



1/ Introduction

JSON (*JavaScript Object Notation*) est un format d'échange de données léger et performant qui a largement remplacé le format **XML** beaucoup plus complexe à manipuler.

Tout comme la table d'une base de données, **JSON** peut représenter des nombres, des booléens, des chaînes, la valeur *null* - structurellement composé d'objets ou de tableaux constitués par ces valeurs (ou d'autres tableaux et objets s'imbriquant).

Par contre, JSON ne représente pas nativement des types de données plus complexes tels que des fonctions, des expressions régulières, des dates, etc. Il peut être généré par PHP et utilisé via JavaScript afin d'afficher dynamiquement des contenus à partir d'AJAX, par exemple.

La structure d'un fichier JSON se compose sous les 2 formes suivantes :

→ Un **objet** commence avec `{` et se termine avec `}`. Les noms uniques sont suivis de `:` et les paires `"nom" : "valeur"` sont séparées par des `,`

→ Un **tableau** est une collection ordonnée de valeurs. Un tableau commence avec `[` et se termine avec `]`. Chaque valeurs sont séparées par des `,`

Une valeur peut être une **chaîne de caractères** entourées par des guillemets doubles, un **nombre**, un **booléen**, la valeur *null*, un **objet** ou un **tableau**.

Exemple de données au format **JSON** :

```
{
  "prenom": "Bob",
  "nom": "Merlin",
  "age": "32",
  "email": "bob@server.com",
  "adresse": {
    "rue": "1, rue de la rue",
    "ville": "Maville",
    "cp": "1234",
    "pays": "Monpays"
  },
  "photos_perso": [
    "01_e5f4d2g1a8.jpg",
    "02_d5r7for8gz.jpg",
    "03_d455z4gs5s.jpg"
  ]
}
```



→ Copier le données ci-dessus et créer un fichier : **data.json** au format UT8 (sans BOM). Respectez l'indentation si elle n'a été pas été mise en forme.

2/ Interprétation du JSON sous PHP

L'interprétation, et donc le traitement des fichiers JSON sous PHP se fait soit par :

- **Objet** : `$data->mavaleur`
- **Association** : `$data[0]["mavaleur"]`

Le format JSON est donc par nature retourné et traité sous forme de tableau.

Exemple d'importation du fichier « **data.json** » avec la fonction `json_decode()` :

```
<?php
$data = file_get_contents('./data.json');
$json = json_decode($data);

echo '<pre>'; //formatage pour l'HTML
print_r($json); //on décompose le tableau pour le debug
echo "<h1>" . $json->email . "</h1>";
?>
```

Le même exemple avec un retour en tableau associatif :

```
<?php
$data = file_get_contents('./data.json');
//TRUE : les valeurs JSON seront retournées dans un tableau associatif :
$json = json_decode($data, TRUE);

echo '<pre>'; //formatage pour l'HTML
print_r($json); //on décompose le tableau pour le debug
echo "<h1>" . $json["email"] . "</h1>";
?>
```



Comparez les deux types de retours précédant. Quelles sont les différences que l'on peut trouver ?

On peut recomposer les données du fichier JSON par association de valeurs :

```
<?php
$data = file_get_contents('./data.json');
$json = json_decode($data, TRUE);

echo "<h1>" . $json["prenom"] . $json["nom"] . "</h1>";
echo "<li>" . "Rue : " . $json["adresse"]["rue"] . "</li>";
echo "<li>" . "Avatar 2 : " . $json["photos_perso"] . "</li>";
?>
```



- En s'inspirant de l'exemple ci-dessus, comment afficher en plus : l'âge, l'email, et la ville ? Comment afficher correctement la photo perso n°2 ? S'aider avec `print_r()`

- Selon le fichier JSON ci-dessous, comment afficher sur PHP la valeur en jaune contenu dans : « *niveau moyen* » ?

```
{
  "jeux": [
    {
      "utilisateur": {
        "nom": "Robert",
        "age": "32",
        "id": "1234"
      },
      "mon_soft": {
        "licence": "acquise",
        "nb_utilisation": "8",
        "niveau": [
          {
            "debutant": "oui",
            "moyen": "encours",
            "senior": "non"
          }
        ]
      }
    }
  ]
}
```

En reprenant l'exemple en page 1 (**data.json**), tester le code ci-dessous :

```
<?php

$data = file_get_contents('./data.json');
$json = json_decode($data, TRUE);
echo '</pre><pre>Elements :<br />';

foreach ($json as $key => $value) {

    echo $key . " ==> " . $value . "<br>";

    if ($key === "adresse") {
        //print_r($value);
        foreach ($value as $ad_key => $ad_value) {
            echo $key . " : " . $ad_key . " ==> " . $ad_value . "<br>";
        }
    }

    if ($key === "photos_perso") {
        //print_r($value);
        foreach ($value as $img_key => $img_value) {
            echo $key . " : " . $img_key . " ==> " . $img_value . "<br>";
        }
    }

}

?>
```



Comment afficher correctement au format HTML le profil complet de l'utilisateur « Bob » avec son avatar n°2 sélectionné ? Choisir trois images au choix pour compléter l'exercice.

3/ Mise à jour d'une donnée JSON

La mise à jour se fait directement sur le tableau importé. Ici la clé étant le « *nom du paramètre* », et la valeur étant « *son contenu* », ex : **ma_clé** = **ma_valeur**.

- Si l'on souhaite modifier une valeur : `$json["clé"] = "nouvelle_valeur";`
- Si l'on souhaite retirer une clé, et donc aussi sa valeur on utilise `unset()` : `unset($json["clé"]);`
- Si l'on souhaite ajouter une clé (avec sa valeur en option) : `$json["clé"]["nouvelle_clé"] = "nouvelle_valeur";`

Exemple avec le fichier « **data.json** » et la fonction `json_encode()` pour transformer le tableau au format JSON, et l'enregistrer avec `file_put_contents()` :

```
<?php

$json_file = "./data.json";

$data = file_get_contents($json_file);

$json = json_decode($data, TRUE);

echo "Valeur courante : Nom = <strong>" . $json["nom"] . "</strong><br />";

$json["nom"] = "Martin"; //on modifie une valeur

echo "Nouvelle valeur : Nom = <strong>" . $json["nom"] . "</strong>";

unset($json["adresse"]["pays"]); //on retire une clé

$json["adresse"]["detail"]="Porte de gauche"; //on rajoute une clé + sa valeur

/*
 * On encode les données au format JSON
 * Option 1: JSON_UNESCAPED_UNICODE : retire les antislashes "\"" pour la lisibilité
 * Option 2: JSON_PRETTY_PRINT : Retourne un code sur plusieurs lignes + indentation
 */

$json_string = json_encode($json, JSON_UNESCAPED_UNICODE|JSON_PRETTY_PRINT);

echo "<br />Nouvelle(s) clé(s):valeur(s) dans <strong>".$json_file."</strong> :<br />";
echo '<pre>'.$json_string."</pre>";

//On sauvegarde le fichier :
if ( file_put_contents($json_file, $json_string) ) {
    echo "Fichier <strong>".$json_file."</strong> enregistré avec succès !";
}

?>
```



- Depuis l'exemple proposé ci-dessous, comment créer un fichier JSON nommé :
« **user.json** » - à partir d'un tableau PHP . Lire son contenu avec **print_r()** :

```
{  
  "id": 1234,  
  "admin": true,  
  "user": {  
    "prenom": "Robert",  
    "pseudo": "Bob",  
    "email": "bob@server.com"  
  }  
}
```



- En début de page 3, à partir de l'exemple de structure JSON, comment :

- modifier la valeur de "nb_utilisation" et la remplacer par "9"
- ajouter dynamiquement une clé dans « *le niveau* », soit :
→ "junior " : "oui "
- remplacer la clé "age" par "pseudo" et définir la valeur de "pseudo" par "Bob"
- mettre à jour le fichier **data.json**.
- avec **foreach()** afficher tous les **noms:valeurs** pairs associés
(sans mise en page de l'html) ?