

Cápsula 3: *Join* de datos en D3.js I

Hola, bienvenidxs a una cápsula del curso Visualización de Información. En esta hablaré sobre el famoso *join* de datos de D3, mecanismo que nos permitirá unir elementos HTML con datos.

Comenzaré por explicar la intención a lograr, que es vincular elementos con datos. ¿Por qué? Porque así, se vincula una marca gráfica, como un rectángulo, con un dato, digamos un número. Así el rectángulo tendrá acceso a dicho número y podría alterar uno de sus canales, como el tamaño, **en base ese número**. ¡Podremos vincular marcas con datos y alterar sus canales a partir de datos!

De ahora en adelante utilizaremos elementos SVG para los ejemplos, ya que generalmente eso es lo que usaremos. También, por el momento usaremos un arreglo de números como los datos a cargar, pero eventualmente **podremos utilizar un arreglo de objetos cualquiera de JavaScript**.

Queremos vincular a un grupo de rectángulos dentro de un elemento "svg" con números. Primero debemos seleccionar los rectángulos que posiblemente ya existen en el elemento "svg".

Por lo que seleccionaremos primero el elemento "svg" contenedor, y luego sobre esa selección seleccionamos todos los elementos "rect" dentro. Como revisado, esto nos retorna una selección con un grupo cuyo padre es el elemento "svg", padre de los rectángulos, y los elementos son cualquier rectángulo dentro de este.

Paso siguiente, podemos llamar al método "data" de selecciones. **Este es el punto en que D3 genera un cruce**, o *join*, entre la selección sobre la cual actuó y los datos que se le entregan a la función.

De esto hay tres casos posibles cuando intenta generar vínculos entre elementos y datos: hay datos que no se le asocian elementos; efectivamente se asocian elementos con datos; o hay elementos que no se le asocian datos.

Por esas posibilidades, el método "data" crea tres objetos selección separados, cada uno capturando los casos mencionados: **enter**, **update** y **exit**. Por defecto, la selección que retorna "data" es *update*, pero llamando a los métodos "enter" y "exit" sobre ella, se retornan las selecciones respectivas.

Los nombres de estas selecciones tienen intención. *Enter* hace alusión a datos que aún no tienen elementos en el documento, y que se le debería agregar o **entrar**. *Exit* hace alusión a elementos que no tienen datos asociados, y que se deberían remover o **salir**. Finalmente

update hace alusión a elementos que se le vincula datos, y posiblemente se **actualizarán** sus propiedades para ser fiel a esos datos.

Por simplicidad al ejemplo, supongamos que todo cae dentro del caso *update*, es decir, contamos con solo elementos vinculados a datos exitosamente. Cómo son cuatro rectángulos que existían dentro de "svg" y se llama a "data" con cuatro números distintos. Cada rectángulo se vincula con un número distinto, el primer rectángulo con el primer dato, el segundo con el segundo, y así...

El método "data" retorna update por defecto. En esta selección encontramos elementos que tienen datos ya asociados. Podemos alterar sus atributos fijando un ancho idéntico en todos los rectángulos, todos una posición vertical común 0 y cambiar su posición horizontal dependiendo de su índice en la selección.

Ahora, cómo estos elementos tienen un dato asociado, podemos usar ese dato para cambiar sus atributos. **El primer argumento de las funciones de alteración hace alusión a dicho dato.** Podemos entonces fijar el alto ("height") de los rectángulos, como que sea el doble del valor de su dato asociado. Esto produciría los rectángulos en pantalla, separados de forma uniforme pero con altura que depende del dato asociado.

Ahora, revisaremos otra de las selecciones del *join*: *exit*. Esto se produce cuando hay elementos sin datos asociados. Esto puede darse cuando no hay suficientes datos en el arreglo comparado con la cantidad de elementos en la selección, por lo que quedan elementos sobrando.

Al extraer esa selección con el método "exit", se obtienen estos elementos sobrantes. Y, mediante el método "remove" es posible removerlos del documento actual.

Finalmente, revisaremos la tercera selección: *enter*. Contrario a *exit*, se produce cuando sobran datos y faltan elementos que asociarlos. Esto suele ocurrir al comienzo de un programa en D3, que comienza sin elementos creados aún. Al llamar a "enter", **retorna una selección única y especial, que contiene un elemento fantasma por cada dato sin pareja.**

Se entregan esos elementos especiales, **para llamar al método "append" inmediatamente**, de forma que cree y agregue elementos que reemplacen estos fantasmas y se asocian a los datos con falta de pareja. El padre de estos elementos es el elemento del grupo de la selección donde actuó.

Tras eso, se cuenta con una selección con elementos vinculados a datos, similar a la que nos entregaba *update* inmediatamente, por lo que es posible alterar atributos de los elementos en base a datos.

Con eso podemos al fin entender el primer código presentado en la introducción a D3. Se crea un elemento "svg" que contendrá todo. Se definen los datos a agregar. Se definen las dimensiones de "svg" en base a los datos. Y finalmente se realiza un *join* de datos que

mediante "enter" y "append" se agregan rectángulos cuya altura depende de los números entregados.

Con eso termina el contenido de esta cápsula. En la siguiente mostraré otros detalles a considerar con el *join* de datos. Recuerda que si tienes preguntas, puedes dejarlas en los comentarios del video para responderlas en la sesión en vivo de esta temática. ¡Chao!