

Índice

1. Capturando Eventos de ratón
2. Añadiendo Controles
3. Usando Widgets
4. Añadiendo Animación
5. Información geoespacial

1. Capturando Eventos de Ratón (I)

- `selection.on(typenames[, listener[, options]])`
 - Documentación: https://github.com/d3/d3-selection/blob/v2.0.0/README.md#selection_on
 - Permite la captura y la gestión (realizar acciones) de eventos (*typenames*), entre ellos el de ratón.
 - Ejemplos típicos de eventos (valores de *typenames*) asociados al ratón:
 - “click”: se lanza el evento al realizar clic sobre la “selección”.
 - “mouseover”: se lanza el evento al pasar el ratón sobre la “selección”.
 - “mouseout”: se lanza el evento al salir de la “selección”.
 - Hay cambios importantes en la V6 con respecto a la V5.
 - Ver: <https://github.com/d3/d3/blob/master/CHANGES.md> y <https://observablehq.com/@d3/d3-selection-2-0#events>
 - Los más importantes:
 - El evento se pasa directamente al “listener”
 - Desaparecen las funciones: `d3.event`, `d3.mouse`, `d3.touch`, Esa información es accedida mediante el evento.
 - Ejemplo de prueba sencillo:

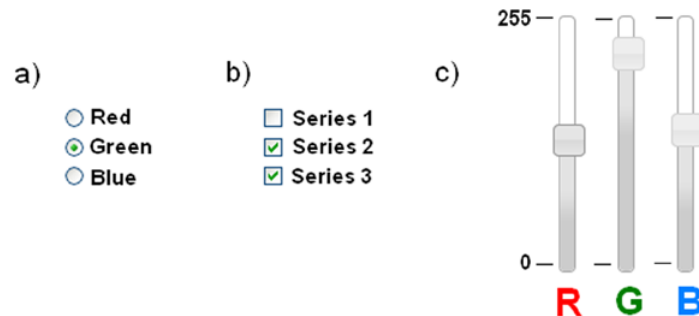
Versiones 4 y 5	Versión 5
<pre> <p>Haz clic en este texto (una vez).</p> <script type="text/javascript"> d3.select("p").on("click", function() { var coords = d3.mouse(this); alert("Clic en X: " + coords[0] + " Y: " + coords[1]); }); </script> </pre>	<pre> <p>Haz clic en este texto (una vez).</p> <script type="text/javascript"> d3.select("p").on("click", function(evento) { //Para ver el objeto asociado al evento console.log(evento); console.log(evento.type); var coords = d3.pointer(evento); alert("Clic en X: " + coords[0] + " Y: " + coords[1]); }); </script> </pre>

1. Capturando Eventos de Ratón (II)

- Más información:
 - En la web de la asignatura:
 - Capítulo correspondiente del libro de Scott Murray:
 - D3_Tema4_Interaccion_MurrayScott_2017_Chapter10Interactivit_InteractiveDataVisual.pdf. Disponible en “[Tema 4. Material adicional - captura eventos ratón](#)”
- Ejemplo de uso:
 - <http://bl.ocks.org/WilliamQLiu/76ae20060e19bf42d774>
 - Está en la versión 3. En la web de la asignatura se ha dejado lo mismo, pero para la versión 6 (fichero “D3_Tema4_Interaccion_Ejemplo_SelectOn_V6.html” de “[Tema 4. Material adicional - captura eventos ratón](#)”)
 - Ahí se puede ver cómo se capturan los tres eventos de ratón típicos: click, mouseover, mouseout.
 - Se puede ver, también, cómo añadir de manera dinámica un elemento y cómo eliminarle mediante su id
 - Otro punto interesante es ver como realizar la operación inversa a “scale”, es decir, como a partir de un valor del rango de salida (pixel), se obtiene el del dominio de entrada.

2. Capturando Eventos de Control

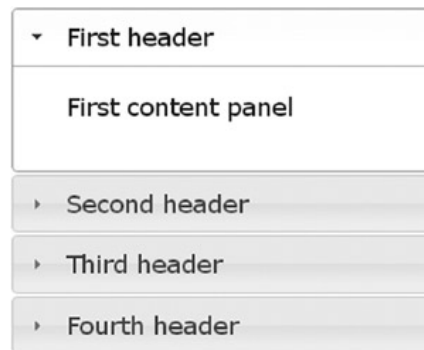
- Otra forma de interactuar sobre el gráfico es mediante la captura de acciones/eventos sobre botones de la pagina html.
 - Considerando como “botón” cualquier elemento html que implique acción del usuario y suponga que el usuario ha elegido (evento tipo “click”) o modificado (evento tipo “change”) alguna opción.
 - Ej. botones, “radio buttons”(a), “check boxes” (b), “sliders” (c), etc



- Esta opción nos permite añadir interactividad mediante controles html estándar
- Ejemplo: En documento [D3 Tema4 LibroCh10 01 Añadiendo Controles.pdf](#) de la página web, extraído del libro de Fabio Nelli “Create Web Charts with D3”
 - Está en la versión 3 de D3 y además usa la biblioteca Javascript JQuery para la captura de eventos. Esto es innecesario, ya que las nuevas versiones de D3 lo permiten.
 - Lo usamos como referencia y para mostrar el mismo ejemplo
 - Los fuentes del ejemplo usando solo D3 (sin JQuery) en las versiones V5 y V6 (hay cosas que cambian mucho en la V6) están en [“Tema 4. Material adicional - captura eventos control”](#)

3. Usando Widgets

- Cuando se tienen varios gráficos, se puede añadir interactividad colocando los gráficos en “acordeón” o en pestañas, de manera que el usuario pueda elegir el gráfico a ver.



Acordeón



Pestañas

- Estas y otras opciones son herramientas de jQuery UI.
- Esto también nos va a permitir crear gráficos que se redimensionen de manera automática ante acciones del usuario y que puedan colocarse dentro de la página web donde el usuario quiera.
- Ejemplos de cómo usar los Widgets están en el documento [D3 Tema4 Gráficos en Widgets.pdf](#) de la página web, extraído del libro de Fabio Nelli “Create Web Charts with D3” (Ed. Apress, 2014).
 - Está desactualizado, pero puede servir de referencia.
 - El Código de los ejemplos está en el fichero (parte “Material Extra”): [Create Web Charts - Código](#)

4. Añadiendo Animación (I)

- No es exactamente interacción, pero añade un componente que puede hacer nuestras visualizaciones más atractivas.
- No es conveniente abusar, ya que puede ralentizar y volver las visualizaciones pesadas.
- La información que aparece aquí se puede ver en el documento [“D3_Tema2_D3Basico_Aptdos3-7.pdf”](#) de la web, páginas 43 y 44, apartado “Transitions”.
 - Es de la versión 3, pero sirve de referencia en cuando al tipo de animación que podemos realizar y qué propiedades se pueden modificar.
 - En este apartado no ha habido grandes modificaciones de la V3 a las siguientes en cuanto a las operaciones a realizar y su sintaxis
- Más información en la web de la asignatura:

[D3_Tema4_Interaccion_MurrayScott_2017_Chapter9UpdatesTransi_InteractiveDataVisual.pdf](#)

4. Añadiendo Animación (II)

- Algunas funciones d3 (I):
 - Vamos a ver la operaciones básicas, que se realizan de la misma manera en todas las versiones
 - `.transition()`: añade animación. Crea una transición suavizada entre un punto origen y un final
 - “punto” puede ser cualquier atributo como una coordenada, un tamaño, un color, etc.
 - interpola todos los atributos modificados entre el punto inicial y final.
 - Si no hay punto final, d3 lo crea.
 - Ej. con punto inicial y final (en pruebasD3_EjemploAnimacionCirculo_VariasVersiones.html, se puede probar como las funciones básicas son las mismas en todas las versiones):


```
var svg = d3.select('body').append('svg').attr('width', 200).attr('height', 200);
svg.append('circle').style('stroke', 'black').style('fill', 'yellow').attr('r', 40).attr('cx', 50).attr('cy', 50)
  .transition().delay(100).duration(4000).attr("r", 10).attr("cx", 150).style("fill", "black");
```
 - Ej. sin punto inicial (del ejemplo de gráfico de barras del tema anterior):


```
svg.selectAll(".bar").data(data).enter().append("rect").transition().attr("class", "bar").
  .attr("x", function(d) { return x(d.country); }).attr("width", x.rangeBand())
  .attr("y", function(d) { return y(d.income/sum); })
  .attr("height", function(d) { return h - y(d.income/sum); }).attr("fill", "grey");
```
 - En [“Tema 4. Material adicional – Animación”](#) de la web se ha dejado otro ejemplo para ver como podemos usar la función `join()` junto con `transition()` (pruebaJoin_Animacion.html/.js)
 - Este ejemplo está basado en el que aparece en <https://observablehq.com/@d3/selection-join>

4. Añadiendo Animación (III)

- Funciones d3 (II):
 - `.transition().duration(ms)`: por defecto la transición dura 250 ms, pero esa duración se puede modificar con esta función. La duración en ms. se pone como argumento.
 - `.transition.ease([valor])`: si se pone una duración alta (la animación se realiza lentamente) se puede observar la dinámica que usa d3 por defecto para la animación: comienza muy lenta, después acelera para luego ralentizarse otra vez hacia el final. La dinámica de la transición se denomina “easing” y se controla mediante esta función.
 - Valor por defecto: “cubic-in-out” (el que se ha descrito).
 - Se puede modificar mediante el argumento “valor” que tiene que ser una función.
 - Funciones ya construidas: linear, circle, elastic, bounce, ...
 - La función que se ponga es invocada en cada “frame”, teniendo en cuenta que el tiempo de animación es normalizado a [0,1].
 - Sin argumento, retorna la función usada
 - `.transition.delay(ms)`: indica un retraso en el comienzo de la animación en ms.
 - Se puede colocar en cualquier lugar (como el resto de funciones de esta transparencia), pero su ubicación lógica es antes de `duration()`. Le da más legibilidad al código

5. Información Geoespacial (I)

- Sin grandes cambios desde la versión 4, salvo mejoras en funcionamiento y actualizaciones en cuanto a formatos de datos.
 - Función básica, que genera los “paths” para “dibujar” un mapa:
 - `d3.geoPath()`
- Se apoya en una versión de la estructura de datos Json especial para información geográfica: [GeoJSON](#)
 - También maneja un formato más moderno y compacto: TopoJson
 - Es una extensión de GeoJson. Usa las propiedades de la topología geoespacial para eliminar redundancias.
 - Ej., una frontera compartida en GeoJson se repite, en TopoJson no.
- Se puede encontrar información en:
 - API D3 [Geographies](#)
 - Capítulo 14. Geomapping, del libro de Scott Murray. Se deja en la web en el documento (apartado “Tema 4. Material adicional - información geográfica”):
[D3_Tema4_Interaccion_MurrayScott_2017_Chapter14Geomapping_InteractiveDataVisual.pdf](#)
- Existen bibliotecas de libre acceso.
 - Es más flexible usar directamente las funciones de D3

5. Información Geoespacial (II)

- SVG: Fundamentos.
 - D3 para visualizar información geográfica se apoya en los “caminos” de SVG; etiqueta “*path*”.
 - Elemento más potente de SVG, ya que permite dibujar cualquier cosa.
 - Atributo más importante: “*d*”
 - Propiedades: M→moveto, L→lineto, H→horizontal lineto, V→vertical lineto, C→curveto, S→smooth curveto, Q→quadratic Bézier curve, T→smooth quadratic Bézier curveto, A→elliptical Arc, Z→closepath
 - Ejemplo simple
 - de https://www.w3schools.com/graphics/svg_path.asp, permite probar y editar los ejemplos

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<svg height="210" width="400">
```

```
<path d="M150 0 L75 200 L225 200 Z"
```

```
style="fill:none;stroke:black"/>
```

```
</svg>
```

```
</body>
```

```
</html>
```



- Probar el ejemplo copiando y pegando el código o mediante opción “try it yourself” del enlace.

5. Información Geoespacial (II)

- Entendiendo cómo funciona `d3.geopath()`
 - Es un generador de caminos (paths) a partir de un fichero con información geográfica (longitud-latitud) en formato GeoJson, aplicando una determinada proyección.
 - Proyección:
 - La información geográfica está en longitud y latitud \Rightarrow 3D (ente otras cosas)
 - Los mapas (visualización) están en 2D \Rightarrow tendremos que aplicar una transformación en la representación original de los datos: proyección
 - Hay múltiples alternativas
 - En d3 se realiza mediante la función `projection()` (https://github.com/d3/d3-geo/blob/v1.11.6/README.md#_projection)
 - El estudio en profundidad excede el ámbito de la asignatura. Más información en:
 - https://d3-wiki.readthedocs.io/zh_CN/master/Geo-Projections/
 - <https://github.com/d3/d3-geo-projection/>
 - Aquí vamos a aplicar la denominada `geoMercator` (<https://github.com/d3/d3-geo/blob/v1.11.6/README.md#cylindrical-projections>)

5. Información Geoespacial (III)

- Entendiendo cómo funciona `d3.geopath()`. Ejemplo simple
 - Vamos a dibujar un triángulo como el del ejemplo de SVG, pero ahora usando las herramientas de `d3` para generar el camino de manera automática.
 - Descargar el fichero [ejHtmlTema4_Mapas_EjemploSimpleGeoPath.html](#). Ahí se muestra:
 - Un ejemplo sencillo de objeto GeoJson (se puede ver como funciona desde la V4 a la V6)
 - Un ejemplo sencillo de visualización de un camino que muestra un triángulo.
 - Analizar y entender su contenido
 - Ejecutarle en el navegador y ver tanto la salida de consola, como el html generado, para ver como la función `geoPath()` ha creado de manera automática el código SVG correspondiente al camino de las coordenadas geográficas.
 - Descargar el fichero [ejHtmlTema4_Mapas_EjemploSimpleGeoPath2.html](#). Ahí se puede ver:
 - Una modificación del ejemplo anterior que muestra como cambiar propiedades del mapa.
 - Un ejemplo sencillo de como calcular y añadir de manera dinámica elementos adicionales al mapa.

5. Información Geoespacial (III)

- Ejemplo Final: dibujando el mapa de Castilla y León (I)
 - Hay muchas fuentes de mapas en formato GeoJson. Por ej. :
 - <https://geojson-maps.ash.ms/>
 - https://github.com/codeforamerica/click_that_hood/blob/master/public/data/spain-provinces.geojson
 - <https://data.opendatasoft.com/>.
 - Aquí se puede obtener una gran variedad de información geográfica. Para obtener mapas:
 - » data → Ajustar lo que queremos mediante “Filters” (ej., España) → Seleccionar lo que se quiera de los conjuntos que aparezcan. Una vez en el conjunto seleccionado, para obtener mapas: “export” → geographic file formats → GeoJson (“Whole”, descarga todo, “Only the nn selected records”, lo seleccionado, si se ha hecho)
 - El mapa de las provincias de Castilla y León que usamos en el ejemplo se ha obtenido del último repositorio

5. Información Geoespacial (IV)

- Ejemplo Final: dibujando el mapa de Castilla y León (II)
 - Descargarse el fichero “[ejHtmlTema4_MapasCyL_d3.zip](#)” (está en [Tema 4. Material adicional - información geográfica](#)).
 - Contiene el código html, el fichero con el objeto GeoJson y el fichero con el código javascript, comentado, para distintas opciones de visualización e interacción con el mapa:
 - *Básico*: se dibuja el fichero original. Problema: tamaño muy pequeño
 - *Aumentado y centrado*: se muestra como calcular de manera dinámica propiedades del mapa para centrarlo y ajustar su tamaño
 - *Accediendo a cada provincia*: se muestra como introducir interacción
 - *Dando color*: se deja como ejercicio. Dar a cada provincia un color diferente
 - Comentarios:
 - El mapa es de tamaño variable. Se ajusta al tamaño de la ventana al ser cargado
 - Cada reescalado del mapa cambia la proyección: hay que recalcular valores de posición
 - Se ha hecho para versiones 4, 5 y 6 de D3. La diferencia está en la parte de leer los datos (función `d3.json()`) de V4 a V5, y en la parte de captura de eventos de V5 a V6
 - Analizar con detenimiento, sobre todo la parte de carga de datos, ya que ahora sí el tiempo de carga es un problema y se aborda de distinta manera entre versiones 4 y 5/6.
 - El resto del código no varía.