

Réalisation de systèmes interactifs à manipulation directe en JavaScript

Ingénierie des systèmes interactifs
L2 MIASHS Parcours informatique (UE MIB0405V)



> typeof NaN	> true==1
< "number"	< true
> 9999999999999999	> true===1
< 10000000000000000	< false
> 0.5+0.1==0.6	> (!+[[]]+[![]]).length
< true	< 9
> 0.1+0.2==0.3	> 9+"1"
< false	< "91"
> Math.max()	> 91-"1"
< -Infinity	< 90
> Math.min()	> []==0
< Infinity	< true
> []+[]	
< ""	
> []+{}	
< "[object Object]"	
> {}+[]	
< 0	
> true+true+true===3	
< true	
> true-true	
< 0	



Ajout d'un capteur d'événement

A. Capter un événement dans le code HTML

(i.e. onévénement: *onmouseover*, *onload*, *onclick*, etc)

```
<button onclick="cliqueBouton()">Cliquez ici</button>
```

- Considéré comme une **mauvaise pratique** : mélanger sémantique et comportement ==> *addEventListener()*

B. *addEventListener()*

- Les événements produits par les actions de l'internaute sont modélisés dans le DOM par des objets **Event**
- Avantages de *addEventListener()* :
 - On peut ajouter **plusieurs listeners** pour un événement sur un même élément
 - On peut **retirer des listeners**
 - **Contrôle plus fin** du déclenchement de l'événement (capture ou remontée, etc)

addEventListener ...

- Pour ajouter un capteur de clics sur un élément elem

(deux arguments)

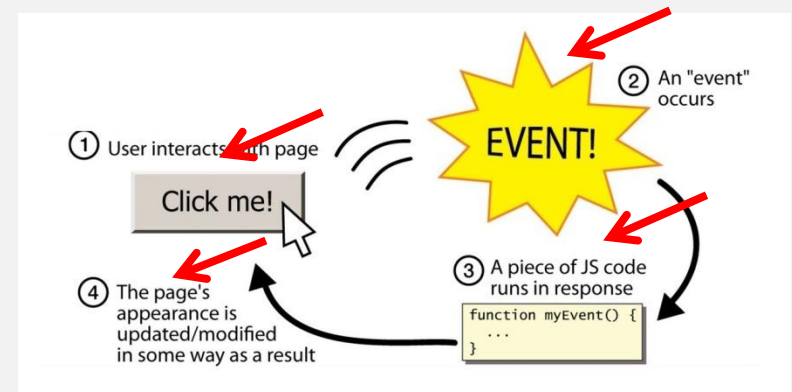
- le type d'événement à capter
- une fonction (callback à exécuter lorsque l'événement se produit)

```
elem.addEventListener ('click', maFonction);
```

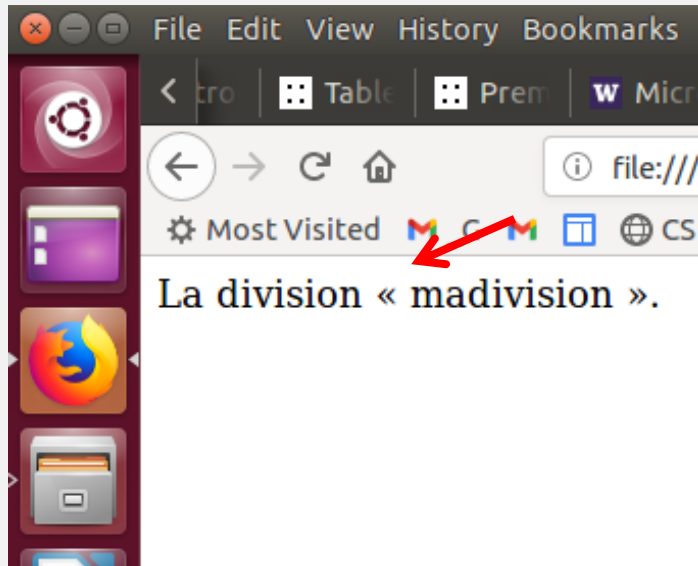
- Déroulement :

1. L'internaute **clique** sur l'élément
2. Le clic génère un objet **Event**
3. Le **capteur** se charge d'appeler sa **fonction callback**, en lui passant l'objet **Event** en paramètre

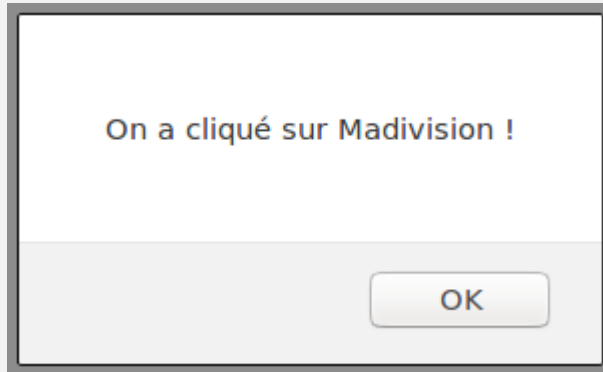
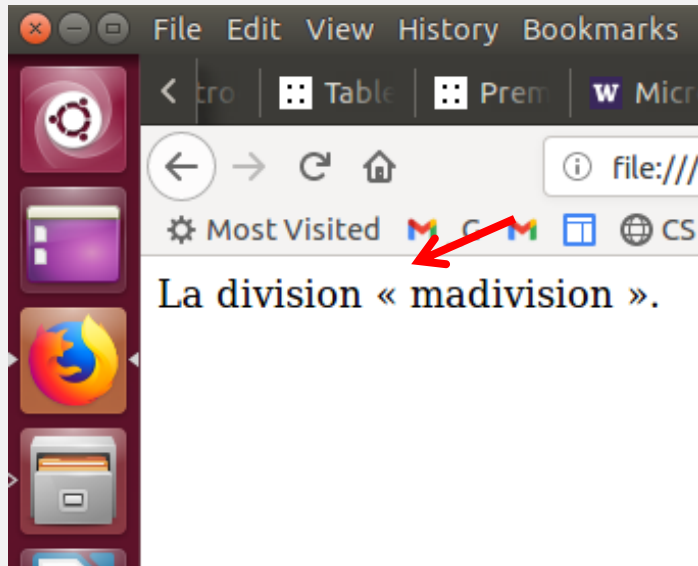
- Exemple ...



addEventListener ...



addEventListener ...



```
<div id=madivision> La division « madivision ». </div>
<script> "use strict";
    function clicSurMadivision (even) {
        alert("On a cliqué sur Madivision !");
        console.log(even);
    }
    let mydivision = document.getElementById("madivision");
    mydivision.addEventListener("click", clicSurMadivision);
</script>
```

Types d'événements

- Types d'événements :
 - **souris** : click, mouseenter et mouseleave, mouseup et mousedown, etc
 - **clavier** : keydown et keyup, keypress, etc
 - **formulaire** : focus, change, blur, submit, input...
 - **autres** : load, unload, scroll, resize, **JavaScript Timing Events**
- Voir recommandation du W3C

www.w3schools.com/jsref/obj_mouseevent.asp

www.w3schools.com/js/js_timing.asp

Projet d'ingénierie de systèmes interactifs

Partie Web

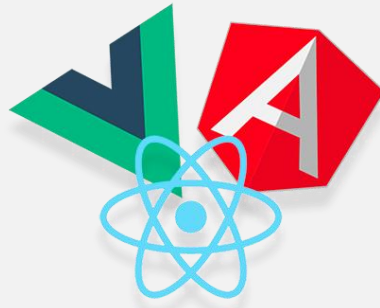


Tamagotchi

- Vous allez réaliser une application Web de Tamagotchi en suivant les éléments donnés dans le sujet de la partie Python (*dépôt sur IRIS*)
 - La fichier *HTML*
 - Les règles *CSS*
 - Le fichier *Javascript* dans lequel la gestion des différentes fonctionnalités est implémentée.



Allez Plus loin : Frameworks JS

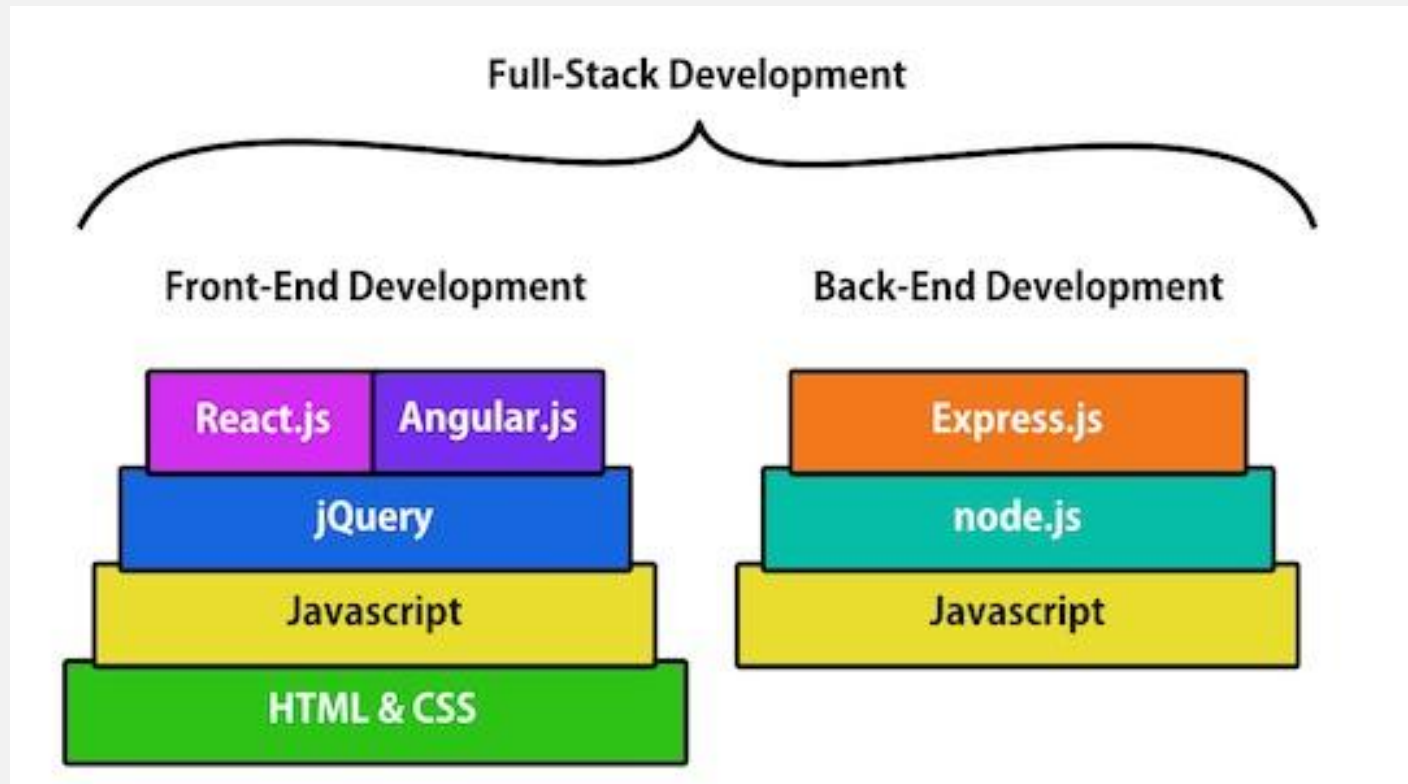


Frameworks JS

- Quelques bibliothèques de développement et de conception pour gagner en productivité,
- La bibliothèque jQuery
- Angular, React
- Côté serveur : nodejs, Express, etc

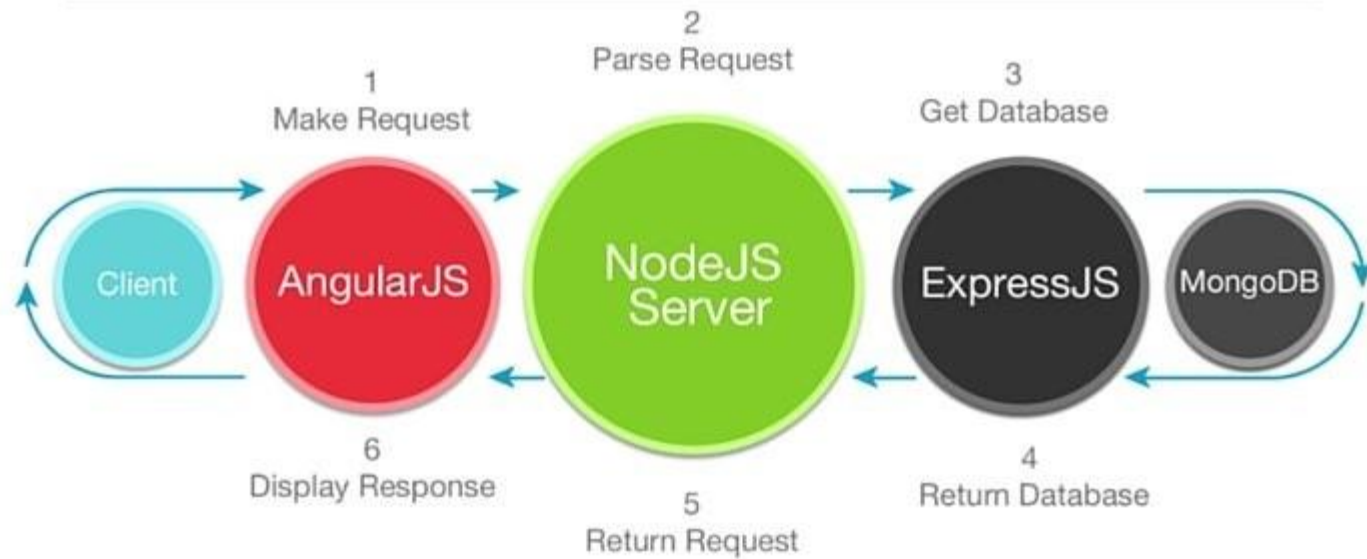


Développement Fullstack



Développement Fullstack

OPTIMIZE YOUR FULL STACK JAVASCRIPT APPLICATIONS



When you know only one programming language but it's capable of doing front end, back end and everything you need..



techviral /TechViral /TheTechViral

Javascript I am looking at you!