



# Université Assane Seck de Ziguinchor

## JavaScript – Le DOM et les événements

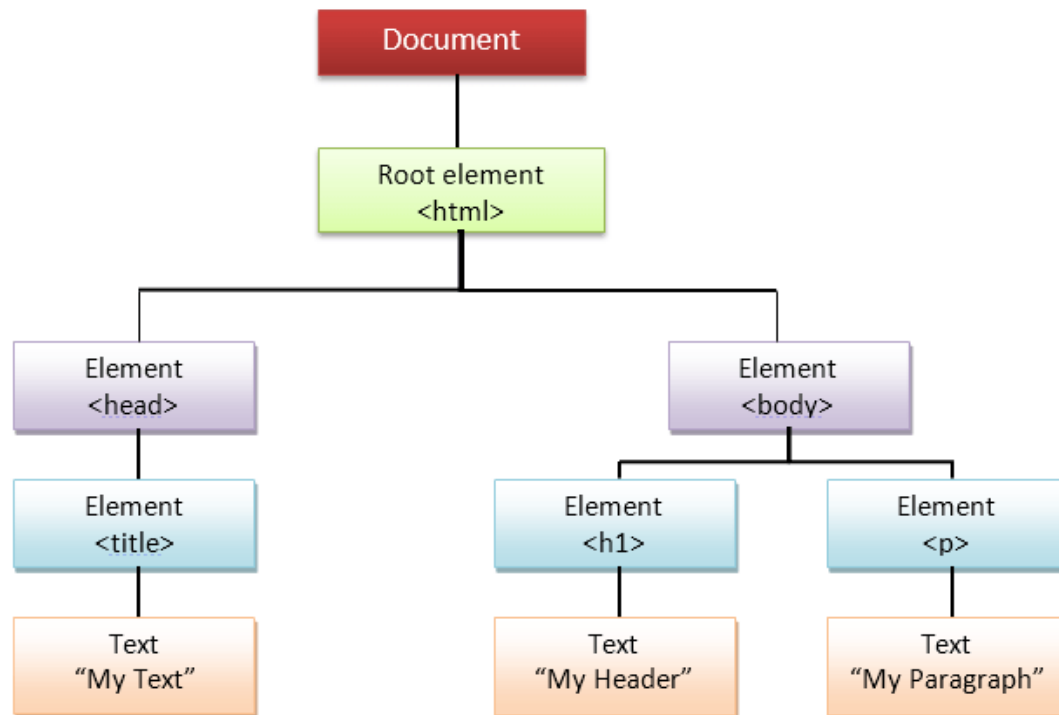
UFR ST – Département Informatique  
Licence 2 Ingénierie Informatique 2021/2022

**Marie NDIAYE**

# Les DOM

# Qu'est-ce que le DOM ?

- ▶ Le modèle d'objet de document (*Document Object Model - DOM*) d'une page Web est créé par le navigateur Web lorsque la page est chargée.
- ▶ Le modèle DOM est créé sous la forme d'un arbre comme ceci :



# JavaScript et le DOM

---

- ▶ JavaScript peut accéder à tous les éléments d'une page Web en utilisant le DOM.
- ▶ JavaScript peut effectuer plusieurs tâches. Il peut :
  - ▶ **créer** de nouveaux éléments et attributs ;
  - ▶ **modifier** les éléments et attributs existants ;
  - ▶ **supprimer** des éléments et attributs existants.
- ▶ JavaScript peut également **réagir aux événements existants** et **créer de nouveaux événements** dans la page.

# Exemple 1

---

```
<html>
<head>
    <title>DOM!!!</title>
</head>
<body>
    <h1 id="un">Bienvenue</h1>
    <p>C'est le message de bienvenue.</p>
    <h2>Technologie</h2>
    <p>C'est la section technologie.</p>
    <script type="text/javascript">
        var text = document.getElementById("un")
            .innerHTML;
        alert("The first heading is " + text);
    </script>
</body>
</html>
```

# Recherche d'éléments HTML

---

Méthode	Description
<code>document.getElementById(id)</code>	Rechercher un élément par ID
<code>document.getElementsByTagName(name)</code>	Rechercher des éléments par nom de balise
<code>document.getElementsByClassName(name)</code>	Rechercher des éléments par nom de classe

# Exemple 3 : HTML

---

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Exemple 3</title>
</head>
<body>
  <p id='p1' class='par'>Premier paragraphe</p>
  <p class='par'>Deuxième paragraphe</p>
  <p class='par'>Troisième paragraphe</p>
  <script type="text/javascript" src="exemple3.js"></script>
</body>
</html>
```

## Exemple 3 : JavaScript

---

```
let texte0 =  
document.getElementById("p1").innerHTML;  
alert(texte0);
```

```
let texte1 =  
document.getElementsByTagName("p")[1].innerHTML;  
alert(texte1);
```

```
let texte2 =  
document.getElementsByClassName("par")[2].innerHTML;  
alert(texte2);
```



# Modification des éléments HTML

---

Propriété	Description
<code>element.innerHTML = new html content</code>	Modifier le contenu HTML interne d'un élément
<code>element.attribute = new value</code>	Modifier la valeur d'attribut d'un élément HTML
<code>element.style.property = new style</code>	Changer le style d'un élément HTML
Méthode	Description
<code>element.setAttribute(attribute, value)</code>	Modifier la valeur d'attribut d'un élément HTML

# Exemple 4 : HTML

---

```
<!DOCTYPE HTML>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Exemple4</title>
</head>
<body>
  <p id='p1' class='par'>Premier paragraphe</p>
  <textarea rows="3" cols="20"></textarea>
  <script type="text/javascript"
    src="exemple4.js"></script>
</body>
</html>
```

# Exemple 4 : JavaScript

---

```
document.getElementById("p1").innerHTML = "Un  
nouveau paragraphe";
```

```
let attributs =  
document.getElementsByTagName("textarea")[0].attribu  
tes;  
attributs[0].value = "10";
```

```
element =  
document.getElementsByTagName("textarea")[0];  
element.setAttribute("cols", "10");  
element.style.color = "red";
```

# Ajout et suppression d'éléments

---

Méthode	Description
<code>document.createElement(element)</code>	Créer un élément HTML
<code>document.removeChild(element)</code>	Supprimer un élément HTML
<code>document.appendChild(element)</code>	Ajouter un élément HTML
<code>document.replaceChild(new, old)</code>	Remplacer un élément HTML
<code>document.write(text)</code>	Écrire dans le flux de sortie HTML

# Exemple 5 : HTML

---

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ajout elements</title>
</head>
<body>
  <ul id="menu">
    <li>Accueil</li>
    <li>A propos</li>
    <li>Contact</li>
  </ul>
  <script type="text/javascript"
src="exemple5.js"></script>
</body>
</html>
```

## Exemple 5 : JavaScript

---

```
let menu = document.getElementById('menu');
```

```
//Créer un nouveau noeud li
```

```
let li = document.createElement('li');
```

```
//Mettre du contenu dans li
```

```
li.textContent = 'Services';
```

```
//Ajouter le nouveau li au menu
```

```
menu.appendChild(li)
```

# Ajout de gestionnaires d'événements

---

Méthode	Description
<code>document.getElementById(id) . onclick = function() {code}</code>	Ajout de code de gestionnaire d'événements à un événement onclick

# Exemple 6 : HTML

---

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Ajout gestionnaire evenement</title>
</head>
<body>
    <input id='col' type="submit" value="Envoyer">
    <script type="text/javascript"
src="exemple6.js"></script>
</body>
</html>
```



## Exemple 6 : JavaScript

---

```
function colorier() {  
    document.body.style.backgroundColor = "red";  
}  
  
document.getElementById("col").onclick = colorier;
```

# Gestionnaires d'évènements

# Les évènements

---

- ▶ Les évènements sont des actions qui se produisent et auxquelles on va pouvoir répondre en exécutant un code.
- ▶ Par exemple, on va pouvoir afficher ou cacher du texte suite au clic d'un utilisateur sur un élément, on change la taille d'un texte lors du passage de la souris d'un utilisateur sur un élément.
- ▶ Les évènements et leur prise en charge sont l'un des mécanismes principaux du JavaScript qui vont nous permettre d'ajouter un vrai dynamisme à nos pages Web.

# Les types d'évènement

---

- ▶ Il existe différents types d'événements :
  - ▶ actions par le clavier ou la souris : click, keyup, mouseover ...
  - ▶ changement d'état : change, focus...
  - ▶ fin ou début de chargement d'un élément sur la page : load

# Gérer les évènements

---

- ▶ Pour écouter et répondre à un évènement, il faut définir des gestionnaires d'évènements.
- ▶ Il existe trois grandes façons d'implémenter un gestionnaire d'évènements :
  - ▶ On peut utiliser la méthode `addEventListener()` (recommandé).
  - ▶ On peut utiliser des propriétés JavaScript liées aux évènements ;
  - ▶ On peut utiliser des attributs HTML de type évènement (non recommandé) ;
- ▶ Un gestionnaire d'évènements est toujours divisé en deux parties :
  - ▶ une partie qui sert à écouter le déclenchement de l'évènement.
  - ▶ une partie gestionnaire en soi qui va être le code à exécuter lorsque l'évènement se produit.

# La méthode **addEventListener()**

---

- ▶ Elle réalise l'abonnement d'une fonction à un événement pour un élément.
- ▶ Utilisation de la méthode
  - ▶ **objet.addEventListener(typeEvenement, fonctionDeclenchee);**
  - ▶ **objet** : objet ciblé (ex : document ou objet récupéré par getElementById()).
  - ▶ **typeEvenement** : chaîne de caractères désignant l'événement concerné.
  - ▶ **fonctionDeclenchee** : fonction appelée (ex : changement image, ou contenu d'une balise)
- ▶ Exemple
  - ▶ Dans le fichier .html
    - ▶ `<input id='col' type="submit" value="Envoyer">`
  - ▶ Dans le fichier .js
    - ▶ `function colorier() { document.body.style.backgroundColor = "red";}`
    - ▶ `document.getElementById("col").addEventListener("click", colorier);`

# Les propriétés JavaScript liées aux événements

---

- ▶ Elles constituent une alternative à `addEventListener()` pour définir un bonnement à une fonction :
- ▶ Utilisation des propriétés JavaScript
  - ▶ **`objet.onevent = fonction`**
- ▶ Exemple :
  - ▶ Dans le fichier `.html`
    - ▶ `<input id='col' type='submit' value='Envoyer'>`
  - ▶ Dans le fichier `.js`
    - ▶ `function colorier() { document.body.style.backgroundColor = "red";}`
    - ▶ `document.getElementById("col").onclick = colorier;`

# Les attributs HTML de type évènement

---

- ▶ Il s'agit de la méthode la plus ancienne à notre disposition mais qui ne devrait plus être utilisée de nos jours.
- ▶ Ces attributs HTML de « type évènement » possèdent souvent le nom de l'évènement qu'ils doivent écouter et gérer précédé par « **on** » comme par exemple **onclick** pour l'évènement « clic sur un élément » ; **onmouseover** pour l'évènement « passage de la souris sur un élément » ; **onmouseout** pour l'évènement « sortie de la souris d'élément » ; etc.
- ▶ Exemple
  - ▶ Dans le fichier .html
    - ▶ `<input id='col' type="submit" value="Envoyer" onclick ="colorier()" >`
  - ▶ Dans le fichier .js
    - ▶ `function colorier() { document.body.style.backgroundColor = "red";}`



# Références

---

- ▶ <https://www.guru99.com/how-to-use-dom-and-events-in-javascript.html>
- ▶ <https://www.pierre-giraud.com/javascript-apprendre-coder-cours/addeventlistener-gestion-evenement/>
- ▶ <https://www.pierre-giraud.com/javascript-apprendre-coder-cours/addeventlistener-gestion-evenement/>
- ▶ [https://profdoc.iddocs.fr/IMG/pdf/billiejoe\\_javascript\\_fiches.pdf](https://profdoc.iddocs.fr/IMG/pdf/billiejoe_javascript_fiches.pdf)
- ▶ [https://perso.univ-rennes1.fr/laurent.dorazio/data/teachings/js/js\\_evenements.pdf](https://perso.univ-rennes1.fr/laurent.dorazio/data/teachings/js/js_evenements.pdf)
- ▶ <https://perso.liris.cnrs.fr/tristan.roussillon/ens/javascript/seance2/evtsJavaScript.pdf>
- ▶ <https://fgrelard.github.io/ens/even/even.pdf>