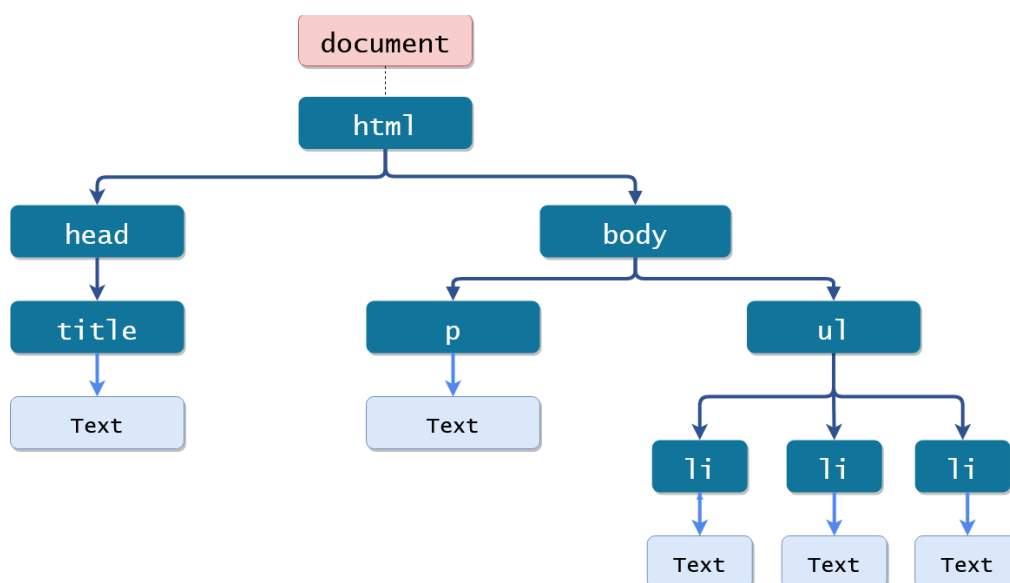


D.O.M

Présentation

Le DOM est donc une arborescence que va fabriquer le navigateur quand il interprète un fichier HTML, comme cité précédemment en JS tout est objet, et dans le DOM nous allons pouvoir retrouver tous les éléments HTML de notre page sous forme d'objets (ayants des propriétés) manipulables par JS.



Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Sélectionner des éléments du DOM

getElement(s)By

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

getElementsByName() - getElementsByClassName()

permet de sélectionner TOUS les éléments par le nom de leur balise (leur tag) html.

```

//? Une fonction type getElement pour récupérer tous les élément selon une certaine
//? balise dans une HTMLCollection
let tousLesP = document.getElementsByTagName('p');
console.log('La HTMLCollection', tousLesP);
//? Quand on a une HTMLCollection on peut accéder à un certains éléments
console.log('le 3e <p> dans la HTMLCollection : ', tousLesP[2]);
//? Une fonction type getElement pour récupérer tous les élément selon une certaine
//? class dans une HTMLCollection
let tousLesSuper = document.getElementsByClassName('super');
console.log(tousLesSuper);

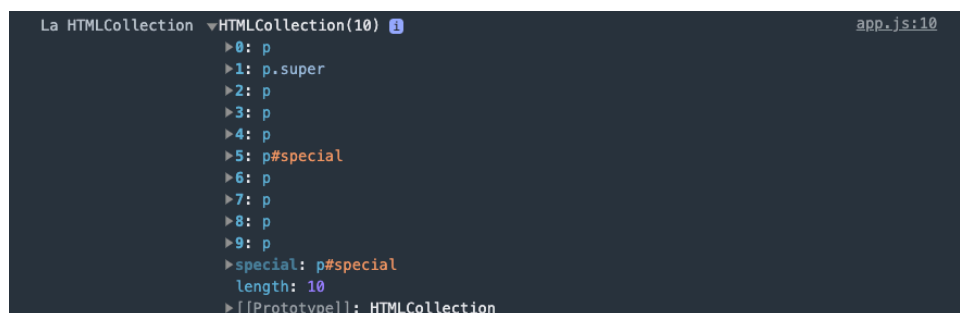
```

La console du navigateur nous affiche un tableau de type HTMLCollection, un tableau qui va contenir tous les paragraphe <p> de notre page. On a accès à toutes les propriétés de ces éléments Html.

(Son contenu, 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 paragraphe <p> de la page)

Défi : dans la console, trouver les propriétés dans lesquelles on retrouve ce que l'on a écrit dans les paragraphes.



Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



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.

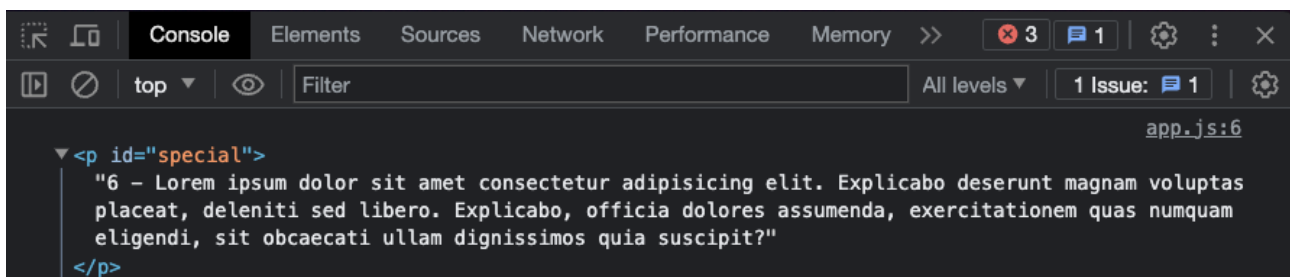
getElementById()

On peut aussi sélectionner 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 HTMLCollection.

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)

```
///? Une fonction type getElement pour récupérer UN élément par son ID
let specialP = document.getElementById('special');
console.log(specialP);
```



Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

querySelector()

Une alternative, les fonctions de type querySelector, qui vont se basées sur la syntaxe de css (rappel : Les selector en CSS)

querySelector()

```
/** Une fonction type querySelector pour récupérer UN élément (le 1er trouvé)
let lePremierP = document.querySelector('p');
console.log('lePremierP via querySelector : ',lePremierP);
```

↓ sélection par l'id (avec # comme en css)

```
/** Une fonction type querySelector pour récupérer UN élément par son ID
let pSpecial = document.querySelector('#special');
console.log('pSpecial querySelector + ID',pSpecial);
```

↓ Sélection par la classe (avec . comme en css)

```
/** Une fonction type querySelector pour récupérer UN élément (le 1er trouvé) par sa
classe
let pSuper = document.querySelector('.super');
console.log('pSuper querySelector + class',pSpecial);
```

querySelectorAll()

Pour récupérer plusieurs éléments (par le nom de balise ou par leurs class css) dans une NodeList

```
/** Une fonction type querySelector pour récupérer TOUS les élément dans une NodeList
let allParagraphes = document.querySelectorAll('p');
console.log('allParagraphes querySelector + balise',allParagraphes);
let allSuper = document.querySelectorAll('.super');
console.log('allSuper querySelector + class',allSuper);
```

```
allSuper querySelector + class ▼NodeList(2) [h2.super, p.super] ⓘ
  ►0: h2.super
  ►1: p.super
  length: 2
  ►[[Prototype]]: NodeList
```

<https://github.com/jefff404/cours-js/tree/20-dom>

Auteur :

Jean-François Pech

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
☑ Sophie POULAKOS
☑ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Placer des éléments du DOM

Une fois que l'on arrive à sélectionner des éléments HTML (toute la page, le body, un titre, une div etc...) on va pouvoir utiliser des fonctions prévues pour gérer l'insertion / le placement de ces éléments dans la page.

insertBefore()

Cette fonction s'utilise sur un élément HTML sélectionné (généralement une <div>, un <form>, une , le <body>, etc...), prend en compte 2 éléments Html en paramètre pour placer l'un avant l'autre

```

//! Placer des elements dans une page web
//? Une fonction type querySelector pour récupérer UN (le 1er trouvé) élément par la
NodeList
let allParagraphes = document.querySelectorAll('p');
let laDiv = document.querySelector('.vide');
let premierH1 = document.querySelector('h1');
//! insertBefore, on selectionne 2 éléments pour placer l'un avant l'autre
document.body.insertBefore(allParagraphes[9],premierH1);

```

Page sans JS :

D.O.M (Placer des éléments)

Document Object Model

- 1 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 2 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 3 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 4 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 5 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

Page avec JS :

10 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

D.O.M (Placer des éléments)

Document Object Model

- 1 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 2 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 3 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?
- 4 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



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.

append() / appendChild()

Généralement on va plutôt utiliser append ou appendChild pour placer un element à la fin d'un autre (placer une div à la fin du body, placer un titre à la fin d'une div, etc...)

Pour l'exemple on va créer une div vide (donc elle ne s'affiche pas de base) avec du style

```
<div class="vide" style="background-color: midnightblue; color: beige;"></div>
```

```
laDiv.append(`Là c'est JS qui ajoute du texte dans la div`);  
// Append plutôt pensé pour ajouter du contenu à la volé au format string  
// si on a crée ou sélectionné un élément que l'on veut placer : ceci peut marcher  
laDiv.append(allParagraphes[4]);  
// Mais on a aussi la fonction appendChild;  
laDiv.appendChild(allParagraphes[0]);
```

Là c'est JS qui ajoute du texte dans la div

5 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

1 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

D.O.M (Placer des éléments)

Document Object Model

2 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

3 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores assumenda, exercitationem quas numquam eligendi, sit obcaecati ullam dignissimos quia suscipit?

4 - Lorem ipsum dolor sit amet consectetur adipisicing elit. Explicabo deserunt magnam voluptas placeat, deleniti sed libero. Explicabo, officia dolores

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

removeChild()

Une fonction qui permet de retirer UN élément HTML du DOM

```
// //! On peut aussi supprimer un élément du DOM  
document.body.removeChild(allParagraphes[9]);
```

Ici on sur le body on supprime le 10^è paragraphe

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Créer - Modifier - Supprimer des éléments du DOM

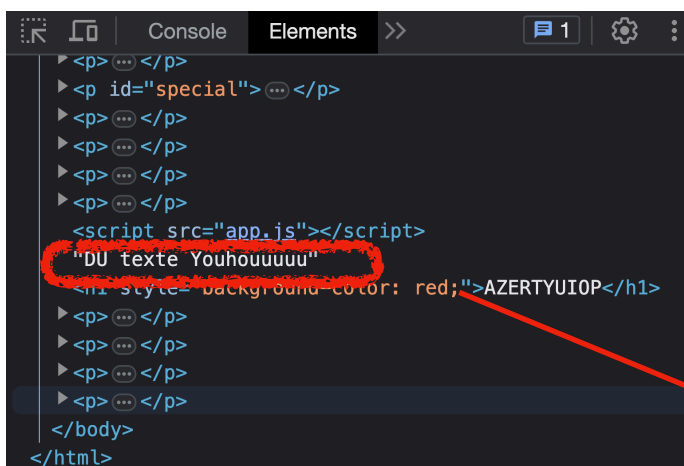
createTextNode()

Une fonction pour créer du texte brut dans une page web

```

//! Créer des éléments de texte
const newTxt = document.createTextNode('DU texte Youhouuuuu');
document.body.append(newTxt); //!Créer c'est bien mais il faut
placer
  
```

Dans l'inspecteur :



10 - Lorem ipsum dolor sit amet
 consectetur adipisicing elit.
 Explicabo deserunt magnam
 voluptas placeat, deleniti sed libero.
 Explicabo, officia dolores
 assumenda, exercitationem quas
 numquam eligendi, sit obcaecati
 ullam dignissimos quia suscipit?

DU texte Youhouuuuu

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

createElement()

Cette fonction permet de créer n'importe quel élément HTML en précisant le nom de sa balise (le tag Name)

Ici on crée un titre h1, on modifie quelques propriétés et on le place dans la page

```
#!/Cr  er n'importe quel element HTML
const newH1 = document.createElement('h1');//phase 1 creation
newH1.innerText = "AZERTYUIOP";//phase2 remplissage
newH1.style.backgroundColor = 'red';
document.body.append(newH1);//phase 3 on place dans la page
```

AZERTYUIOP

⚠ Ne pas oublier de placer les   l  ments cr   s dans la page web 🧐

Auteur :

Jean-Fran  ois Pech

Relu, valid   & vis   par :

☒ J  r  me CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date cr  ation :

03/03/2023

Date r  vision :

10/03/2023



Toute reproduction, repr  sentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque proc  d   que ce soit est interdite sans l'autorisation expresse,   crite et pr  alable de l'ADRAR.

Exercice : Créer des éléments - Afficher un profil utilisateur

Nous avons cet objet userData et via JS on doit l'afficher dans la page web

```

//! EXO 20.1
//TODO: via JS afficher le profil utilisateur dans la page web
const userData = {
  name: 'John delavega',
  email: 'john.doe@example.com',
  age: 25,
  dob: '08/02/1989',
  active: true,
  img: 'https://www.boredpanda.com/blog/wp-content/uploads/2022/06/funny-low-cost-
  cosplay-pics-62a744d39c80a__700.jpg'
};
  
```



Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Solution Possible

```
// JS qui va customiser la div du profile utilisateur
let divUser = document.querySelector('.userProfile');
divUser.style.backgroundColor = `background-color: #4158D0`;
divUser.style.backgroundImage = `linear-gradient(43deg, #4158D0 0%, #C850C0 46%, #FFCC70 100%)`;
divUser.style.color = `white`;
divUser.style.width = '500px';
divUser.style.margin = 'auto';
divUser.style.padding = '2rem';
//JS crée une image, renseigne la src , modif taille
const imgTemplate = document.createElement('img');
imgTemplate.src = userData.img;
imgTemplate.style.height = '500px';
imgTemplate.style.width = '500px';
divUser.append(imgTemplate);
// JS crée le titre du name
const nameTemplate = document.createElement('h1');//phase 1 creation
nameTemplate.innerText = userData['name'];
divUser.append(nameTemplate);
// JS crée le titre du email
const emailTemplate = document.createElement('h2');//phase 1 creation
emailTemplate.innerText = userData.email;
divUser.append(emailTemplate);
// JS crée le titre du age
const ageTemplate = document.createElement('h2');//phase 1 creation
ageTemplate.innerText = userData.age;
divUser.append(ageTemplate);
// JS crée le titre du dob
const dobTemplate = document.createElement('h2');//phase 1 creation
dobTemplate.innerText = userData.dob;
divUser.append(dobTemplate);
// JS crée le titre du active
const activeTemplate = document.createElement('h2');//phase 1 creation
activeTemplate.innerText = userData.active;
divUser.append(activeTemplate);
```

🤔 À optimiser avec une ou plusieurs fonctions ???????

Auteur :

Jean-François Pech

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
☑ Sophie POULAKOS
☑ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Exercice : Créer des éléments - fonction d'ajout de texte

```
#!/ EXO 20.1  
//TODO: Créer une f° ajouterTexte qui prend 2 paramètres : pseudo et duTexte  
//TODO: La fonction a pour but :  
//TODO: de créer puis remplir et enfin placer un paragraphe contenant pseudo et  
duTexte, dans la page  
// TODO (Bonus) : Dans le paragraphe le pseudo est affiché en gras
```

exemple d'utilisation :

```
ajouterTexte('Daniel', 'Gracia');  
ajouterTexte('Jarry', 'Borne');  
ajouterTexte('JCVD', 'OK');  
ajouterTexte('Dongue', 'Rodrigue');
```

Dans la page web :

Daniel - Gracia

Jarry - Borne

JCVD - OK

Dongue - Rodrigue

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Solution possible

```
function ajouterTexte(unPseudo, duTexte){  
  const nouveauTexte = document.createElement('p');  
  nouveauTexte.innerHTML = `<strong>${unPseudo}</strong> - ${duTexte}`;  
  document.body.append(nouveauTexte);  
};
```

Auteur :

Jean-François Pech

Date création :

03/03/2023

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date révision :

10/03/2023



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.

Modifier les attributs des éléments du DOM

Accéder aux attributs

Modifier attribut style

Comme pour les objets, une fois qu'on a sélectionné ou crée un élément HTML, on va pouvoir accéder à ses propriétés, certaines de ces propriétés vont nous permettre de modifier des attributs HTML.

Par exemple on peut accéder à l'attribut style des éléments et ensuite accéder à toutes les propriétés Css (! les propriétés Css ont une syntaxe en camelCase en JS)

```
let firstTitle = document.querySelector('h1');  
console.log(firstTitle);  
  
firstTitle.style.backgroundColor = 'blue';  
firstTitle.style.color = 'beige';
```

D.O.M (Modif Attributs)

Et si on vérifie via l'inspecteur du navigateur :

```
<h1 class="laClasse" style="background-color: blue; color: beige;">  
D.O.M (Modif Attributs)</h1> == $0
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Modifier attribut class

Un autre attribut utile pour pouvoir gérer dynamiquement le style des éléments, c'est l'attribut `className` dans lequel on renseigne le nom d'une classe CSS que l'on veut ajouter à l'élément. Admettons que nous avons déjà préparé une classe CSS

```
.laClasse{  
    color: chartreuse;  
}
```

```
let firstTitle = document.querySelector('h1');  
console.log(firstTitle);  
  
firstTitle.className = 'laClasse';
```

D.O.M (Modif Attributs)

Dans l'inspecteur du navigateur :

```
<h1 class="laClasse">D.O.M (Modif Attributs)</h1>
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Fonction classList

Une autre option pour pouvoir gérer les classes c'est de passer par la propriété classList, qui donne accès à principalement trois fonction natives permettant d'ajouter, de supprimer ou de toggle des class CSS sur un élément HTML. (Il ya aussi une fonction replace pour remplacer des classes)

Bien sûr cela sous entend que nous avons déjà préparé les classes CSS.

Modifier ou créer n'importe quel attribut

Avec la fonction `setAttribute()` qui s'utilise sur un élément HTML, on va pouvoir lui renseigner 2 paramètres, en premier quel attribut on veut modifier ou créer (si l'attribut existe déjà côté html ce dernier est modifier sinon il est crée), cette fonction permet aussi (dans des cas spécifiques) de créer les attributs de votre choix (beaucoup de framework on des fonctionnalités basés sur des attribut spécifique côté template HTML.

Dans l'exemple ci-dessous on part du principe qu'on a une classe css nommée « democlass »

```
element.setAttribute("class", "democlass");
```

Ici exemple dans le cas ou dans notre variable element (admettons nous avons sélectionné un lien <a>) on pourrait modifier son attribut **href**

```
element.setAttribute("href", "www.google.com");
```

Inspecter le D.O.M via le navigateur pour vérification

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Gérer les évènements du DOM

Pour pouvoir prétendre à des application totalement « réactive » il va nous falloir cette dernière étapes à savoir, le fait de pouvoir réagir lorsque l'utilisateur déclenche un évènement dans la page. (Le click sur un bouton, le fait de scroller ou encore envoyer un formulaire).

La fonction **addEventListener** qui prend 2 arguments en paramètres, une chaine de caractère pour mentionner quel évènement on doit surveiller (click, scroll, etc...) et en second paramètre, une fonction qui va réunir plusieurs instructions.

```
/** Mode f° => (anonyme + fléchée)
let leH1 = document.querySelector('#mainTitle');

leH1.addEventListener('click',()=>{
    console.log('ok ca click');
});
```

Dans l'exemple ci-dessus on sélectionne un titre h1 via son id, ensuite sur cet élément HTML on écoute le click et en réaction on fait un affichage en console. (Ici on utilise une fonction fléchée en second paramètre)

```
/** Mode fonction anonyme classique
leH1.addEventListener('click',function(){
    console.log('ok ca click');
});
```

Ci-dessus avec une syntaxe classique de fonction mais également anonyme (généralement on utilise des fonctions anonymes dans les addEventListener, on a pas vraiment besoin de rappeler ces fonctions dans d'autres parties du programme), on peut aussi exécuter une fonction déclarée ailleurs dans le programme comme ci-dessous ↓

```
// ? la fonction est en dehors
function reactClick(){
    console.log('ok ca click');
}

leH1.addEventListener('click',reactClick());
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Exercice : réagir au click

Dans une page web mettre en place un titre h1, faire en sorte que lorsqu'on click sur le titre cela modifie son texte

D.O.M Events

Une fois cliqué :



Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Solution possible

```
let leH1 = document.querySelector('#mainTitle');
let selectTitle = false; //! Un booléen pour savoir si le titre a été cliqué ou non
leH1.addEventListener('click',()=>{
  console.log('ok ca click');
  leH1.innerText = selectTitle ? '-- 🤔 --': 'D.O.M Events'; //? condition ternaire si
  selectTitle est vrai alors smiley sinon dom Events
  selectTitle = !selectTitle; // ? à chaque click le booléen passe à son inverse
  (pour faire le re click)
});
```

Auteur :

Jean-François Pech

Relu, validé & visé par :

☑ Jérôme CHRETIENNE
 ☑ Sophie POULAKOS
 ☑ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.

Exercice réagir au click 2

Dans une page web mettre en place 1 titre et trois liens ou boutons
En css créer une classe

En utilisant les principales fonction de classList ,via JS, faire en sorte que le premier lien AJOUTE la classe css au titre de la page, le second lien SUPPRIME la classe et le troisième lien fais un TOGGLE de la classe sur le titre.

D.O.M Events

[Ajouter Classe](#) [Suppr Classe](#) [Toggle Classe](#)

Avec la classe appliquée

D.O.M Events

[Ajouter Classe](#) [Suppr Classe](#) [Toggle Classe](#)

Auteur :

Jean-François Pech

Relu, validé & visé par :

☒ Jérôme CHRETIENNE
☒ Sophie POULAKOS
☒ Mathieu PARIS

Date création :

03/03/2023

Date révision :

10/03/2023



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.