

JavaScript

Gestion des évènements, AJAX

Pierre Genthon - 2019

Avant de commencer...

- Pierre, 25
 - Mon Job : Dev web & mobile
 - Mon dada : RoR, React, React Native
 - Mes autres dadas :
 - La musique (Pink Floyd, Yes, Genesis, Queen, Birdy Nam Nam, ...)
 - les films (Kubrick, Tarantino, Dolan, Dupieux, ...)
 - La nourriture
 - Voyager !
 - pierregenthon.pro@gmail.com
- Et vous, qui êtes-vous ?

Au menu

- Présentation et live coding
 - Mini Intro à JS
 - Manipulation du DOM
 - Gestion des évènements
 - AJAX
 - Gestion de l'asynchronisme en JS
- Atelier : développement d'une petite application utilisant un 'autocomplete' pour rechercher des communes et afficher des infos détaillées sur ces dernières en récupérant des données en AJAX

(Mini) intro à JavaScript

- Créé en 1995 par Brandon Eich pour Netscape
- Évolution continue depuis (spec ECMAScript)
- Manipulation du DOM en direct
- Interactions : réagir à des événements ('click', 'keydown', 'focus', 'blur', ...)
- Gestion de la couche réseau (AJAX, Service Workers, ...)
- Manipulation des autres APIs Web (Storage, Geoloc., Canvas, Audio & Video, ...)

=> Bloc élémentaire des “SPA”

Manipulation du DOM

- getElementById
- querySelectorAll
- createTextNode
- innerHTML = ...
- style = ...
- appendChild
- removeChild
- setAttribute
- ...

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

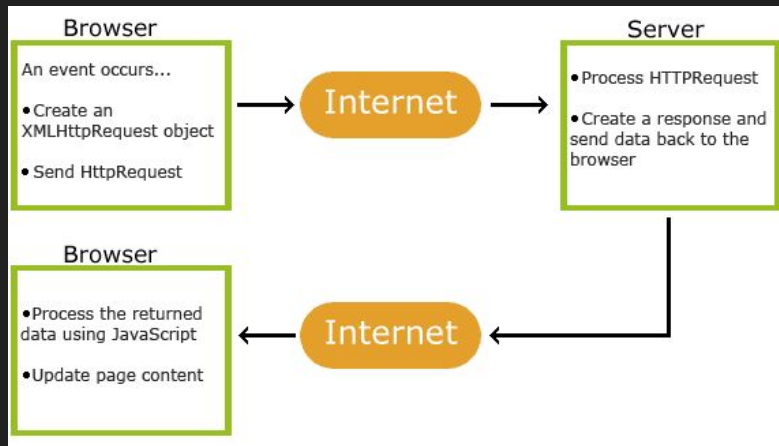
Gestion des évènements

```
<button onclick="alert('clicked !');" />
```

(à éviter !)

```
myButtonElement.addEventListener('click', function () { alert('clicked'); })
```

AJAX



Plusieurs manières de faire :

- utiliser XMLHttpRequest directement ou `ActiveXObject("Microsoft.XMLHTTP")`
- **utiliser `fetch()` qui fait partie du standard moderne**
- utiliser une autre bibliothèque (jQuery, Axios, Apisauce, Superagent...)

Gestion de l'asynchronisme

“Callback hell / Pyramid of doom”

```
doSomething(function(result) {  
  doSomethingElse(result, function(newResult) {  
    doThirdThing(newResult, function(finalResult) {  
      console.log('Got the final result: ' + finalResult);  
    }, failureCallback);  
  }, failureCallback);  
}, failureCallback);
```

Promise

```
doSomething()  
  .then(function(result) {  
    return doSomethingElse(result);  
  })  
  .then(function(newResult) {  
    return doThirdThing(newResult);  
  })  
  .then(function(finalResult) {  
    console.log('Got the final result: ' + finalResult);  
  })  
  .catch(failureCallback);
```

- Utilisation de bibliothèques comme bluebird, ...
- Les Promises sont la base de la nouvelle syntaxe “async/await” (ES7)
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Using_promises
- <https://pouchdb.com/2015/05/18/we-have-a-problem-with-promises.html>

Sources

- <https://developer.mozilla.org>
- <https://www.w3schools.com>
- <https://fr.wikipedia.org>
- <https://medium.com>

Atelier

Tout est ici

<https://github.com/ComicScrip/ajax-en-seine>