

Cápsula 2: Selecciones en D3.js II

Hola, bienvenidxs a una cápsula del curso Visualización de Información. En esta hablaré más en detalle de selecciones en D3.js.

La idea general de selecciones es simple, es agrupar elementos y dar una forma de alterarlos usando ideas de HTML y CSS. Selecciones también es la pieza de D3 que permite introducir datos al documento, de forma que podamos representarlos con elementos HTML o SVG. Para eso, internamente las selecciones tienen mecanismos que permiten hacerlo.

En esta cápsula explicaré más los mecanismos internos de selecciones, para introducir en la próxima cápsula el uso de datos con ellas. **Es posible que no entiendas inmediatamente lo expuesto en esta y las próximas cápsulas, por lo que te recomiendo ir revisando con ejemplos de código para ver distintas situaciones.** Puedes usar la base los ejemplos usados en estas cápsulas.

La primera aclaración que hay que hacer sobre selecciones, es que no son una simple colección de elementos, si no que en realidad, **son una colección de grupos de elementos.** Podrías pensarlo como un arreglo, de arreglos de elementos. No es un contenedor que directamente tiene elementos almacenados, sino que almacena uno o más colecciones llamadas grupos, y estos grupos contienen elementos.

Los ejemplos que revisamos creando selecciones mediante “d3.selectAll” y “d3.select” tienen la similitud de que siempre crean una selección de **un solo grupo**, y los resultados de la selección se contienen en ese único grupo.

Cada grupo se define por un elemento padre que contiene a los elementos dentro del grupo. En el caso de “d3.select” y “d3.selectAll”, el grupo siempre es definido por el elemento “html”, que representa el documento completo.

Ahora, sucede que es posible crear selecciones de múltiples grupos, y cuyos elementos de grupo no son “html”. Para eso, se puede llamar a “selectAll”, pero sobre un objeto selección ya existente (es decir, no directamente “d3.select” o “d3.selectAll”).

Entonces, consideremos comenzamos con una selección que busca todas las listas “ul” de un documento. Como se llama con “d3.selectAll”, sabemos que los elementos encontrados pertenecen todos al mismo grupo del documento completo.

Si se llama a “selectAll” **sobre esa selección**, para buscar ítems de listas “li”, sucederá que para cada elemento (que en este caso son “ul”) se buscan **dentro** de sí elementos “li” para generar una nueva selección.

Ahora, “selectAll” tiene la particularidad de que al ser llamado **sobre una selección**, utiliza cada elemento previo de la selección y crea un grupo por cada uno, y agrupa los elementos encontrados en ellos. Entonces eso genera una selección de múltiples grupos (uno por cada elemento “ul” previamente seleccionado), donde cada grupo puede tener múltiples elementos “li”.

Por otro lado, llamar a “select” sobre una selección produce un resultado distinto. **No crea grupos nuevos desde los elementos de la selección anterior**, si no que mantiene los mismos grupos pero los elementos son reemplazados con el único resultado que retorna.

Cuando una selección anidada no encuentra nada, por lo general deja grupos sin elementos o espacios vacíos dentro de grupos. Estos **no** son problemas al usar métodos de modificación como “attr” o “style”, ya que estos omiten dichos vacíos.

Entonces estas dos llamadas se diferencian en la cantidad de grupos que generan y cuantos elementos se seleccionan por grupo. Si se imprimen mediante “console.log”, puedes ver en la consola de un navegador que tienen atributos que describen la estructura explicada.

Al usar “select” hay un solo grupo y un padre asociado (“html”), pero al usar “selectAll” hay múltiples grupos, uno por cada “ul” inicialmente seleccionado. Intenta probar con selecciones e imprimiéndolas para ver su estructura.

Otro detalle a considerar, es qué métodos que agregan nuevos elementos sobre una selección se comportan como “select”. Es decir, mantienen el grupo de la selección, pero los elementos son reemplazados por los nuevos elementos que se agregan. Puede parecer raro en primera instancia, pero la gracia de preservar grupos es que se preservan también otras cosas, como datos asociados.

Eso es lo que introduciremos en la próxima cápsula, como asociar datos a elementos mediante selecciones.

Con eso termina el contenido de esta cápsula. Recuerda que si tienes preguntas, puedes dejarlas en los comentarios del video para responderlas en la sesión en vivo de esta temática. ¡Chao!