

# Javascript

Naviguer dans le DOM

# Plan

- Naviguer entre les nœuds
  - C'est quoi un noeud?
  - Informations sur les nœuds
    - **nodeType**
    - **nodeName**
  - **Nœud parent**
    - **parentNode** et **parentElement**
  - **Nœuds enfants**
    - **firstChild** et **firstElementChild**
    - **lastChild** et **lastElementChild**
    - **childNodes** et **children**
  - **Nœuds frères**
    - **nextSibling** et **nextElementSibling**
    - **previousSibling** et **previousElementSibling**

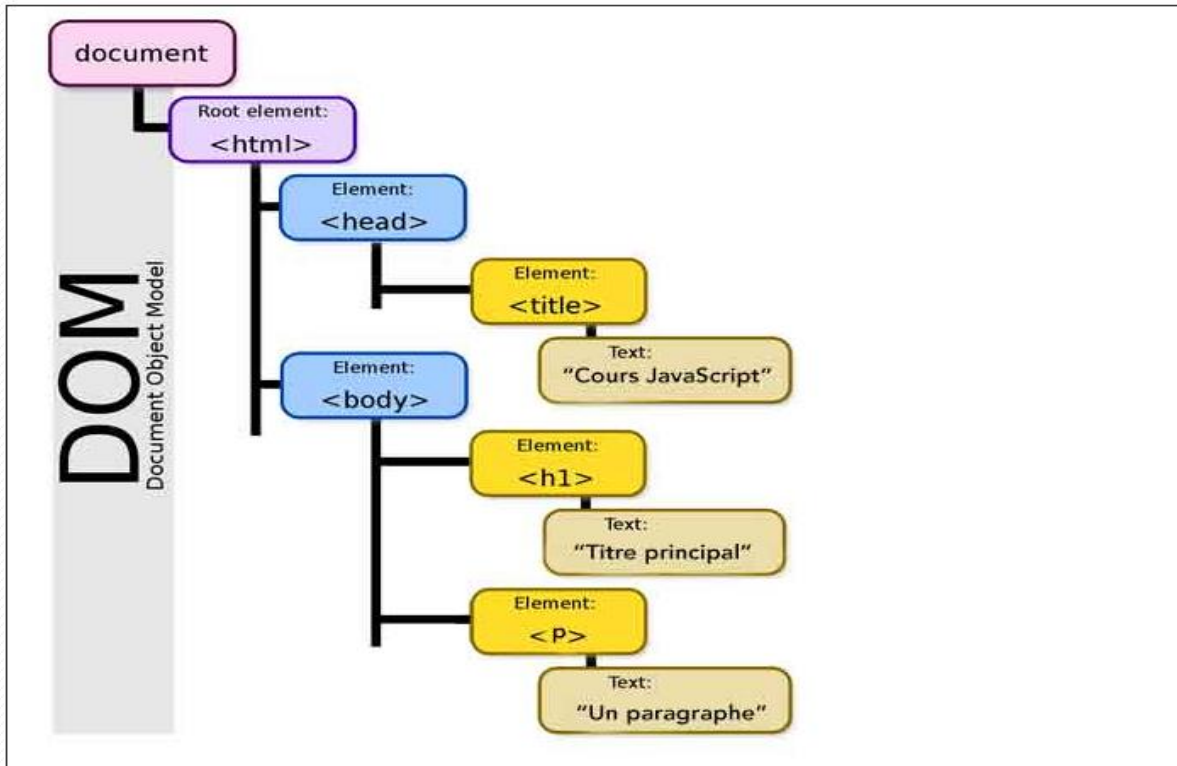
# Structure du DOM

- 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(2)

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



# C'est quoi un nœud?

- Chaque branche dans l'arbre du DOM se termine par ce qu'on appelle un **nœud**.
- Le terme « **nœud** » est un terme qui sert à désigner tous les objets contenus dans le DOM. **A l'extrémité de chaque branche** du DOM se trouve un nœud.

# Informations sur les nœuds (nodeType)

- La propriété **nodeType** sert à vérifier le *type* d'un nœud.
- **nodeType** retourne un nombre, qui correspond à un type de nœud. Il existe plusieurs types de nœuds , les plus utilisés :

Numéro	Type de nœud
1	Nœud élément comme <p> , <img>....
3	Nœud texte: le text actuel d'un élément
8	Nœud commentaire
9	Nœud doctype

# Informations sur les nœuds (nodeName)

- La propriété **nodeName** sert à récupérer le *nom* d'un nœud: **nom de la balise** s'il s'agit d'un nœud **Element**, ou **#text** s'il s'agit d'un nœud textuel.

## Exemple:

```
var par = document.getElementById('p1'); //paragraphe
```

```
alert("le type de nom"+par.nodeType );
```

```
alert("le nom du nœud"+par.nodeName.toLowerCase());
```

# Accéder aux éléments parents

- Pour accéder à un élément parent d'un autre élément, on utilise:
  - **parentNode**: récupère le parent d'un élément que ce soit un élément ou autre
  - **parentElement**: récupère le parent élément (balise)



# Naviguer entre les nœuds (nœud parent)

- La propriété **parentElement** permet d'accéder à l'élément parent d'un élément.
- *Exemple :*

```
<div>  
<p id="p1">Ceci est un paragraphe !</p>  
</div>
```

Le code suivant va nous récupérer le parent du paragraphe (c'est-à-dire le div):

```
Var mondiv=document.getElementById("p1").parentElement;
```

# Accéder aux éléments enfants

- Pour accéder à un élément enfants, il existe les propriétés suivantes:
  - **firstChild**: accède au premier enfant que ce soit un nœud element ou text ou autre
  - **firstElementChild**: accède au premier nœud élément enfant
  - **lastChild**: accède au dernier enfant que ce soit un nœud élément ou text ou autre
  - **lastElementChild**: accède au dernier nœud élément enfant
  - **childNodes** : retourne un tableau contenant la liste des nœud enfants (que ce soit élément ou text,...) d'un élément.
  - **Children**: retourne un tableau contenant la liste des nœud enfants de type élément.

# Les noeuds enfants

- **Exemple:**

`<p id="p1">`

Un peu de texte, `<a>`un lien`</a>` et `<strong>`une portion en emphase`</strong>`  
`</p>`

**Le code suivant va récupérer le premier et le dernier enfants de l'élément paragraphe:**

```
var x= document.getElementById('p1').firstElementChild;
```

```
var y = document.getElementById('p1').lastElementChild;
```

```
alert(x.nodeName.toLowerCase()); //affiche a
```

```
alert(y.nodeName.toLowerCase()); // affiche strong
```

# Naviguer entre les nœuds (children)

- Exemple:

```
<div>  
<p id="p1">Un peu de texte <a>et un lien</a></p>  
</div>
```

```
var D= document.getElementById("div");  
var tab= D.children;    //tab est un tableau qui contient la liste des  
                        enfants éléments du div  
  
alert(tab.length);    //1  
alert(tab[0].nodeName) // p
```

# Les nœuds frères

- Les nœuds frères sont les nœuds qui se trouvent dans le même niveau.

`<p id="p1">`

Un peu de texte, `<a>`un lien`</a>` et `<strong>`une portion en emphase`</strong>`

`</p>`

- Dans l'exemple dessus `<a>` et `<strong>` sont des nœuds frères.

# Les nœuds frères

- Il existe 4 propriétés pour accéder aux nœuds frères:
  - **nextSibling**: accède au nœud suivant (nœud element ou text,..)
  - **nextElementSibling**: accède au nœud Element suivant
  - **previousSibling**: accède au nœud précédent (nœud element ou text,..)
  - **previousElementSibling**: accède au nœud Element précédent

# Les nœuds frères

## *(nextSibling et nextElementSibling)*

- La propriété **nextSibling** et **nextElementSibling** est une propriété qui permet d'accéder au nœud **suivant**.
- Exemple

```
<div>  
<p id="p1">Un peu de texte, <a>un lien</a> et <strong>une portion en  
emphase</strong></p>  
</div>
```

```
<script>  
var monpara= document.getElementById('p1');  
var premier= monpara.firstElementChild;  
var suivant= premier.nextElementSibling;  
alert(suivant.firstChild.data);  
</script>
```

# Les nœuds frères (*previousSibling*)

- La propriété **previousSibling** est une propriété qui permet d'accéder au nœud précédent.
- Exemple :

```
<div>  
<p id="p1">Un peu de texte, <a>un lien</a> et <strong>une portion en  
emphase</strong></p>  
</div>
```

```
<script>  
var monpara= document.getElementById('p1');  
var dernier= monpara.lastElementChild;  
var precedent= dernier.previousElementSibling;  
alert(precedent.data);  
</script>
```



# Récupérer la valeur d'un nœud (**nodeValue** et **data** )

- La propriété **nodeValue** et la propriété **data**, permettent de récupérer le contenu d'un nœud de type text

Exemple:

```
var monpara= document.getElementById("p1");  
var premier = monpara.firstChild;  
var dernier = monpara.lastChild;  
alert(premier.nodeValue);  
alert(dernier.firstChild.data);
```

Utilisé pour les nœud qui ne sont de type element . Sinon, on utilise **innerHTML** ou **textContent**.