

Martin DELSINNE

Gestionnaire Data / Développeur

martin.delsinne@geovendee.fr

INITIATION À OPENLAYERS



OpenLayers

- 1 – Nouvelle carte `ol.Map`
- 2 – Ajouter des éléments à la carte `ol.control`
- 3 – Ajouter des couches à la carte `ol.layer`
- 5 – Afficher une popup `ol.Overlay`

Nouvelle carte ol.Map

1. Se rendre sur le site d'OpenLayers - <https://openlayers.org/>
 - Go **QuickStart!**
 - Copier le code HTML dans votre page « index.html » - Sauvegarder ! **CTRL + S** puis ouvrir la page web dans un navigateur

```
<script type="text/javascript">
var map = new ol.Map({
  target: 'map',
  layers: [ new ol.layer.Tile({ source: new ol.source.OSM() }) ],
  view: new ol.View({ center: ol.proj.fromLonLat([37.41, 8.82]), zoom: 4 })
});
</script>
```

- Parcourir l'API et la classe **ol.Map**
 - <https://openlayers.org/en/latest/apidoc/>
- Ajouter l'évènement **click** sur la carte

```
map.on('click', function(event){a
```



Nouvelle carte ol.Map

2. Organiser son code

- Déplacer le style CSS dans style.css et importer la balise **<link>** dans *index.html*
`<link rel="stylesheet" href="css/style.css" type="text/css" >`
- Déplacer le script JS dans script.js et importer la balise **<script>** dans *index.html*
`<script src="js/script.js" type="text/javascript"></script>`

Vérifier que ça fonctionne toujours...

- Redéfinir l'étendue initiale via la propriété center de l'objet ol.View appartenant à Map

```
view: new ol.View({ // Vue initiale
  center: ol.proj.fromLonLat([2.294481, 48.858370]), // Paris WGS84
  zoom: 6
})
```

Nouvelle carte ol.Map

3. Commenter son code !

- La déclaration **const** indique une constante dont la valeur ne peut pas être modifiée par des réaffectations ultérieures

```
// Raster OpenStreetMap
var osm = new ol.source.OSM();

// Map OpenLayers
const map = new ol.Map({
  target: 'map', // nom de la <div> HTML
  // Tableau des couches de la carte
  layers: [
    // 1er élément : Nouvelle couche tuilée OSM
    new ol.layer.Tile({source: osm})
  ],
  view: new ol.View({ // Vue initiale
    center: ol.proj.fromLonLat([2.294481, 48.858370]), // Paris WGS84
    zoom: 6
  })
});
```

Ajouter des éléments à la carte

ol.control

1. La propriété **controls** de l'objet **Map** définit la liste des contrôles (fonctionnalités) attachés à la carte

- Ajouter le control **OverviewMap** :

```
// Tableau des contrôles de la carte
controls: ol.control.defaults().extend([
  // 1er élément : mini-map - vue générale
  new ol.control.OverviewMap({
    layers:[
      new ol.layer.Tile({source: osm})
    ]
  })
]),
```

- La méthode **extend()** indique que l'on ajoute des contrôles (passés en paramètre) à ceux par défaut (Zoom, Rotate et Attribution)

- Ajouter d'autres contrôles : **FullScreen** et **ScaleLine** :

```
new ol.control.FullScreen() // Plein-écran
new ol.control.ScaleLine() // Echelle
```

Ajouter des éléments à la carte

ol.control

2. Afficher les coordonnées X Y

- Ajouter une **<div>** à la page HTML

```
<div id="map" class="map"></div> <!-- carte -->
<div id="mouse-position"></div> <!-- coordonnées -->
```

- Dans **script.js**, créer un nouveau control **ol.control.MousePosition** et l'ajouter au tableau des contrôles de la carte

```
// 4ème élément : Coordonnées de la souris
new ol.control.MousePosition({
  coordinateFormat : ol.coordinate.createStringXY(4),
  projection: 'EPSG:4326',
  className: 'custom-mouse-position',
  target : document.getElementById('mouse-position')
})
```

Ajouter une couche raster ol.layer.Tile

1. Ajouter une couche « WMTS »

- <https://openlayers.org/en/latest/examples/wmts-ign.html>
- <https://wxs.ign.fr/pratique/geoportail/wmts?SERVICE=WMTS&VERSION=1.0.0&REQUEST=GetCapabilities>

```
/**
 * IGN : Service WMTS
 * Retourne des images précalculées (tuiles) suivant une grille prédéfinies
 */
var resolutions = [];
var matrixIds = [];
var proj3857 = ol.proj.get('EPSG:3857');
var maxResolution = ol.extent.getWidth(proj3857.getExtent()) / 256;

for (var i = 0; i < 18; i++) {
  matrixIds[i] = i.toString();
  resolutions[i] = maxResolution / Math.pow(2, i);
}

var tileGrid = new ol.tilegrid.WMTS({
  origin: [-20037508, 20037508],
  resolutions: resolutions,
  matrixIds: matrixIds
});
```

```
var ign = new ol.source.WMTS({
  url: 'https://wxs.ign.fr/pratique/geoportail/wmts',
  layer: 'GEOGRAPHICALGRIDSYSTEMS.MAPS',
  matrixSet: 'PM',
  format: 'image/jpeg',
  projection: 'EPSG:3857',
  tileGrid: tileGrid,
  style: 'normal'
});

map.addLayer(
  new ol.layer.Tile({source: ign})
);
```


Ajouter une couche raster ol.layer.Tile

2. Ajouter une couche « Tiled ArcGIS MapServer »

- <https://openlayers.org/en/latest/examples/arcgis-tiled.html?q=arcgis>
- https://server.arcgisonline.com/arcgis/rest/services/World_Imagery/MapServer

```
/**
 * ArcGIS REST Service
 * Service de carte tuilé ArcGIS Server
 */
var arcgis = new ol.source.TileArcGISRest({
  url: 'https://server.arcgisonline.com/arcgis/rest/services/World_Imagery/MapServer'
});

map.addLayer(
  new ol.layer.Tile({
    extent: [134987.79195151184, 6175929.419557666, 428505.98056670104, 6319172.410564141],
    source: arcgis
  })
);
```



Ajouter une couche raster ol.layer.Tile

3. Visibilité des couches

- Ajouter les « checkbox » dans la page HTML pour gérer la visibilité des couches sur la carte

```
<!-- visibilité des couches des couches -->
<div>
  <input type="checkbox" id="cbOSM" onclick="setVisibility(this)" checked><label>OpenStreetMap</label>
  <input type="checkbox" id="cbWMTS" onclick="setVisibility(this)" ><label>WMTS</label>
  <input type="checkbox" id="cbTileArcGISRest" onclick="setVisibility(this)" checked><label>Orthophoto</label>
</div>
```

☒ OpenStreetMap ☐ WMTS ☒ Orthophoto

- Définir la visibilité par défaut à false pour la couche IGN
 - Propriété **visible** de l'objet ol.source.WMTS
visible: **true**
- Définir l'opacité à 0.7 pour l'orthophoto
 - Propriété **opacity** de l'objet ol.source.TileArcGISRest
opacity: **0.7**



Ajouter une couche raster ol.layer.Tile

- Créer la fonction setVisibility(checkbox) dans le fichier JS

```
function setVisibility(checkbox) {  
  map.getLayers().forEach(function (layer) { // Parcourir les couches  
    switch (checkbox.id) { // selon l'id de la checkbox, on agit sur la bonne couche  
      case 'cbOSM': // OpenStreetMap  
        if (layer.getSource() == osm) { layer.setVisible(checkbox.checked); }  
        break;  
      case 'cbWMTS': // IGN  
        if (layer.getSource() == ign) { layer.setVisible(checkbox.checked); }  
        break;  
      case 'cbTileArcGISRest': // ArcGIS  
        if (layer.getSource() == arcgis) { layer.setVisible(checkbox.checked); }  
        break;  
    }  
  })  
}
```



- › Utiliser le debugger (**F12**)
- › Ajouter un point d'arrêt et parcourir la fonction pas à pas
- › Inspecter l'objet **layer** dans la boucle **forEach**

Ajouter une couche vecteur

ol.layer.Vector

1. Ajouter une couche vecteur à la carte

```
var vectorLayer = new ol.layer.Vector({  
  source: new ol.source.Vector({ // Source de ma couche vecteur  
    // tableau d'entités  
    features: [  
      // Création d'une entité  
      // Objet de type ol.Feature initialisé avec une géométrie et un nom  
      new ol.Feature({  
        geometry: new ol.geom.Point(ol.proj.fromLonLat([2.294481,48.858370])),  
        name: "Paris"  
      })  
    ]  
  })  
});  
map.addLayer(vectorLayer); // Ajout de la couche vecteur à la carte --  
  method 'addLayer()'
```

Ajouter une couche vecteur

ol.layer.Vector

2. Définir l'apparence des entités de la couche

```
/**
 * Création d'un style 'Icon'
 * @type {module:ol/style/Icon~Options}
 */
var iconStyle = new ol.style.Style({
  image: new ol.style.Icon({
    src: 'data/icon.png' // mon icône
  })
});

// Parcourir les entités de la couche
// <Array>.forEach (ES6) : parcours du tableau
vectorLayer.getSource().getFeatures().forEach(function(feature) {
  feature.setStyle(iconStyle);
})
```



Ajouter une couche vecteur

ol.layer.Vector

3. Ajouter des entités à la couche

```
monTableau.forEach(function (element) { // <Array>.forEach (ES6) : parcours du tableau
  // Exclure les villes 'undefined'
  if (element.ville != undefined) {
    // Créer une nouvelle entité instance de la classe Feature
    var feature = new ol.Feature({
      // Propriété geometry : ol.geom.Point
      geometry: new ol.geom.Point(ol.proj.fromLonLat(
[element.getCoordinates().x, element.getCoordinates().y]
      )),
      // Propriété name : String
      name: element.ville
    })
    // Appliquer le style
    feature.setStyle(iconStyle);
    // Ajouter l'entité à la source de la couche
    vectorLayer.getSource().addFeature(feature);
  }
})
```

Afficher une popup ol.Overlay

1. Ajouter un bloc HTML

```
<!-- PopUp -->
<div id="popup" class="ol-popup">
  <a href="#" id="popup-closer" class="ol-popup-closer" onclick="closePopup()"></a>
  <div id="popup-content"></div>
</div>
```

2. Créer l'objet JavaScript et l'ajouter à la carte

```
<!-- Création d'un objet ol.Overlay pour ancrer la popup sur la carte. -->
var popup = new ol.Overlay({
  element: document.getElementById('popup') // <div>
});
map.addOverlay(popup);
```

3. fermer la popup au clic sur X

```
function closePopup() {
  popup.setPosition(undefined); // masquer la popup
}
```

Afficher une popup ol.Overlay

4. Ajouter/Modifier l'événement click sur la carte

```
map.on('click', function (event) {  
  // Récupérer l'entité ol.Feature  
  var feature = map.forEachFeatureAtPixel(event.pixel,  
    function (feature) {  
      return feature;  
    });  
  if (feature) { // Si une entité retournée  
    var coordinates = feature.getGeometry().getCoordinates();  
    popup.setPosition(coordinates); // Position de la popup  
    //Contenu de la popup  
    document.getElementById('popup-content').innerHTML =  
      "<p style='margin:0'>" + feature.get('name').replace(/\n/g, '<br>') + "</p>";  
  } else {  
    if (document.getElementById('popup') != null) { closePopup(); }  
  }  
});
```


Afficher une popup ol.Overlay

1. Parcourir les exemples OpenLayers 3

- <https://openlayers.org/en/v3.20.1/examples/>

2. Ajouter les contrôles : **ZoomSlider** et **ZoomToExtent**

3. Ajouter de nouvelles interactions :

- **Rotation**
- **Dessin et/ou mesure**

4. Afficher les coordonnées x,y dans la popup dans la popup



Retrouver l'ensemble du cours, ressources et exercices à l'adresse suivante :

<https://github.com/geovendee/formation-prog-20212022>

MERCI

