

# Notation JSON

---

```
var personne = {  
  prenom: "p",  
  nom: "n"  
} ← initialiseur d'objet.
```

Les objets peuvent être imbriqués:

```
var personne= {  
  acteur: {  
    prenom: "Louis",  
    nom: "de Funés"  
  },  
  role: "don Salluste"  
}
```

- 
- Un objet : ensemble de couples nom/valeur non ordonnés. Un objet commence par { et se termine par } .
  - Chaque nom est suivi de :(deux-points) et les couples nom/valeur sont séparés par ,
  - Un tableau : collection de valeurs ordonnées. Un tableau commence par [ et se termine par ]. Les valeurs sont séparées par , (virgule).

# Utilisation de JSON

---

## JSON en PHP

- Initialisation du service:
  - `$json = new Services_JSON();`
- Encodage
  - `$json->encode($villes);`
- Décodage
  - `$json->decode($villes)`
- JSON en javascript
  - Bibliothèque: <http://www.json.org/json.js>
  - Array et Object: `toJSONString()` retourne une représentation JSON.
  - Dans le prototype String ajout de la méthode `parseJSON(chaine)` analyse une chaîne JSON et renvoie un objet javascript, false si la chaîne est incorrecte.

# exemple

---

- Php:

```
1 <?php
2 include('json.php');//http://pear.php.net/pepr/pepr-proposal-
  show.php?id=198.
4 $p=$_REQUEST["pays"];
5
6 $villes=$pays[$p];
7
8 // instantiation du service JSON
9 $json = new Services_JSON();
10
11 // Convertir la variable complexe
12 echo $json->encode($villes);
13 ?>
```

- **Javascript**

- t=xhr.responseText.parseJSON();