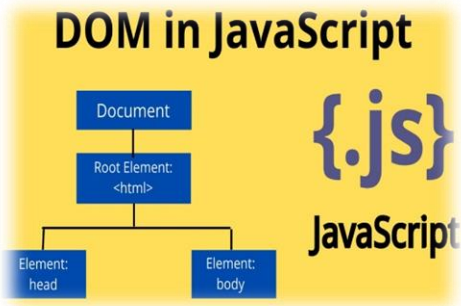
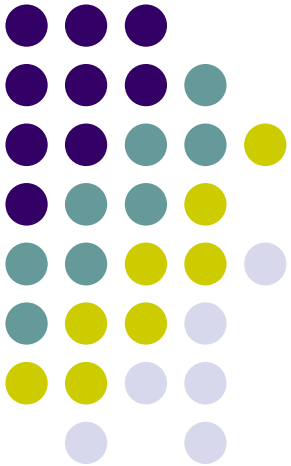
JavaScript

DOM





Franck LAMY – 2022/2023 – BTS SIO2

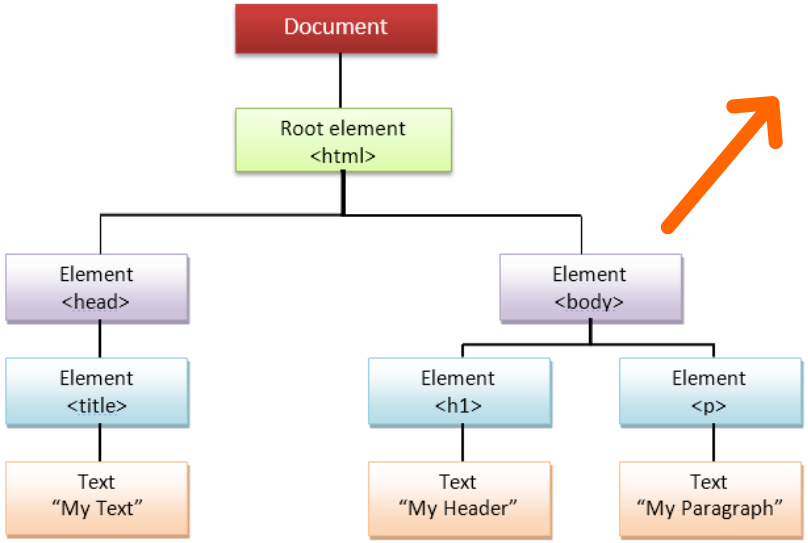
Objectifs

- **Introduire la notion de DOM**
- **Manipuler le DOM**

DOM

- Document Object Model
- Créé par le navigateur au chargement de la page
- DOM représente une page sous la forme d'une arborescence

```
<html lang="fr">
<head>
   <title>My text</title>
</head>
<body>
   <h1>My header</h1>
   <p>My paragraph</p>
</body>
</html>
```



Noeud

Propriétés

Méthodes

DOM

DOM est une interface de programmation (API) permettant à des scripts (javascript) de lire et de manipuler de contenu d'une page HTML

DOM

Grâce au DOM, on peut :

- Accéder aux différents éléments de la page
- Modifier des éléments
- Créer ou supprimer des éléments
- Modifier le style des éléments
- Interagir avec l'utilisateur
- Etc.

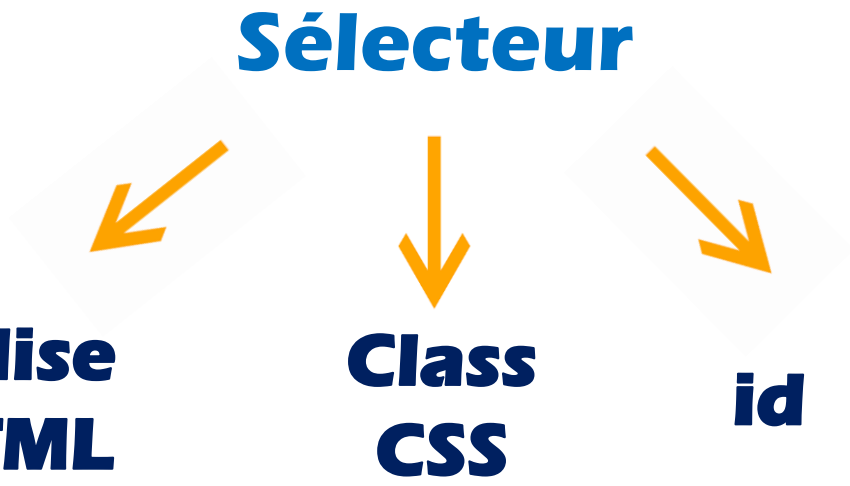
L'objet Document

Document est un **objet** qui représente la **page entière**

Document propose des **méthodes** permettant d'accéder aux **éléments** dans la page

querySelector()

Méthode permettant de rechercher le premier élément correspondant à un **sélecteur**

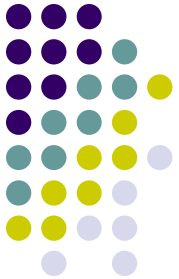


querySelector()

index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
      <title>Cours DOM</title>
</head>
<body>
  <h2>Présentation</h2>
  <div class="id1">Salut</div>
  <div class="id2">
    <p id="promo">BTS SIO2</p>
    <p>Bienvenue dans le cours sur le DOM </p>
  </div>
</body>
</html>
```

querySelector()



```
const element = document.querySelector("div");
console.log(element);
```

```
const element = document.querySelector(".id2");
console.log(element);
```

```
const element = document.querySelector("#promo");
console.log(element);
```

```
const element = document.querySelector(".id2 > p");
console.log(element);
```

Autres méthodes

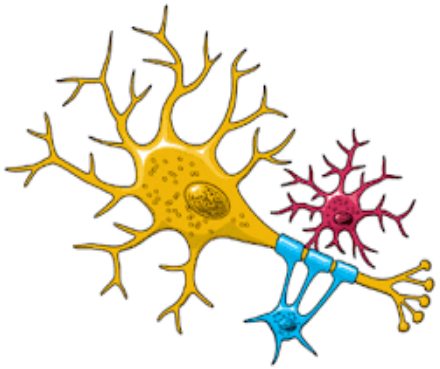
Document propose d'autres méthodes pour rechercher des éléments

getElementById
rechercher un élément via son id

getElementsByClassName
rechercher des éléments via une class

getElementsByTagName
rechercher des éléments via une balise

A vos neurones !



Afficher (sur la console) tous les éléments de la page qui sont des paragraphes

Contenu d'un élément

2 propriétés principales

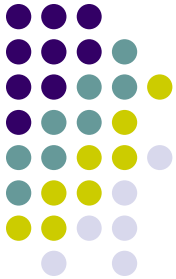
innerHTML

texte représentant un contenu HTML

textContent

texte non interprété en HTML

Les styles d'un élément



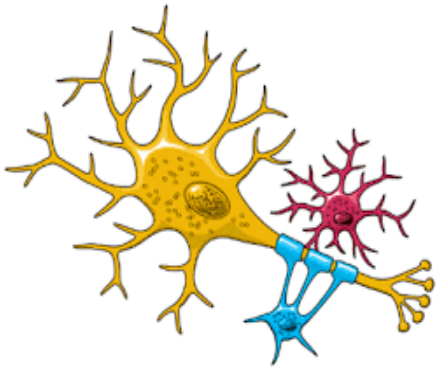
Propriété style

style

représente le style d'un élément

```
const paragraphe = document.querySelector("#promo");
paragraphe.style.color = "blue";
```

A vos neurones !



Modifier le style du paragraphe (#promo) de manière à :

- Mettre le texte en vert et en gras
- Mettre une bordure de 2px rouge

Les classes d'un élément

S'il y a beaucoup de style à modifier, il est préférable de passer par une **classe CSS**

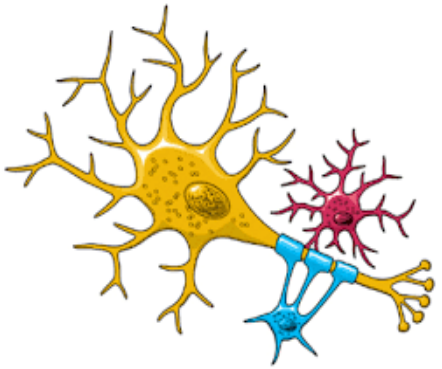
Propriété classList

classList

fournit des fonctions permettant de gérer les classes associées à un élément

add(<string>, [<string>, ...]) : ajoute la ou les classes spécifiées ;
remove(<string>, [<string>, ...]) : supprime la ou les classes spécifiées ;
contains(<string>) : vérifie si la classe spécifiée est contenue par cet élément ;
replace(<old>, <new>) : remplace l'ancienne classe par la nouvelle classe.

A vos neurones !



Modifier le style du paragraphe (#promo) de manière à :

- **Mettre le texte en vert et en gras**
- **Mettre une bordure de 2px rouge**

En utilisant une classe CSS !

p-vert-bordure-rouge

Les événements

- Gérer les **interactions** avec l'**utilisateur** au sein de la page
- **Ecouter** les **événements** afin de les **traiter**

Les événements

Evénement : c'est une réaction à une action émise par l'utilisateur
Ex : clic sur un bouton

Evénement

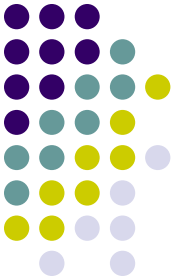
Nom

(click, mousemove,...)

Callback

(fonction qui traite l'événement)

Les événements



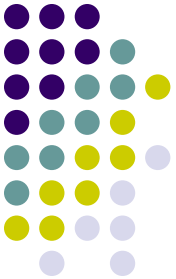
Pour être averti qu'un événement se déclenche, il faut l'écouter

Utilisation de la fonction :

element.addEventListener(event,callback)

permet d'être à l'écoute de tous types d'événements sur l'élément

Les événements



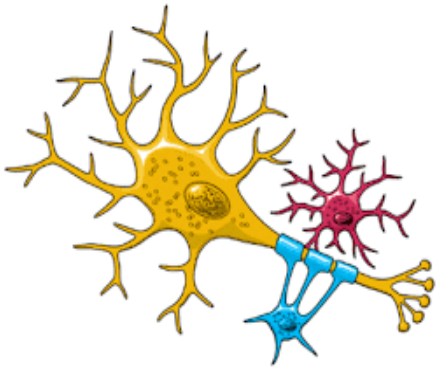
```
<button class="btn">Clic</button>
```

Nom événement

```
btn.addEventListener("click", () => {
    console.log("Bouton cliqué");
});
```

Callback

A vos neurones !

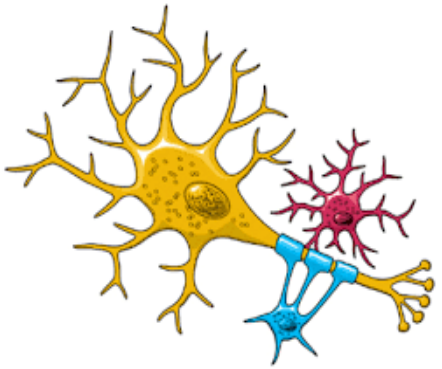


Modifier le style du paragraphe (#promo) de manière à :

- **Mettre le texte en vert et en gras**
- **Mettre une bordure de 2px rouge**

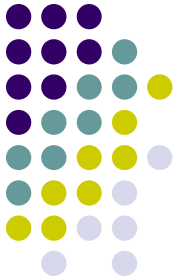
Lors du clic sur le bouton !

A vos neurones !



Créer un nouveau bouton Reset permettant de remettre "à zéro" le style du paragraphe (#promo)

Récupérer l'événement



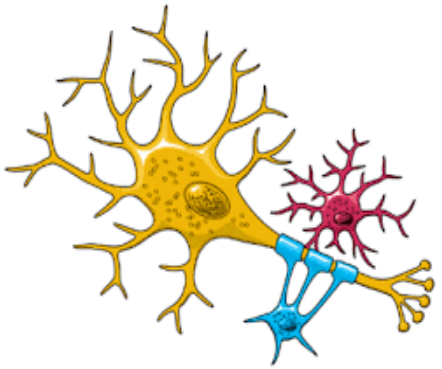
Possibilité de récupérer toutes les informations liés à l'événement

```
<button class="info">Infos</button>
```

```
const infos = document.querySelector(".info");
infos.addEventListener("click", (e) => {
    console.log(e);
})
```

Evénement

A vos neurones !



```css
.bordure-bleu {
    border: 2px solid blue;
    width: 100px;
    height: 20px;
    text-align: center;
}
```

Créer une div `<div class="coord bordure-bleu"></div>`

Afficher dans cette div les coordonnées de la souris lors de chaque déplacement de celle-ci dans le document !