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	`"texte"`
Number	`2`
Boolean	`true`, `false`
Null	`null`
Array	`[ "valeur1", "valeur2" ]`
Object	`{ "attribut1": "valeur1", "propriété2": "valeur2" }`

JSON: Exemple

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

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,
          "type": "route",
          "name": "Rue du Moléson"
        },
      "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.html

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.html

Exercices

Faites les exercices `ol-2_coordonnees.html` et `ol-3_popup.html`. Pour l'exercice 3, Visual Studio avec "Go Live" de Live Server permettra de lire le fichier GeoJSON. Sans Visual Studio, vous aurez besoin d'un serveur web.

Quelques exemples pour aller plus loin

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