

## Quelques exercices pour mieux comprendre la librairie D3.js

### 1. Création d'une div avec D3.js:

<pre>1 2 3</pre>	HTML	<pre>1 #text{ 2   color:#fff; 3 }</pre>
<pre>1 // On peut aussi créer une div avec d3.js 2 var myDiv = d3.select("body").append("div"); 3 4 myDiv.style("background-color", "#000") 5   .attr("id", "text") 6   .text("Hello World");</pre>	JavaScript	Result Hello World

### 2. Création d'un cercle

<pre>1 var canvas=d3.select("body") 2   .append("svg") 3   .attr("width",140).attr("height",80); 4 var circle= canvas.append("circle") 5   .attr("cx", 60).attr("cy",50) 6   .attr("r", 20) 7   .attr("fill", "red"); 8</pre>	JavaScript	Result 
---	------------	--

- Pour créer le cercle on donne son positionnement horizontal cx et son positionnement vertical cy ainsi que son rayon r et sa couleur rouge
- Les trois premières lignes permettent de créer un élément SVG et l'ajouter au body du html en précisant son positionnement
- les lignes de 4 à 7 permettent de rajouter le cercle dans le canvas

### 3. Manipuler un rectangle SVG

<pre>1 // On crée un élément svg qu'on ajoute au body du html 2 var svg = d3.select("body").append("svg"); 3 4 // On ajoute en ensuite un rectangle qu'on ajoute dans le svg 5 svg.append("rect") 6   .attr("x",100) 7   .attr("y",100) 8   .attr("width",200) 9   .attr("height",100) 10  .style("fill", "#000") 11  .style("stroke", "red") 12  .style("stroke-width", "10")</pre>	JavaScript	Result 
--	------------	--

d3.select("body").append("svg"): création d'un élément de type SVG et l'ajouter au body du html

svg.append("rect"): ajout d'un objet SVG <rect>

#### 4. Gérer des données au format JSON et dessiner en fonction de ces données

The screenshot shows a code editor with three panels. The top-left panel is labeled 'HTML' and is empty. The top-right panel shows CSS code: 

```
1 text {
2   font: 10px sans-serif;
3 }
```

. The bottom-left panel is labeled 'JavaScript' and contains the following code: 

```
1 var data = [
2   {"x":190,"y":34,"r":20,"color":"#FDAE6B","name":"cercle 1"},
3   {"x":150,"y":89,"r":39,"color":"#FDD0A2","name":"cercle 2"},
4   {"x":267,"y":234,"r":38,"color":"#A1D99B","name":"cercle 3"},
5   {"x":100,"y":34,"r":29,"color":"#31A354","name":"cercle 4"},
6   {"x":87,"y":89,"r":21,"color":"#3182BD","name":"cercle 5"},
7   {"x":160,"y":248,"r":25,"color":"#FD8D3C","name":"cercle 6"},
8   {"x":78,"y":150,"r":35,"color":"#A1D99B","name":"cercle 7"},
9   {"x":345,"y":156,"r":32,"color":"#9ECAE1","name":"cercle 8"},
10  {"x":233,"y":117,"r":24,"color":"#C6DBEF","name":"cercle 9"}
11 ];
12
13 var svg = d3.select("body").append("svg");
14
15 var nodes = svg.selectAll(".node")
16   .data(data)
17   .enter()
18   .append("g")
19   .attr("class","node")
20   .attr("transform", function(d)
21     { return "translate(" + d.x + "," + d.y + ")"; });
22
23
24 nodes.append("circle")
25   .attr("r", function(d) {return d.r})
26   .style("fill",function(d){return d.color;});
27
```

. The bottom-right panel shows the rendered output: a collection of nine circles of various colors and sizes arranged in a cluster.

Les données au format JSON (JavaScript Object Notation).

[

```
{ "x":190,"y":34,"r":20,"color":"#FDAE6B","name":"cercle 1"},
{ "x":150,"y":89,"r":39,"color":"#FDD0A2","name":"cercle 2"},
{ "x":267,"y":234,"r":38,"color":"#A1D99B","name":"cercle 3"},
{ "x":100,"y":34,"r":29,"color":"#31A354","name":"cercle 4"},
{ "x":87,"y":89,"r":21,"color":"#3182BD","name":"cercle 5"},
{ "x":160,"y":248,"r":25,"color":"#FD8D3C","name":"cercle 6"},
{ "x":78,"y":150,"r":35,"color":"#A1D99B","name":"cercle 7"},
{ "x":345,"y":156,"r":32,"color":"#9ECAE1","name":"cercle 8"},
{ "x":233,"y":117,"r":24,"color":"#C6DBEF","name":"cercle 9" }
```

]

C'est un tableau [] qui contient des objets {} eux-mêmes contenant 5 objets:  
x, y, r, color, name

- `svg.selectAll(".node")`: permet de sélectionner les éléments définis par le selecteur css passé en argument. `selectAll("p")` va sélectionner tous les éléments présents dans le DOM.
- `.data(data)`: La méthode `data()` permet d'attacher les données à des éléments sélectionnés par la méthode `selectAll()`.
- `.enter()`: permet d'attacher les données à des éléments et créer les éléments sélectionnés par `selectAll()`. Si on n'utilise pas `enter()` aucun élément ne sera créé