

OpenLayers 2

JSON

JSON signifie **J**ava**S**cript **O**bject **N**otation. C'est une syntaxe permettant de représenter des **objets JavaScript** sous forme de texte. Les types suivants sont reconnus:

Types	Notation
String	<code>"texte"</code>
Number	<code>2</code>
Boolean	<code>true</code> , <code>false</code>
Null	<code>null</code>
Array	<code>["valeur1", "valeur2"]</code>
Object	<code>{ "attribut1": "valeur1", "propriété2": "valeur2" }</code>

JSON: Exemple

Objet JavaScript	JSON
<pre>let car = { brand: "Reliant", model: "Regal", year: 1962, isOldTimer: true, peopleInside: ["Jon Doe", "Jane Does"], doors: { front: 2, rear: 0 }, airConditioning: null };</pre>	<pre>{ "brand": "Reliant", "model": "Regal", "year": 1962, "isOldTimer": true, "peopleInside": ["Jon Doe", "Jane Does"], "doors": { "front": 2, "rear": 0 }, "airConditioning": null }</pre>

En JSON, les propriétés et les valeurs de type string sont obligatoirement **entre double-guillemets** et les fonctions ne sont pas supportées.

Le **GeoJSON** est du **JSON**. Le SRID d'un GeoJSON est 4326 (WGS84). Voir geojson.io

```
{  
  "type": "FeatureCollection",  
  "features": [{  
    "type": "Feature",  
    "properties": {"id": 28},  
    "geometry": {
```

```

    "type": "LineString",
    "coordinates": [
      [6.92930594086647,47.00624259401693],
      [6.929318010807037,47.00623253424974],
      [6.9293421506881705,47.006233906036286],
      [6.929344832897186,47.00624305127904]
    ]
  },{
    "type": "Feature",
    "properties": {},
    "geometry": {
      "type": "MultiPoint",
      "coordinates": [
        [6.929314658045769,47.0062517392582],
        [6.9293394684791565,47.00625265378224]
      ]
    }
  }
] //features
}

```

OpenLayers: couches vectorielles

Pour définir une couche vectorielle [ol/layer/Vector](#), vous devrez fournir une source vectorielle [ol/source/Vector](#) qui aura ces propriétés:

- `format`:
 - [ol/format/GeoJSON](#)
 - [ol/format/KML](#)
 - [ol/format/GPX](#)
- `url`: source du fichier

Voir *ol-06_couches_vecteur*

Projections

OpenLayers ne connaît que deux projections par défaut:

- EPSG 4326: WGS84 Long., Lat.
- EPSG 3857: Web / Spherical Mercator

Pour pouvoir utiliser le système de coordonnées suisses: **EPSG 2056**, nous allons le déclarer à OpenLayers à l'aide de la librairie **Proj4js**:

- Import de la librairie **Proj4js**
- Déclaration de la projection à l'aide de Proj4js et du site [epsg.io](#), par exemple
- Inscription de la projection auprès d'OpenLayers

Voir *ol-07_projections*

Quelques exemples pour aller plus loin

- **GetFeatureInfo**: voir *ol-08_get_feature_info_text* et *ol-09_get_feature_info_gml*
- **Styles & Labels**: voir *ol-10_labels*
- **Couche WFS**: voir *ol-11_couches_wfs*

- **Couche WMTS de Swisstopo:** voir `ol-11_couches_wmts`
- **Couche ArcGIS REST:** voir `ol-11_couches_arcgis`

OpenLayers: page des exemples

De nombreux exemples fonctionnels sur <https://openlayers.org/en/latest/examples/>

- Popup
- Select
- Animation

Exercices

Faites les exercices `o1-2_coordonnees` et `o1-3_popup`.