



# DATA VIZ

CM #1 du 14/12/2022

Antonin ROSA-MARTIN - IDU7

# Contenu du cours

**01**

---

"Quiz"

**02**

---

Quelques cas personnels

**03**

---

Présentation de D3.js

**04**

---

Présentation de DC.js

# QUIZ

Qu'est ce qui est  
important dans un  
**dashboard** ?

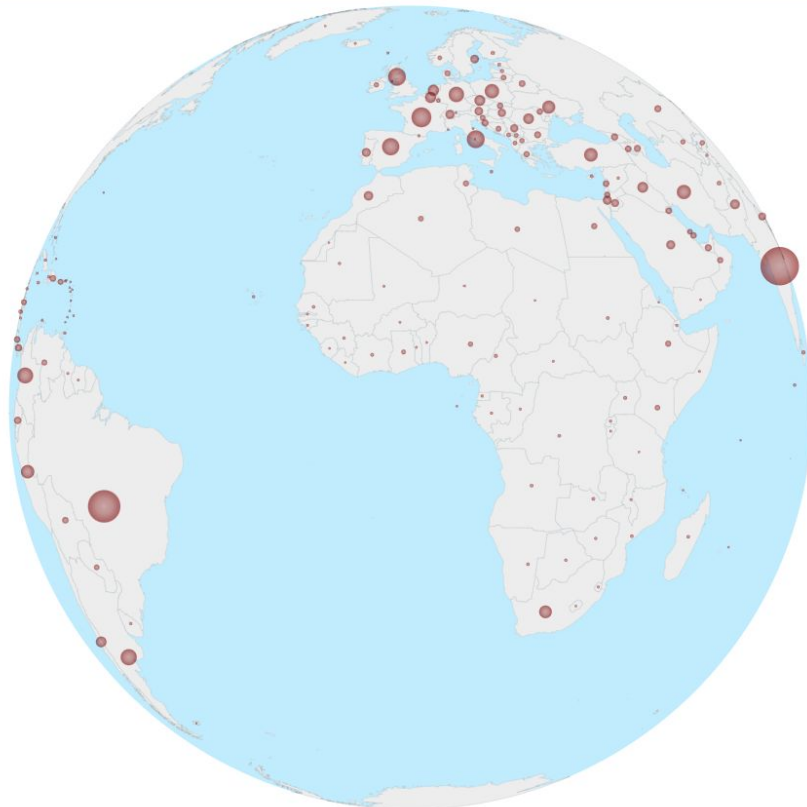
Avec quels **langages**  
peut-on faire de la  
dataviz ?

Comment créer des  
**dashboards** interactifs ?



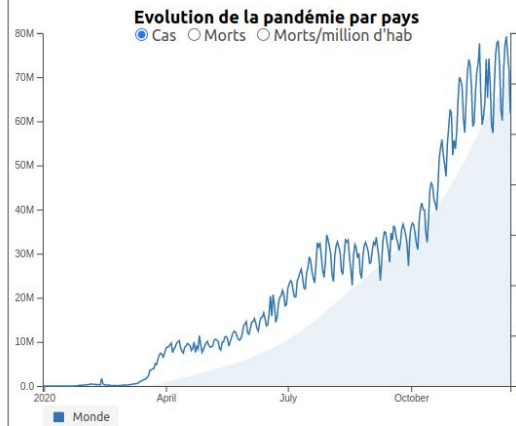
# CAS PERSONNELS





Rechercher pays

Pays	Morts	Cas	Morts/million d'hab.
États-Unis	299 177	16 256 754	909
Inde	143 355	9 884 100	105
Brésil	181 402	6 901 952	860
Russie	46 941	2 653 928	322
France	57 911	2 377 618	864
Royaume-Uni	64 170	1 849 403	963



**Période des données**

31/12/2019 - 14/12/2022

**Légende de la Carte**

**Projection géographique**

● Orthographique ○ Mercator

**Donnée des cercles**

● Cas ○ Morts

**Volume de cas**

● 63 976  
● 779 290  
● 3 734 575  
● 16 256 754

**Morts par million d'habitant**

0  
0.0 à 157.1  
157.1 à 469.0  
469.0 à 812.6  
812.6 à 1 567.0  
1 567.0 ou plus

**Légende du Graphique**

**Granularité des données**

● Jours  
○ Semaines  
○ Mois

**Signification des courbes**

— Evolution par jour  
■ Valeurs cumulée



## Sites visibles

- ACEBI - Pays de la Loire
- DAHER - Pays de la Loire
- Naval Group - Pays de la Loire
- Cetim - Pays de la Loire
- Mauric - Pays de la Loire
- Bureau Mauric - Pays de la Loire
- Bessé - Pays de la Loire
- Segula technologies - Pays de la Loire
- Agence Ship st - Pays de la Loire
- Technicatome - Pays de la Loire
- Engie Axima - Pays de la Loire
- Wärtsilä - Pays de la Loire
- Nexeya - Pays de la Loire
- Socomec - Pays de la Loire
- Segula technologies - Pays de la Loire
- EMC2 - Pays de la Loire
- Airbus - Pays de la Loire
- DAHER - Pays de la Loire
- Sirehna - Pays de la Loire
- Dassault Systemes -



## Légende

### Nickel

Importation d'oxide (en T)



361 472  
271 104  
180 736  
90 368

Pays concernés

### Les usines de production de batteries

Localisation Usine (73)

### Compagnies minières

Cobalt (30)  
 Nickel (33)  
 Graphite (21)  
 Manganèse (15)  
 Lithium (24)



Co

Ni

C

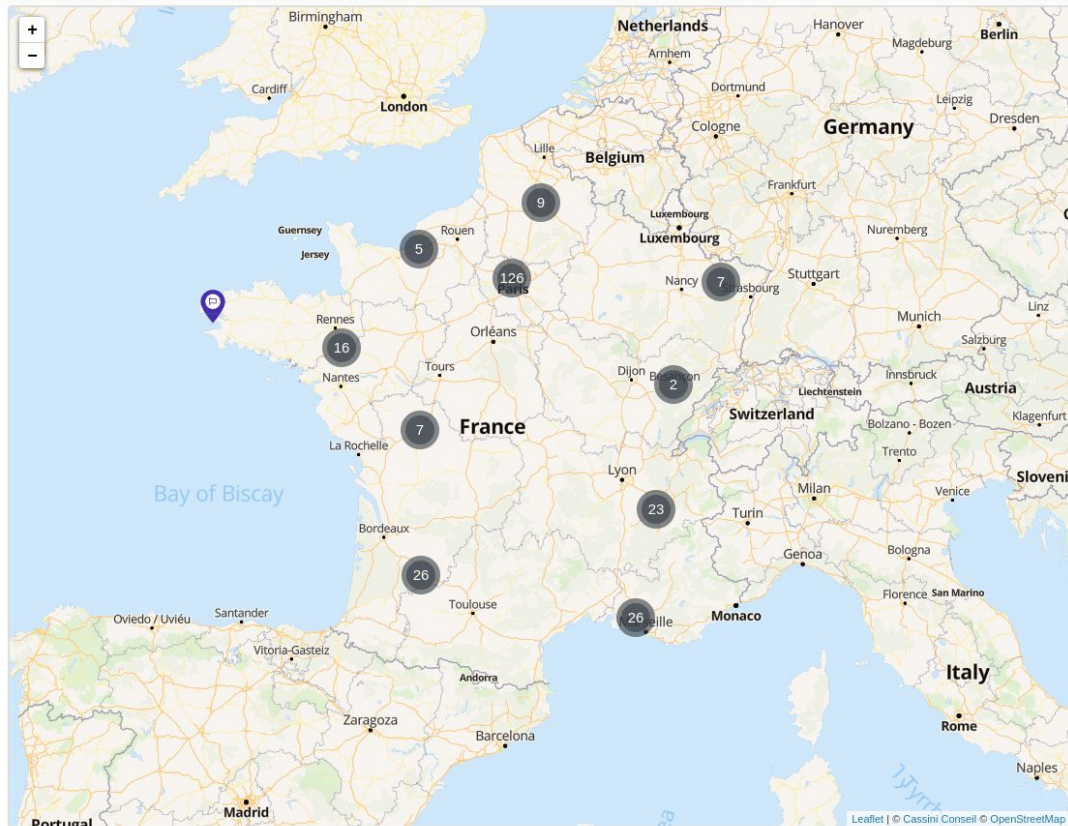
Mn

Li

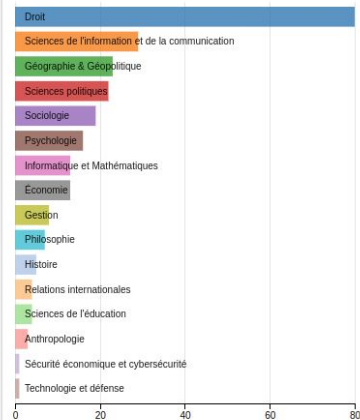


CASSINI PIRELLA





## Spécialité ⓘ



## Régions ⓘ

France ▾

## Chercheurs ⓘ

Rechercher un chercheur

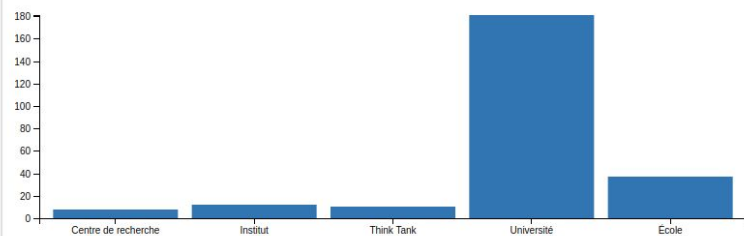
## Proximité ⓘ



## Volume de publications ⓘ



## Répartition par établissement





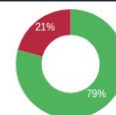
## Publications ⓘ

1,321 publications sélectionnées sur 1,321

Titre	Chercheur	Spécialité du chercheur	Année	Langue	Thématique
# Affaire Benalla: déconstruction d'une polémique sur le rôle de la communauté Twitter «russophile» dans le débat politique français	COLIN Gérard	Géographie & Géopolitique	2020	Français	RS et influence médiatique
(Cyberharcèlement : Sortir de la violence, à l'école et sur les écrans	BERENGERE Stassin	Sciences de l'information et de la communication	2019	Français	Cyberharcèlement
(cyberharcèlement. Sortir de la violence, à l'école et sur les écrans	BERENGERE Stassin	Sciences de l'information et de la communication	2019	Français	Cyberharcèlement
17 Key Takeaways From Our Response to the EU White Paper on AI	CHRISTAKIS Théodore	Droit	2020	Anglais	Intelligence artificielle
A Systemic Approach to Interpersonal Relationships and Activities among French Teleworkers	VAYRE Emilie	Psychologie	2014	Anglais	E-work et E-learning
A cartography of web tracking using dns records	GRUMBACH Stéphane	Géographie & Géopolitique	2019	Anglais	Données numériques et datasphère
A close look at France's new Military Cyber Strategy	DELERUE François	Droit	2019	Anglais	Géopolitique et cyberstratégie
A market of black boxes: The political economy of Internet surveillance and censorship in Russia	ERMOSHINA Ksenia	Sciences de l'information et de la communication	2021	Anglais	Souveraineté numérique
A study of vulnerabilities associated with excessive internet use in adolescence	GALAN Clémentine	Psychologie	2016	Anglais	Cyberpsychologie
A systematic literature review of Blockchain-enabled smart contracts: platforms, languages, consensus, applications and choice criteria	DHAIOUIR Samya	Gestion	2020	Anglais	Economie numérique
AI for Security Purposes. The Proposals of the French White Paper on Internal Security	CHRISTAKIS Théodore	Droit	2020	Anglais	Géopolitique et cyberstratégie
Accord du 14 janvier 2018 et zones blanches : fin d'une certaine fracture numérique ?	GUERRIER Claudine	Droit	2018	Français	Droit numérique
Addiction à Internet à l'adolescence et chez le jeune adulte : déterminants cliniques de l'usage problématique	GALAN Clémentine	Psychologie	2019	Français	Cyberpsychologie
Addressing the State : reconfiguration des normes et des processus de légitimité à Taiwan : du réseau de hackers militants au Yuan Exécutif	BOURGEOIS Fiorella	Sciences politiques	en cours	Français	Droit numérique
Adoption et impact de la Blockchain dans la transformation digitale de l'industrie 4.0 et de la Supply-chain: Qualité, sécurité et gouvernance des 'smart contracts'	DHAIOUIR Samya	Gestion	en cours	Français	Economie numérique
Algorithmic Projections and Cyberspace	BERTHIER Thierry	Informatique et Mathématiques	2013	Anglais	Données numériques et datasphère
Alternatives for the Internet: A Journey into Decentralised Network Architectures and Information Commons	MUSIANI Francesca	Sociologie	2020	Anglais	Données numériques et datasphère
Amateurisme cartographique et néonographie à l'heure du web	RAKIS Hervé	Géographie & Géopolitique	2010	Français	Géopolitique et

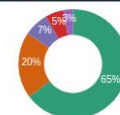
## Langue

Anglais  
Français

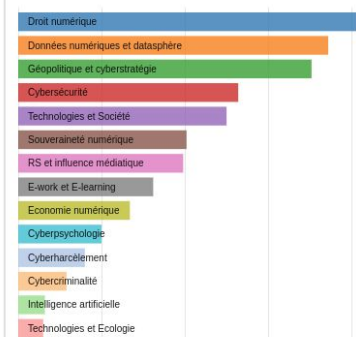


## Type d'établissement ⓘ

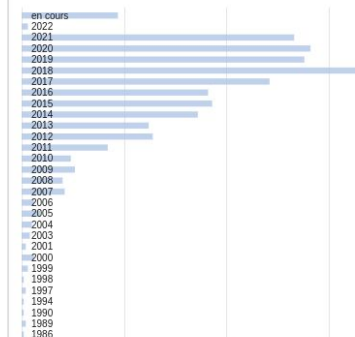
Université  
Ecole  
Institut  
Centre de recherche  
Think Tank



## Thématique ⓘ



## Année de publication



2D Graph 3D Graph

☐ Montrer / cacher les labels

Couleurs des nœuds par:

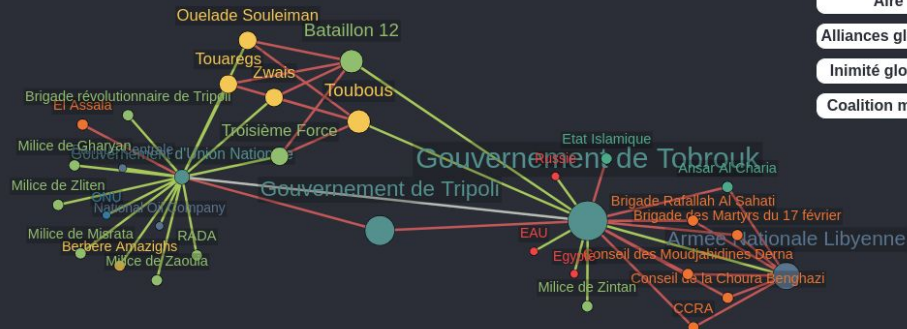
Type

Aire

Alliances globales

Inimité globales

Coalition milices



Couleur des nœuds

En fonction de "type"



Couleur des liens



Degré des nœuds



Nombre total de nœuds

32

Choix des alliances

Pro-Gouvernement d'Union Nationale

Pro-Gouvernement de Tobrouk

Pro-Gouvernement de Tripoli

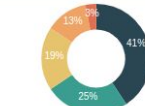
Soutiens

Opposants

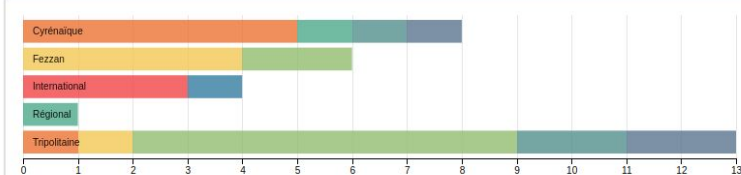
Répartition des acteurs par types



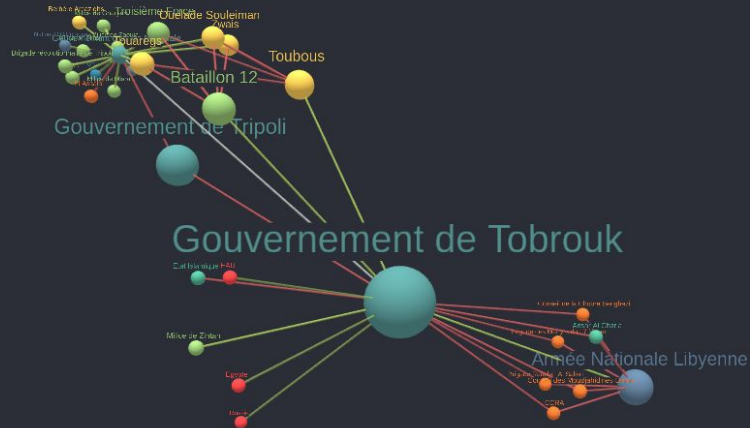
Répartition des acteurs par aires



Répartition des acteurs par "aire" par type



2D Graph 3D Graph

☐ Montrer / cacher les labels

## Couleur des noeuds

### En fonction de "type"

- Etat étranger
- Milice Islamiste
- Groupe ethnique
- Milice Autre
- Milice Djihadiste
- Gouvernement local
- Institution locale
- Organisation
- Autres

### Couleur des liens

- Soutien  
Oppose  
Processus d'accord

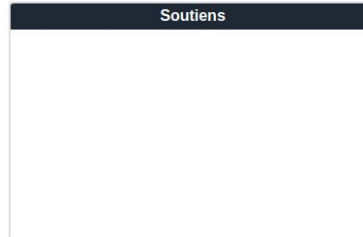
**Nombre total de nœuds**

32

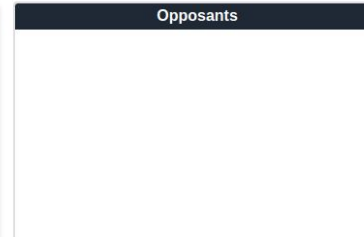
### Choix des alliances

Pro-Gouvernement d'Union  
NationalePro-Gouvernement de  
TobroukPro-Gouvernement de  
Tripoli

## Soutiens



## Opposants



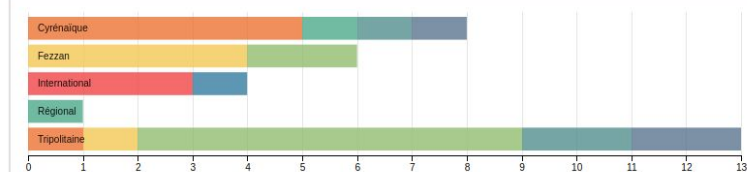
### Répartition des acteurs par types



### Répartition des acteurs par aires



### Répartition des acteurs par "aire" par type



# D3.JS

D3 est l'acronyme de Data-Driven Documents. D3.js est une bibliothèque JavaScript permettant de manipuler des documents basés sur des données. D3.js est un cadre de visualisation de données en ligne, dynamique et interactif.

La bibliothèque D3.js nous permet de manipuler des éléments d'une page web dans le contexte d'un ensemble de données. Ces éléments peuvent être des éléments HTML, SVG ou Canvas et peuvent être introduits, supprimés ou modifiés en fonction du contenu de l'ensemble de données. Il s'agit d'une bibliothèque permettant de manipuler les objets DOM.

Il faut inclure D3.JS avec la ligne suivante dans le fichier HTML

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

<https://www.tutorialspoint.com/d3js/index.htm>

<https://d3js.org/>

# D3.js – SVG



**Matriciel**  
.jpeg .gif .png



**Vectoriel**  
.svg

SVG signifie Scalable Vector Graphics, en clair c'est un format de données utilisé pour définir des graphiques vectoriels.

Ce format est basé sur du XML et permet de définir des éléments graphiques pour le web.

Certaines formes géométriques de base sont déjà gérées telles que les rectangles ou les ellipse. Il est également possible d'obtenir n'importe quelle forme à l'aide des paths qui sont des chemins et vous permettent de tracer ce que vous souhaitez.



Run >

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg height="100" width="100">
```

```
<circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
```

```
Sorry, your browser does not support inline SVG.
```

```
</svg>
```

```
</body>
```

```
</html>
```



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg width="400" height="110">
```

```
<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />
```

```
Sorry, your browser does not support inline SVG.
```

```
</svg>
```

```
</body>
```

```
</html>
```





```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg width="400" height="180">
```

```
<rect x="50" y="20" width="150" height="150" style="fill:blue;stroke:pink;stroke-width:5;fill-opacity:0.1;stroke-opacity:0.9" />
```

Sorry, your browser does not support inline SVG.

```
</svg>
```

```
</body>
```

```
</html>
```



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg height="150" width="500">
```

```
<ellipse cx="240" cy="100" rx="220" ry="30" style="fill:purple" />
```

```
<ellipse cx="220" cy="70" rx="190" ry="20" style="fill:lime" />
```

```
<ellipse cx="210" cy="45" rx="170" ry="15" style="fill:yellow" />
```

Sorry, your browser does not support inline SVG.

```
</svg>
```

```
</body>
```

```
</html>
```



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg height="200" width="500">
```

```
<polyline points="20,20 40,25 60,40 80,120 120,140 200,180" style="fill:none;stroke:black;stroke-width:3" />
```

Sorry, your browser does not support inline SVG.

```
</svg>
```

```
</body>
```

```
</html>
```



# D3.js – Selection

D3.JS permet de sélectionner un ou plusieurs éléments dans le DOM grâce aux fonctions:

- `d3.select(<query selector>)` => la première sélection trouvée ou null
- `d3.selectAll(<query selector>)` => une liste des sélections trouvées (si aucune alors ça sera une liste vide)

Le `<query selector>` est une chaîne de caractère pouvant être construite comme sur ce site: [https://www.w3schools.com/cssref/css\\_selectors.php](https://www.w3schools.com/cssref/css_selectors.php)

`d3.select("svg")` => me renvoie le premier élément qui est un **svg**

`d3.select("#id1")` => me renvoie le premier élément qui a l'id **id1**

`d3.select(".class1")` => me renvoie le premier élément qui a la classe **class1**

`d3.selectAll(".class2")` => me renvoie tous les éléments qui ont la classe **class2**

`d3.selectAll("circle")` => me renvoie tous les éléments qui sont des éléments **circle**

`d3.selectAll("input:checked")` => me renvoie tous les éléments qui sont des éléments **input** et qui sont **checked** (les checkbox qui sont cliquées)

`d3.select("#maViz").selectAll("*")` => me renvoie absolument tous les éléments qui sont présents dans l'élément qui a pour id **maViz**

# D3.js – Modification d'une sélection

Quand vous avez sélectionné un ou plusieurs élément(s) vous pouvez modifier cette/ces selection(s) avec des fonctions spéciales donc les plus utiles :

```
let selection = d3.select("#monElement")
```

- `const span = selection.append("span")` => ajoute un élément **span** à selection et on garde l'élément ajouté span dans la variable span
- `span.text("Ma span")` ou `selection.append("span").text("Ma span")` => ajoute un élément **span** à selection et met son texte à **Ma span** (on peut chaîner les instructions)
- `selection.append("div").html("<span>Ma span</span>")` => ajoute une **div** et met son contenu HTML à **<span>Ma span</span>** (revient à ajouter un span et lui mettre son texte à **Ma span**).
- `span.attr("id", "span1")` => met l'attribut id de span à span1 (du coup on pourra y accéder via `d3.select("#span1")`)
- `span.style("color", "blue")` => met le style color de span à blue, son texte sera donc bleu
- `span.classed("maClass", <true | false>)` => ajoute la classe maClass de span si true et l'enlève si false
- `span.classed("maClass")` => renvoie true si span a la classe maClass ou false sinon

# D3.js – Jointure des données

La grande force de D3.JS est de lier de la donnée à un ou plusieurs éléments par le biais d'une sélection puis un ajout de données.

`<ul id="monUI"></ul>` => HTML de la page

```
d3.select("#monUI").selectAll(".mesLi").data([1, 2, 3]).enter().append("li").classed("mesLi", true).text(function(d) { return d; });
```

On récupère l'élément avec l'id `monUI` et on veut sélectionner tous les éléments qui ont la classe `mesLi` dedans (aucun pour le moment, mais ça nous permet de faire une sélection de liste). On lie la liste `[1,2,3]` à cette sélection puis on `entre` dans la cette sélection multiple. Pour chaque valeur de la liste on `ajoute` un élément `li` en lui ajoutant la classe `mesLi` et en lui mettant comme `texte` la valeur de la liste à l'indice de l'élément

`<ul id="monUI">` => HTML après la ligne

```
  <li class="mesLi">1</li>
```

```
  <li class="mesLi">2</li>
```

```
  <li class="mesLi">3</li>
```

```
</ul>
```

## D3.js – Autre

On peut presque tout faire avec D3.JS, la librairie peut même remplacer jQuery.

Une liste des choses possible avec D3.JS

- Création de SVG et d'images canvas (exportables en PNG)
- Animations
- Création de graphiques (de façon native ou avec des librairies comme DC.js)
- Création de graphes
- ...

# DC.js

DC.js est une bibliothèque graphique construite au-dessus de D3.js et fonctionne nativement avec crossfilter, qui est une autre bibliothèque JavaScript populaire utilisée pour explorer des millions d'enregistrements dans une courte période du côté client. DC.js est une bibliothèque JavaScript utilisée pour créer des tableaux de bord interactifs en JavaScript.

Il faut inclure D3.js, crossfilter et DC.js pour l'utiliser

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/dc/4.2.7/style/dc.min.css"/>  
<script src="https://d3js.org/d3.v7.min.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/crossfilter/1.3.12/crossfilter.min.js"></script>  
<script src="https://cdnjs.cloudflare.com/ajax/libs/dc/4.2.7/dc.min.js"></script>
```

<https://www.tutorialspoint.com/dcjs/index.htm>

<https://dc-js.github.io/dc.js/>

<https://github.com/square/crossfilter/wiki/API-Reference>



# DC.js – crossfilter

Considérons un objet crossfilter chargé avec une collection de fruits et de légumes

```
const fruits = crossfilter ([
  { name: "Pomme", type: "fruit", count: 3 },
  { name: "Carotte", type: "legume", count: 8 },
  { name: "Poire", type: "fruit", count: 50 },
  { name: "Salade", type: "legume", count: 40 }
]);
```

const filtering = fruits.dimension(function(d) { return d.type; }); => Si nous voulons filtrer par type

filtering.filter("fruit"); => Ne garder que les fruits

console.log(console.log(filtering.top(Infinity))) => Permet de montrer TOUS les éléments filtrés par type où le type est fruit

```
[{"name":"Poire","type":"fruit","count":50},{"name":"Pomme","type":"fruit","count":3}]
```

const grouping = filtering.group(); => Création d'un groupe (Map - Reduce) sur notre dimension

console.log(grouping.reduceCount().top(Infinity)); => Permet d'avoir le nombre d'élément dans chaque type

```
[{"key":"legume","value":2},{"key":"fruit","value":2}]
```

# DC.js – Les graphiques

Tous les graphiques se construiront de la même façon:

- Création d'un crossfilter avec la donnée.
- Création d'une dimension à représenter (suivant votre cas d'étude et le type de graphique).
- Création d'un groupe à partir de la dimension à représenter (suivant le type de graphique).
- Création d'un graphique qui répond à ce que vous voulez montrer avec le "chartGroup" ( Dans DC.js, les graphiques peuvent être regroupés en un seul ensemble. Tous les graphiques d'un groupe sont censés partager le même jeu de données Crossfilter. Ils sont rendus et redessinés simultanément.).
- Ajout de la dimension au graphique
- Ajout du groupe au graphique
- Ajout des options spécifiques au graphique

# DC.js – Quelques exemples

La meilleure façon de comprendre comment ça marche est d'aller voir des exemples et de pratiquer, la seule partie compliqué de DC.js est de savoir quoi prendre en dimension et comment construire les groupes.

<https://dc-js.github.io/dc.js/examples/>

- <https://dc-js.github.io/dc.js/examples/adjustable-threshold.html>
- <https://dc-js.github.io/dc.js/examples/boxplot-enhanced.html>
- <https://dc-js.github.io/dc.js/examples/brush-ordinal.html>
- <https://dc-js.github.io/dc.js/examples/filtering-removing.html>
- <https://dc-js.github.io/dc.js/examples/heat.html>
- <https://dc-js.github.io/dc.js/examples/horizon-chart.html>
- <https://dc-js.github.io/dc.js/examples/range-series.html>
- <https://dc-js.github.io/dc.js/examples/sunburst-cat.html>
- Recherchez sur google !