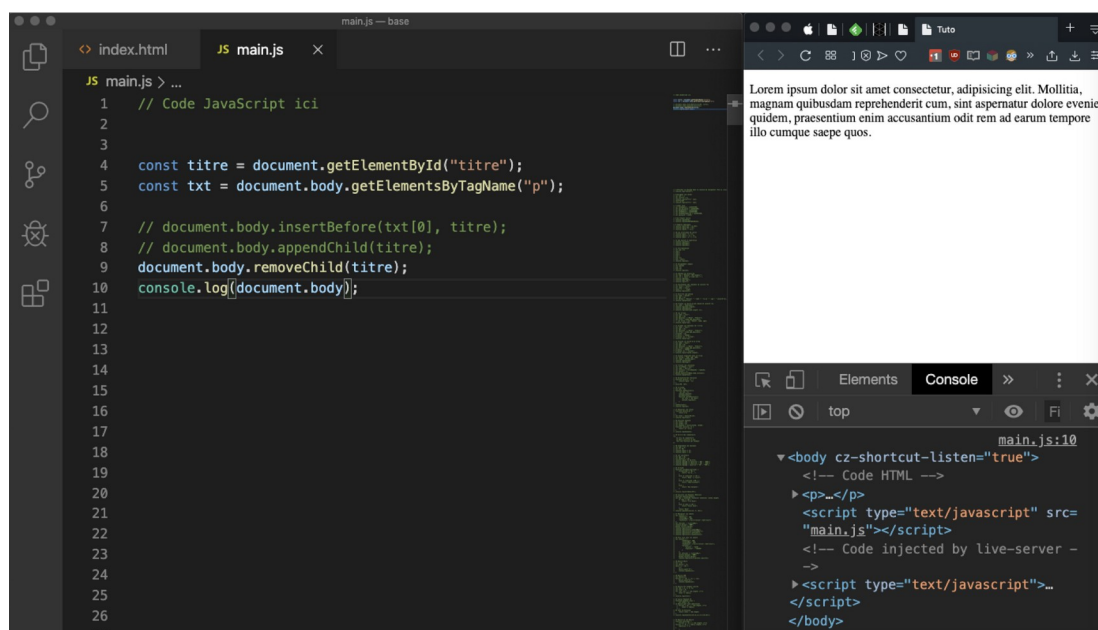
Ici en utilisant cette fonction sur document.body je vais placer le titre à la fin de la balise <body>

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



```

1 // Code JavaScript ici
2
3
4 const titre = document.getElementById("titre");
5 const txt = document.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

```

main.js:10

```

<body cz-shortcut-listen="true">
  <!-- Code HTML -->
  <p>...</p>
  <script type="text/javascript" src="main.js"></script>
  <!-- Code injected by Live-server -->
  <script type="text/javascript">...
</script>
</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

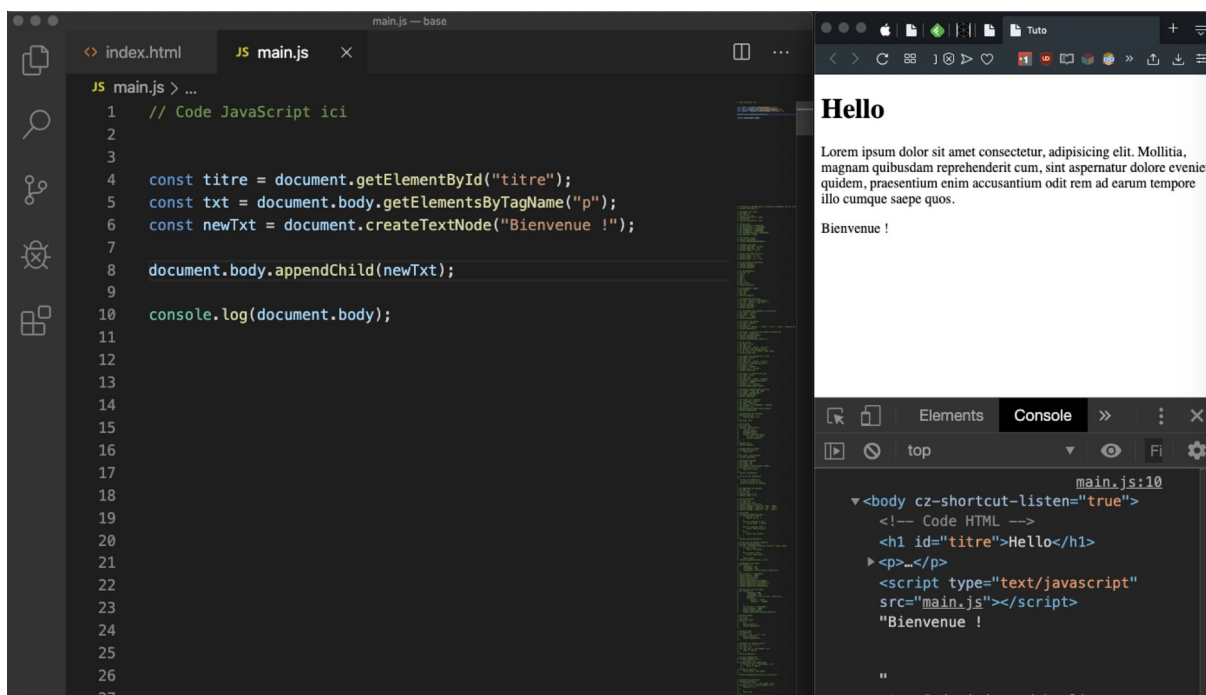
## DOM JS : Créer un élément Html

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.



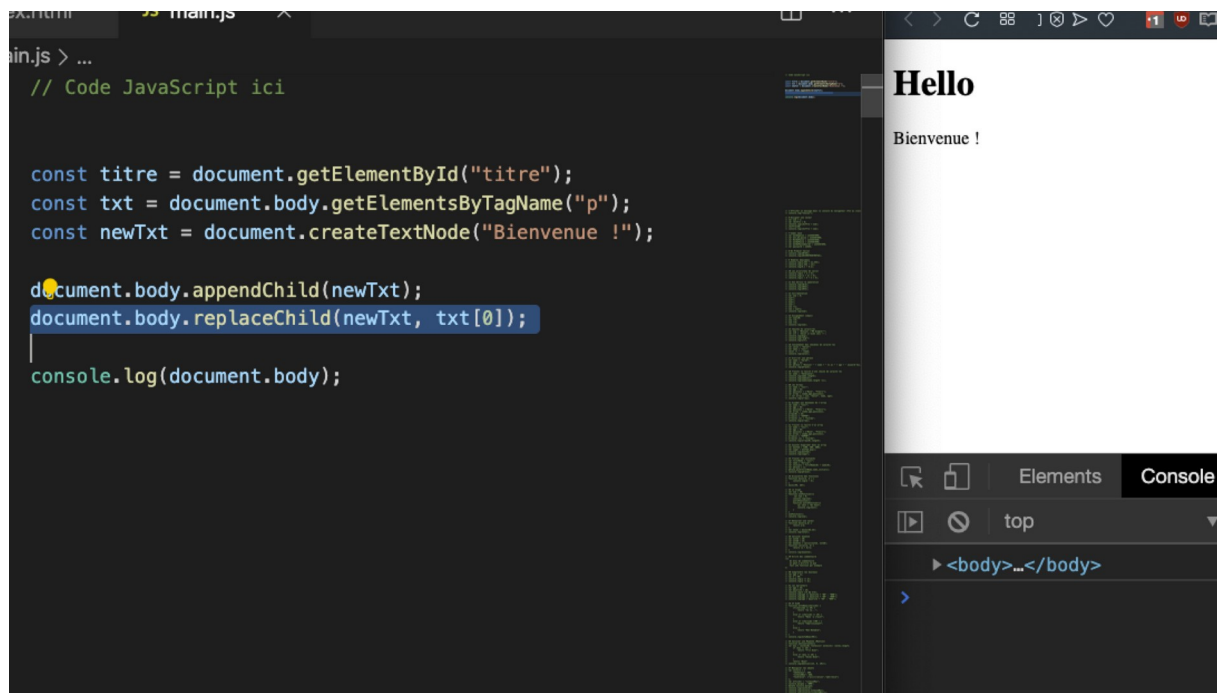
L6 on crée un nouvel élément de texte dans la variable newTxt, L8 on n'oublie pas de placer l'élément dans la page ici dans le <body>.

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.



Ici dans le body notre variable newText 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 pourrions 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");
```

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		



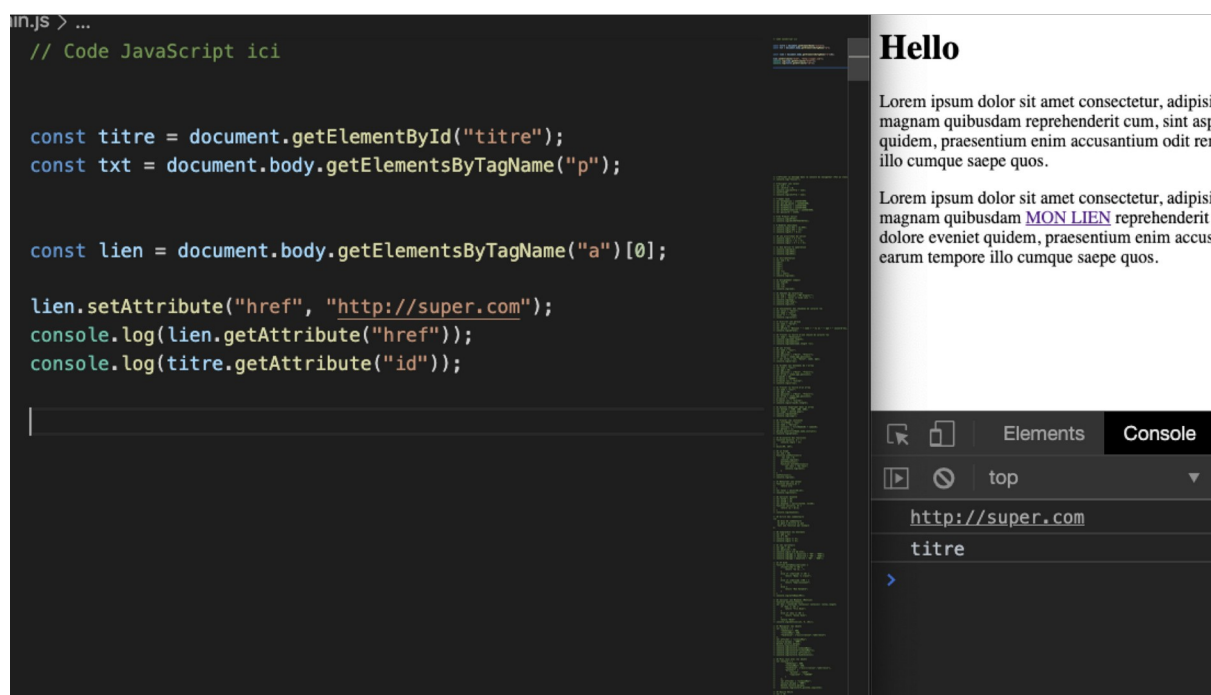
# JS

## 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 a web browser with a dark theme. The console on the left displays 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];

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

```

The browser's DOM tree on the right shows the following changes:

- The `href` attribute of the first `a` tag has been updated to `http://super.com`.
- The `id` attribute of the `h1` tag has been updated to `titre`.

The browser's content area displays the text "Hello" followed by two paragraphs of Lorem Ipsum text. The first paragraph contains the text "MON LIEN" which is highlighted in blue.

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		
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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.

## JS

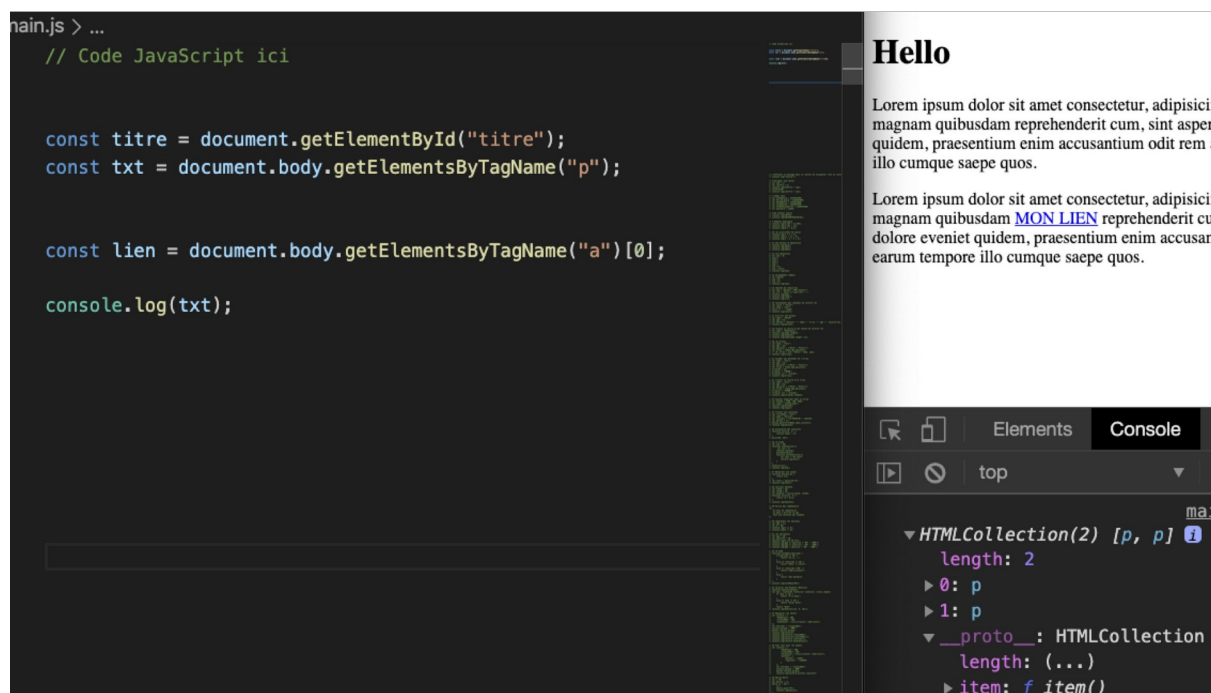
### DOM JS : Transformer une liste d'élément Html en tableau

#### 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()



The screenshot shows a web browser with a page titled "Hello". The page contains two paragraphs of Lorem ipsum text. The first paragraph is followed by a link labeled "MON LIEN". The browser's developer tools are open, showing the console with the following code and output:

```
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);
```

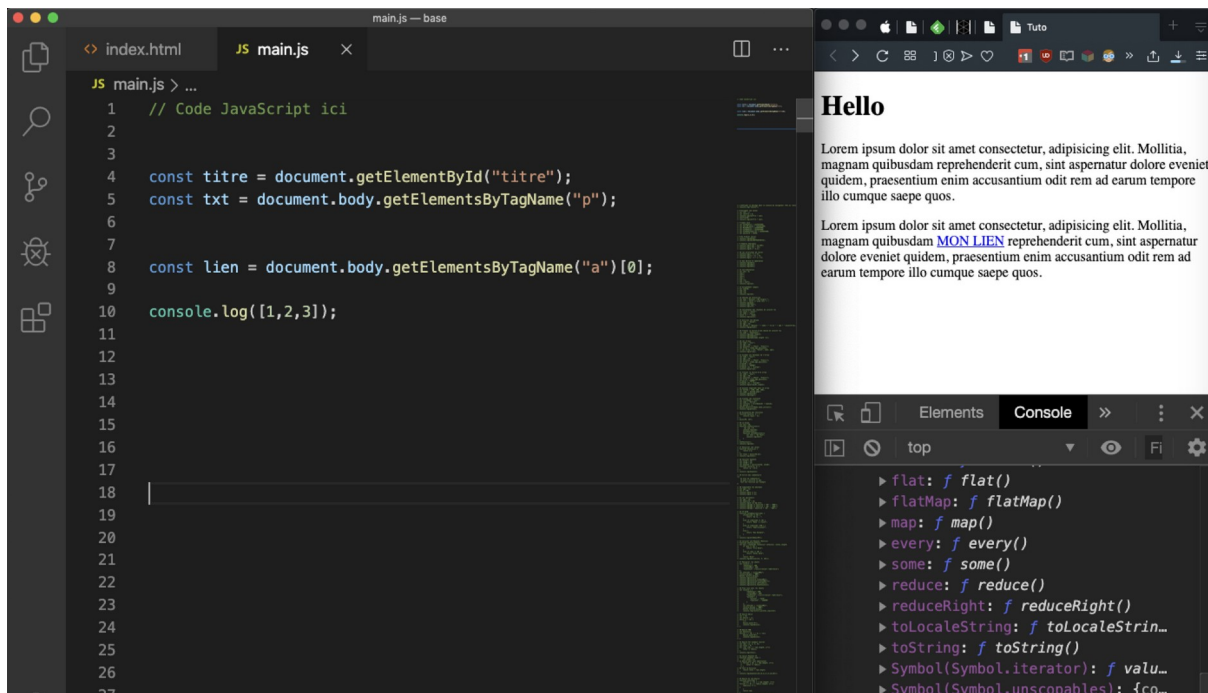
The console output shows an HTMLCollection of length 2, containing two paragraph elements (p).

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		
	Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT		
	Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM		

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.

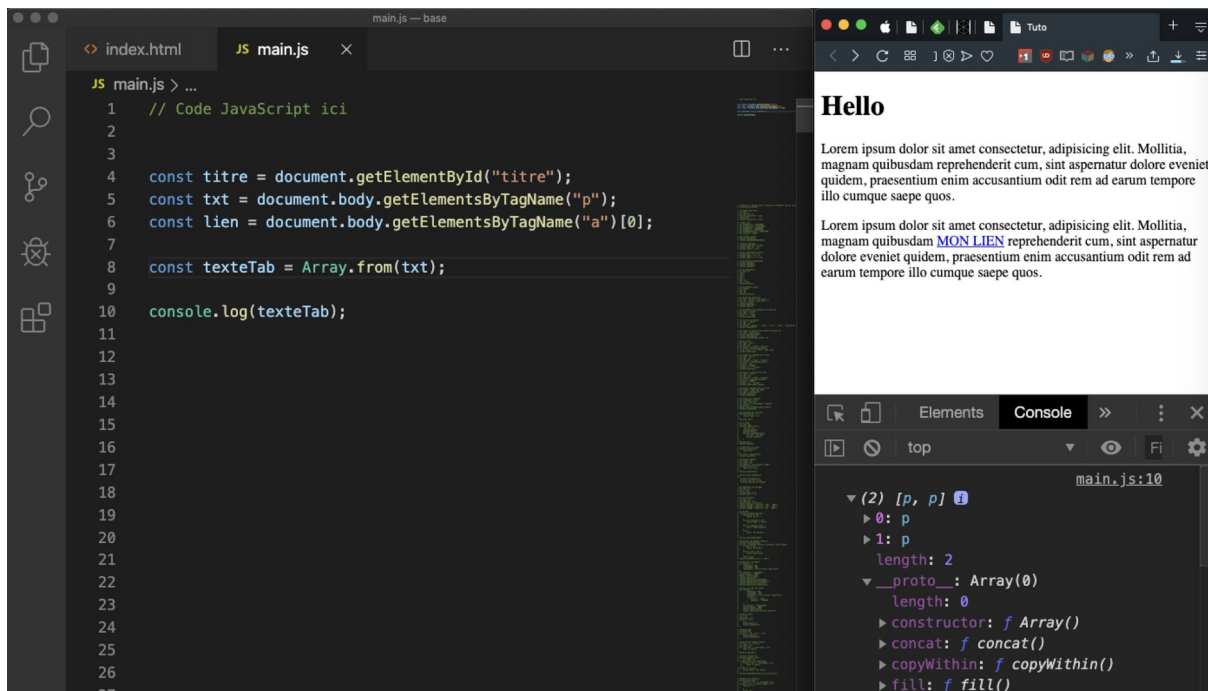
# JS



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



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);
```

# Hello

LOL HACKED

LOL HACKED

Exemple ou 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		



L'ECOLE  
REGIONALE DU  
NUMERIQUE



GRANDE  
ÉCOLE DU  
NUMÉRIQUE



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

### DOM JS : Sélectionner des éléments (alternative)

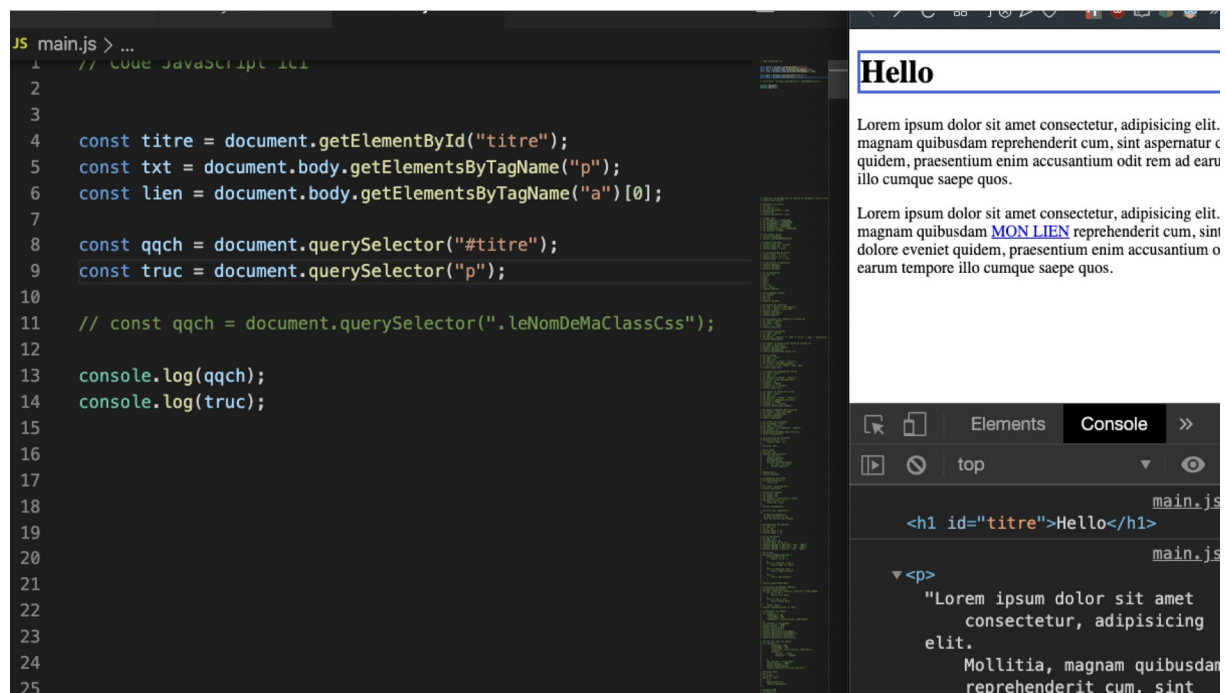
Nous avons déjà vu comment nous pouvons sélectionner des éléments Html via JS, avec les méthodes comme `getElementsByTagName()`, `getElementById()` etc...

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 selecteurs 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

```

Elements Console >>

top

main.js

```

<h1 id="titre">Hello</h1>

```

main.js

```

<p>
  "Lorem ipsum dolor sit amet
  consectetur, adipisicing
  elit.
  Mollitia, magnam quibusdam
  reprehenderit cum, sint

```

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.querySelector("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

```

### Hello

Lorem ipsum dolor sit amet consectetur, adi magnam quibusdam reprehenderit cum, sint quidem, praesentium enim accusantium odit illo cumque saepe quos.

2Lorem ipsum dolor sit amet consectetur, ad magnam quibusdam [MON LIEN](#) reprehendere eveniet quidem, praesentium enim ac earum tempore illo cumque saepe quos.

Elements Console

top

```

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

```

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		



L'ECOLE  
REGIONALE DU  
NUMERIQUE



GRANDE  
ÉCOLE DU  
NUMÉRIQUE



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			