



D3.JS INITIATION – Niveau 1

Alice MARCHAL Roch JACQUOT
Duy NGUYEN Houda BOUJEDDAYN

26/10/2018

Sommaire

- Présentation de l'outil
- Pourquoi d3.js
- Typologie des diagrammes
- Installation
- Connaissances préalables
 - Chargement des données
 - Création d'un SVG
 - Sélection et manipulation
 - Lier les données à des balises
 - Spécifications des échelles D3

Présentation de l'outil

D3.js (Data-Driven Documents) est une bibliothèque graphique de JavaScript servant à :

- o Afficher des données numériques sous une forme graphique et dynamique
- o Dessiner des graphiques
- o Visualiser les données
- o Développer les tableaux de bord en temps réel

Il est important de noter que D3.js :

- o Ne fournit pas de graphiques prédéfinis
- o Ne peut pas être utilisé pour dessiner les graphiques 3D

Pourquoi d3.js

D3.js permet de :

- o Créer sa propre dataviz from scratch à partir de mockups
- o Adapter un des nombreux exemples D3.js disponibles sur <https://d3js.org/>
- o Contribuer à une bibliothèque de visualisation basée sur D3.js (c3js, nvd3, ..)
- o Intégrer une visualization ou une bibliothèque basée sur D3.js dans un framework (react, angular) sous forme de component

Il est caractérisé par :

- o Facilité de manipulation du DOM d'une page web à la JQuery, aussi bien en SVG qu'en Canvas
- o Facilité de créer des axes, labels et autres finitions indispensables à une dataviz professionnelle
- o Interpolation des différents états du DOM pour faire des transitions animées
- o Création des layouts complexes de graphes ou de Treemap

Typologie des diagrammes



Installation

**Gestionnaire de paquets officiel
de Node.js :**

- Pour NPM → `npm install d3`
- Pour Yarn → `yarn add d3`



**Chargement direct à partir de
d3js.org :**

```
<script src = "https://d3js.org/d3.v5.js"> </ script>
```

Connaissances préalables

- **HTML/ CSS** : langages standards pour la mise en page web;
- **SVG** (*Scalable Vector Graphics*) : format de données pour les graphiques vectoriels;
- **Javascript** : langage de programmation orienté web qui contient des librairies telles D3.js.



```
HTML
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>Bar Chart</title>
6     <!-- Utiliser D3.js ici -->
7     <script src="https://d3js.org/d3.v5.min.js"></script>

CSS
1 <style>
2   html, body {
3     margin: 0;
4     height: 100%;
5   }

JS
1 console.log();
2
3 const = "test";
```

Structure du fichier HTML de base

D3 est une bibliothèque de fonctions JavaScript, utilisée du côté client et appelée depuis un fichier HTML.

Le fichier HTML de base aura la forme suivante :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Fichier de base D3</title>
6 <script src="http://d3js.org/d3.v3.js" type="text/javascript"></script>
7 <script type="text/javascript">
8   function draw(data) {
9     // le code de dessin viendra se placer ici
10  }
11 </script>
12 </head>
13 <body>
14 <script type="text/javascript">
15   d3.json("data/data.json", draw);
16 </script>
17 </body>
18 </html>
```


Structure du fichier HTML de base

Description du code source

Ligne 6 : Appel de la bibliothèque.

Lignes 7 à 11 : Le script de dessin proprement-dit.

Lignes 14 à 16 : Le script de chargement des données, qui lancera le dessin une fois terminé. Notez son positionnement dans le `<body>`.

Avec cette organisation, la page HTML se charge, lance la récupération des données (qui peut être longue s'il s'agit d'un flux) puis exécute le code de réalisation du graphisme. Ainsi, le programme se lancera une fois que tous les éléments nécessaires (éléments de la page et données) seront disponibles.

Chargement des données

Chargement des données directement dans le code JavaScript

Les données peuvent être chargées directement dans le code JavaScript en utilisant un tableau [] qui contient des objets {} eux-mêmes contenant deux couples variable: valeur.

```
[  
  {"mois": "janvier", "valeur": 5},  
  {"mois": "février", "valeur": 10},  
  {"mois": "mars", "valeur": 13},  
  {"mois": "avril", "valeur": 19},  
  {"mois": "mai", "valeur": 21},  
  {"mois": "juin", "valeur": 25},  
  {"mois": "juillet", "valeur": 22},  
  {"mois": "août", "valeur": 18},  
  {"mois": "septembre", "valeur": 15},  
  {"mois": "octobre", "valeur": 13},  
  {"mois": "novembre", "valeur": 11},  
  {"mois": "décembre", "valeur": 12}  
]
```

Chargement des données

Chargement des données à partir d'un fichier externe

Les données peuvent être également chargées à partir de nombreux types de fichiers externes à l'aide de commandes telles que `d3.csv`, `d3.json`, `d3.tsv`.

Les fonctions D3 prennent également en charge les fonctions de rappel pour traiter les données ou les cas d'erreur résultants.

```
1 var dataset;  
  
2 d3.csv("nom_fichier.csv", function(error, data){  
3   if (error){  
4     console.log(error);  
5   }else{  
6     dataset = data;  
7     console.log(data);  
8   }  
9   });
```

Création d'un SVG

Définition de la longueur et la largeur du graphe

```
9  fonctiondraw(dataset){  
10  var larg =500;  
11      var haut =100;  
12      var barPadding =1;//Padding des barres
```

Création de l'élément SVG

```
14      //Creation de l'élément SVG  
15      var svg = d3.select("body")  
16      .append("svg")  
17      .attr("width", larg)  
18      .attr("height", haut);
```

La fonction de dessin (lignes 9 à 12), va d'abord établir les valeurs de variables globales : les dimensions du graphes et la largeur de l'intervalle entre deux barres (padding).

*Le bloc entre les lignes 15 et 18 va ajouter un élément de type SVG au **<body>** de la page en précisant ses dimensions.*

Sélection et manipulation

Sélection d'éléments

`d3.select()` et `d3.selectAll()` peuvent être utilisés pour accéder aux éléments HTML par nom, classe, id ou de nombreux autres sélecteurs CSS.

`d3.select()` sélectionne uniquement le premier élément correspondant aux sélecteurs CSS ou HTML, tandis que `d3.selectAll()` sélectionne tous les éléments correspondants.

Modifier les éléments sélectionnés

La modification des propriétés des sélections peut se faire avec les opérateurs `attr()`, `text()`, `style()`.

Lier les données à des balises

```
1 var dataset = [ 5, 10, 15, 20, 25 ];  
  
2 d3.select("body") // Trouver la balise body dans DOM  
3   .selectAll("p") // Sélectionner tous les paragraphes du DOM  
4   .data(dataset) // Compter et parser les données  
5   .enter() // Créer un nouvel élément lié aux données  
6   .append("p") // Ajouter un élément en tant que paragraphe  
7   .text("New paragraph!"); // Ajouter une valeur text
```

Spécification des échelles D₃

Les échelles sont des fonctions qui font correspondre un domaine en entrée à une plage de sortie.

Les générateurs d'échelles de D₃ sont accessibles avec **d₃.scale** suivi du type d'échelle :

```
var scale = d3.scale.linear();
```

Les échelles peuvent aussi être ordinales et temporelles.

Notez que la plage () ne doit pas nécessairement être un ensemble de nombres; il peut aussi s'agir de couleurs ou de cordes.

Sources

Ressources Tutoriels :

- Pie Chart using D3.js: [Lien](#)
- Getting Started with Data Visualization Using JavaScript and the D3 Library: [Lien](#)

Références Documentation :

- Introduction à D3.js et aux documents pilotés par les données: [Lien](#)
- Interactive Visualization for the Web: [Lien](#)
- D3.js Tutorial Slides by Jane Hoffswell & Kanit [Lien](#)
- GitHub [Lien](#)