

# Langage de Script JavaScript

## Partie 3 : AJAX & JSON

Pr. Mohammed SALIHOUN

salihoun.med@gmail.com

3<sup>ème</sup> Année - Ingénierie en Informatique et Réseaux

3 novembre 2024

## 1. Introduction à AJAX

- Communication Synchrone
- AJAX
- Communication Asynchrone
- L'objet XMLHttpRequest
- Déroulement d'une requête Ajax
- Sécurité en Ajax

- L'objet simpleAjax

## 2. Introduction au format JSON

- JSON : JavaScript Object Notation
- JavaScript et les objets
- Syntaxe JSON
- Ecrire et lire du JSON
- JSON en PHP
- Envoi de valeur JSON en PHP

## 1. Introduction à AJAX

- Communication Synchrone
- AJAX
- Communication Asynchrone
- L'objet XMLHttpRequest
- Déroulement d'une requête Ajax
- Sécurité en Ajax

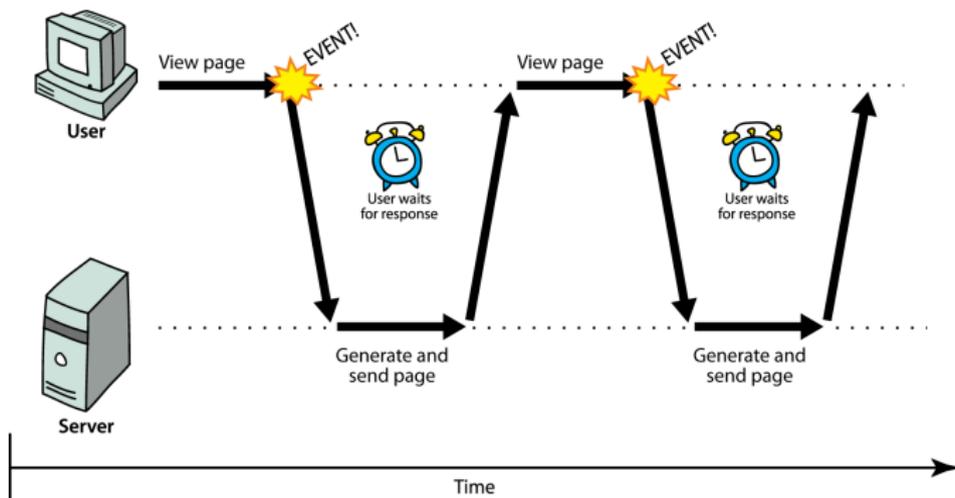
## - L'objet simpleAjax

## 2. Introduction au format JSON

- JSON : JavaScript Object Notation
- JavaScript et les objets
- Syntaxe JSON
- Ecrire et lire du JSON
- JSON en PHP
- Envoi de valeur JSON en PHP

# Introduction à AJAX

## Communication Synchrone

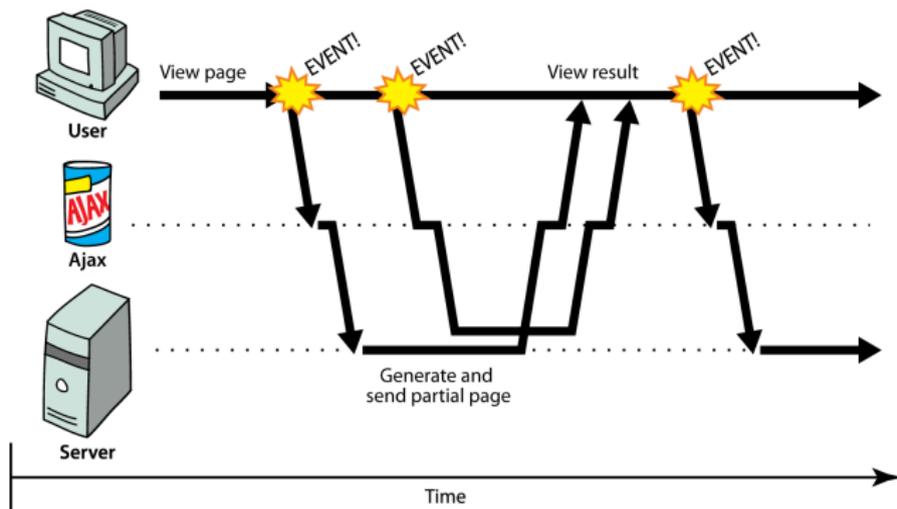


- Synchrone : l'utilisateur doit attendre le chargement de la page.
- Le modèle classique des pages web (cliquer, attendre, rafraîchir). éléments.

- Application web : un site web dynamique qui imite le comportement d'une application de bureau.
  - Offre une interaction continue à l'utilisateur plutôt qu'une suite de pages différentes.
  - Exemples : Gmail, Google Maps, Google Docs and Spreadsheets, Flickr, A9.
- Ajax : Asynchronous JavaScript and XML
  - Pas un langage de programmation mais une façon d'utiliser JavaScript.
  - Permet de récupérer des données du serveur en tâche de fond.
  - Permet de mettre à jour une page dynamiquement sans la recharger.
  - Un modèle différent du modèle "cliquer-attendre-rafraîchir".

# Introduction à AJAX

## Communication Asynchrone



- Asynchrone : l'utilisateur peut interagir avec la page pendant le chargement de nouvelles données.
  - Un autre modèle pour les pages web.

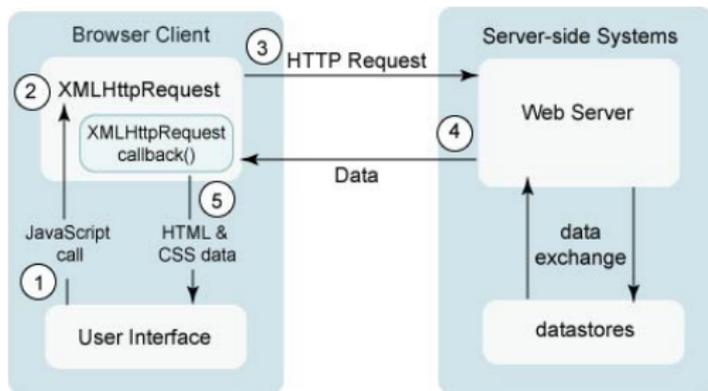
# Introduction à AJAX

## L'objet XMLHttpRequest

- JavaScript contient l'objet `XMLHttpRequest` qui peut envoyer des requêtes au serveur.
  - Disponible dans tous les navigateurs.
  - Sauf dans les vieilles versions (en particulier IE6-).
  - Pourtant inventé par Microsoft pour les liens entre IE et Outlook...
- Les requêtes sont asynchrones (exécutées en tâche de fond).
- Les données récupérées peuvent être utilisées pour modifier le HTML via JavaScript/DOM.

# Introduction à AJAX

## Déroulement d'une requête Ajax



- 1 L'utilisateur clique et cet évènement appelle une fonction.
- 2 Cette fonction fabrique un objet XMLHttpRequest.
- 3 L'objet XMLHttpRequest invoque un script sur le serveur.
- 4 Le script produit de la donnée et la transmet à l'objet XMLHttpRequest.
- 5 XMLHttpRequest déclenche un évènement à la réception des données.
  - La fonction associée à cet évènement est appelée.
  - On appelle cette fonction un callback.
- 6 Le callback traite les données reçues et modifie le HTML en conséquence.

- Ajax ne fonctionne qu'au travers d'un serveur web.
  - (*ne fonctionne pas en mode `file`*).
- Ajax ne peut effectuer des requêtes que vers le même serveur qui a servi la page dans laquelle il se trouve.
  - `http://www.foo.com/a/b/c.html` ne peut faire des requêtes Ajax qu'à l'adresse `http://www.foo.com`.

Un constructeur écrit pour simplifier les requêtes Ajax.

### HTML (Exemple)

```
new simpleAjax(url, method, params, onSuccess, onFailure)
```

- `url` : le script invoqué côté serveur (la requête).
- `method` : la methode GET ou POST ( `"get"` ou `"post"` ).
- `params` : les paramètres de la requête (une chaîne de caractères de la forme `param1=val1&param2=val2&...`
- `onSuccess` : la fonction appelée au retour de la requête (et qui va traiter la valeur retournée).
- `onFailure` : la fonction appelée en cas d'erreur.

Les exemples suivants sont fournis pour comprendre comment utiliser l'objet simpleAjax.

- ❶ Exemple 1 : Comment récupérer périodiquement des informations (aléatoires) depuis le serveur (requêtes Ajax sans paramètre).
- ❷ Exemple 2 : Comment échanger des informations simples avec le serveur (requêtes Ajax avec paramètres, sans JSON).
- ❸ Exemple 3 : Comment échanger des informations structurées avec le serveur (requêtes Ajax avec paramètres et en JSON).

Pour tester ces exemples, téléchargez l'archive [exemples-ajax.zip](#) et dézippez-la dans votre répertoire web.

## 1. Introduction à AJAX

- Communication Synchrone
- AJAX
- Communication Asynchrone
- L'objet XMLHttpRequest
- Déroulement d'une requête Ajax
- Sécurité en Ajax

- L'objet XMLHttpRequest

## 2. Introduction au format JSON

- JSON : JavaScript Object Notation
- JavaScript et les objets
- Syntaxe JSON
- Ecrire et lire du JSON
- JSON en PHP
- Envoi de valeur JSON en PHP

# Introduction au format JSON

JSON : JavaScript Object Notation

JavaScript Object Notation (JSON) est un format de données pour encoder des valeurs JavaScript.

- Inventé par le gourou JavaScript **Douglas Crockford** (Yahoo).
- Intégré dans tous les navigateurs et disponible sous forme de bibliothèque dans tous les langages de programmation.
- Très populaire et très utilisé à cause de sa simplicité.

Les fonctions `setTimeout` and `setInterval` retournent une valeur unique qui identifie le *timer*.

- Cette valeur est passée en paramètre à la fonction `clearTimeout` ou `clearInterval` pour arrêter et détruire le *timer*.

# Introduction au format JSON

## JavaScript et les objets

### JS (Modèle)

```
let name = {  
  fieldName : value,  
  ...  
};
```

### JS (Exemple)

```
let pt = {  
  x : 4,  
  y : 3 };  
pt.z = -1;  
alert("(" + pt.x + ", " + pt.y + ", " + pt.z + ")"); // (4, 3, -1)
```

- En JavaScript, on peut créer des objets sans créer une classe (juste en utilisant les caractères '{' et '}').
- On peut leur ajouter dynamiquement des attributs (comme `z`).

# Introduction au format JSON

## JavaScript et les objets

### JS (Exemple)

```
var person = {  
  name : "Philip J. Fry",           // string  
  age : 23,                         // number  
  "weight" : 172.5,                 // number  
  friends : ["Farnsworth", "Hermes", "Zoidberg"], // array  
  getBeloved : function() { return this.name + " loves Leela"; }  
};  
alert(person.age);                  // 23  
alert(person["weight"]);            // 172.5  
alert(person.friends[2]);           // Zoidberg  
alert(person.getBeloved());         // Philip J. Fry loves Leela
```

- Un objet peut avoir des méthodes qui utilise `this` comme référence à l'objet sur lequel on les appelle.
- On peut accéder aux attributs par `.fieldName` ou `["fieldName"]`.

# Introduction au format JSON

## Exemple de donnée JSON

### JSON (Exemple)

```
{
  "private" : "true",
  "from" : "Chakib Hamid (chakib@hamid.com)",
  "to" : [
    "Anis Rochdi (anis@rochdi.com)",
    "Aya Sophia (aya@sophia.com)"
  ],
  "subject" : "Tomorrow's \" Birthday Bash \" event!",
  "message" : {
    "language" : "english",
    "text" : "Hey guys, don't forget to call me this weekend!"
  }
}
```

# Introduction au format JSON

## Exemple de donnée JSON

### JS (Exemple)

```
var student = { // pas d'affectation
  "first_name" : 'Bart', // chaînes entre " (et non ')
  last_name : "Simpson", // attributs entre "
  "birthdate" : new Date("April 1, 1983"), // pas d'instance de classe
  "enroll" : function() { // pas de fonction
    this.enrolled = true;
  }
};
```

- Les chaînes doivent être délimitées avec le caractère `"`.
- Les attributs (les champs) doivent être placés entre des `"`.
- Types interdits : `Function`, `Date`, `RegExp`, `Error`.
- Types autorisés : `Number`, `String`, `Boolean`, `Array`, `Object`, `Null`.
- Il existe des validateurs pour JSON : [JSONLint](#), [JSON Formatter & Validator](#), [Free Formatter](#), [JSON Validator](#).

# Introduction au format JSON

## Exemple de donnée JSON

- `JSON.parse(string)` : Convertit une chaîne JSON en valeur JavaScript et retourne cette valeur.
- `JSON.stringify(object)` : Convertit une valeur JavaScript en chaîne JSON et retourne cette chaîne.

Pour envoyer un paramètre complexe au serveur :

- 1 On fabrique un objet JavaScript qui encode le paramètre.
- 2 A l'aide la fonction `JSON.stringify` on convertit l'objet JavaScript en chaîne JSON.
- 3 On effectue un requête Ajax au serveur avec la chaîne JSON en paramètre.

Pour recevoir la réponse complexe du serveur :

- 1 On reçoit la donnée en retour au format JSON.
- 2 A l'aide la fonction `JSON.parse` on convertit la chaîne JSON en objet JavaScript.
- 3 On utilise cette valeur pour mettre à jour le HTML.

# Exemple d'utilisation de XMLHttpRequest

## Chargement de JSON en JavaScript

### JS (Exemple)

```
var xhttp = new XMLHttpRequest();
xhttp.onload = function() {
  if (this.status == 200) {
    try {
      let json = JSON.parse(xhttp.responseText);
      let clients = json.clients;
      document.querySelector(#clients).innerHTML = "";
      for (let c = 0; c < clients.length; c++) {
        document.querySelector(#clients).innerHTML +=
          "<option>" + clients[c].nom + "</option>";
      }
    } catch (e) {
      // TODO : traitement en cas d'erreur
    } finally {
      // TODO : programme à executer quelque soit la situation
    }
  }
}
```

# Introduction au format JSON

## JSON en PHP

- `json_decode(string)` : Convertit une chaîne JSON en valeur PHP sous la forme d'un tableau associatif et retourne cette valeur (comme `JSON.parse` en JavaScript).
- `json_encode(object)` : Convertit une valeur PHP (valeur primitive, tableau associatif ou objet) en chaîne JSON et retourne cette chaîne (comme `JSON.stringify` en JavaScript).
- **NB** : `json_encode` convertit les tableaux associatifs en objets et les tableaux indicés en tableaux.

# Introduction au format JSON

Envoi de valeur JSON en PHP

## PHP (Exemple)

```
<?php
$data = array(
    "library" => "Odegaard",
    "category" => "fantasy",
    "year" => 2012,
    "books" => array(
        array("title" => "Harry Potter",
            "author" => "J.K. Rowling"),
        array("title" => "The Hobbit",
            "author" => "J.R.R. Tolkien"),
        array("title" => "Game of Thrones",
            "author" => "George R. R. Martin"),
    )
);
header("Content-type : application/json");
echo <em>json_encode($data)</em>;
?>
```

# Introduction au format JSON

Envoi de valeur JSON en PHP

Le script précédent produit la valeur JSON suivante :

## JSON (Exemple)

```
{  
  "library" : "Odegaard",  
  "category" : "fantasy",  
  "year" : 2012,  
  "books" : [  
    "title" : "Harry Potter", "author" : "J.K. Rowling",  
    "title" : "The Hobbit", "author" : "J.R.R. Tolkien",  
    "title" : "Game of Thrones", "author" : "George R. R. Martin",  
    "title" : "Dragons of Krynn", "author" : "Margaret Weis"  
  ]  
}
```