

# Javascript

## DOM

# Plan

- Définition du DOM
- Naviguer dans le document
  - Structure du DOM
  - Accéder aux éléments
    - getElementById
    - getElementsByName
- Accéder au contenu des éléments et le modifier
  - innerHTML
  - innerText
  - textContent
  - Value
- Certaines propriétés des éléments d'un formulaire

# Définition du DOM

- Le **Document Object Model** (abrégé **DOM**) est une interface qui offre aux langages de programmation, comme le `javaScript`, la possibilité de manipuler un document à balises (HTML et XML).
- La manipulation consiste à accéder aux éléments du document pour les **lire** ou pour la **modification** du **contenu** du document et de **son style**.

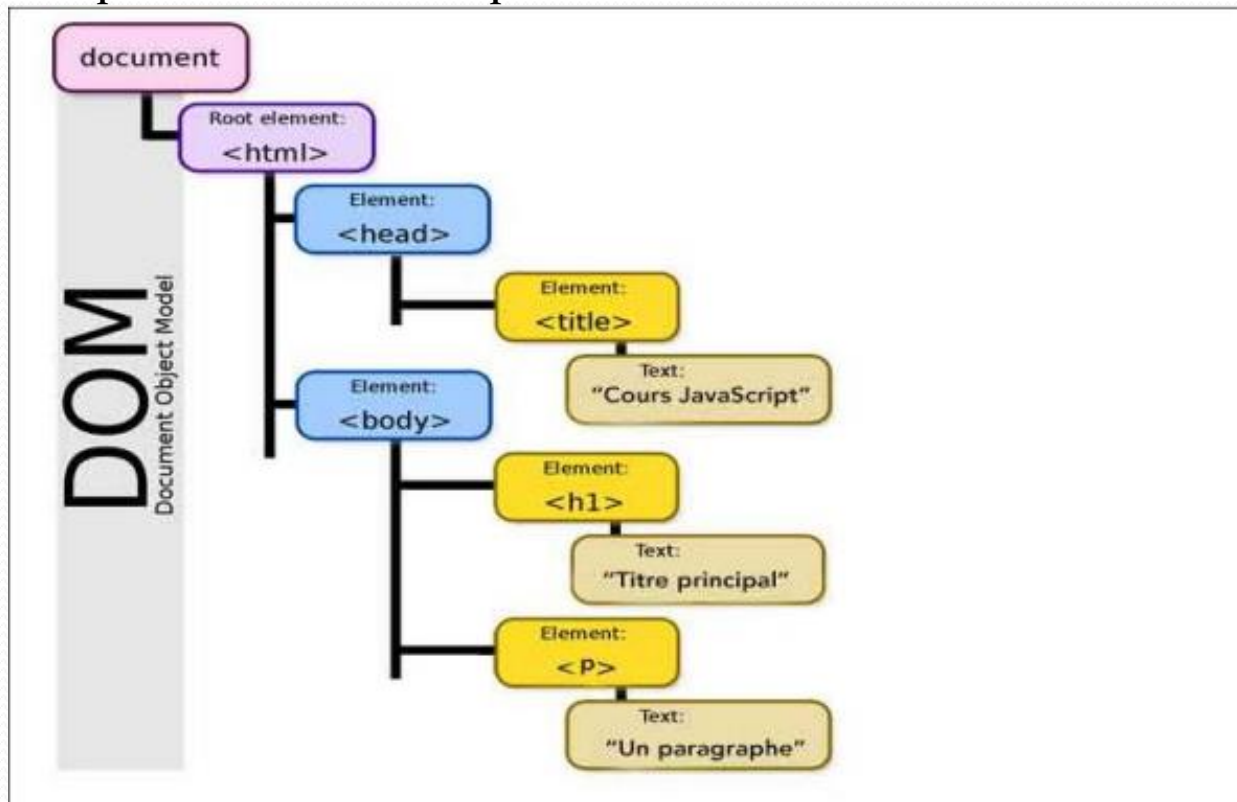
# Structure du DOM

- Le DOM est une représentation structurée du document sous forme « d'arbre » créée automatiquement par le navigateur.
- Soit le code html suivant :

```
<!DOCTYPE html>
<html>
<head>
  <title>Cours Javascript </title>
</head>
<body>
<h1>Titre principal</h1>
<p>Un paragraphe</p>
</body>
</html>
```

# Structure du DOM

- Lorsqu'on demande au navigateur d'afficher cette page, il crée automatiquement un DOM qui ressemble à ceci :



# Accéder aux éléments d'un document (**getElementById**)

- Cette méthode permet d'accéder à un élément en connaissant son ID qui est simplement l'attribut id de l'élément.
- Exemple: Un peu de texte et un lien

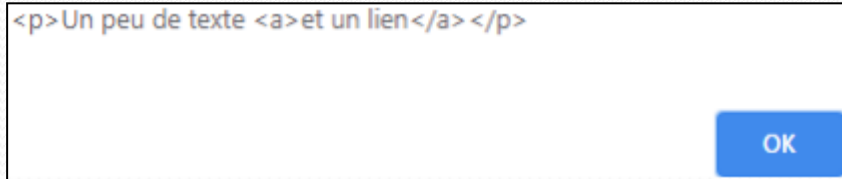
```
<div id="myDiv">  
<p>Un peu de texte <a>et un lien</a></p>  
</div>  
<script>  
var div = document.getElementById('myDiv');  
alert(div);  
</script>
```

# innerHTML

- innerHTML permet de récupérer le code HTML enfant d'un élément sous forme de texte. Ainsi, si des balises sont présentes, innerHTML les retournera sous forme de texte
- Exemple:

```
<body>  
<div id="myDiv">  
<p>Un peu de texte <a>et un lien</a></p>  
</div>  
<script>  
var div = document.getElementById('myDiv');  
alert(div.innerHTML);  
</script>  
</body>
```

- Le résultat est :



<p>Un peu de texte <a>et un lien</a> </p>

OK

# innerHTML

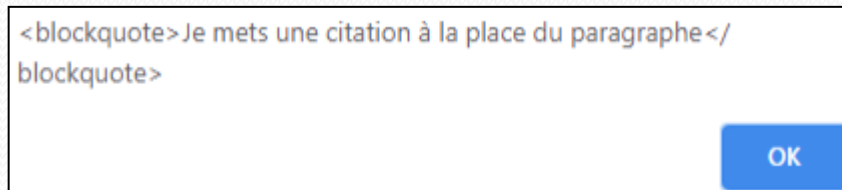
- innerHTML permet également de modifier le contenu d'un élément, en lui affectant le contenu souhaité.

- *Exemple :*

```
document.getElementById('myDiv').innerHTML = 'Je mets  
une citation à la place du paragraphe';
```

```
alert(div.innerHTML);
```

Résultat :





# innerText et textContent

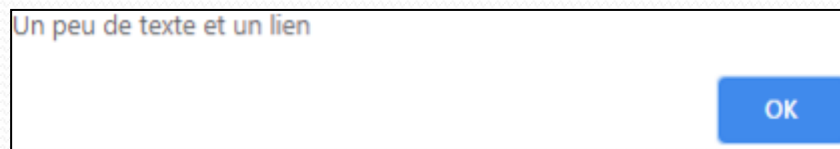
- La propriété **innerText** a aussi été introduite dans Internet Explorer mais elle n'a jamais été standardisée et elle n'est pas supportée par tous les navigateurs.
- Le fonctionnement **d'innerText** est le même **qu'innerHTML** excepté le fait que seul le texte est récupéré, et non les balises.
- La propriété **textContent** est la version standardisée d'innerText.
- Cette propriété fonctionnent exactement de la même manière que innerText, la seule différence est que elle est reconnue par tous les navigateurs

# textContent

- Exemple :

```
<body>
<div id="myDiv">
  <p>Un peu de texte <a>et un lien</a></p>
</div>
<script>
var div = document.getElementById('myDiv');
alert(div.textContent);
</script>
</body>
```

Résultat :



# value

- La propriété **value** va nous permettre de **récupérer la valeur d'un élément d'un formulaire** (**input** ou **select** ou textarea, case à côcher, radio bouton...) et aussi de modifier sa valeur.

# value

- Exemple:

```
<body>
<form>
<input type="text" id="x" ><br>
<input type="text" id="y"><br>
<input type="button" value="valider"><br>
</form>

<script>
let cont=document.getElementById("x").value;
alert(x)
</script>

</body>
```

# Accéder aux éléments d'un document(**getElementsByName**)

- Cette méthode permet de récupérer les éléments qui possèdent un attribut **name** que vous spécifiez.
- Et **retourne un tableau des objets**, la première case va contenir le premier objet, la deuxième case va contenir le deuxième objet et ainsi de suite.
- **Exemple :**

```
<input type="radio" name="rad" value="1" >choix numéro 1 <br>  
<input type="radio" name="rad" value="2" checked="checked">choix numéro 2 <br>  
<input type="radio" name="rad" value="3">choix numéro 3 <br>
```

## **Script :**

```
var tab=document.getElementsByName("rad");
```

```
Var x=tab.length; // x est égal à 3
```

# Les cases à côcher

- Pour vérifier si une case à côcher est cochée on utilise la propriété **checked**. Cette propriété retourne **true** ou **false** selon si la case est cochée ou non.
- Pour récupérer la **valeur** d'une case à côcher on utilise la propriété **value**.

- **Exemple :**

```
<input type="checkbox" id="ch1" value="Etudiant" />Etudiant  
if (document.getElementById("ch1").checked==true) {  
  alert(document.getElementById("ch1").value);  
}
```

# Les radios boutons

- La caractéristique des radios boutons c'est que un certains nombre appartiennent à un même groupe et ont le même nom.
- Alors comment cibler un radio bouton? Le nom commun à ces radio boutons va être **un tableau indicé, selon le nombre de radios qu'il y'a dans le groupe.**

# Les radios boutons

- Pour vérifier si un radio bouton est coché on utilise la propriété **checked**.
- Pour récupérer la valeur d'un radio bouton ou de la modifier on se sert de la propriété **value**.
- Exemple:

```
<input type="radio" name="rad" value="F" />F  
<input type="radio" name="rad" value="H" />H
```

```
//vérifier que le deuxième radio est coché, si oui afficher sa valeur  
if (document.getElementsByName("rad")[1].checked==true){  
  alert(document.getElementsByName("rad")[1].value);  
}
```



# Zone de liste (select)

- La propriété **selectedIndex**: propriété qui permet de récupérer l'indice de l'élément sélectionné ou modifier sa valeur.
- La propriété **value** : récupère la valeur de l'option sélectionnée.
- La propriété **length**: retourne le nombre des options de la zone de liste.

# Exemple

- Soit la zone de liste suivante:

```
<select id="sel">  
  <option>option1</option>  
  <option>option2</option>  
  <option>option3</option>  
</select>
```

- Code js:

```
alert(document.getElementById("sel").selectedIndex);  
alert(document.getElementById("sel").value);  
alert(document.getElementById("sel").length); // 3  
document.getElementById("sel").selectedIndex=2; //sélectionne option3
```

# Donner le focus à un élément

- on utilise la méthode **focus()** pour donner le focus de la souris à un élément du formulaire :

```
document.getElementById("Monid").focus();
```