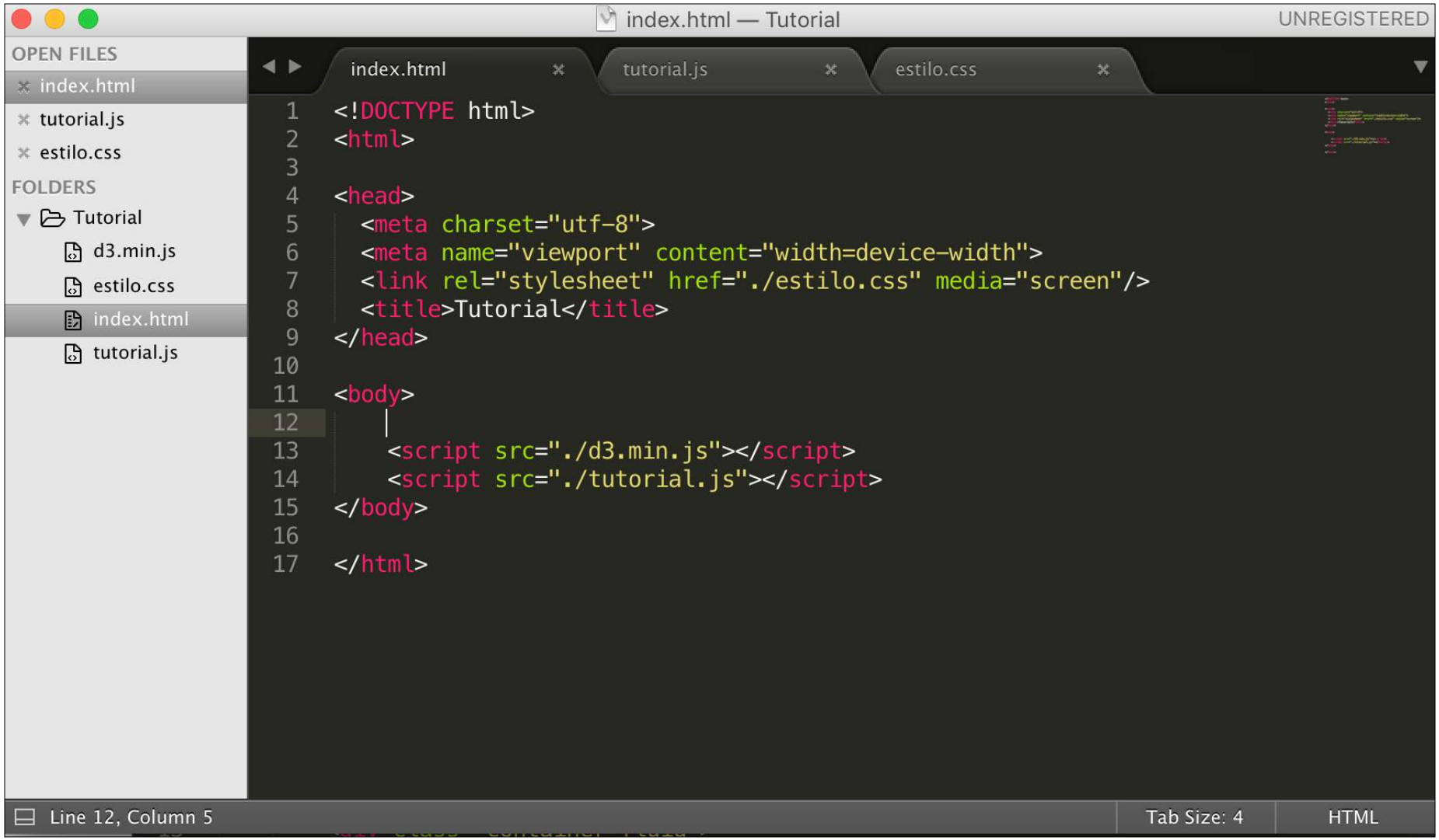

HTML


CSS


SVG


JS

# JS


index.html — Tutorial
UNREGISTERED
OPEN FILES
index.html
tutorial.js
estilo.css
FOLDERS
Tutorial
d3.min.js
estilo.css
index.html
tutorial.js
index.html
tutorial.js
estilo.css
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width">
7 <link rel="stylesheet" href="./estilo.css" media="screen"/>
8 <title>Tutorial</title>
9 </head>
10
11 <body>
12
13 <script src="./d3.min.js"></script>
14 <script src="./tutorial.js"></script>
15 </body>
16
17 </html>
Line 12, Column 5
Tab Size: 4
HTML

# Variables y tipos

```
a = 21;

b = a * 2;

console.log( b );
```

```
var a;
typeof a;               // "undefined"

a = "hello world";
typeof a;               // "string"

a = 42;
typeof a;               // "number"

a = true;
typeof a;               // "boolean"

a = null;
typeof a;               // "object" -- weird, bug

a = undefined;
typeof a;               // "undefined"

a = { b: "c" };
typeof a;               // "object"
```

# Objetos

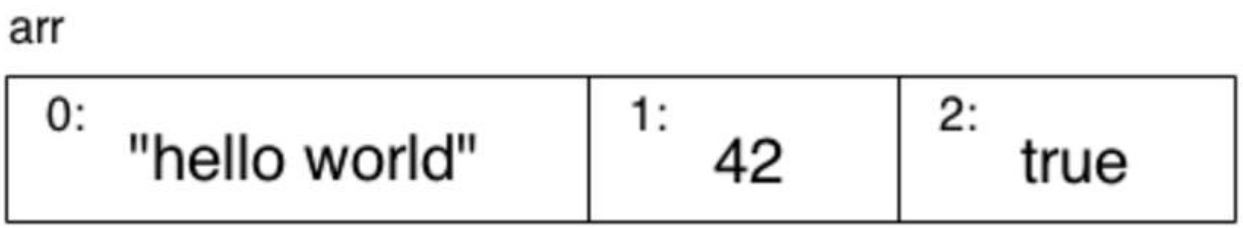
```
var obj = {
    a: "hello world",
    b: 42,
    c: true
};

obj.a;      // "hello world"
obj.b;      // 42
obj.c;      // true

obj["a"];   // "hello world"
obj["b"];   // 42
obj["c"];   // true
```

obj

| a: "hello world" | b: 42 | c: true |
|---|---|---|

# Arrays o arreglos

```
var arr = [
    "hello world",
    42,
    true
];

arr[0];          // "hello world"
arr[1];          // 42
arr[2];          // true
arr.length;      // 3

typeof arr;      // "object"
```



# Codicionales

```
const ACCESSORY_PRICE = 9.99;

var bank_balance = 302.13;
var amount = 99.99;

amount = amount * 2;

// can we afford the extra purchase?
if ( amount < bank_balance ) {
    console.log( "I'll take the accessory!" );
    amount = amount + ACCESSORY_PRICE;
}
// otherwise:
else {
    console.log( "No, thanks." );
}
```

`&&` (and), `||` (or)

`==` (loose-equals), `===` (strict-equals), `!=` (loose not-equals), `!==` (strict not-equals),

# Loops

```
while (numOfCustomers > 0) {
    console.log( "How may I help you?" );

    // help the customer...

    numOfCustomers = numOfCustomers - 1;
}

// versus:

do {
    console.log( "How may I help you?" );

    // help the customer...

    numOfCustomers = numOfCustomers - 1;
} while (numOfCustomers > 0);
```

```
for (var i = 0; i <= 9; i = i + 1) {
    console.log( i );
}
// 0 1 2 3 4 5 6 7 8 9
```

# Funciones

```
const TAX_RATE = 0.08;

function calculateFinalPurchaseAmount(amt) {
    // calculate the new amount with the tax
    amt = amt + (amt * TAX_RATE);

    // return the new amount
    return amt;
}

var amount = 99.99;

amount = calculateFinalPurchaseAmount( amount );

console.log( amount.toFixed( 2 ) );     // "107.99"
```

# Funciones dentro de funciones, scope

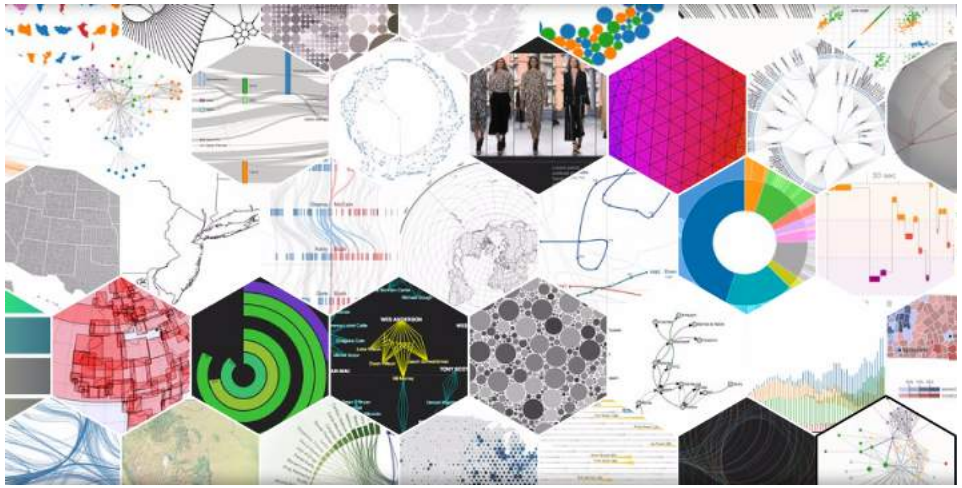
```
function outer() {
    var a = 1;

    function inner() {
        var b = 2;

        // we can access both `a` and `b` here
        console.log( a + b );   // 3
    }

    inner();

    // we can only access `a` here
    console.log( a );           // 1
}

outer();
```

# Visualizaciones de datos en la web

## Graficar y pintar cosas en la web

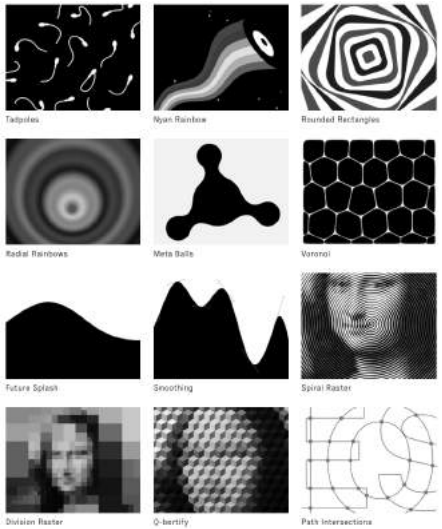
### SVG

(d3js.org)



### Canvas

Paper.js (paperjs.org)



# Data-Driven Documents

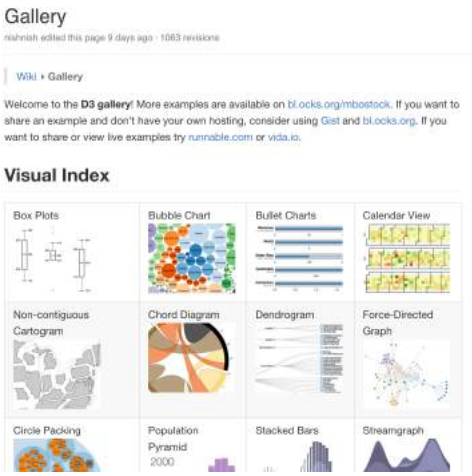
Documento HTML + Datos

D3js no es una librería para hacer gráficas

D3js no es una capa de gráficos
(D3 ni siquiera pinta nada)

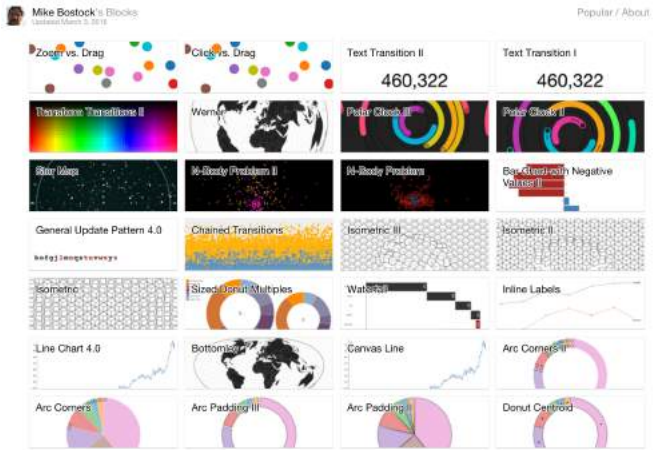
# Data-Driven Documents

## Miles de ejemplos




d3js.org

http://bl.ocks.org/mbostock





http://christopheviau.com/d3list/gallery.html

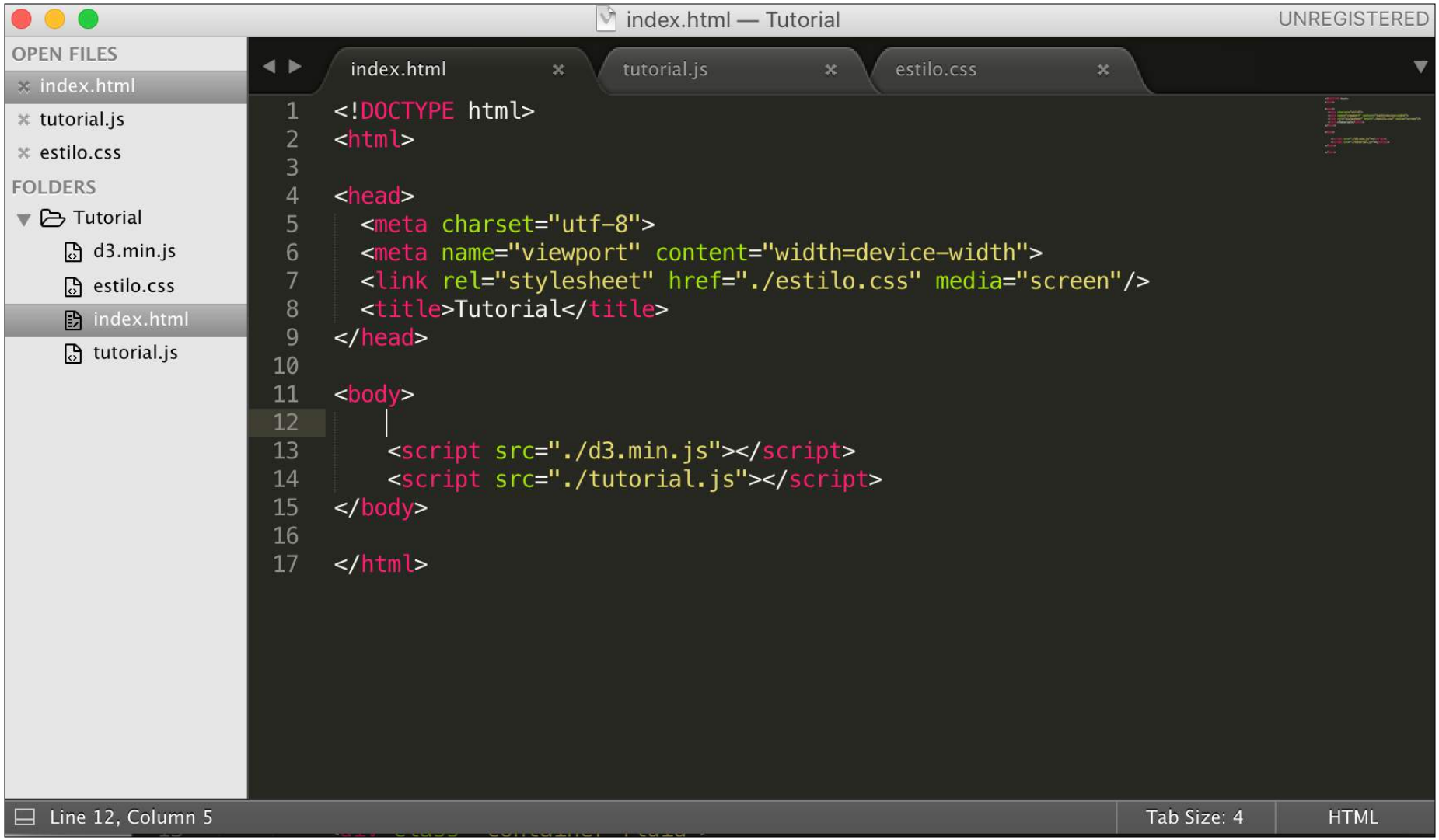
*Miles de tutoriales


HTML


CSS


SVG


JS


index.html — Tutorial
UNREGISTERED
OPEN FILES
index.html
tutorial.js
estilo.css
FOLDERS
Tutorial
d3.min.js
estilo.css
index.html
tutorial.js
index.html
tutorial.js
estilo.css
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width">
7 <link rel="stylesheet" href="./estilo.css" media="screen"/>
8 <title>Tutorial</title>
9 </head>
10
11 <body>
12
13 <script src="./d3.min.js"></script>
14 <script src="./tutorial.js"></script>
15 </body>
16
17 </html>
Line 12, Column 5
Tab Size: 4
HTML

```
<p>Hola</p>
<p>Soy Irving</p>
<p>Estoy aprendiendo D3</p>
```

Hola

Soy Irving

Estoy aprendiendo D3

# Selectores

```
<p id="saludo" class="algo">Hola</p>
<p class="algo">Soy Irving</p>
<p class="algo">Estoy aprendiendo D3</p>
```

# Selectores

```
#saludo {
    background-color: pink
}
```

```
.algo {
    color: blue
}
```

Hola

Soy Irving

Estoy aprendiendo D3

# Selectores

```
#foo            // <any id="foo">
foo             // <foo>
.foo            // <any class="foo">
[foo=bar]       // <any foo="bar">
foo bar         // <foo><bar></foo>
```

# Seleccionando con D3

```
d3.select("#saludo").style("font-size","50px")

d3.selectAll(".algo").style("color","purple");
```

# Seleccionando puedo agregar cosas al DOM

```
1 | d3.select("body").append("h1").html("My beautiful
  | text")
```

```
1 | d3.select("body").append("p").html("Look at me, I'm
2 | a paragraph.")
3 | d3.select("body").append("p").html("And I'm another
  | paragraph!")
  | d3.select("body").append("p").html("Woohoo! number
  | 3 baby")
```

# Seleccionando puedo quitar cosas del DOM

```
d3.selectAll("p").remove()
```

# ¿Que hace esta instrucción?

```
1 | d3.select("p").append("span").html("and I'm a
  span!")
```

# Estilos con D3

```
1 | d3.select("p").style("color","red")
```

```
1 | d3.select("span").style("color","blue")
```

# Atributos con D3

```
d3.select("body").append("p").html("First
paragraph");
d3.select("body").append("p").html("Second
paragraph").attr("class","p2");
d3.select("body").append("p").html("Third
paragraph").attr("id","p3");
```

```
d3.select(".p2").html("I'm classy");
d3.select("#p3").html("I've got ideas");
```

## Seleccionando todos, SelectAll

```
d3.selectAll("p").style("font-weight","bold");
```

## Seleccionando progresivamente

```
d3.select("p").select("span").style("font-
weight",null);
```

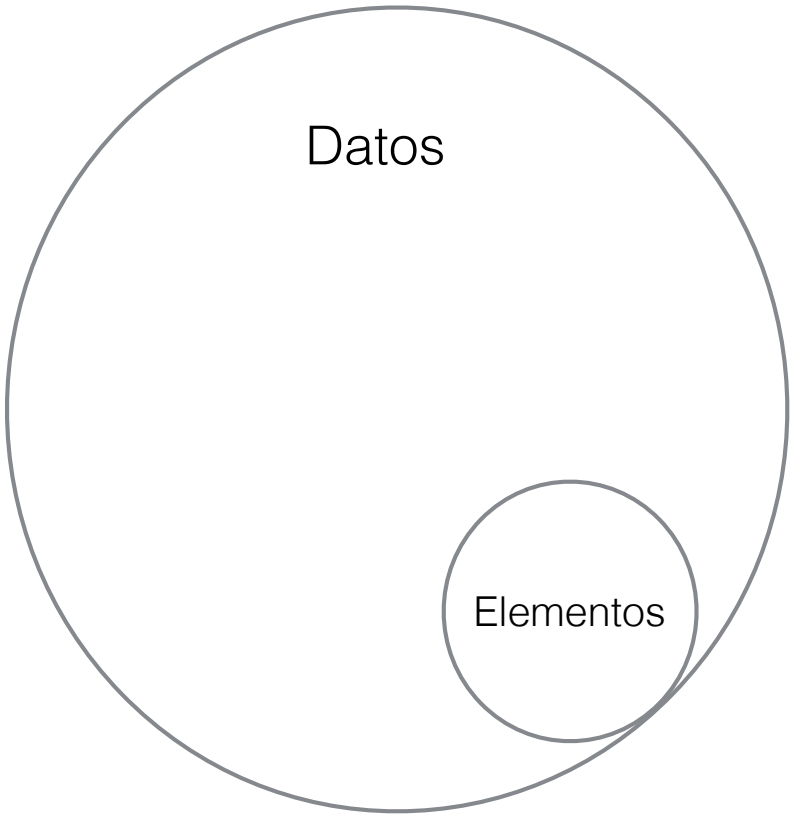
# Datos

```
var fs=["10px","20px","30px"];
```

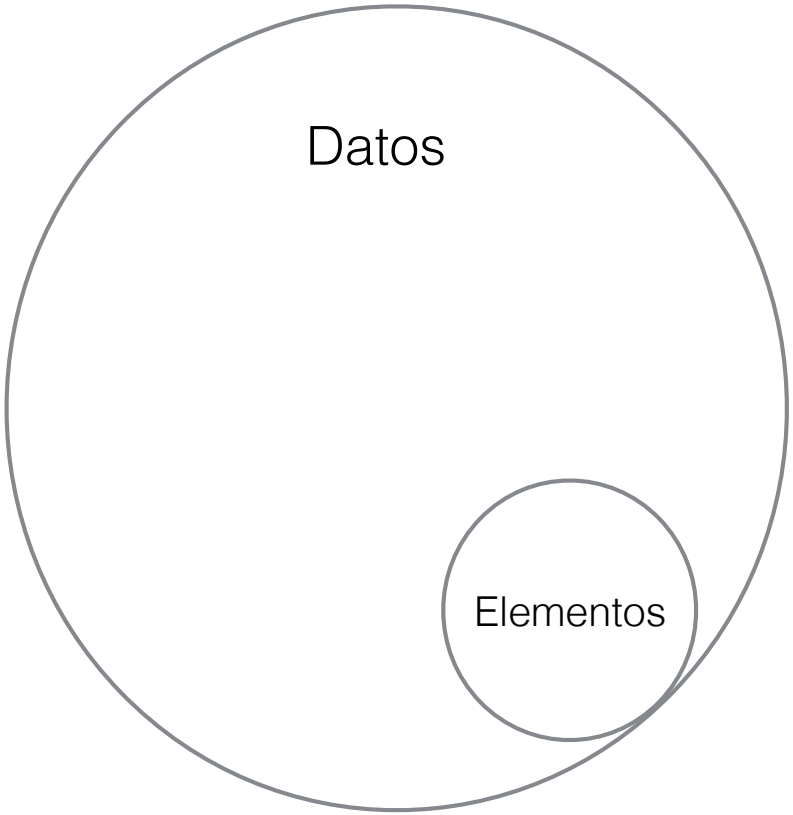
```
d3.selectAll("p").data(fs).style("font-
size",function(d) {return d;})
```

```
d3.selectAll("p").style("font-size",function(d,i)
{return 10*(i+1)+"px";})
```

# ¿Que pasa si no tenemos igual número de datos y de elementos?

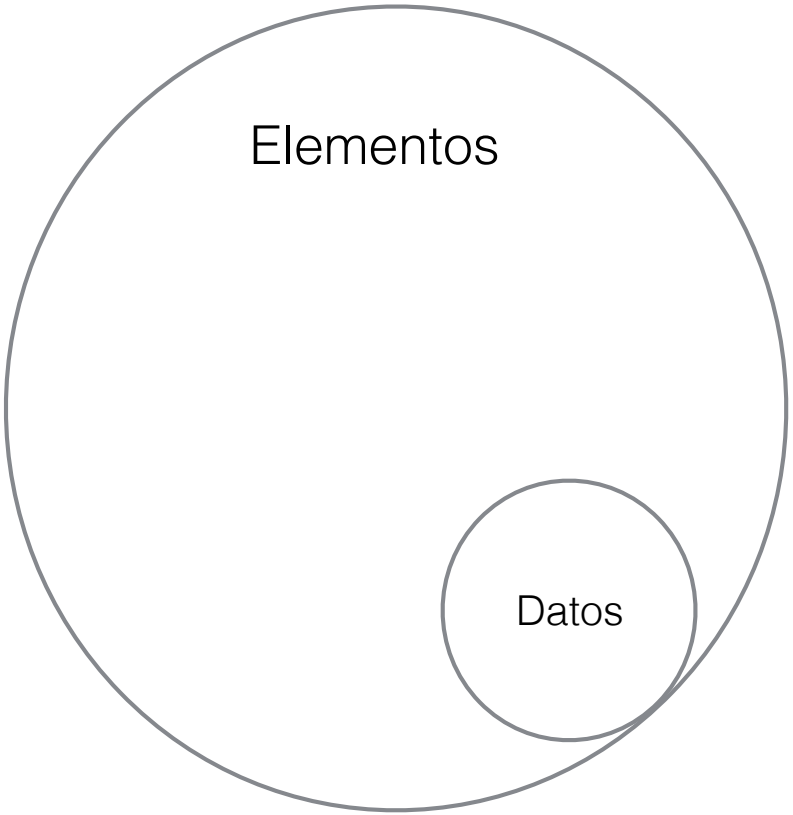



Más datos que elementos



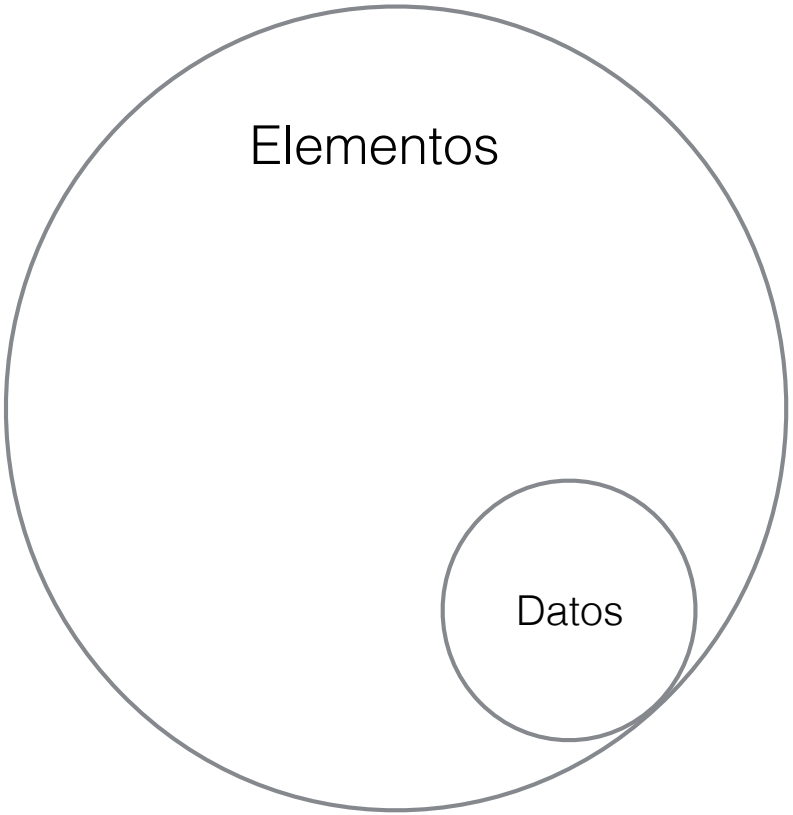


```
d3.select("body").selectAll("*").remove();

var textos = ["texto uno","texto dos","texto tres"];

var ps = d3.select("body")
          .selectAll("p")
          .data(textos)
          .enter()
          .append("p")
          .html(function(d){return d;});
```





```
var texto2 = ["Hola mundo"];

d3.selectAll("p")
    .data(texto2)
    .html(function(d){return d;});
```



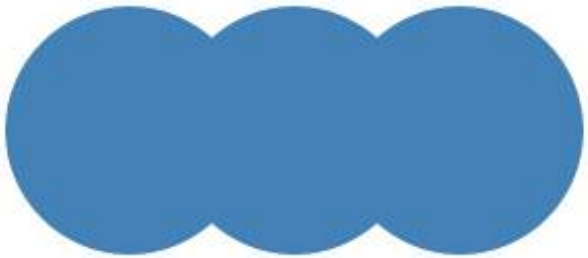


```
d3.selectAll("p")
    .data(texto2)
    .exit()
    .style("color","red");
```

```
d3.selectAll("p")
    .data(texto2)
    .exit()
    .remove();
```

```svg
<svg width="720" height="120">
    <circle cx="40" cy="60" r="10"></circle>
    <circle cx="80" cy="60" r="10"></circle>
    <circle cx="120" cy="60" r="10"></circle>
</svg>
```

```
var circle = d3.selectAll("circle");
circle.style("fill","steelblue");
circle.attr("r",30);
```

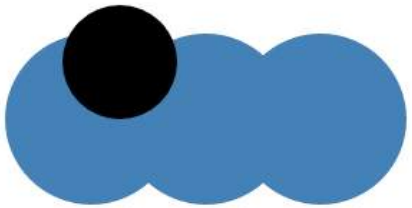


# Agregar elementos al DOM

```
d3.select("body")
        .append("p")
        .attr("class","otro")
        .html("Vamos bien!")
```

```
d3.select("svg")
        .append("circle")
        .attr("cx",50)
        .attr("cy",40)
        .attr("r",20);
```

Vamos bien!



Datos!!!

```
var datos = [5,10,8,30];
```

```
var circleN = d3.selectAll("circle");
```

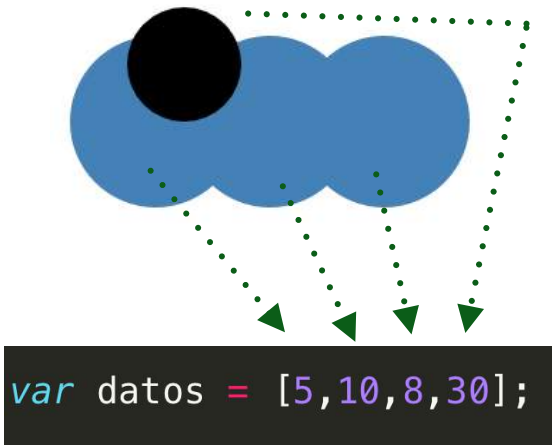
Esto es lo más importante!!!

```
circleN.data(datos);
```

Esto es lo más importante!!!

```
circleN.data(datos);
```

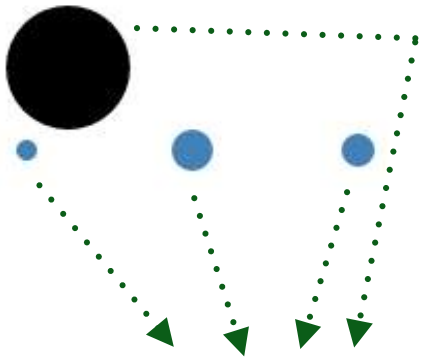
Amarre entre los círculos y los datos



¿Que podemos hacer con los datos amarrados?

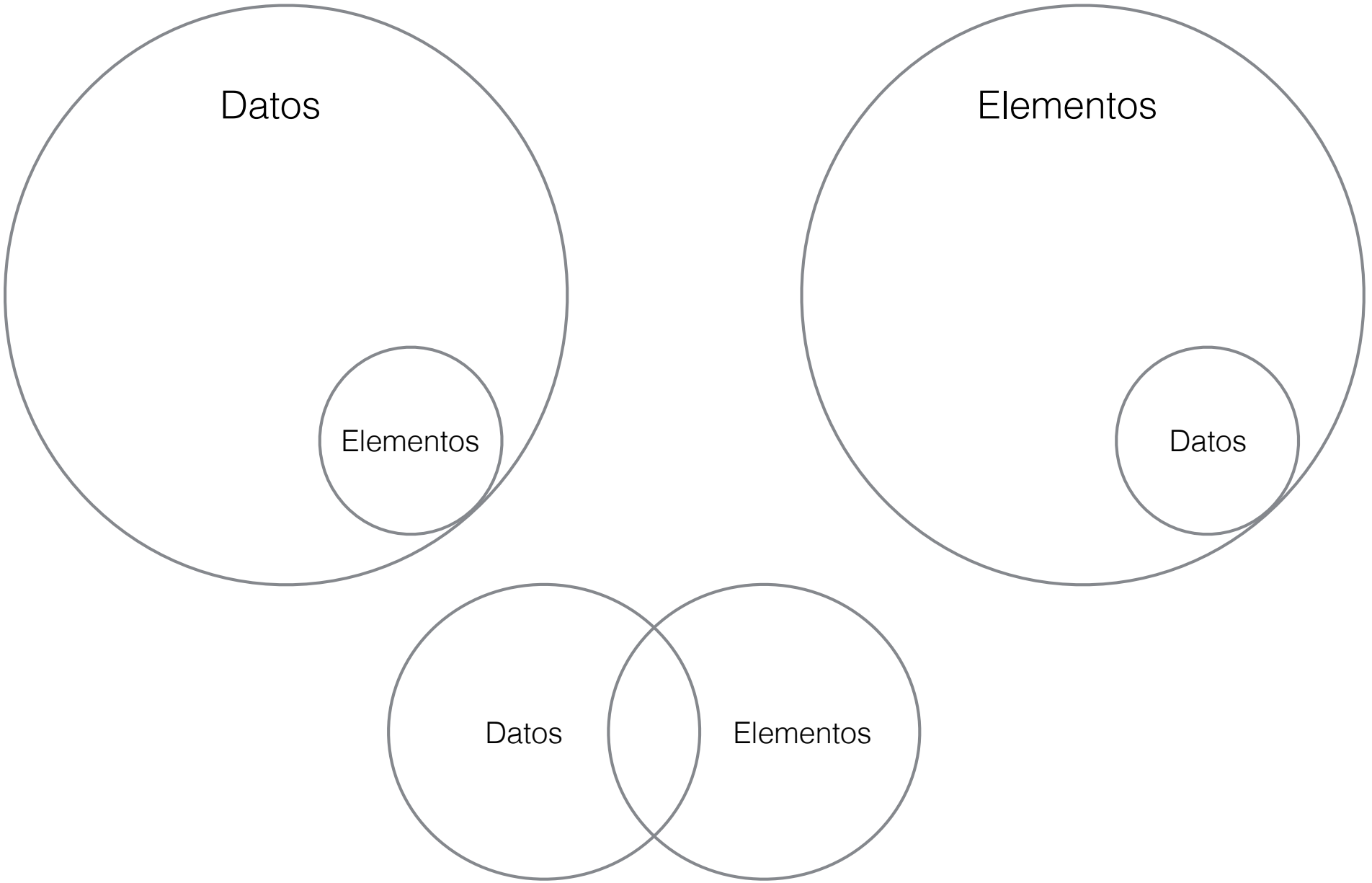
Podemos usarlos para modificar cualquier cosa!

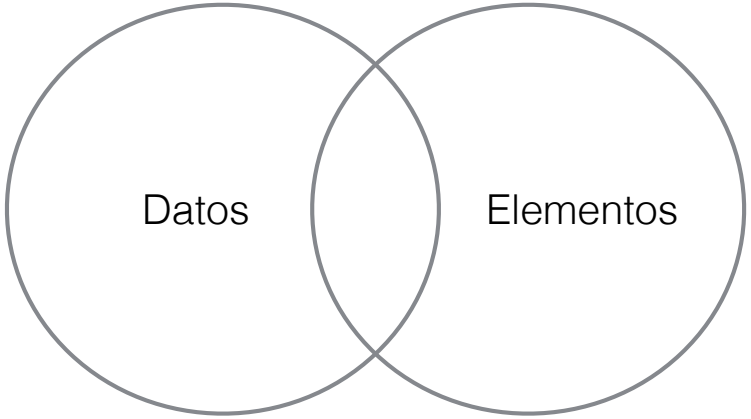
```
circleN.attr("r",function(d,i){return 0.5 * d });
```



```
var datos = [5,10,8,30];
```

# ¿Que pasa si no tenemos igual número de datos y de elementos?








```
d3.select("svg").selectAll("circle")
    .data([10, 11, 12, 13, 14, 15])
  .enter().append("circle")
    .style("fill","red")
    .attr("cy", 70)
    .attr("cx", function(d, i) { return i * 10 + 3; })
    .attr("r", function(d) { return Math.sqrt(d); });
```


```
d3.select("svg").selectAll("circle")
    .data([3,2])
  .exit().remove();
```