

# 420-1WE-BB

## Web dynamique

JSON

Tirées des notes de Jean-François Brodeur

# Les objets en Javascript

- ▶ Un objet est un ensemble de propriétés et une propriété est une association entre un nom (aussi appelé clé) et une valeur.
- ▶ La valeur d'une propriété peut être une fonction, auquel cas la propriété peut être appelée « méthode ».
- ▶ En plus des objets natifs fournis par l'environnement, il est possible de construire ses propres objets.
- ▶ Ex.: 

```
personne = {nom: "Jean-François Brodeur",  
            métier: "professeur"};  
console.log(personne)
```

# Qu'est que le format JSON

- ▶ Le JavaScript Object Notation (JSON) est un format standard utilisé pour représenter des données structurées de façon semblable aux objets Javascript, qui a été popularisé par Douglas Crockford..
- ▶ Il est habituellement utilisé pour structurer et transmettre des données sur des sites web (par exemple, envoyer des données depuis un serveur vers un client afin de les afficher sur une page web ou vice versa).

# JSON

- ▶ Pour accéder aux valeurs des propriétés on utilise deux notations.
  - ▶ `personne.nom`
  - ▶ `personne["nom"]`
- ▶ La seconde notation pourra être utilisée lorsque la propriété sera elle-même une variable

`prop="nom";`

`personne[prop];`

# JSON

package.json X



```
1  {  
2    "name": "myapplication",  
3    "description": "some description here",  
4    "version": "0.0.1",  
5    "private": true,  
6    "scripts": {  
7      "start": "node ./bin/www"  
8    },  
9    "dependencies": {  
10     "express": "~4.12.2",  
11     "jade": "~1.9.2"  
12   }  
13 }
```

# JSON

- ▶ Le JSON se présente sous la forme d'une chaîne de caractères —utile lorsque vous souhaitez transmettre les données sur un réseau.
- ▶ Il a donc besoin d'être converti en un objet JavaScript natif lorsque vous souhaitez accéder aux données.
- ▶ Ce n'est pas vraiment un souci puisque le JavaScript fournit un objet global JSON disposant des méthodes pour assurer la conversion entre les deux.

# JSON

- ▶ Convertir une chaîne de caractères en un objet natif se nomme analyse syntaxique (parsage) tandis que le contraire porte le nom de la linéarisation (stringification).
- ▶ Un objet JSON peut être stocké dans son propre fichier qui se présente simplement sous la forme d'un fichier texte avec l'extension `.json` et le MIME type `application/json`.

# Structure du JSON

- ▶ Un objet JSON n'était ni plus ni moins qu'un objet Javascript tout à fait normal et c'est généralement le cas.
- ▶ Un objet JSON accepte comme valeur les mêmes types de données de base que tout autre objet Javascript — chaînes de caractères, nombres, tableaux, booléens et tout autre objet littéral.



# Structure du JSON

- Cela vous permet de hiérarchiser vos données comme ceci :

```
{ "squadName": "Super hero squad",  
  "homeTown": "Metro City",  
  "formed": 2016,  
  "secretBase": "Super tower",  
  "active": true  
}
```

# Les tableaux JSON

- un tableau peut également être un objet JSON valide, par exemple :

```
[ { "name": "Molecule Man",  
  "age": 29,  
  "secretIdentity": "Dan Jukes",  
  "powers": [ "Radiation resistance", "Turning tiny", "Radiation blast" ] },  
  { "name": "Madame Uppercut",  
    "age": 39,  
    "secretIdentity": "Jane Wilson",  
    "powers": [ "Million tonne punch", "Damage resistance", "Superhuman reflexes" ]  
  } ]
```

# Accéder aux éléments du tableaux

- Le code tableau de la page précédent est une notation JSON parfaitement valide. Vous n'aurez qu'à accéder aux éléments de votre tableau en commençant avec un index, par exemple : `[0]["powers"][0]`.

# Règles des éléments JSON

- ▶ Un objet JSON est uniquement un format de données — il ne contient que des propriétés mais pas de méthodes.
- ▶ La notation JSON nécessite l'usage des guillemets pour être valide. Il est obligatoire d'utiliser des guillemets et non les apostrophes autour des chaînes de caractères et des noms de propriétés.
- ▶ Une simple virgule ou un double point mal placé peut rendre votre fichier JSON invalide et non fonctionnel. Soyez très attentif aux données que vous utilisez (bien que le JSON généré automatiquement par un programme sera moins enclin à contenir des erreurs, à partir du moment où le programme est codé correctement).
- ▶ Vous pouvez utiliser une application comme JSONLint pour valider votre code JSON.

# Règles JSON

- ▶ Dans l'absolu, le JSON peut prendre la forme de n'importe quel type de données qui serait valide pour être contenu dans du JSON et non juste des tableaux ou des objets. Ainsi, par exemple, une simple chaîne de caractères ou un nombre serait un objet JSON valide.
- ▶ Contrairement au JavaScript dans lequel les propriétés (keys) non entourées de guillemets peuvent être utilisées, en JSON, seules les chaînes de caractères entourées de guillemets peuvent être utilisées en tant que propriétés.

# Exercices

- ▶ Manipuler le JSON avec un exemple
- ▶ <https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON>
- ▶ Aller vers le centre et effectuer les exercices de manipulation du JSON