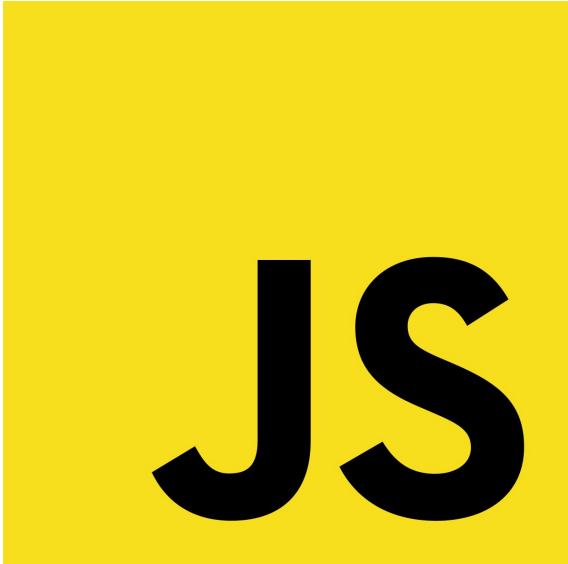




La Plateforme

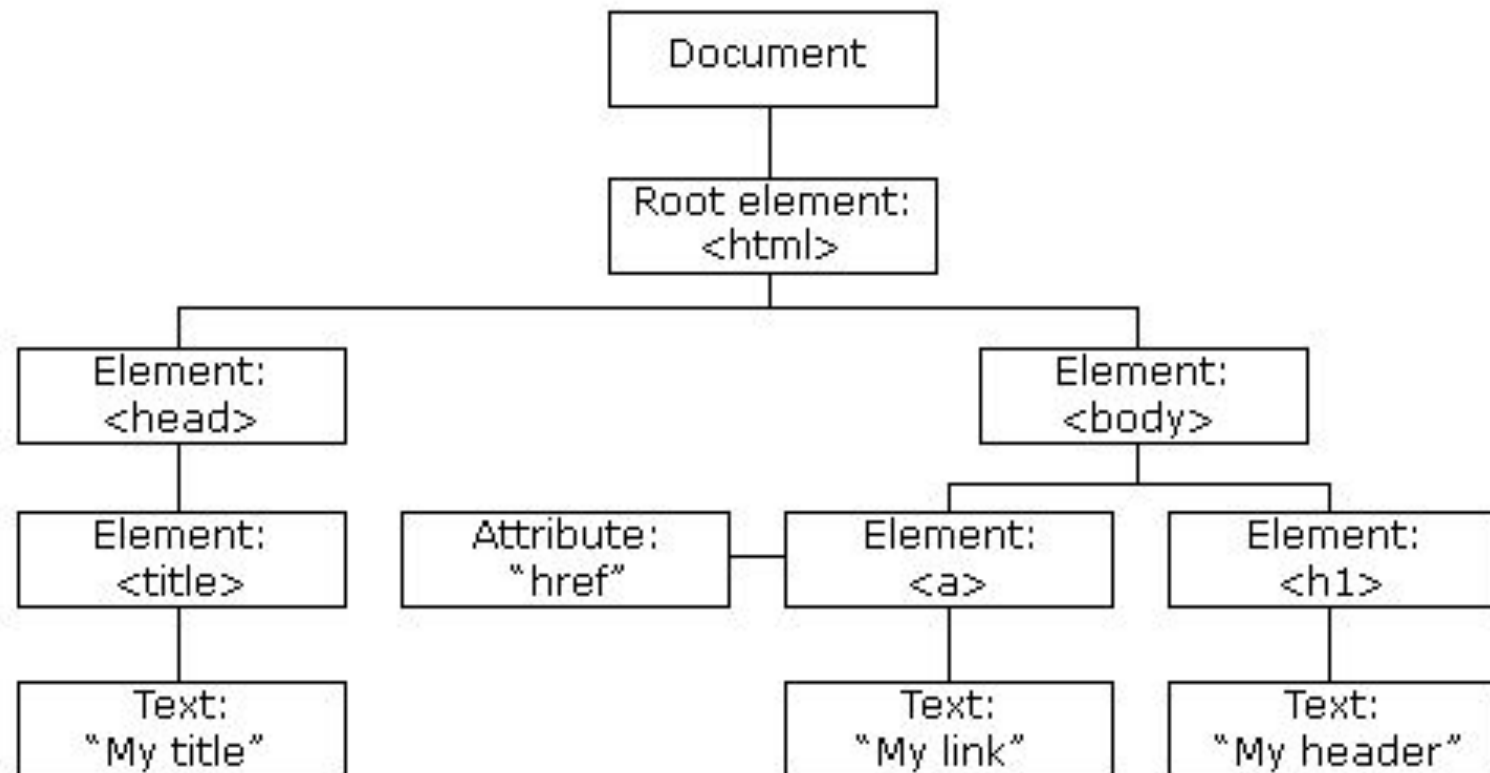
/ JAVASCRIPT - Jour 2 - JS ++



JS

- Notion de Document Object Model
- Le DOM doit être chargé avant toute modification de celui-ci.
- En JS on peut : créer des éléments, écouter des évènements, modifier des élément ciblés, modifier des attributs (classes..etc.

Qu'est ce que le DOM ?



La notion de DOM et chargement des scripts

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.

Si vous utilisez JavaScript pour manipuler des éléments du DOM, votre code ne fonctionnera pas si le JavaScript est chargé et analysé avant le code HTML sur lequel vous voulez opérer.



```
1 document.addEventListener("DOMContentLoaded", function() {  
2   ...  
3 });
```

Introduction aux évènements

Les événements sont des actions ou des occurrences qui se produisent dans le système que vous programmez et dont le système vous informe afin que vous puissiez y répondre d'une manière ou d'une autre si vous le souhaitez.

Il y a beaucoup de types différents d'événements qui peuvent se produire, par exemple :

- L'utilisateur clique avec la souris sur un certain élément ou en place le curseur sur un certain élément.
- L'utilisateur appuie sur une touche du clavier.
- L'utilisateur redimensionne ou ferme la fenêtre du navigateur.
- Une page web finissant de se charger.
- Un formulaire en cours de soumission
- Une vidéo en cours de lecture, en pause ou en fin de lecture.
- Une erreur qui survient.

Liste exhaustive des évènements : <https://developer.mozilla.org/fr/docs/Web/Events>

Exemple 1 :

Créer un élément

Etape. 1 - cibler un élément

HTML

```
1 <main class="container">
2 <section id="section1"></section>
3 </main>
```

JS

```
1 /* 1 - Je cible par son id l'élément <section></section> que j'ai crée sur l'index.php : */
2 const section1 = document.querySelector("#section1");
3 /* On peut aussi écrire : document.getElementById('section1') */
```


Etape. 2 - créer un élément title1 : <h1> et son contenu texte

JS

```
1 /* 2 - Je crée un nouvel élément <h1></h1> et je crée un texte dedans */  
2 const title1 = document.createElement("h1");  
3 title1.textContent = "Titre de la Section 1 inséré grâce à un.textContent";
```

Pour l'instant il n'est pas encore rattaché au DOM ...

Etape. 3 - j'ajoute l'élément title1 à l'élément section1

JS

```
1 /* 3 - J'ajoute ensuite cet élément à l'élément section1 précédemment crée' */  
2 section1.appendChild(title1);
```

HTML

```
1 <section id="section1">  
2     <h1>Titre de la Section 1 inséré grâce à un.textContent</h1>  
3 </section>
```

Exemple 2 :

Ecouter un évènement

Etape. 1 - cibler un élément

HTML

```
1 <button id="bouton">Cliquez ici !</button>
```

JS

```
1 const btn = document.querySelector("#bouton");
```

Etape. 2 - écouter l'évènement onClick sur le bouton

JS

```
1  /* 1 - Je vais écouter un évènement click sur le bouton et déclencher une fonction à cet évènement */  
2  btn.addEventListener("click", () => {  
3      console.log("click");  
4  });
```

Affichage dans la console à chaque click : "click"

Exemple 3 :

Modifier des attributs (href...)

Etape. 1 - cibler un élément

HTML

```
1 <!DOCTYPE html>↵
2 <html lang="en">↵
3 <head>↵
4   ...<meta charset="UTF-8">↵
5   ...<link id="theme" rel="stylesheet" href="style-light.css">
```

JS

```
1 const link = document.querySelector("#theme");
```

Etape. 2 - modifier l'attribut href pour charger un autre css

Ici je modifie cet attribut au “click” sur un bouton que j’ai précédemment ciblé et appelé “btn”

```
1 btnTheme.addEventListener("click", () => {  
2     link.href = "style-light.css";  
3 });
```


Et pleins d'autres possibilités
...

LIENS INCONTOURNABLES

https://www.w3schools.com/js/js_htmlDOM.asp

Liste exhaustive des évènements :

<https://developer.mozilla.org/fr/docs/Web/Events>

Questions ?