Javascript &

Le format JSON

Définition

JSON : **J**ava**S**cript **O**bject **N**otation

- JSON est un format de données textuelles dérivé de la notation des objets du langage JavaScript.
- Il permet de représenter de l'information structurée comme le permet XML.
- Créé par Douglas Crockford entre 2002 et 2005, il est décrit par la RFC 7159 de l'IETF.

JSON, Javascript Object Notation

- JavaScript Object Notation ;
- Initialement créé pour la sérialisation et l'échange d'objets JavaScript ;
- Langage pour l'échange de données structurées;
- Format texte indépendant du langage de programmation utilisé pour le manipuler.

Utilisation première : échange de données dans un environnement Web (par exemple applications Ajax)

Extension : sérialisation et stockage de données

Les bases de JSON

Structure de base : paire clef-valeur (key-value)

"titre": "The Social network"

Qu'est-ce qu'une valeur ? On distingue les valeurs atomiques et les valeurs complexes (construites)

Valeurs atomiques : chaînes de caractères (entourées par les classiques guillemets), nombres (entiers, flottants) et valeurs booléennes (true ou false).

"year": 2010

"oscar": false

Les bases de JSON (suite)

Valeurs complexes : les objets.

Un objet est un ensemble de paires clef-valeur.

Au sein d'un ensemble de paires, une clef apparaît au plus une fois (NB : les types de valeurs peuvent être distincts).

```
{"last_name": "SIMIER", "first_name": "Philippe"}
```

Un objet peut être utilisé comme valeur (dite complexe) dans une paire clef-valeur.

```
"teacher": { "last_name": "SIMIER",
             "first_name": "Philippe",
              "birth_date": 1960
 }
```

Les bases de JSON (suite)

Valeurs complexes : les tableaux.

Un tableau (array) est une liste de valeurs (dont le type n'est pas forcément le même).

"actors": ["Eisenberg", "Mara", "Garfield", "Timberlake"]

Imbrication sans limite :

- tableaux de tableaux,
- tableaux d'objets contenant eux-mêmes des tableaux, etc.

Les bases de JSON (suite)

Un document est un objet. Il peut être défini par des objets et tableaux imbriqués autant de fois que nécessaire.

```
{   "title": "The Social network",
     "summary": "On a fall night in 2003, Harvard undergrad and \n
                      programming genius Mark Zuckerberg sits down at his \n
                     computer and heatedly begins working on a new idea. (...)",
       "year": 2010,
       "director": {"last_name": "Fincher",
                      "first_name": "David"},
       "actors": [
             {"first_name": "Jesse", "last_name": "Eisenberg"},
             {"first_name": "Rooney", "last_name": "Mara"}
             ]

 }
```

JSON

- JSON est léger
- Il est possible d'en générer très facilement du côté du serveur - éventuellement après interrogation d'une base de données - car il s'agit bien d'un format texte pur.
- Il est facile à parser pour n’importe quel langage de programmation Javascript Php C C++ Python

JSON & Javascript

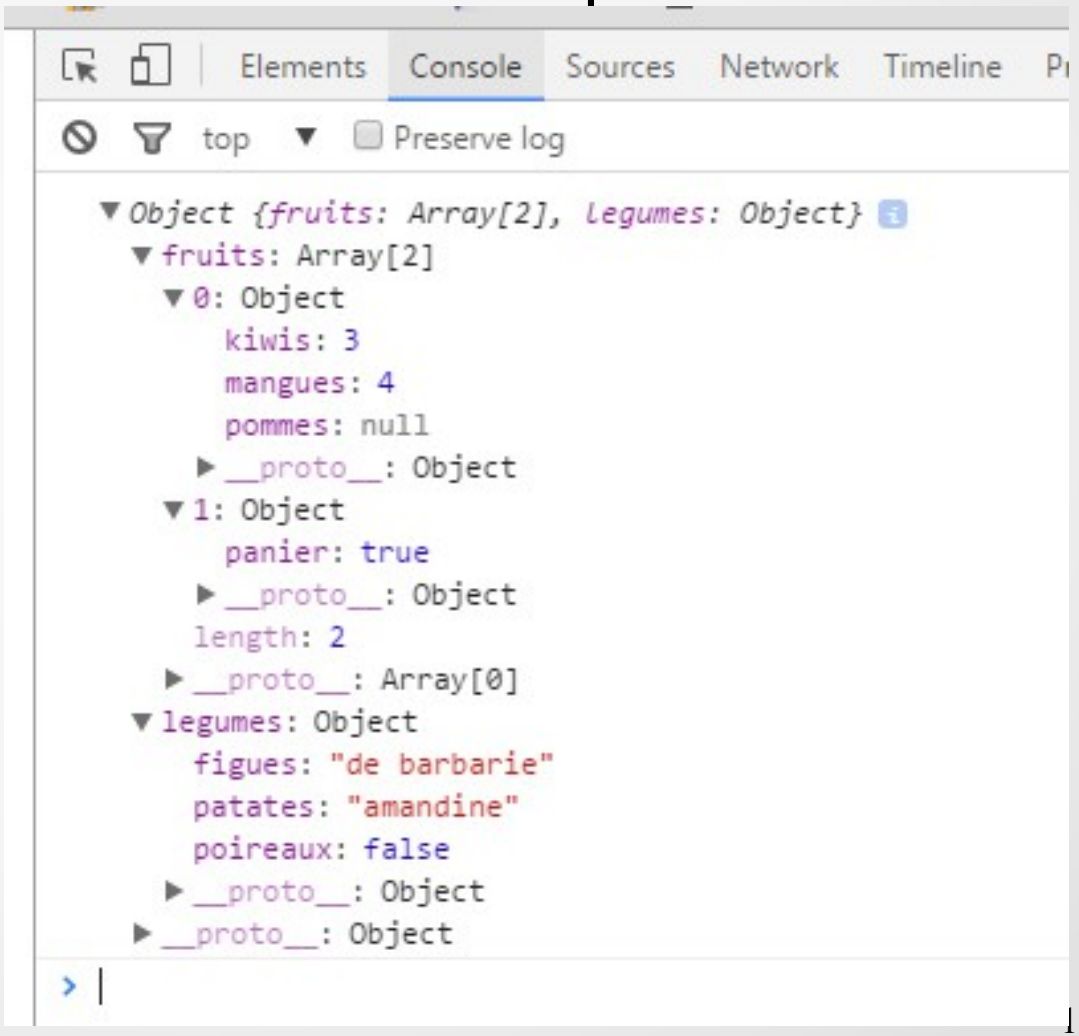
- Avec JavaScript, une déclaration peut être effectuée de la façon suivante pour stocker un objet dans une variable :

```
var courses = {
 "fruits": [
      { "kiwis": 3,
              "mangues": 4,
             "pommes": null
      },
       { "panier": true },
      ],

 "legumes":

      { "patates": "amandine",

       "figues": "de barbarie",

      "poireaux": false

       }

 };
```

Json & Javascript

On peut visualiser le contenu de la variable sous forme d'arborescence grâce à la console JavaScript _

```
console.log(courses);
```



chaînes JSON et parsing natif

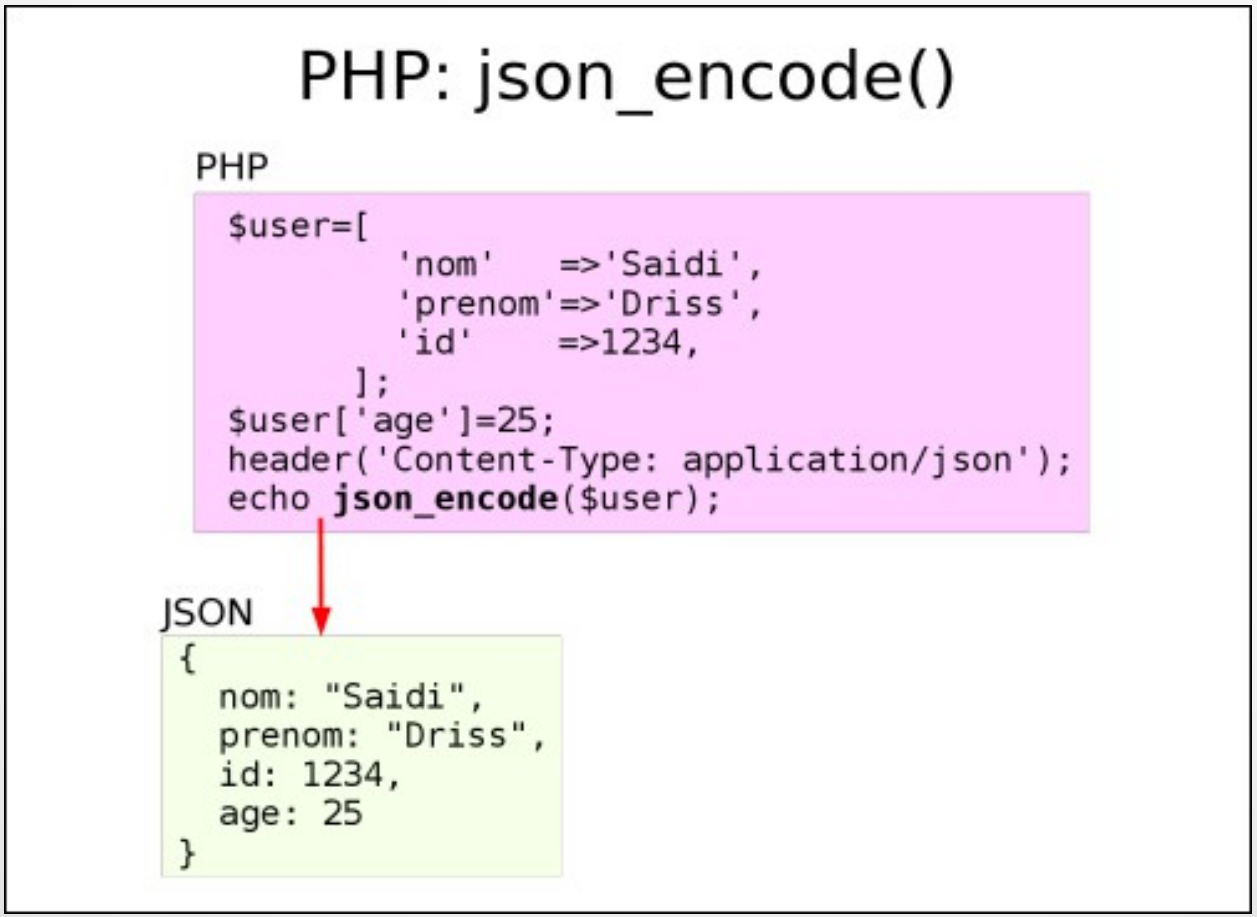
La plupart des navigateurs récents intègrent un parseur natif

Exemple :

```
var textejson = '{"liste":[{"kiwis": 3},{"pommes" : "golden"}], "where" : "Supermarché"}';

var courses = JSON.parse(textejson);


console.log(courses);
```

Json & php

- Tous les principaux langages fournissent des méthodes pour encoder et décoder le JSON.



PHP exemple 1 tableau d'entiers

- <?php

  $array = [1, 2, 3];

  echo json_encode(new ArrayValue($array), JSON_PRETTY_PRINT);

  ?>

- L'exemple ci-dessus va afficher :

  [1, 2, 3]

PHP tableau associatif

- Les tableaux associatifs fonctionnent sur le même principe que les tableaux, sauf qu'au lieu de numéroter les cases, elles sont étiqueter en leur donnant à chacune un nom différent.
- <?php // $coordonnees
$coordonnees = array (
    'prenom' => 'Jean',
    'nom' => 'Dupont',
    'adresse' => '3 Rue du Paradis',
    'ville' => 'Marseille');
?>

Exemple tableau associatif

- <?php // coordonnees de Dupont
  $coordonnees = array (
  'prenom' => 'Jean',
  'nom' => 'Dupont',
  'adresse' => '3 Rue du Paradis',
  'ville' => 'Marseille');

  echo json_encode($array, JSON_PRETTY_PRINT);
  ?>

- **L'exemple va afficher :**

  { "prenom": "Jean", "nom": "Dupont", "adresse": "3 Rue du Paradis", "ville": "Marseille" }

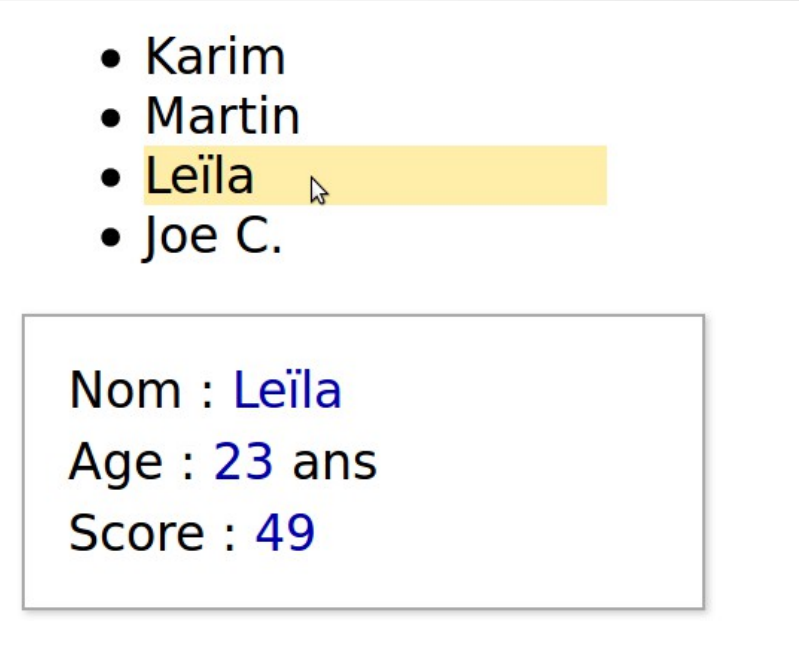
Exemple 2 tableau associatif

- function afficheListeRegion() { // connexion BD

```
$bdd = connexionBD();
$requete = $bdd->query("select * from regions order by region_nom;");
$tabRegion=array();
while ($tab = $requete->fetch()) {
    array_push($tabRegion, array( 'idRegion'=>$tab['regions_id'], 'nomRegion'=> utf8_encode($tab['region_nom'])));
}
```

Exercices JSON : noms

- Dans cet exercice, on veut chercher des informations sur un joueur quand on clique sur son nom.
- Ces informations (nom, age, score) sont sur le serveur.
- Il faut donc utiliser une requête AJAX.



GET ou POST ?

- On veut chercher des informations sur un joueur quand on clique sur son nom.

- Quelle méthode faut-il utiliser pour la requête AJAX ?

  Réponse ?

•Créez le fichier html suivant

```
<!DOCTYPE html>
<html>
	<head>
		<meta charset="utf-8">
		<title>utilisateurs</title>
		<link type="text/css" rel="stylesheet" href="utilisateurs.css"/>
		<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>
		<script src="utilisateurs.js"></script>
</head>
<body>
	<!---------------------->
		<ul id="utilisateurs">
			<li data-uid="karim" >Karim </li>
			<li data-uid="martin">Martin</li>
			<li data-uid="leila" >Leïla </li>
			<li data-uid="joe"   >Joe C.</li>
		</ul>
	<div id="affichage">
		<p>Nom :   <span id="nom"  ></span>          </p>
		<p>Age :   <span id="age"  ></span> ans     </p>
		<p>Score : <span id="score"></span>         </p>
	</div>
</body>
</html>
```

Créer le fichier utilisateurs.css

```
body { font-family: sans; }

#utilisateurs {
	width: 150px;
	cursor: pointer;
}

#utilisateurs li:hover{
	background-color: #fea;
}

#affichage {
	background-color: white;
	width: 200px;
	border: 1px solid #aaa;
	box-shadow: 1px 1px 2px rgba(0,0,0,.2);
	padding: 10px; }

#affichage p { margin: 4px; }

#affichage span { color: #00a; }
```

utilisateurs.php (à compléter)

```
<?php
// Une liste d'utilisateurs, juste pour l'exemple.
// En pratique on chercherait dans une base de données.

$utilisateurs=array(
   'joe'   =>array('nom'=>'Joe C.','score'=>34,'age'=>22),
   'martin'=>array('nom'=>'Martin','score'=>3 ,'age'=>7 ),
   'karim' =>array('nom'=>'Karim' ,'score'=>45,'age'=>19),
   'leila' =>array('nom'=>'Leïla' ,'score'=>49,'age'=>23),

);
```

…… A compléter

Dans cet exercice le serveur veut envoyer du JSON au client.

En regardant sur votre cours (ou sur le web), quel type MIME faut-il utiliser ?

utilisateurs.js

```
console.log("Ce programme JS vient d'être chargé");
$(document).ready(function()
{
console.log("Le document est pret");

        $('#utilisateurs li').    À compléter

console.log("La mise en place est finie. En attente d'événements...");
});
```

Pour aller plus loin

- Tout sur JSON : http://json.org/
- Un validateur de documents JSON : http://jsonlint.com/