



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

Table des matières

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

| 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

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 | 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

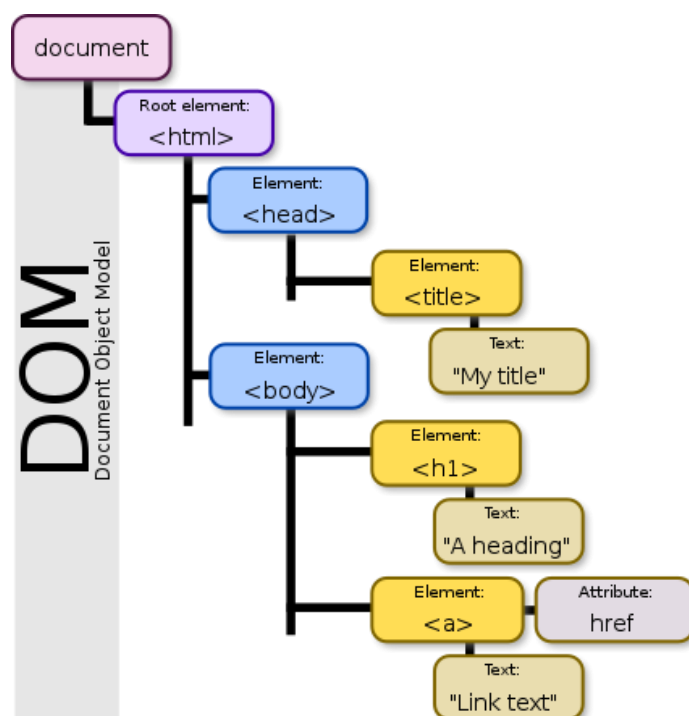
Le **Document Object Model (DOM)** : Le DOM est un modèle de document chargé dans le [navigateur](#). La représentation du document est un arbre nodal. Chaque nœud représente une partie du document (par exemple, un [élément](#), une chaîne de caractères ou un commentaire). (MDN : <https://developer.mozilla.org/fr/docs/Glossary/DOM>)

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...

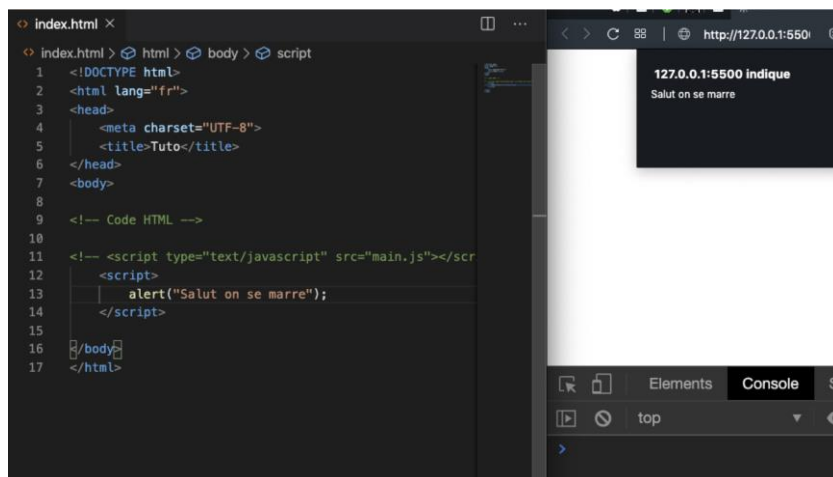


| Auteur(s) | Relu, validé et visé par : | Date de création : | Date dernière MAJ : |
|-----------|---|---|---------------------|
| | Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE | | |
| | Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX | Toute reproduction, représentation, diffusion 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 :

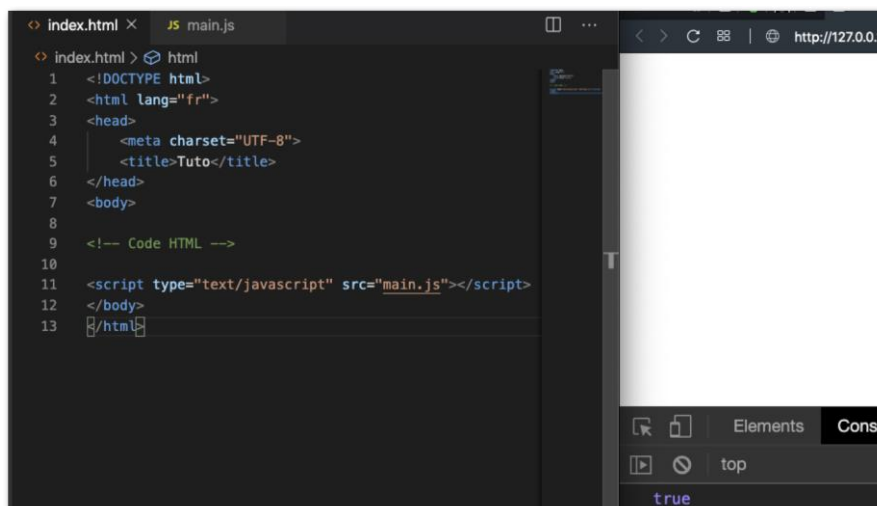
On peut coder du JS directement dans un fichier html entre des balises `<script></script>` :



```

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 recommandée sera d'avoir un fichier JS et de le relier au fichier html :



```

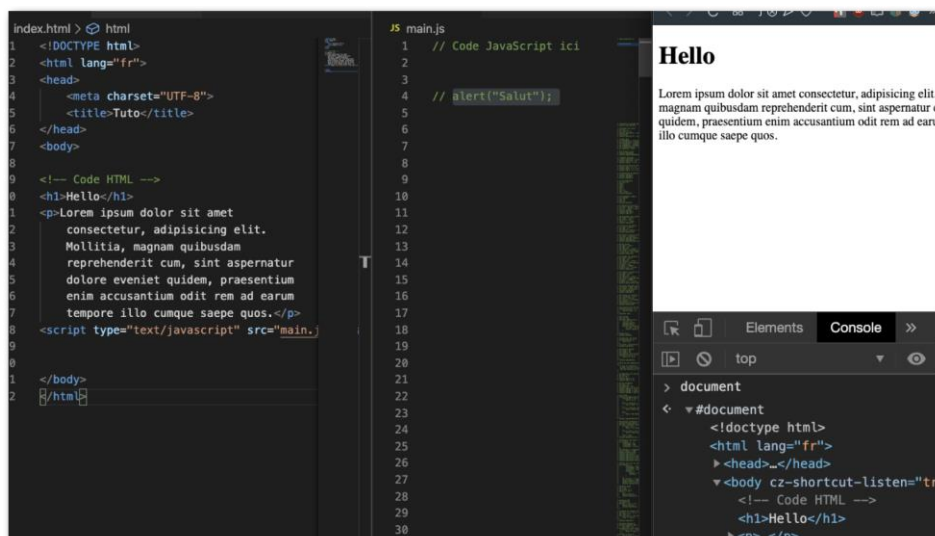
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Tuto</title>
6 </head>
7 <body>
8
9 <!-- Code HTML -->
10
11 <script type="text/javascript" src="main.js"></script>
12 </body>
13 </html>
  
```

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



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

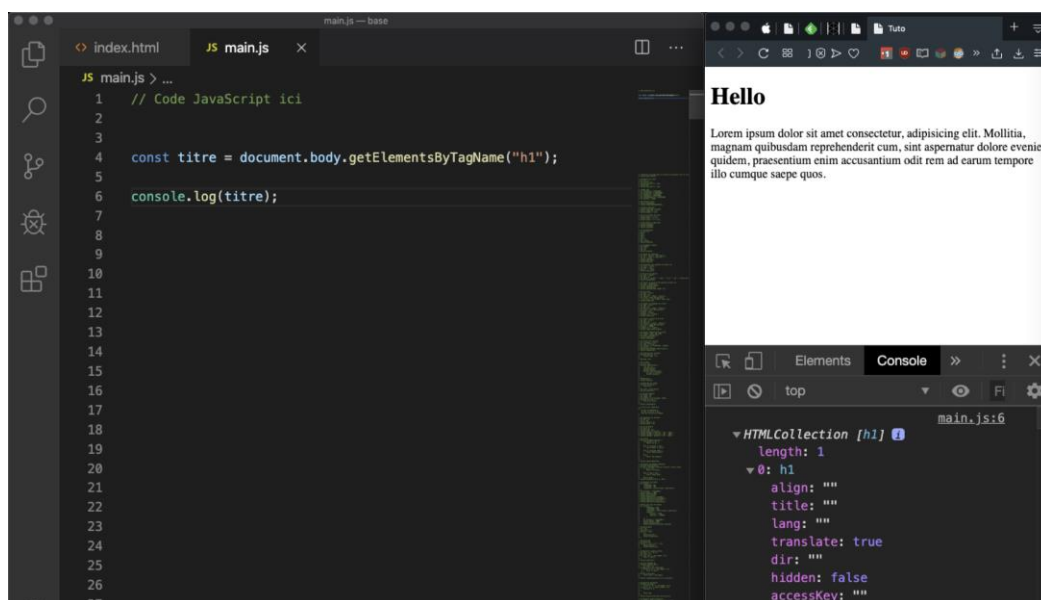
DOM JS : Sélectionner un élément Html

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

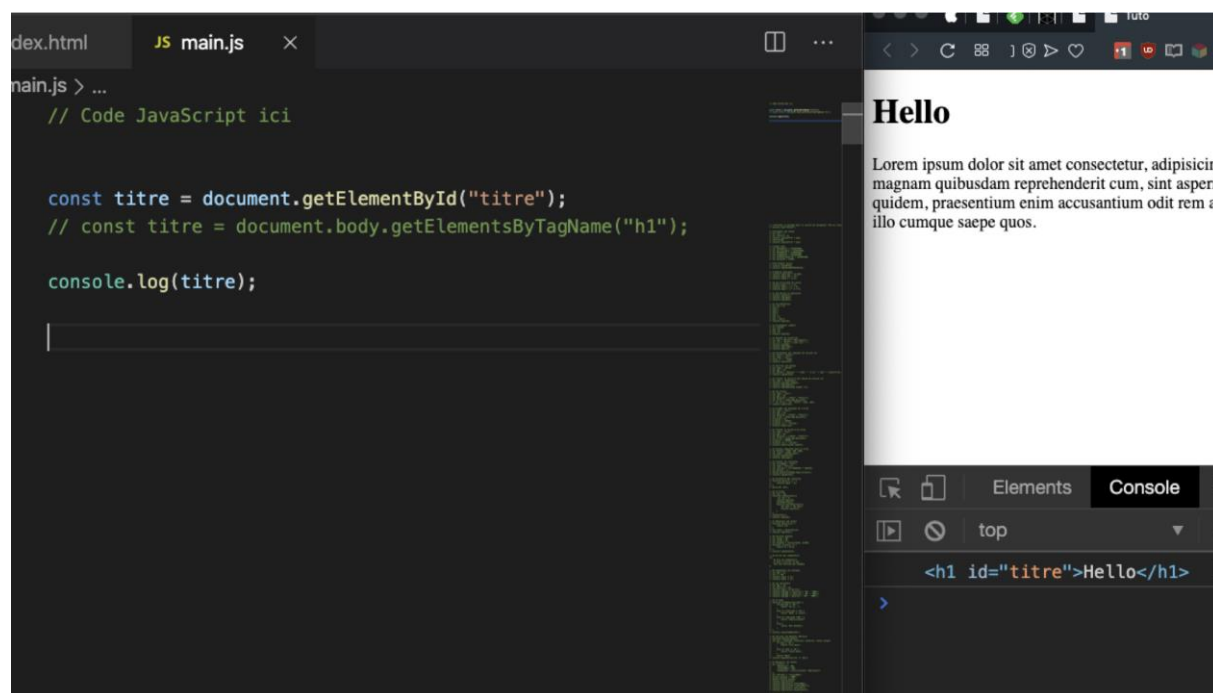
Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

JS

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 | | |
| | 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

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 | | |

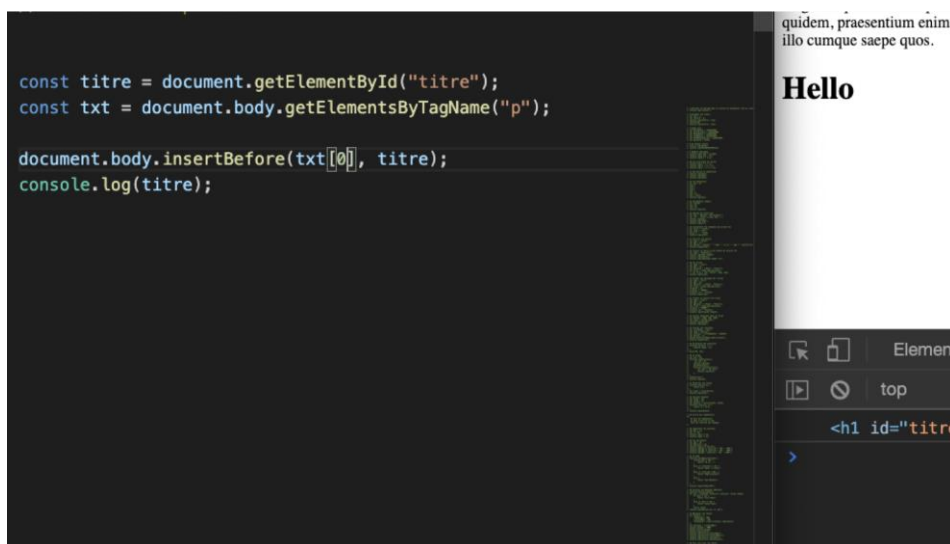
JS

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^{er} paragraphe avant le titre.

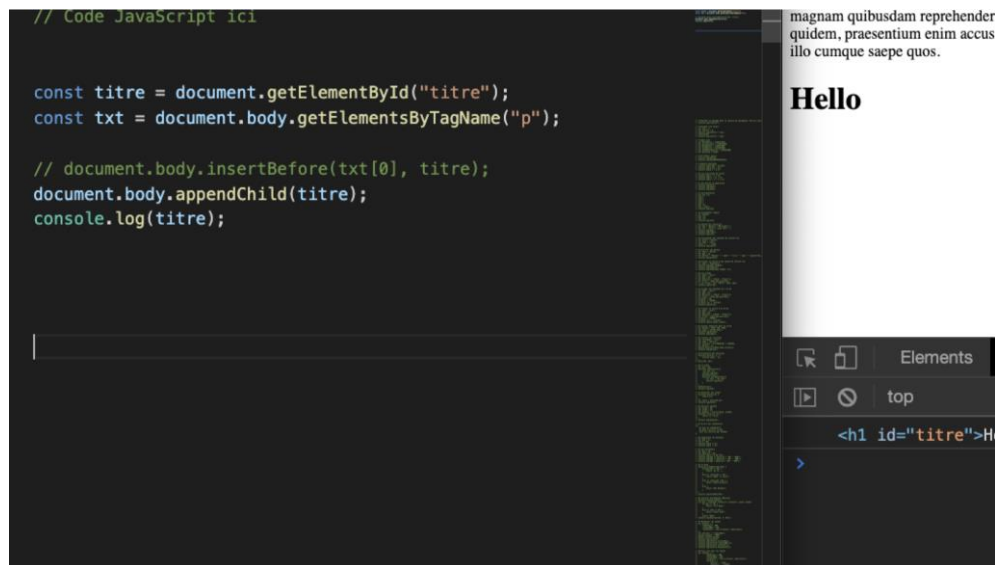
L'utilisation de cette fonction : on lui renseigne quel élément on veut placer (txt[0]) et avant quel éléments on veut l'insérer (titre)

| 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.



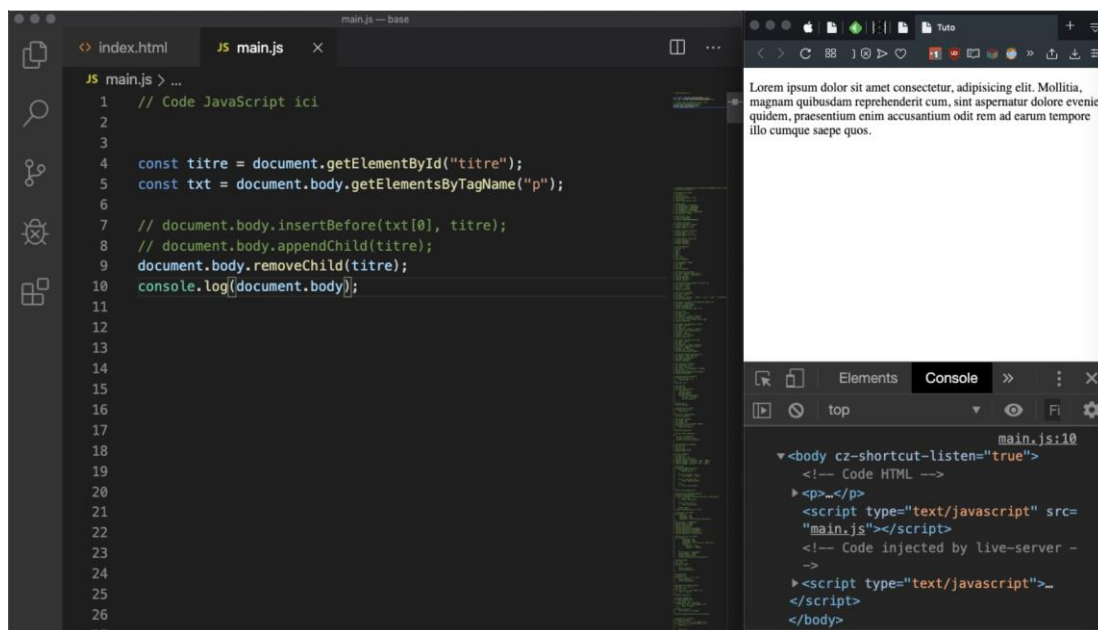
Ici en utilisant cette fonction sur document.body je vais placer le titre à la fin de la balise <body>

| 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



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

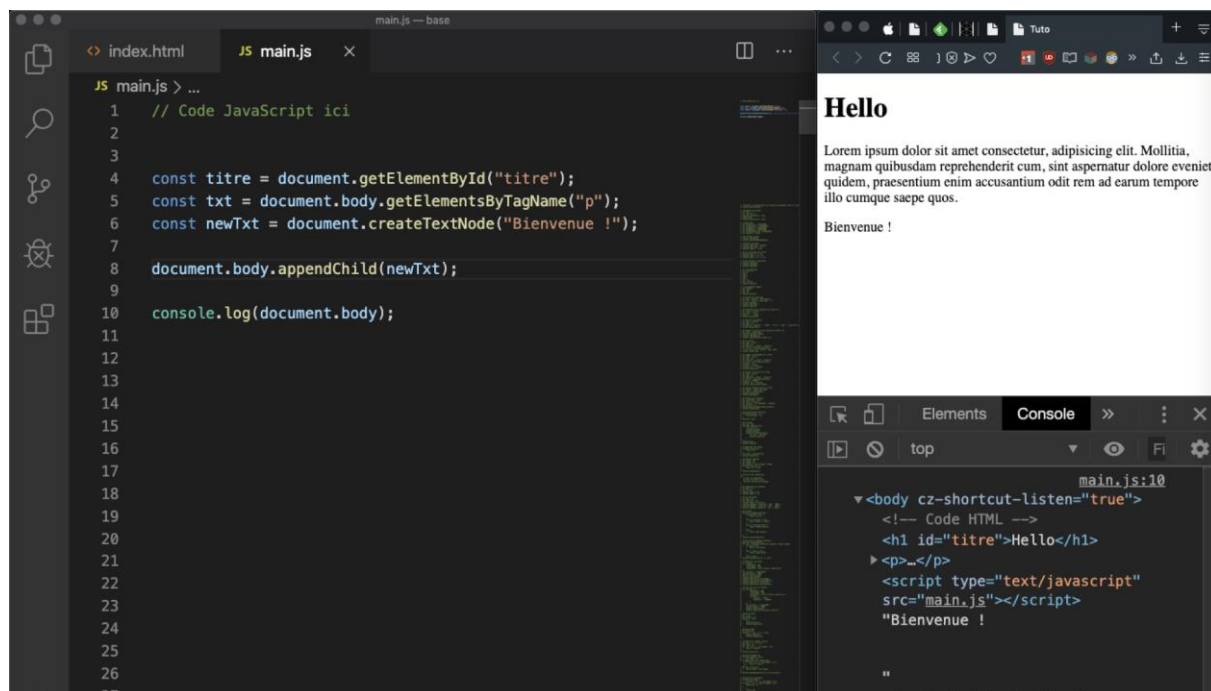
DOM JS : Créer un élément Html

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>.

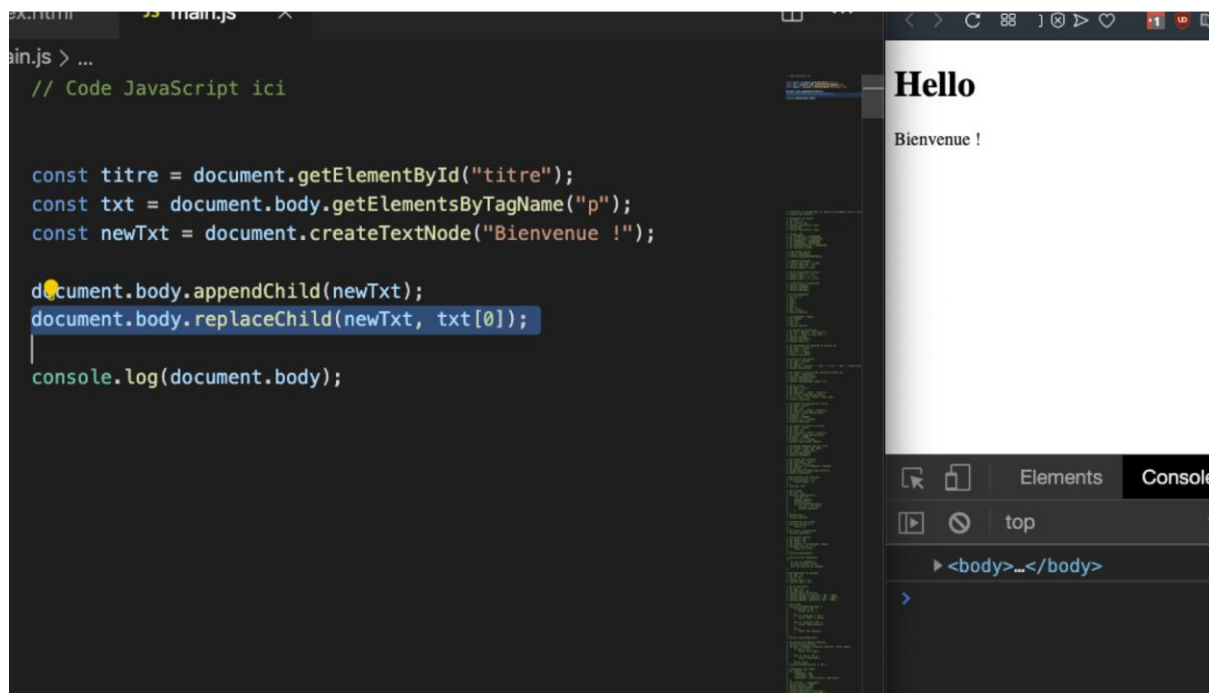
| Auteur(s) | Relu, validé et visé par : | Date de création : | Date dernière MAJ : |
|-----------|---|--------------------|---------------------|
| | Jérôme CHRETIENNE : Resp. Secteur NUMERIQUE OCCITANIE | | |
| | Florence CALMETTES : Coordinatrice Filière SYST. & RESEAUX | | |
| | Marc CECCALDI : Coordinateur 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

replaceChild()

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



Ici dans le body notre variable newText remplacera le 1^{er} 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 | | |
| | Marc CECCALDI : Coordinateur Filière DEVELOPPEMENT | | |
| | Sophie POULAKOS : Coordinatrice Filière WEBDESIGN / PPNUM | | |

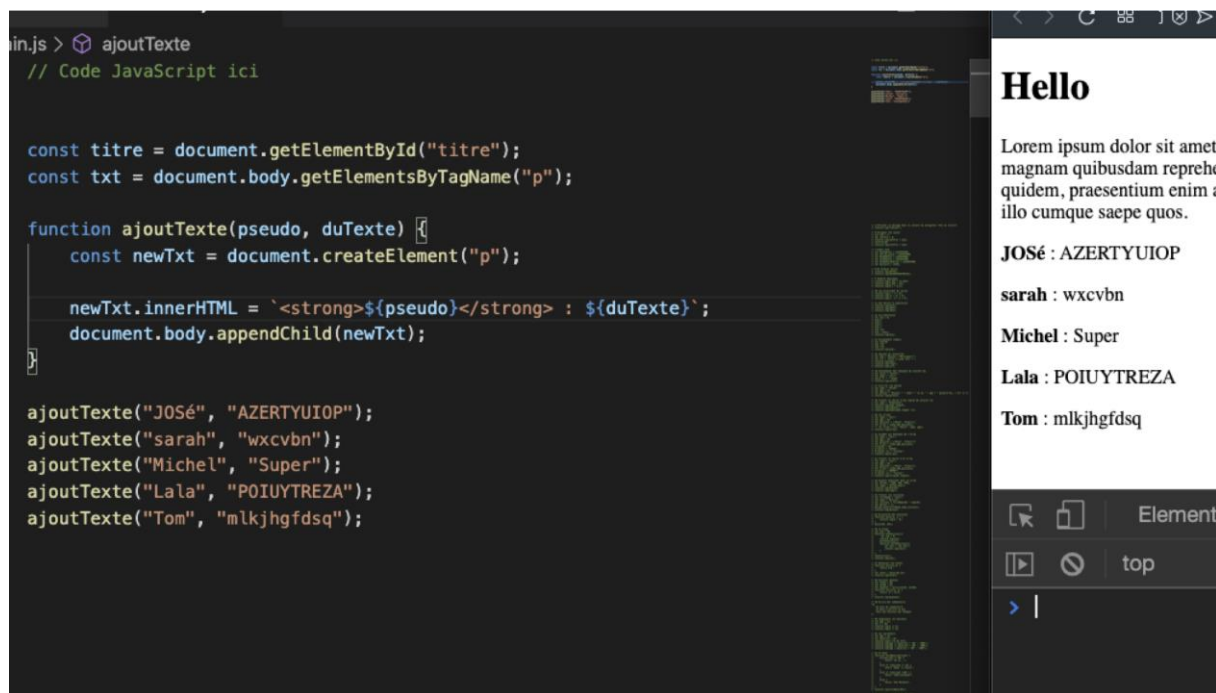
Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.

JS

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.



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 | | |
| | 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. | | | |

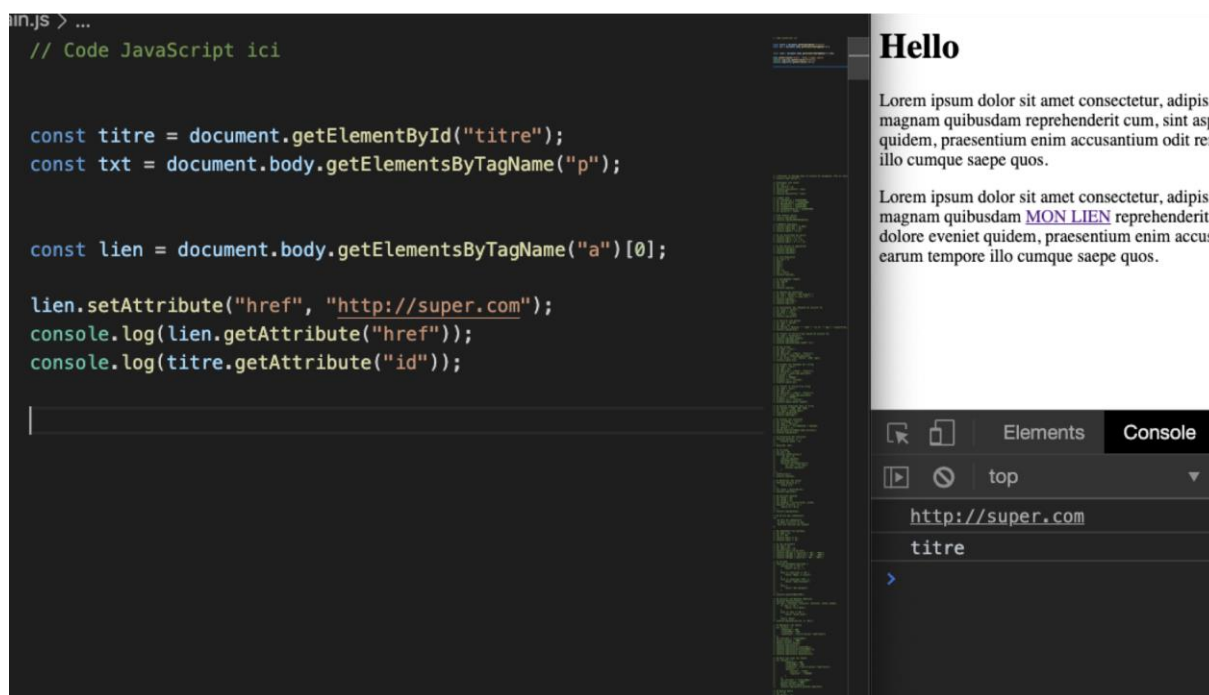
JS

DOM JS : 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 window shows a page with the heading "Hello" and two paragraphs of Lorem Ipsum text. The first paragraph contains a link "MON LIEN". The console output shows the href attribute of the link being set to "http://super.com" and the id attribute of the title being "titre".

Ici je récupère le 1^{er} 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 | | |