

Visualisation de données

03 - Introduction à D3.js

22 février 2024

Noemi Romano
noemi.romano@heig-vd.ch

Semaine passée

Semaine passée

Fonction Explicative ou exploratoire

Semaine passée

Fonction Explicative ou exploratoire

Formes Statique ou interactive

Semaine passée

Fonction Explicative ou exploratoire

Formes Statique ou interactive

SVG Formes, dessins, transformation et animations

D3

 by  Observable



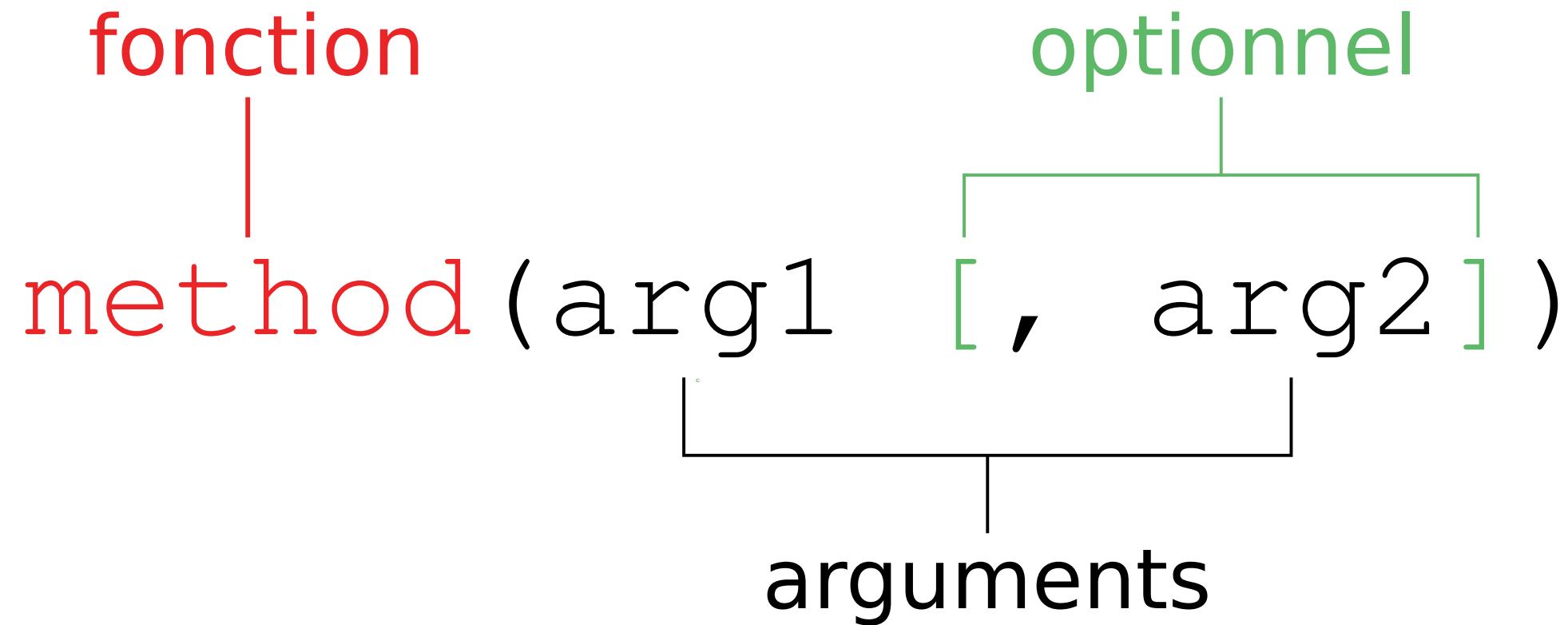
- **Data-Driven Approach** : vous liez vos données directement aux éléments du DOM
- **Manipulation du DOM** : vous associez des données à des éléments HTML existants (ou d'en créer de nouveaux)
- **Construction de Graphiques** : vous créez des éléments graphiques en langage SVG à partir de données



The JavaScript library for
bespoke data
visualization

↳ d3

Documentation



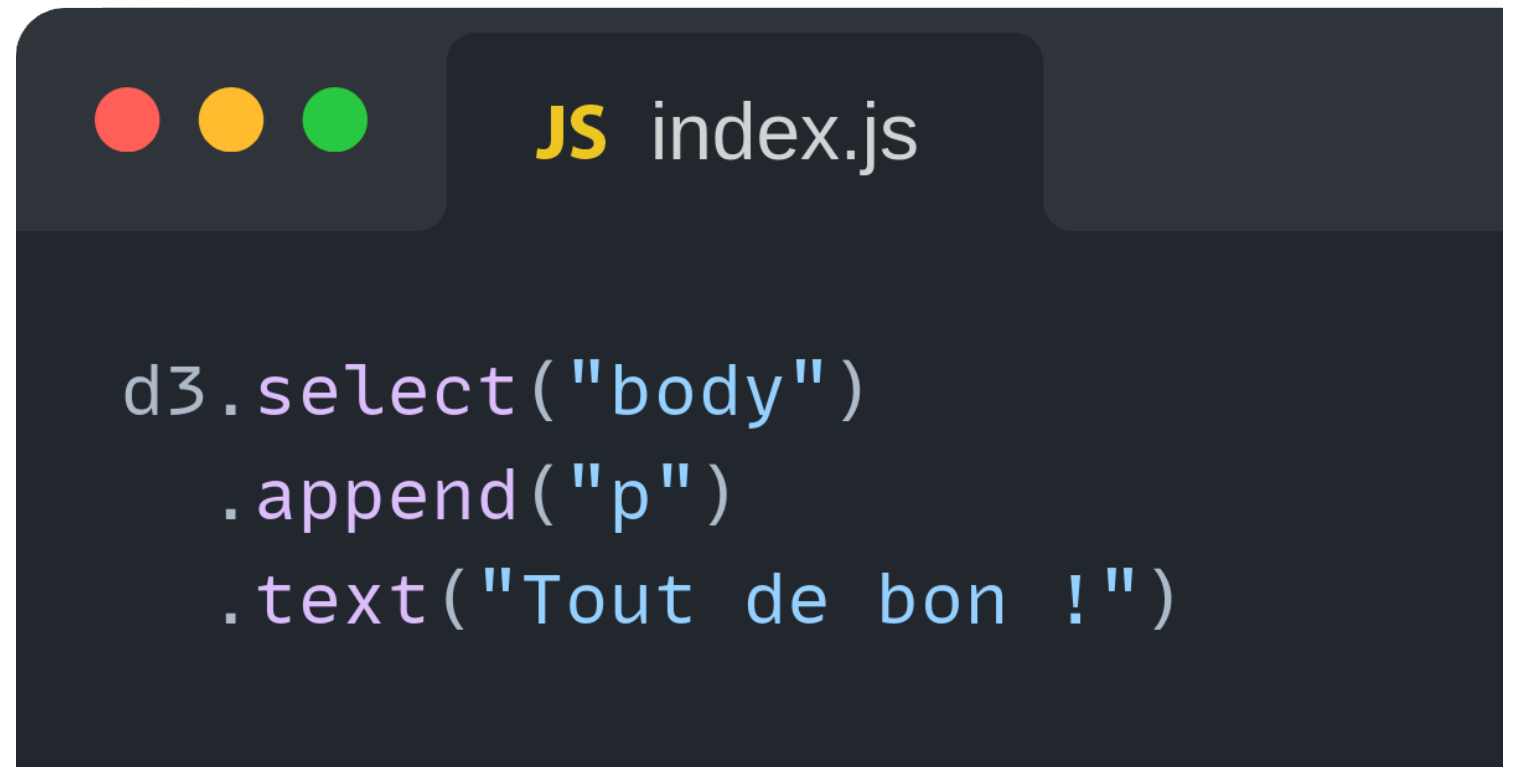
↳ Github: API reference

Syntaxe

Chaînage de méthodes

- Enchaîner plusieurs méthodes sur une sélection ou un objet D3
- Chaque méthode renvoie un objet qui peut être manipulé par d'autres méthodes

`objet.methode1().methode2().methode3()`



```
d3.select("body")
  .append("p")
  .text("Tout de bon !")
```

Tout de bon !

d3-selection

Installation

```
npm install d3-selection
```

Sélectionner

d3.select(selector)

d3.selectAll(selector)

- **d3.select** : méthode qui permet de sélectionner un élément unique du DOM en fonction d'un sélecteur CSS
- **d3.selectAll** : méthode qui permet de sélectionner tous les éléments correspondant au sélecteur CSS donné
- **selector** : le sélecteur CSS utilisé pour sélectionner les éléments du DOM

	CSS	Exemple
type	element	.select('h1')
class	.class	.select('.class')
identifiant	#id	.select('#id')

↳ [d3-selection - Selecting elements](#)

Modifier

selection.attr(name[, value])

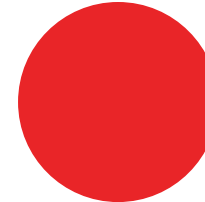
- **attr**: méthode qui permet de modifier les attributs des éléments sélectionnés
- **name**: le nom de l'attribut à modifier
- **value** (optionnel): la valeur à assigner à l'attribut

```
index.html<svg id="mon-svg">
  <circle cx="50%" cy="50%" r="50"></circle>
</svg>
```

```
JS index.jsimport { select } from 'd3-selection';

const monSvg = select("#mon-svg");
const cercle = monSvg.select("circle");

cercle.attr("fill", "red");
```



↳ d3-selection - Modifying elements

Créer

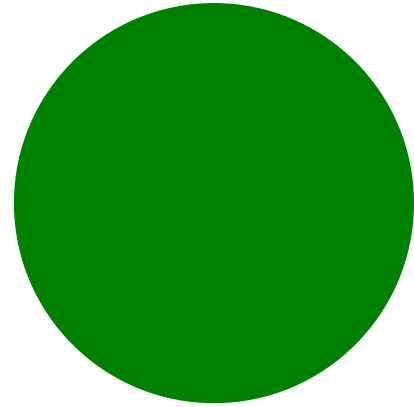
selection.append(element)

- **append** : méthode qui permet d'ajouter de nouveaux éléments DOM à une sélection existante
- **element** : le type d'élément à créer (balise HTML)

JS index.js

```
const WIDTH = 500
const HEIGHT = 800

const cercle = select("body")
    .append("svg")
    .attr("width", WIDTH)
    .attr("height", HEIGHT)
    .append("circle")
    .attr("cx", "30%")
    .attr("cy", "40%")
    .attr("r", "100")
    .attr("fill", "green")
```



↳ d3-selection - Modifying elements

Événements

selection.on(eventType, callback)

- **on** : méthode qui permet d'attacher des gestionnaires d'événements à une sélection d'éléments du DOM
- **eventType** : le type d'événement que vous souhaitez gérer (par exemple, *click*, *mouseover*, *keydown*, etc.).
- **callback** : la fonction de rappel (callback) qui sera exécutée lorsque l'événement se produit.

```
JS index.js

import { select } from "d3-selection";

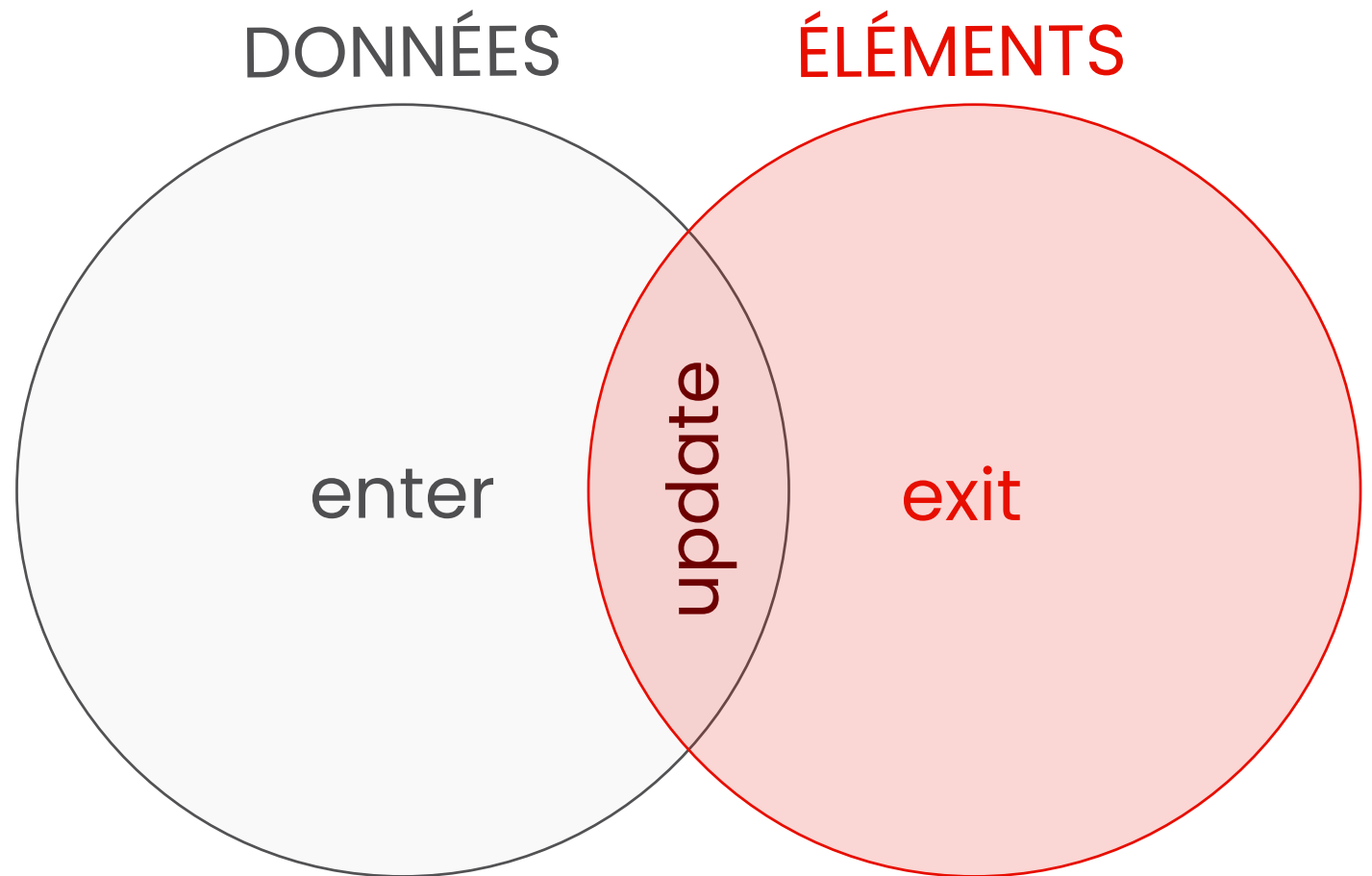
select("#mon-svg")
  .select("circle")
  .on("click", function () {
    select(this).attr("fill", "green");
  })
  .on("mousemove", function (e) {
    console.log("x: " + e.clientX + ", y:" + e.clientY);
  });
```

↳ d3-selection - Handling events

Données

selection.data(tableauDonnees[,key])

- **data** : méthode qui permet de lier les données aux éléments du DOM existants.
- **tableauDonnees** : tableau de données à lier aux éléments du DOM.
- **key** (optionnel): fonction ou une valeur qui spécifie comment faire correspondre les données aux éléments du DOM existants en utilisant une clé. Si la clé n'est pas fournie, la correspondance est faite par indice.

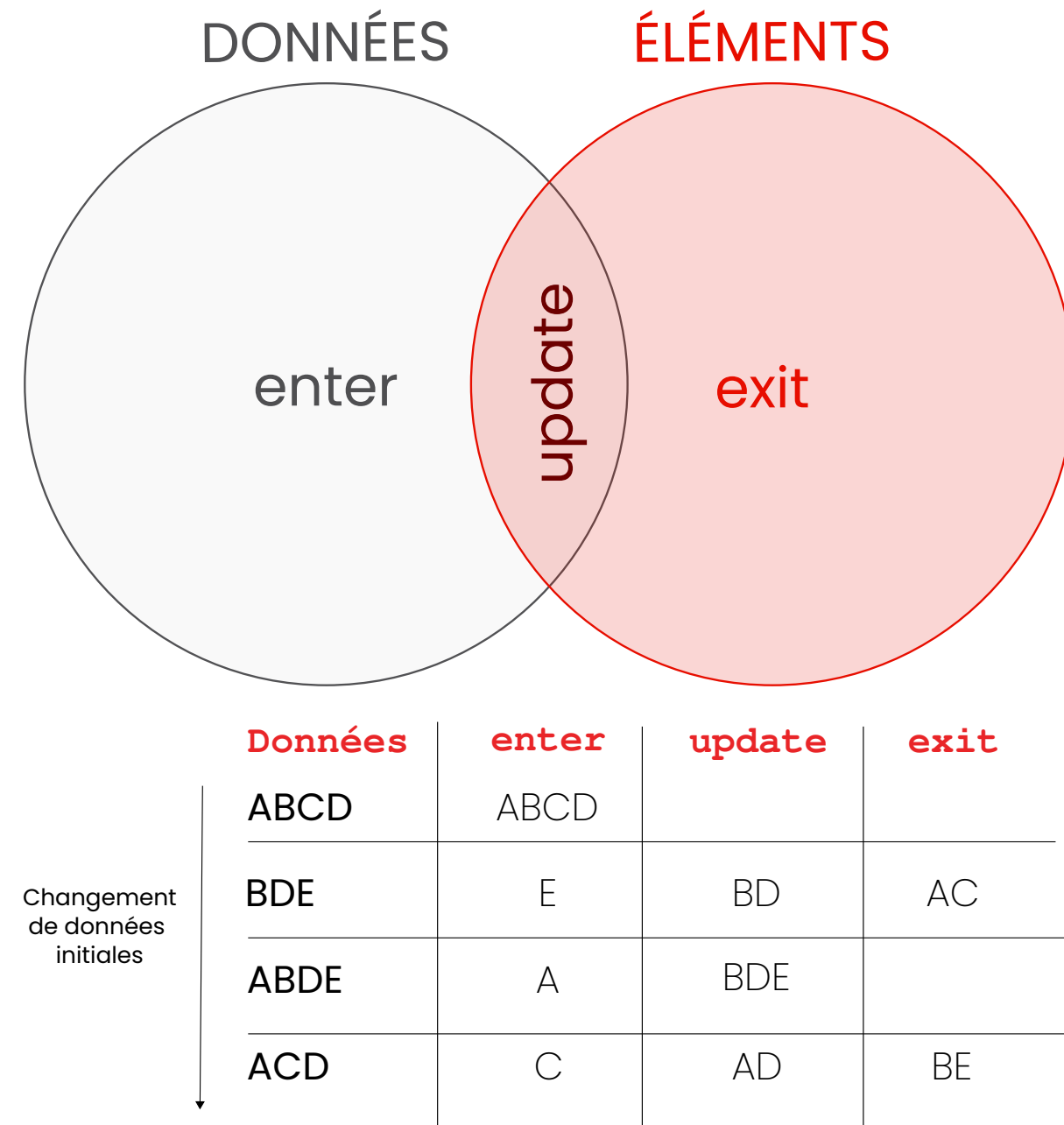


↳ d3-selection - Joining data

Données

`selection.data(data).join(enter, update, exit)`

- **join** : méthode introduite dans D3.js version 6, simplifiant la gestion des nouvelles données, des mises à jour et des éléments à supprimer
- **enter** : représente les nouveaux éléments à créer pour correspondre aux nouvelles données.
- **update** : représente les éléments existants qui correspondent aux données actuelles et qui doivent être mis à jour.
- **exit** : représente les éléments sans correspondance avec les données actuelles et qui doivent être supprimés.



↳ [d3-selection - Joining data](#)

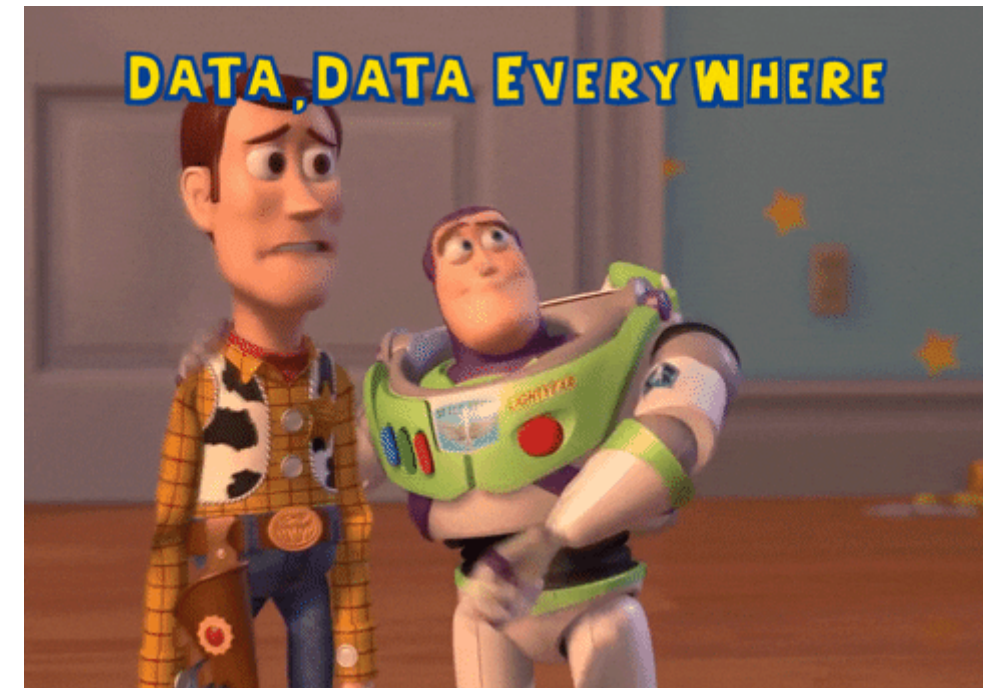
Examples

↳ Observable: d3-selection

Projet

Thématique (8 mars)

- Former les groupes (2-3 personnes par groupe)
- Trouver un jeu de données
- Créer un dossier github pour le projet
- Suivre directives [ici](#)



Examples

- Titanic
- Global gender gap
- The Rythm of food
- Satellites
- Gender and Language

Exercices

↳ [Github - comem-visualdon/exercices](#)