CARTOGRAPHIE ET PLUS AVEC D3.JS

ATELIER THÉMATIQUE EN VISUALISATION DE DONNÉES

Antoine Béland

12 octobre 2018

PLAN DE L'APRÈS-MIDI

1. Format de données GeoJSON
2. Projections avec D3.js
3. Rendu d'une carte avec D3.js
4. Exemples avancés avec D3.js
5. Mise en pratique

FORMAT DE DONNÉES GEOJSON

FORMAT DE DONNÉES GEOJSON

- Format qui encode des données géospatiales au format JSON
- Permet d'encoder plusieurs données, dont des points, des lignes et des polygones
- Est utilisé par plusieurs outils, dont Leaflet et D3.js

FORMAT DE DONNÉES GEOJSON

- Les données doivent **obligatoirement** posséder l'attribut **`type`**
- Habituellement, le type de données utilisées est un **`Feature`** ou un **`FeatureCollection`**
- Chaque **`Feature`** est composé des attributs **`geometry`** et **`properties`**

GEOJSON — EXEMPLE

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [ 102.0, 0.5 ]
      },
      "properties": {
        "name": "île quelque part"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "LineString",
        "coordinates": [ [ 102.0, 0.0 ], [ 104.0, 0.0 ], [ 105.0, 1.0 ] ]
      },
      "properties": {
        "name": "Frontière quelque part"
      }
    }
  ]
}
```

Exemple tiré de GeoJSON · Wikipedia, 2018

GEOJSON — OBJETS GÉOMÉTRIQUES

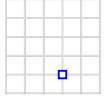
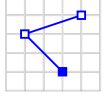
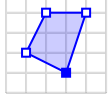
Type	Exemples
 Point	`{ "type": "Point", "coordinates": [30, 10] }`
 Segments	`{ "type": "LineString", "coordinates": [ [30, 10], [10, 30], [40, 40] ] }`
 Polygones	`{ "type": "Polygon", "coordinates": [ [ [35, 10], [45, 45], [15, 40], [10, 20], [35, 10] ], [ [20, 30], [35, 35], [30, 20], [20, 30] ] ] }`

Tableau tiré de GeoJSON · Wikipedia, 2018

GEOJSON — ENSEMBLES GÉOMÉTRIQUES

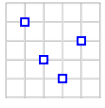
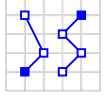
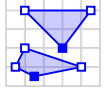
Type	Exemples
 Ensemble de points	`{ "type": "MultiPoint", "coordinates": [ [10, 40], [40, 30], [20, 20], [30, 10] ] }`
 Lignes brisées	`{ "type": "MultiLineString", "coordinates": [ [[10, 10], [20, 20], [10, 40]], [[40, 40], [30, 30], [40, 20], [30, 10]] ] }`
 Ensemble de polygones	`{ "type": "MultiPolygon", "coordinates": [ [ [ [30, 20], [45, 40], [10, 40], [30, 20] ] ], [ [ [15, 5], [40, 10], [10, 20], [5, 10], [15, 5] ] ] ] }`

Tableau tiré de GeoJSON · Wikipedia, 2018

GEOJSON — EXEMPLE

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
        [ [35, 10], [45, 45], [15, 40], [10, 20], [35, 10] ],
        [ [20, 30], [35, 35], [30, 20], [20, 30] ]
        ]
      },
      "properties": {
        "name": "Un polygone quelque part"
      }
    }
  ]
}
```

▶ Démo

GEOJSON — TOPOJSON

- TopoJSON est une extension du format GeoJSON
- Il arrive souvent de rencontrer ce format de fichier puisqu'il permet une **meilleure compression**
- Une fois les données chargées, celles-ci sont converties au format GeoJSON

D3.JS — PROJECTIONS

PROJECTIONS

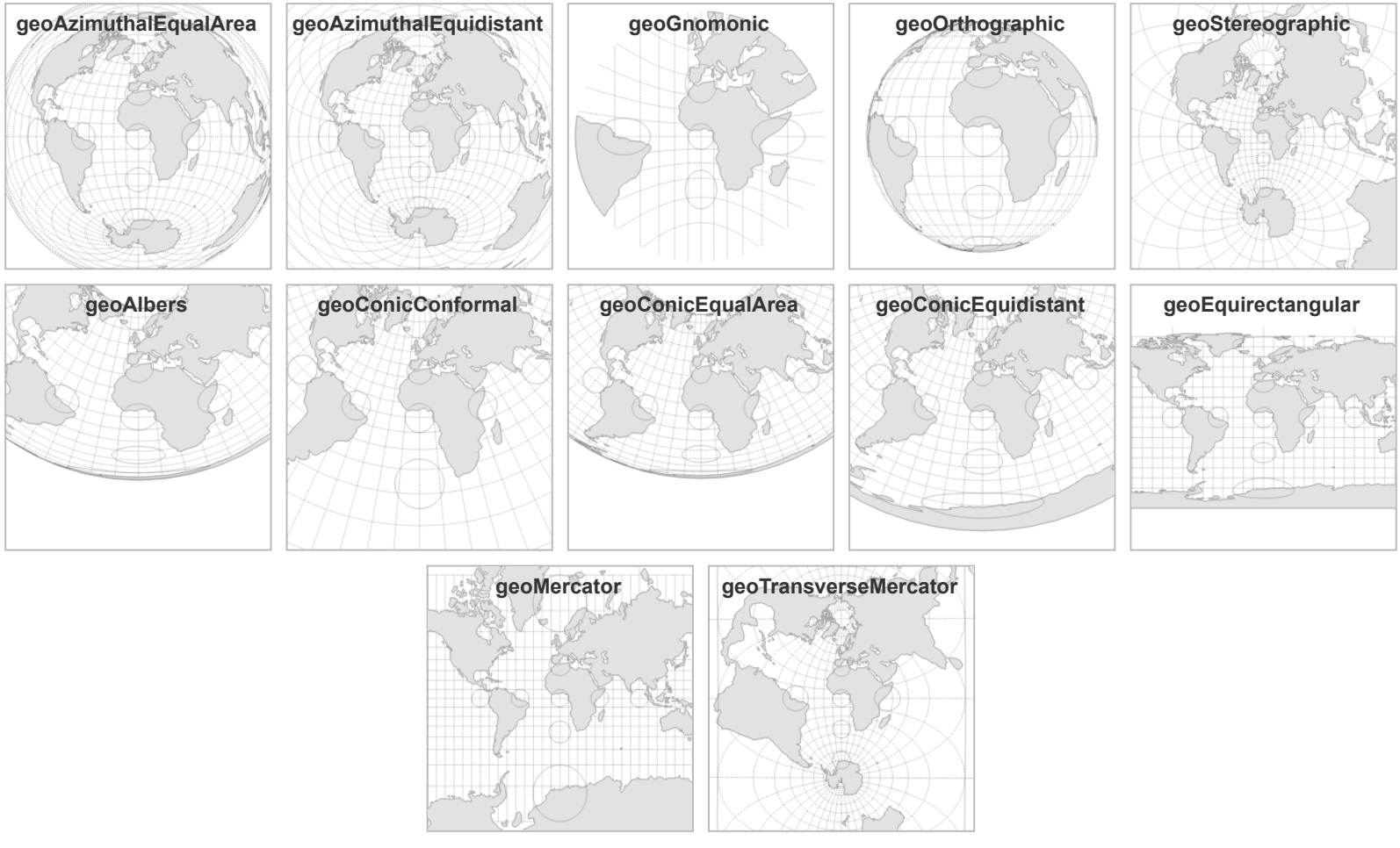
- La fonction de projection doit transformer des coordonnées géographiques en coordonnées *x* et *y*
- D3.js fournit **plusieurs fonctions** pour calculer des projections (azimutale, composite, conique et cylindrique)
- Les projections choisies seront utilisées par la suite pour dessiner la carte

PROJECTIONS — TYPES PRINCIPAUX

- `d3.geoAzimuthalEqualArea`
- `d3.geoAzimuthalEquidistant`
- `d3.geoGnomonic`
- `d3.geoOrthographic`
- `d3.geoStereographic`
- `d3.geoAlbers`
- `d3.geoConicConformal`
- `d3.geoConicEqualArea`
- `d3.geoConicEquidistant`
- `d3.geoEquirectangular`
- `d3.geoMercator`
- `d3.geoTransverseMercator`

Liste tirée de *Geographic — Projections* · *D3 in Depth*, 2016

PROJECTIONS — TYPES PRINCIPAUX

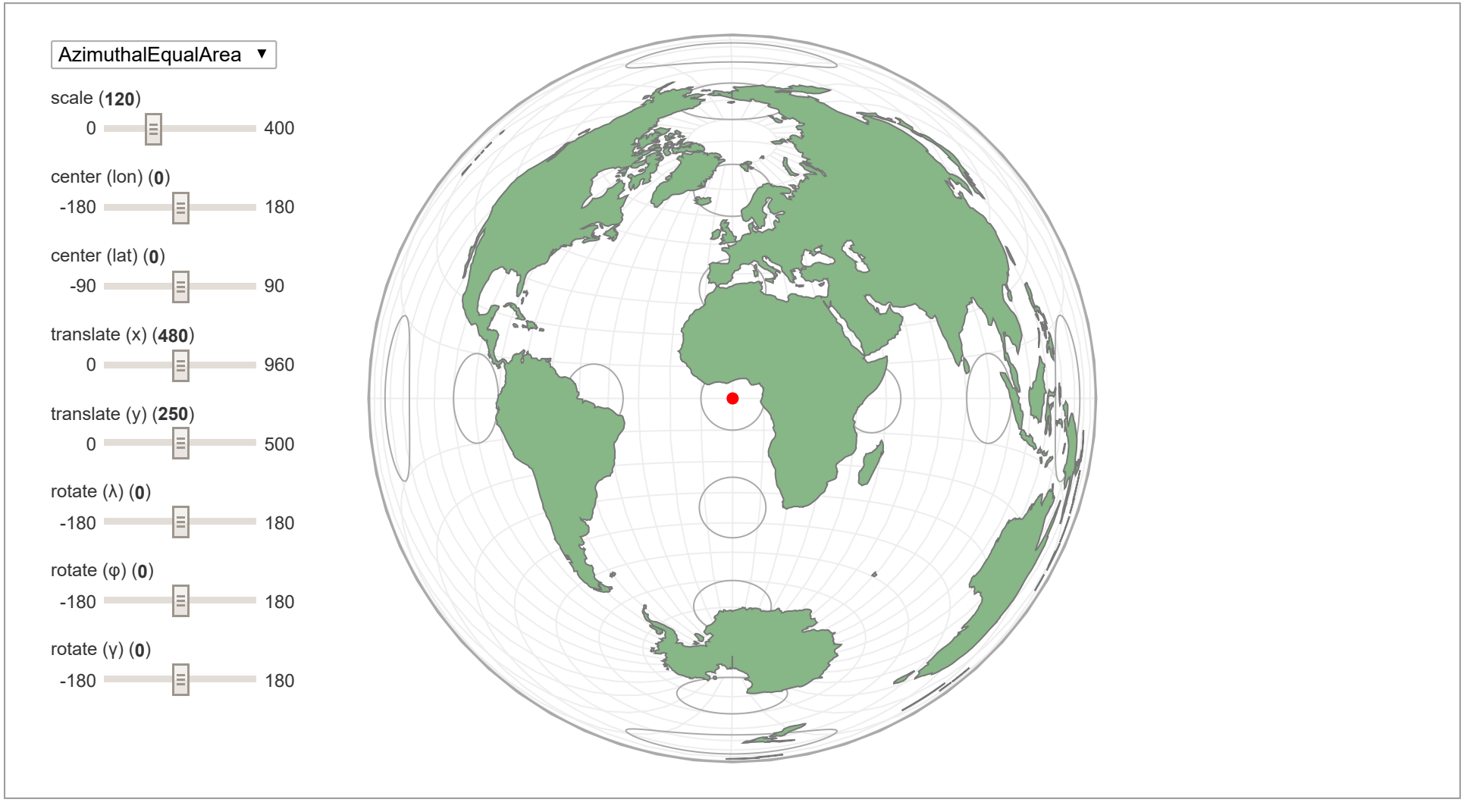


Images tirées de *Geographic — Projections* · *D3 in Depth*, 2016

PROJECTIONS — CONFIGURATION

- `scale` (facteur d'échelle de la projection)
- `center` (centre de la projection [long., lat.])
- `rotate` (rotation de la projection [λ, φ, γ])
- `translate` (position [en pixels] du centre de la projection)

PROJECTIONS — EXEMPLE



Outil *Projection explorer* · *D3 in Depth*, 2017

PROJECTIONS — POUR EN SAVOIR PLUS

- Tutoriel sur l'utilisation des projections avec D3.js
- Documentation complète sur les projections avec D3.js

D3.JS — RENDU D'UNE CARTE

RENDU D'UNE CARTE

- Une fois la bonne projection configurée, il est assez simple de faire le rendu d'une carte avec D3.js
- La fonction `d3.geoPath` permet de réaliser cette tâche assez facilement

RENDU D'UNE CARTE — EXEMPLE

```
const svg = d3.select('svg');
d3.json('canada.geojson').then(data => {
  const projection = d3.geoMercator() // Création de la projection
    .rotate([100, -45])
    .center([5, 20])
    .scale(900)
    .translate([450, 400]);

  const path = d3.geoPath(projection); // Initialisation de "geoPath"
  svg.selectAll('path')
    .data(data.features)
    .enter()
    .append('path')
    .attr('d', path)
    .style('fill', 'white')
    .style('stroke-width', 1)
    .style('stroke', 'black')
});
```

▶ Démo

D3.JS — EXEMPLES AVANCÉS

EXEMPLES AVANCÉS

- D3.js est une bibliothèque **relativement complexe** qui demande un certain temps d'apprentissage
- Dans le cas où l'on souhaite uniquement réaliser des visualisations simples, **d'autres outils** existent
- D3.js offre sa pleine puissance pour la création de visualisations **personnalisées**

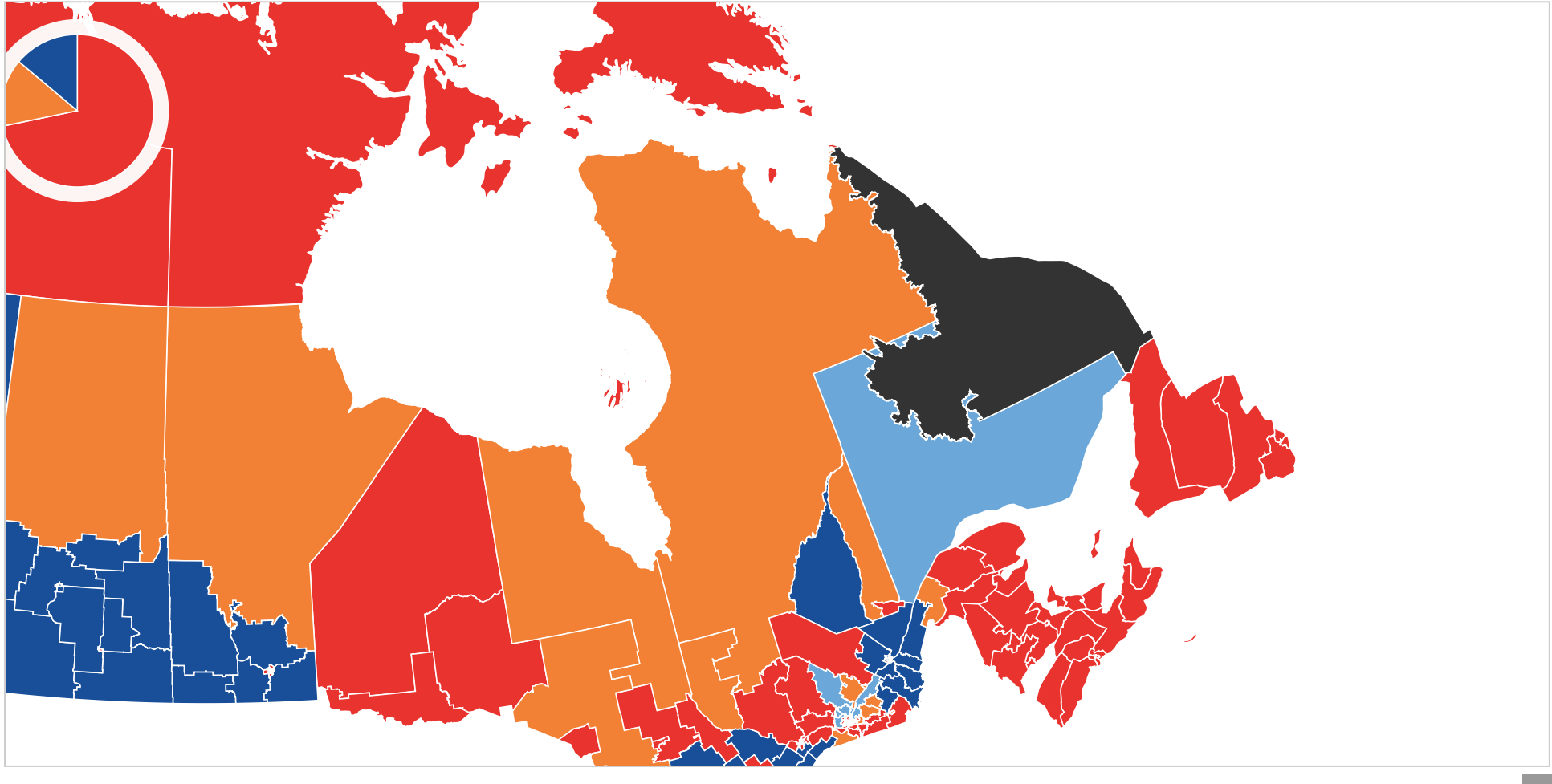
EXEMPLES AVANCÉS

- *Extensive Data Shows Punishing Reach of Racism for Black Boys*
- *Hurricane Irma Is One of the Strongest Storms In History*
- Les pluies de l'ouragan « Irma » en une carte
- Les promesses des partis sont-elles financièrement réalisables?
- Quels sont les risques d’un voyage sur Mars?

MISE EN PRATIQUE

MISE EN PRATIQUE

- Réaliser une carte choroplèthe et un *pie chart* à partir des résultats des élections fédérales de 2015.



MISE EN PRATIQUE — DONNÉES

- Il est nécessaire d'utiliser **deux jeux de données** pour réaliser l'exercice
- Le premier, au format **TopoJSON**, permet de récupérer les polygones à dessiner sur la carte
- Le deuxième, au format **CSV**, permet d'obtenir les résultats pour les différentes circonscriptions

Données CSV

MISE EN PRATIQUE — DONNÉES

- Pour vous simplifier la tâche, les données provenant du fichier CSV ont été **restructurées**

```
[
  {
    id: 12345,
    name: "Nom de la circonscription",
    winnerParty: "Le parti gagnant",
    candidates: [
      {
        name: "Le nom du candidat",
        votes: 12345,
        party: "Le parti du candidat"
      },
      // Suite du tableau listant les candidats...
    ]
  },
  // Suite du tableau listant les circonscriptions...
]
```

MISE EN PRATIQUE — DIRECTIVES

- Définir la **projection** de la carte à utiliser
- Dessiner les polygones associés aux circonscriptions sur la **carte**
- Permettre de **sélectionner** une circonscription quelconque afin d'afficher les résultats
- Afficher les **résultats** d'une circonscription avec un *pie chart*

MISE EN PRATIQUE

- Pour débuter, téléchargez le dossier ZIP contenant le code de départ pour l'exercice
- Complétez, par la suite, le fichier **script.js**
- Utilisez **Firefox** ou un serveur web local pour tester votre script*

Télécharger ▶ Correction

* Google Chrome bloque les requêtes asynchrones réalisées dans un fichier local si celui-ci n'est pas hébergé sur un serveur local