jDev
Vos solutions numériques
open-sources et SIG

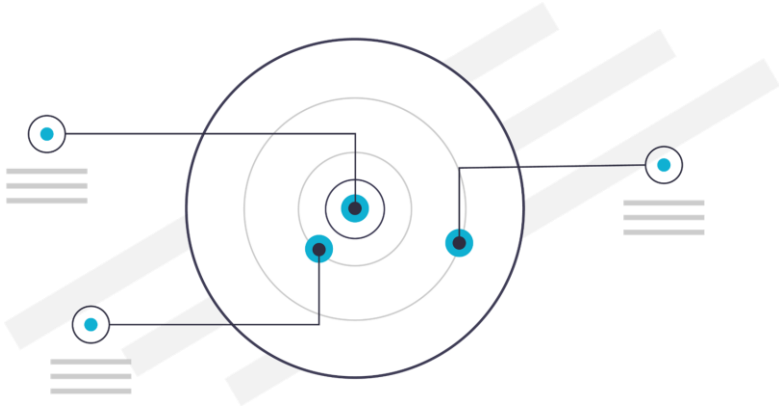
Formation

TP2
Pratiquons JavaScript avec Mviewer

05/10/2021

TP2

Présentation



Objectifs :

> Comprendre du code existant

> Modifier un code existant

> Réutiliser et adapter du JavaScript existant

> Découvrir les outils du navigateur

> Manipuler la carte dans le navigateur

TP2

Contexte

Vous disposez des application du répertoire /demo et /apps.

Nous utiliserons l'éditeur de texte Visual Studio Code pour ouvrir les fichiers.

Nous utiliserons le navigateur comme environnement JavaScript.

TP2

Comprendre le JavaScript dans Mviewer

1. Identifier du JavaScript

Etape 1

- Observer les fichiers du répertoire C:\xampp\htdocs\mviewer\demo\heatmap
- JavaScript = extension .js

Etape 2

- Ouvrez le répertoire C:\xampp\htdocs\mviewer\js
- Ouvrez le fichier C:\xampp\htdocs\mviewer\index.html

Que remarquez-vous des lignes 460 à 487 ?

TP2

Comprendre le JavaScript dans Mviewer

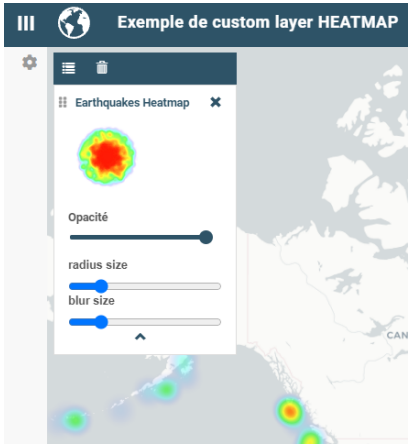
1. Identifier du JavaScript

Etape 3

- Aller sur l'URL http://localhost/mviewer/?config=demo/heatmap/config.xml
- Retrouvez le répertoire de cette carte
- Ouvrez le fichier config.xml de cette carte
- Observez la <layer>, le type et l'URL

Etape 4

- Pour cette même carte, observez la ligne 28 du config.xml
- Ouvrez le répertoire indiqué
- Essayez de faire le lien entre le volet de la couche « Earthquakes Heatmap » et les deux fichiers présents



jDev

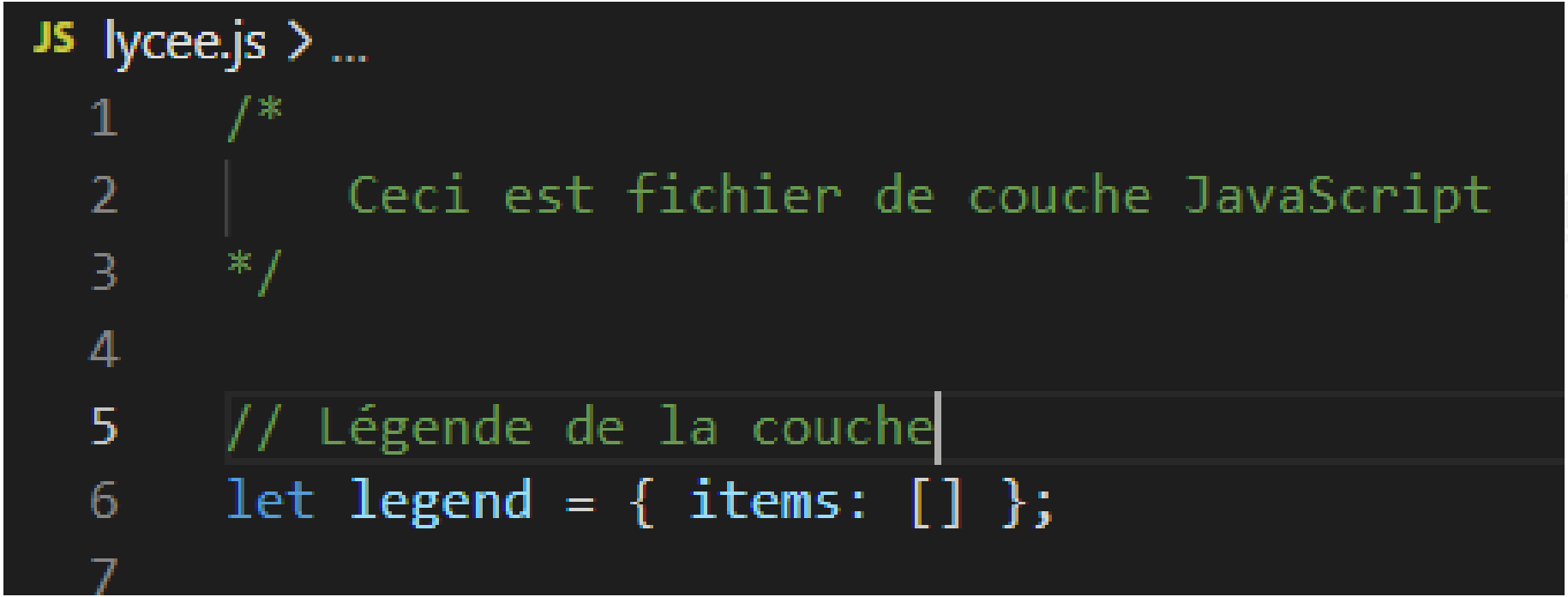
TP2

Comprendre le JavaScript dans Mviewer

1. Identifier du JavaScript

Le JavaScript est donc utilisé dans Mviewer pour :

1. Faire fonctionner Mviewer
2. Afficher des couches sur la carte
3. Créer des interactions entre l'utilisateur, les interfaces et les couches (IHM)



Etape 3

- Ouvrez le fichier lycee.js
- Nous allons commenter les lignes en utilisant // ou bien /*text*/ devant la ligne comme ceci :

TP2

Comprendre le JavaScript dans Mviewer

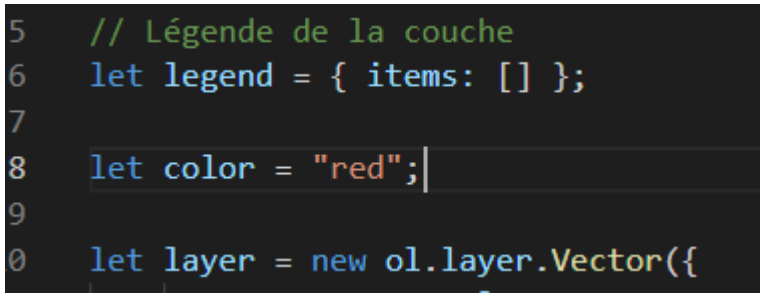
2. Comprendre et manipuler du code existant

Etape 4

La couleur des points n'est pas très visible, nous allons isoler la couleur dans une variable.

Une variable commence par « let » ou « var » et sert à stocker des informations en mémoire pour les réutiliser.
Il faut utiliser des doubles guillemets pour ajouter du texte dans une variable.

1. Ajouter cette ligne à la ligne 8 précédée et suivie d'un saut de ligne (touche « Entrée »)



Une variable commence par « let » / « var », prend un nom et est suivie d'un « = ».
On peut y mémoriser du texte, un nombre, une liste etc...

Doc en ligne

jDev

TP2

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 4

2. Remplacer la ligne 18 de l'image 1 par l'image 2

```
15        style: new ol.style.Style({
16            image: new ol.style.Circle({
17                fill: new ol.style.Fill({
18                    color: 'red'
19                }),
20                radius: 5
21            })
22        })
```

```
15        style: new ol.style.Style({
16            image: new ol.style.Circle({
17                fill: new ol.style.Fill({
18                    color: color
19                }),
20                radius: 5
21            })
22        })
```

TP2

Comprendre le JavaS

2. Comprendre et manip

Etape 5

Faisons de même pour la taille des po

1. Ajouter une variable « let radiusSi
2. Remplacer la valeur « 5 » par le nc

```
let color = "red";
let radiusSize = 5;

let layer = new ol.layer.Vector({
        source: new ol.source.Vector({
            url: "https://ows.region-bretagne
            format: new ol.format.GeoJSON()
        }),
        style: new ol.style.Style({
            image: new ol.style.Circle({
                fill: new ol.style.Fill({
                    color: color
                }),
                radius: radiusSize
            })
        })
});
```

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 6

Nous aimerions maintenant changer la couleur selon la valeur d'un attribut et ne plus avoir une seule couleur.

1. Ajoutons une « fonction » qui permet d'exécuter des instructions plusieurs fois, dès que nécessaire, sans réécrire plusieurs fois la même chose
2. Ajouter donc une fonction nommée « pointStyle » comme ceci :

```
11 const pointStyle = function () {
12
13 };
14
```

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Nous souhaitons que la fonction donne un résultat.

- Nous allons donc ajouter « return » à la fin et indiquer la couleur à retourner par défaut.

```
11  function pointColor () {
12      return "red"
13  };
14
```

- Comme une variable, nous allons appeler la fonction pour obtenir la couleur

```
style: new ol.style.Style({
    image: new ol.style.Circle({
        fill: new ol.style.Fill({
            color: pointColor()
        }),
        radius: radiusSize
    })
})
```

Une fonction commence par « function » et prend un nom. Elle est suivie de parenthèse et d'accolade ouvrante ET fermante.

jDev

Doc en ligne

TP2

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 7

Notre fonction retourne toujours la même couleur, mais nous souhaitons changer selon la valeur d'attributs « secteur_li » qui permet de définir le secteur du lycée (privé / public).

- Remplacer le code A par le code B *(vous pouvez le copier)*

A

```
style: new ol.style.Style({
    image: new ol.style.Circle({
        fill: new ol.style.Fill({
            color: pointColor()
        }),
        radius: radiusSize
    })
})
```

B

```
style: function(feature) {
    return new ol.style.Style({
        image: new ol.style.Circle({
            fill: new ol.style.Fill({
                color: pointColor()
            }),
            radius: 9
        })
    })
}
```

Qu'avons-nous fait ?

jDev

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 8

Une fonction peux prendre un « paramètre » qui sera utilisable dans les instructions de la fonction. Les instructions sont placées entre les { }.

- Pour accéder aux attributs de notre feature, nous donnons à la fonction pointColor un paramètre « feature ».

```
style: function(feature) {
    return new ol.style.Style({
        image: new ol.style.Circle({
            fill: new ol.style.Fill({
                color: pointColor(feature)
            }),
            radius: radiusSize
        })
    })
}
```

2. Comprendre et manipuler du code existant

Etape 9

Le paramètre passé dans la fonction, il faut maintenant qu'on l'utilise dans les instructions !

- Ajouter à la fonction pointColor un paramètre « entiteGeo »

```
function pointColor (entiteGeo) {
    return "red"
};
```

L'entité étant maintenant accessible dans les instructions, nous allons parcourir les propriétés de l'entité pour connaître sa valeur d'attribut. Pour cela, on utilisera le code suivant :

```
entiteGeo.get("secteur_li");
```

- Mémorisez la valeur dans une variable au sein de la fonction (ATTENTION au ; de fin de ligne !!)

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 10

La valeur étant connue, nous devons la comparer pour savoir quoi faire selon la valeur.
On appel cela une condition !

- Avant le return, ajouter une condition comme ceci pour avoir du bleu si le secteur est public (il faut === comme opérateur d'égalité stricte)

```
if (valeurSecteur === "Public") {
    return "blue";
} else {
    return "red";
}
```

- Rajouter des lignes de commentaire pour décrire ce qu'il se passe avec //
- Rafraîchir la carte et voir que le style à changé !

TP2

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 11

Nous pouvons utiliser un opérateur conditionnel pour réaliser une condition sans utiliser de if / else.

Cette syntaxe est plus courte et nous permet également de ne pas utiliser de fonction !

- Commentez la ligne 30

```
image: new ol.style.Circle({
    fill: new ol.style.Fill({
        //color: pointColor(feature)
    }),
```

- Ajouter en dessous un opérateur conditionnel comme ceci :

```
fill: new ol.style.Fill({
    // color: pointColor(feature)
    color: entiteGeo.get("secteur_li") === "Public" ? "blue" : "red"
}),
```

Comment lire cette ligne ?

Comprendre le JavaScript dans Mviewer

2. Comprendre et manipuler du code existant

Etape 12

Explications du reste du code.

Solution jusqu'à cette étape ➔ fichier lycee.step2.js

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Etape 1

Reprenez le code précédent.

Nous allons nous inspirer d'une couche existante pour obtenir un nouveau style.

- Ouvrez en plus le fichier customlayers/cluster.js
- Observez la ligne 74 ➔ A votre avis, que fait ce code ?
- Observer la ligne 84 et la fonction existante en ligne 60

jDev

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Etape 2

Nous allons représenter les lycées avec un cluster.

Nous réutiliserons donc cet exemple pour notre exercice.

- Copiez la fonction dans le fichier lycee.js
- Créez une variable layerUrl avec l'URL de la couche du fichier lycee.js

```
const layerUrl = "https://ows.region-bretagne.fr/geoserver/rb/wfs?SERVICE=WFS&VERSION=1.0.0&REQUEST=GETFEATURE&TYPENAME=lycee&outputFor
```

- Supprimer la couche (commence par « let layer =« et fini par « ; ») dans le fichier lycee.js
- Copiez la ligne 73 à 86 vers le fichier lycee.js et adaptez la comme sur la page suivante

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

```
let layer = new ol.layer.Vector({
    source: new ol.source.Cluster({
        geometryFunction: function(feature) {
            return new ol.geom.Point(ol.extent.getCenter(feature.getGeometry().getExtent()));
        },
        distance: 50,
        source: new ol.source.Vector({
            url: layerUrl,
            format: new ol.format.GeoJSON()
        })
    }),
    style: clusterStyle

});
```

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Etape 3

- Essayez le code en rafraîchissant la carte

Que constatez-vous ?

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Etape 3

- Appuyez sur la touche F12
- Observez l'onglet « console » et observez…

Que pensez-vous de ces messages ?

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Etape 4

- Copiez la fonction « mainStyle » du fichier cluster.js en haut du fichier lycee.js
- Testez la carte et afficher le message d'erreur si besoin…

Que pensez-vous de ces messages ?

- Remplacez ensuite le code A par le code B et testez la carte

```
if (size == 1) {
    return uniqueStyle;
} else {
    return manyStyle(radius, radius2, size);
}
```

A

```
return manyStyle(radius, radius2, size);
/*if (size == 1) {
    return uniqStyle
} else {
    return manyStyle(radius, radius2, size);
}*/
```

B

jDev

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Bonus

- Nous souhaitons maintenant voir un style propre pour les clusters ne contenant qu'un seul lycée

- Nous allons réutiliser et adapter notre fonction pointColor.

- Remplacez le code A par le code B

```
return manyStyle(radius, radius2, size);
/*if (size == 1) {
    return uniqStyle
} else {
    return manyStyle(radius, radius2, size);
}*/
```

A

```
if (size == 1) {
    return pointColor(feature.get("features")[0]);
} else {
    return manyStyle(radius, radius2, size);
}
```

B

jDev

TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Bonus

- Remplacez la fonction pointColor que nous avons créé précédemment par ce code :

```
function pointColor (entiteGeo) {
    // on recherche la valeur de l'attribut "secteur_li"
    let valeurSecteur = entiteGeo.get("secteur_li");
    let color = "red";
    if (valeurSecteur === "Public") {
        color = "blue";
    }
    return [new ol.style.Style({
        image: new ol.style.Circle({
            radius: 10,
            fill: new ol.style.Fill({
                color: color
            })
        })
    })]
};
```

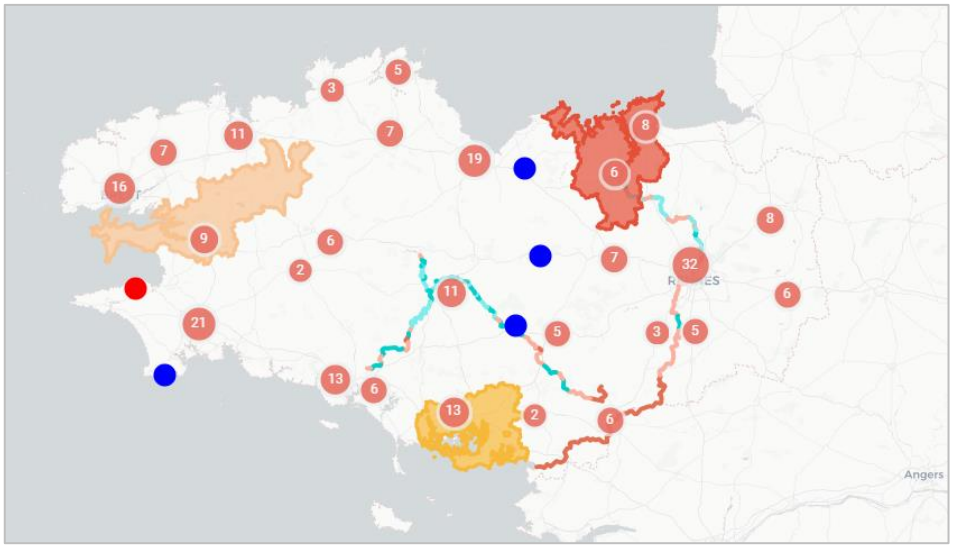
TP2

Comprendre le JavaScript dans Mviewer

3. Réutiliser du code existant

Bonus

- Commentez le code en ajoutant votre lignes d'explications (//)



jDev

CONTACT

Pour toute question, n'hésitez pas à nous contacter

pierre.jego@jdev.fr
gaetan.bruel@jdev.fr
agathe.adam@jdev.fr

CRÉDITS