



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
Université Ibn Khaldoun – Tiaret
Faculté des mathématiques et d'informatique
Département d'Informatique



AJAX, JSON ET JQUERY

ZERROUKI DJAMEL

djameljimmizerrouki@gmail.com

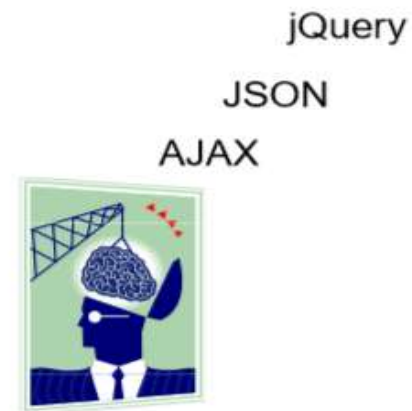
Année universitaire 2018-2019

Introduction générale



LE BUT DE CETTE PRÉSENTATION

- Comprendre JSON
- Comprendre et savoir utiliser jQuery
- Comprendre et savoir utiliser AJAX
- Comprendre l'utilisation d'AJAX et de JSON avec jQuery



JSON-I

- Le texte est un format de transmission de données sur Internet
- Besoin de transmettre des données complexes (structurées)
 - Informations sur le livre, informations sur le film, etc.
- Idée:
 - Encoder les données
 - Transmettre au format texte
 - Décoder les données

JSON-2

- Il signifie «JavaScript Object Notation» est un format d'échange de données basé sur du **texte**.
- **JSON** est construit sur deux structures:
 - Objet: Collection **nom** et **valeur** de paires
 - Tableau: une **liste** ordonnée de valeurs

JSON-3

■ Exemples:

```
{  
  "name" : "djamel"  
}
```

```
["zerrouki ", "Djamel", 06691234567]
```

```
{  
  "course-name" : "XML",  
  "times": ["8-9", "11-12", "14-15"]  
}
```

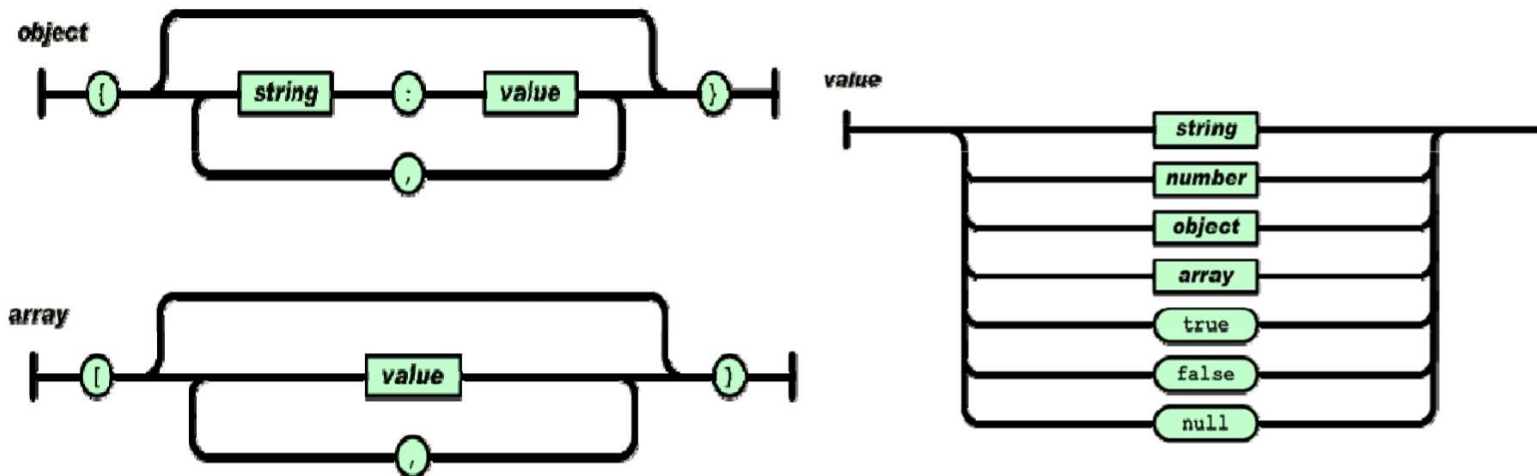


Figure: Les diagrammes de la syntaxe de JSON [1]

JSON-4

- Pourquoi le format JSON est-il plus facile à évaluer?

```
Var a = {  
  "John":  
  {  
    "label": "John",  
    "data": [[1880, 0.081536], [1881, 0.08098], [1882, 0.078308]]  
  },  
  "James":  
  {  
    "label": "James",  
    "data": [[1880, 0.050053], [1881, 0.05025], [1882, 0.048278]]  
  }  
}
```

- Pour obtenir la valeur 1880 marquée en rouge:
 - Nous pouvons utiliser **a.John.data [0] [0]**

JQUERY-I



JQUERY-2

- jQuery: Une bibliothèque JavaScript a été publiée en 2006
- Idée:Assembler un ensemble de fonctions
 - Naviguer dans un document
 - Gestion des événements
 - Sélectionner des éléments DOM
 - Créer des animations
 - Développement Ajax

JQUERY-3

Usage:

- **Étape 1:** Téléchargez la dernière version de jQuery sur [jQuery.com](https://jquery.com).
- **Étape 2:** liez le fichier .js téléchargé.
 - `<script type = "text / javascript" src = "jquery-3.3.1.js"> </ script>`
- **Étape 3:** écrivez votre code.

JQUERY-4

- **APIs :**
 - jQuery core
 - Selectors
 - Attributes
 - Traversing
 - Manipulation
 - Css
 - Events
 - Effects
 - Ajax
 - Utiliti
 - Utilities
 - jQueryUI

CORE

- **jQuery functions:**
 - * `$(expression, [context])`
- **`$(html)`**
 - * `$('<div id="loading">Loading...</div>')`
- **`$(elements)`**
 - * `$(documentbody).css('background' 'red');`

ATTRIBUTES

- **Attr:**

attr(name), attr(properties) attr(key,value), removeAttr(name).

- **Class:**

addClass(class), removeClass(class)

- **HTML:**

html(), html(value)

- **Text:**

text(), text(value)

- **Value:**

val(), val(value)

EVENTS & EFFECTS

■ Events:

✓ Page load:

- ready(fn)

✓ Event handling:

- bind(type, fn), unbind(type, fn), trigger(event)

✓ Event helpers:

- click(), click(fn), mousedown(fn), mouseout(fn), ...

■ Effects:

✓ Basics:

- show() show(speed) hide() toggle() toggle(speed)

✓ Fading:

- fadeIn(speed), fadeOut(speed), fadeTo(speed, opacity)

AJAX

Qu'est-ce que l'AJAX ?



AJAX

- Il signifie “Asynchronous JavaScript and XML”
- Ce n'est pas une nouvelle technologie. C'est plusieurs technologies.
 - présentation basée sur les normes: [XHTML](#) et [CSS](#) .
 - affichage dynamique et interaction: [DOM](#) .
 - échange et manipulation de données: [XML](#) et [XSLT](#) .
 - récupération de données asynchrone: [XMLHttpRequest](#) .
 - liant tout ensemble: [JavaScript](#) .

utilisation de **AJAX** sur **JQUERY** et **JSON**

```
$.ajax({  
  method: "POST",  
  url: "some.php",  
  data: { name: "Djamel", location: "Tiaret" }  
})  
.done(function( msg ) {  
  alert( "les données sauvegarder : " + msg );  
});
```

Réalisation



Simple exemple

- Nous avons utilisé le site <https://jsonplaceholder.typicode.com/> qui permet de retourner les données sur format JSON .
- l'URL: <https://jsonplaceholder.typicode.com/posts> qui permet de retourner 100 posts "Object" de format **JSON** ,sur la forme :

```
{  
  "userId": 1,  
  "id": 2,  
  "title": "qui est esse",  
  "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea  
dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui  
aperiam non debitis possimus qui neque nisi nulla"  
}
```

Page home.html

The screenshot shows a web browser window with the address bar displaying `localhost/projects/jQueryProject/home.html#`. The page content includes a blue button labeled "Get Posts" and a list of 20 items, each starting with a double bullet point. The first item is "sunt aut facere repellat provident occaecati excepturi optio reprehenderit". The second item is "qui est esse", which is highlighted in yellow. The remaining items are standard Lorem Ipsum text. To the right of the list is a table with two columns: "Title" and "Body". The first row of the table has "qui est esse" in the "Title" column and "est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla" in the "Body" column. The "Title" cell is also highlighted in yellow.

Get Posts

- sunt aut facere repellat provident occaecati excepturi optio reprehenderit
- qui est esse
- ea molestias quasi exercitationem repellat qui ipsa sit aut
- eum et est occaecati
- nesciunt quas odio
- dolorem eum magni eos aperiam quia
- magnam facilis autem
- dolorem dolore est ipsam
- nesciunt iure omnis dolorem tempora et accusantium
- optio molestias id quia eum
- et ea vero quia laudantium autem
- in quibusdam tempore odit est dolorem
- dolorum ut in voluptas mollitia et saepe quo animi
- voluptatem eligendi optio
- eveniet quod temporibus
- sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio
- fugit voluptas sed molestias voluptatem provident
- voluptate et itaque vero tempora molestiae
- adipisci placeat illum aut reiciendis qui
- doloribus ad provident suscipit at
- asperiores ea ipsam voluptatibus modi minima quia sint
- dolor sint quo a velit explicabo quia nam
- maxime id vitae nihil numquam
- autem hic labore sunt dolores incidunt
- rem alias distinctio quo quis

Title	Body
qui est esse	est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

Merci pour
votre attention



Questions

