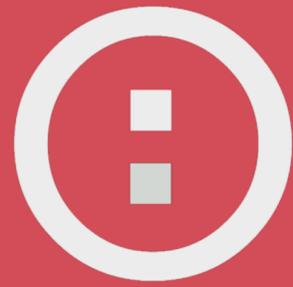


`simplonBoulogne = Javascript;`

DOM



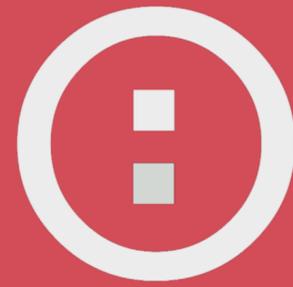


# Qu'est-ce que le DOM ?

Document Object Model (DOM)

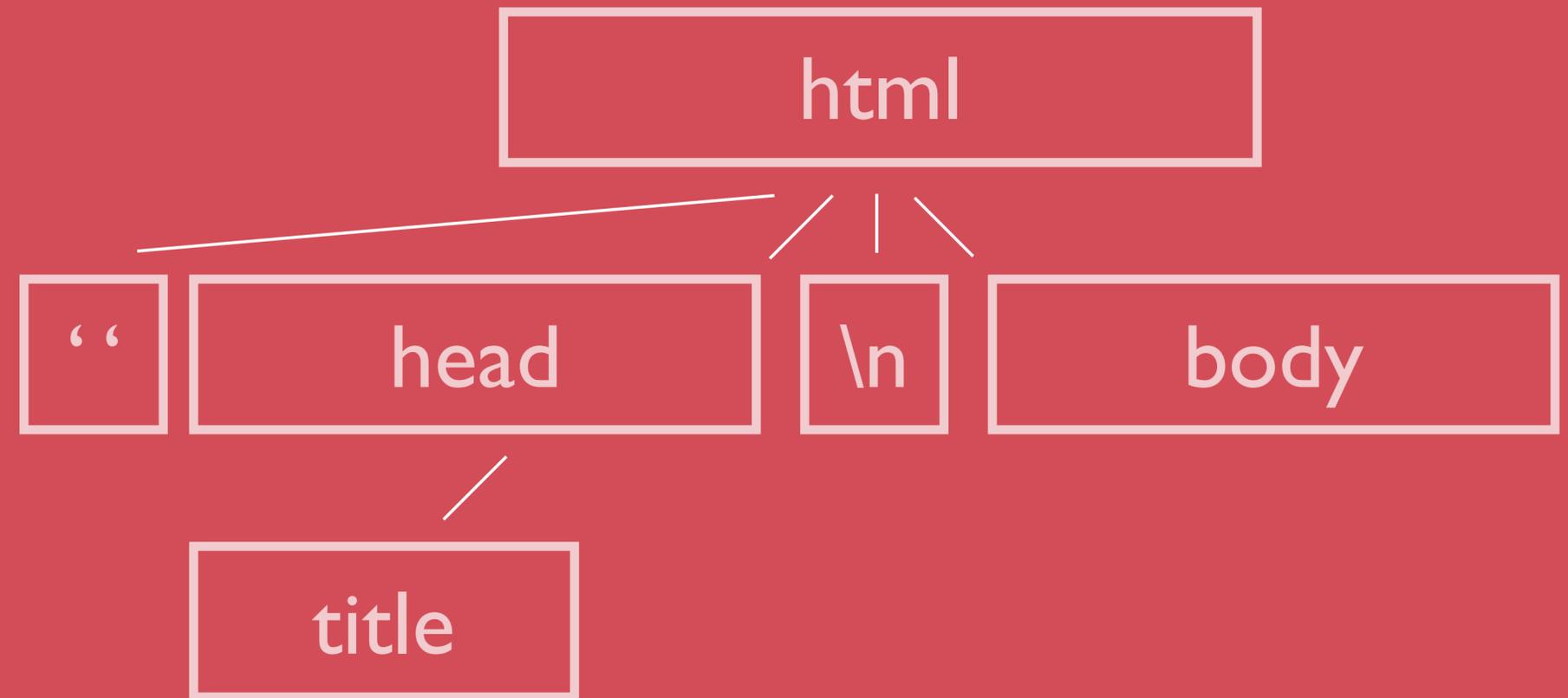
Ensemble d'interfaces normalisées permettant d'exposer le contenu d'un document HTML à un langage de programmation.

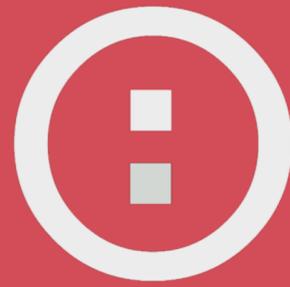
Les documents dans le DOM ont une structure en arbre.



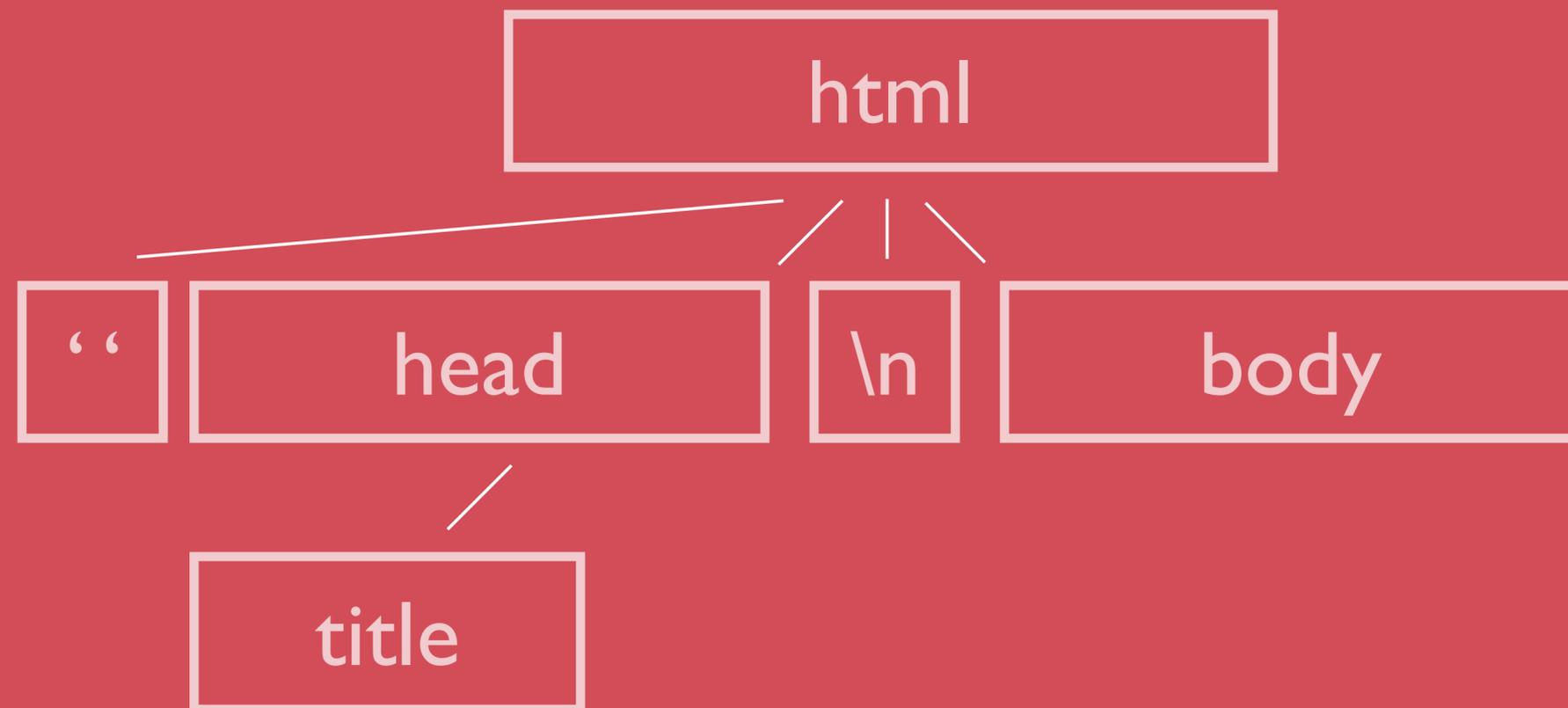
# Représentation en arbre

```
<html>  
  <head>  
    <title></title>  
  </head>  
  <body>  
  </body>  
</html>
```





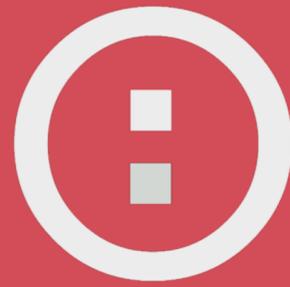
# Représentation en arbre



Attention,

Même les espaces  
blanc sont  
considérés comme  
des éléments donc  
un noeud !

Merci Sarah :)



# Exemple

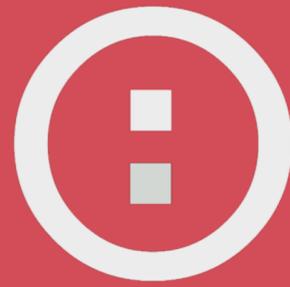
```
<input type="text" name="address">
```

Object

Propriété

Valeur

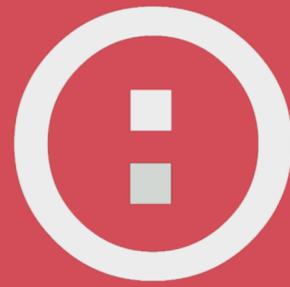




# Inspecter le DOM

Vous pouvez inspecter le DOM d'un document HTML via l'inspecteur d'élément dans vos navigateurs respectifs.

**Attention**, dans votre code, le javascript doit encore être chargé en dernier, sinon votre DOM ne sera pas chargé avant le script...

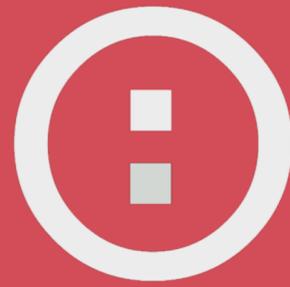


# Accéder à des éléments du DOM

## Méthode 1 - Récupération par noeud

```
var e = document.body.childNodes[1];
```

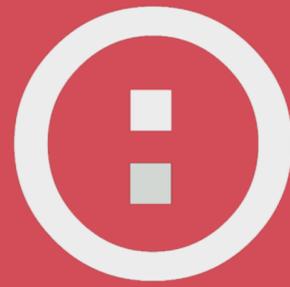
Trouve l'élément au premier noeud après body



# Accéder à des éléments du DOM

## Méthode 2 - Récupération par id

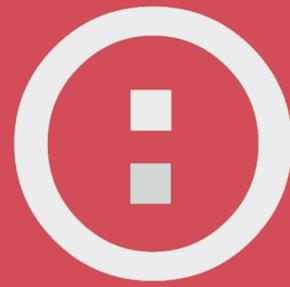
```
var e = document.getElementById('img1');
```



# Accéder à des éléments du DOM

Méthode 3 - Avec l'attribut « name »

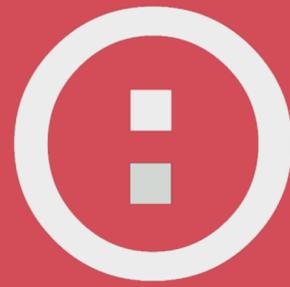
```
<form name="myForm" action="" >  
  <input type="button" name="myButton" />  
</form>
```



# Accéder à des éléments du DOM

Méthode 3 - Avec l'attribut « name »

```
var btn = document.myForm.myButton;
```

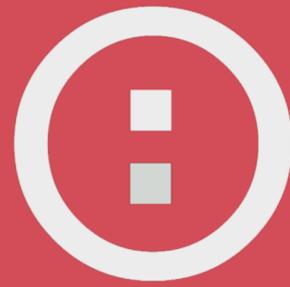


# Accéder à des éléments du DOM

## Méthode 4 - Récupération par class

```
document.getElementsByClassName('rouge test')
```

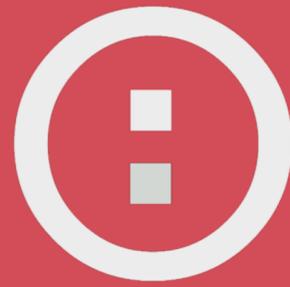
Trouve tous les éléments ayant les classes « rouge » et « test »  
Retourne un tableau d'éléments avec des propriétés universelles



# innerHTML

Remplacer le contenu HTML d'un l'élément

```
d = document.getElementById('paragraphe');  
d.innerHTML('Lorem Ipsum');
```



# Propriété « style »

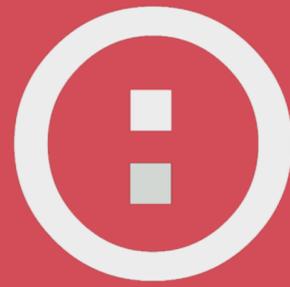
```
var p =  
document.getElementById("paragraphe");  
p.style.color = "red";  
p.style.top = "30px";
```



# Propriété style

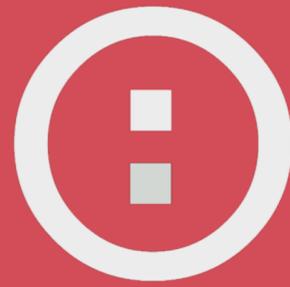
Attention, lorsqu'un attribut CSS comporte un trait d'union, on l'écrit en camelCase !

Quelques exceptions existent... si une erreur apparaît c'est que cet attribut CSS est déjà réservé, il faut alors faire une recherche



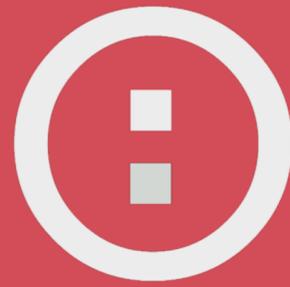
# Quelques méthodes()

```
var p =  
document.getElementById("paragraphe");  
p.toString();  
p.cloneNode(true); //p.cloneNode(false)  
p.appendChild();  
p.removeChild(<node>);  
p.replaceChild(<newNode> <oldNode>);
```



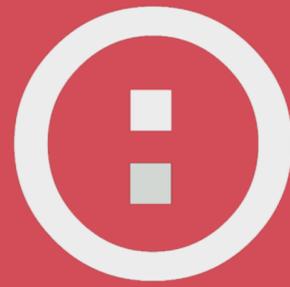
# Lire la valeur d'un attribut

```
<input id="nameRonds" type="text"/>  
p = document.getElementById("nameRonds");  
var typeP = p.getAttribute("type");  
console.log(typeP);
```



# Écrire la valeur d'un attribut

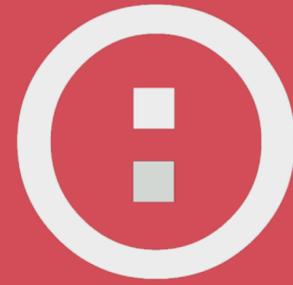
```
<input id="nameRonds" type="text" />  
p = document.getElementById("nameRonds");  
p.setAttribute("style", "color: grey;");  
p.setAttribute("placeholder", "Nom du joueur  
rond");
```



# Challenge

Écrire un bout de code JS qui permet de lire le texte de la première cellule d'un tableau HTML.

Remplacer cette cellule par le texte « Bonjour » et lui ajouter du style directement en JS.



# Ressources

- <https://jsfiddle.net/>
- <https://developer.mozilla.org/fr/docs/Web/JavaScript>