

Tutorial D3.js



Data-Driven Documents

08 – Datos desde ficheros Y Líneas



Importar datos desde ficheros

```
d3.csv(url[, row], callback)
```

```
d3.json(url[, callback])
```

```
d3.tsv(url[, row], callback)
```

```
d3.xml(url[, callback])
```

<https://github.com/d3/d3/blob/master/API.md#requests-d3-request>



Líneas y SVG Path

`d3.line().x([x]).y([y])`

`d3.path()`

Shapes:

<https://github.com/d3/d3/blob/master/API.md#shapes-d3-shape>

SVG Paths:

<https://github.com/d3/d3/blob/master/API.md#paths-d3-path>

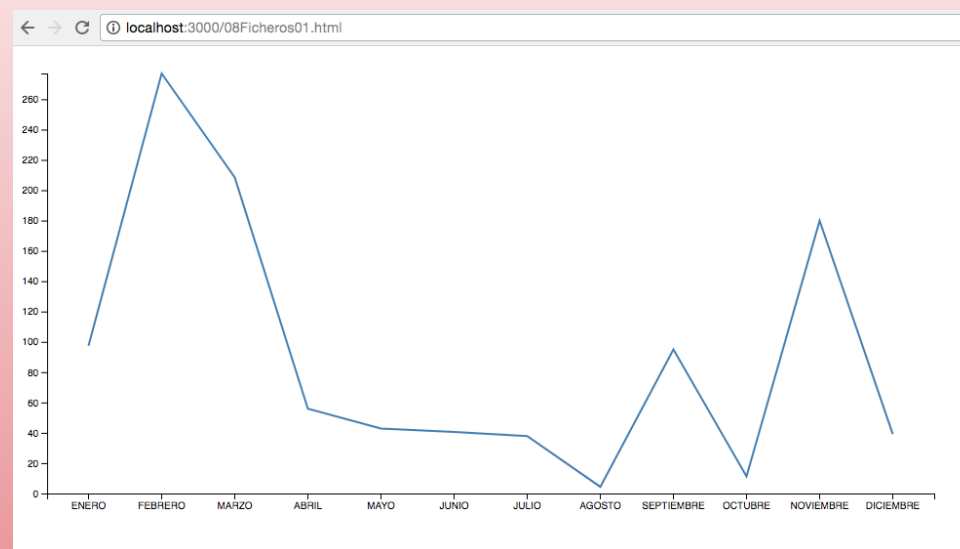
SVG Paths and D3.js:

<https://www.dashingd3js.com/svg-paths-and-d3js>



Datos desde Ficheros y Líneas (I)

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>08 Ficheros 01</title>
6   <style> /* Estilo para la línea */
7     .linea {
8       fill: none;
9       stroke: steelblue;
10      stroke-width: 2px;
11    }
12  </style>
13  <script type="text/javascript" src="http://d3js.org/d3.v4.min.js"></script>
14  <script type="text/javascript">
15
16    // Función para dibujar gráfico de barras
17    function grafico(){
18
19      // Establecer Tamaño y márgenes
20      var ancho_svg = 960,
21          alto_svg = 500,
22          margen = {sup: 20, dch: 20, izq: 30, inf: 50},
23          ancho_graf = ancho_svg - margen.izq - margen.dch,
24          alto_graf = alto_svg - margen.sup - margen.inf;
25
26
27      // Definir Escala para la cantidad de precipitaciones.
28      // De momento solo el rango porque para definir
29      // el dominio necesitamos los datos
30      var escala_y = d3.scaleLinear()
31        .range([alto_graf, 0]);
32
33
34      // Definir Escala para los meses.
35      // De momento solo el rango porque para definir
36      // el dominio necesitamos saber los meses :-))
37      var escala_x = d3.scaleBand()
38        .rangeRound([0, ancho_graf]);
39
40      // Definir la línea y los valores x e y que va a tener
41      var valorLinea = d3.line()
42        // El valor x viene determinado por la posición que ocupa cada mes.
43        // Llamar directamente a la escala nos daría la posición x donde comienza cada mes.
44        // Step nos da el ancho de cada mes, por eso si le sumamos la mitad del ancho,
45        // nos queda en el centro de cada mes.
46        .x(function(d) { return (escala_x(d.Mes) + escala_x.step()/2); })
47        // El valor y lo sacamos de la cantidad de lluvia que viene en la columna
48        // Precipitaciones una vez convertido mediante su escala.
49        .y(function(d) { return escala_y(d.Precipitaciones); });
50
```





Datos desde Ficheros y Líneas (II)

```
51 // Definir y agregar el svg a la página
52 // Agregar un grupo y trasladar el eje de coordenadas
53 var svg = d3.select("body").append("svg")
54   .attr("width", ancho_svg)
55   .attr("height", alto_svg)
56   .append("g")
57   .attr("transform",
58     "translate(" + margen.izq + "," + margen.sup + ")");
59
60 // Obtener los datos de un csv con nombre de columnas Mes y Precipitaciones
61 d3.csv("https://raw.githubusercontent.com/irekalde/youtube/master/D3/lluvia.csv", function(d) {
62   // Le indicamos que Precipitaciones es un numero
63   // Le decimos que pase a mayusculas los meses
64   d.Precipitaciones = +d.Precipitaciones;
65   d.Mes = d.Mes.toUpperCase();
66   return d;
67 }, function(error, datos) {
68   if (error) throw error;
69   // Definir el dominio de la escala de los meses
70   // Lee los valores de la columna mes
71   escala_x.domain(datos.map(function(d) { return d.Mes;}));
72
73   // Definir el dominio de la escala de la lluvia
74   // que va de 0 al valor máximo contenido en la columna precipitaciones.
75   escala_y.domain([0, d3.max(datos, function(d) {return d.Precipitaciones;})]);
76
77   // Definir el eje X de los meses
78   var eje_x = d3.axisBottom()
79     .scale(escala_x);
80
81
82   // Definir el eje Y de las precipitaciones
83   var eje_y = d3.axisLeft()
84     .scale(escala_y);
85
86   // Mostrar en consola para ver las diferencias.
87   console.log(datos);
88   console.log([datos]);
89
90   // Crear el path y añadirle los valores
91   // Añadimos un elemento path al svg
92   svg.append("path")
93   // Le presentamos los datos es necesario incluir corchetes
94   .data([datos])
95   // Le indicamos la clase para que le aplique el estilo definido.
96   .attr("class", "línea")
97   // Aquí le indicamos que va a ser una línea y sus valores.
98   .attr("d", valorlínea);
99
```

```
99
100 // Añadimos el eje x para ello creamos un grupo
101 svg.append("g") // Añadir un grupo para el eje
102   .attr("transform", "translate( 0 , " + alto_graf + " )")
103   // Desplazar el eje de coordenadas
104   .call(eje_x); // Llamar el eje x
105
106 // Añadimos el eje y para ello creamos un grupo
107 svg.append("g") // Añadir un grupo para el eje
108   .attr("transform", "translate( 0 , 0 )")
109   // No es necesarios desplazar el eje de coordenadas
110   .call(eje_y); // Llamar el eje y
111
112 });
113
114
115 </script>
116 </head>
117 <!-- Llamada al cargar body de la función diagrama -->
118 <body onload="grafico()">
119
120 </body>
121 </html>
```