Funciones de conversión en Javascript, métodos addEventListener y querySelector

Conversión de tipo de dato entero: función Number()

```
>> Number("123")
<- 123
>>  Number("abc12")
<- NaN
>> Number(true)
<- 1
>> Number("21.03")
<- 21.03
```

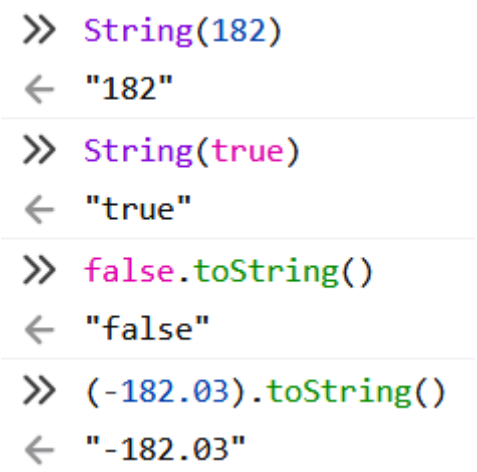
Cuando esta función tiene como argumento un string con caracteres numéricos (o flotantes) retorna un entero con los mismos dígitos que el string, tal como en el primer ejemplo. En cambio si ese string contiene caracteres no numéricos, retornará NaN (Not a Number). Si tiene como argumento como booleano, para true retornará 1 y para false 0.

Veamos unos casos de conversión implícita en javascript, es necesario entender esta lógica para evitar cometer errores.

```
>> "2" + "3"
<- "23"
>> "9"*"2"
<- 18
>> "4"/"2"
<- 2
>> "1"-"1"
<- 0
>> "17.2"-"0.2"
<- 17
```

A excepción de la suma, las demás operaciones aritméticas, los operandos que sean strings con caracteres numéricos (o flotantes) se trabajaran como si fuesen de un tipo de dato numérico.

Conversión a String

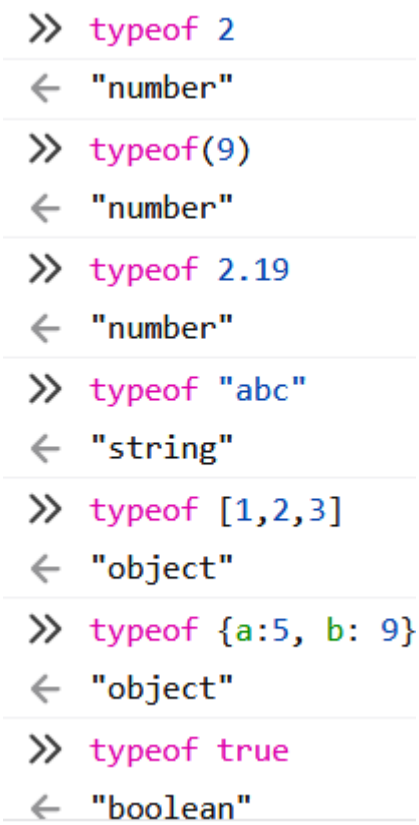


La función String() se encarga de retornar el valor de su argumento como un string. Los booleanos, enteros y flotantes tienen un método llamado toString() que retorna el mismo valor como un string.

Operador typeof

Este operador nos sirve para poder identificar el tipo de dato de una variable o un valor. Retorna un string distinto según cualquiera de los siguientes casos:

- Para los enteros y flotantes retornará "number"
- Para los strings retornará "string"
- Para los booleanos retornará "boolean"
- Para los arrays y objetos retornará "object"
- Si se coloca como operando un nombre que no coincida con el de ninguna variable o función creada retornará "undefined"



Método addEventListener()

Sabemos que podemos mandar a invocar funciones escritas en código javascript para que sucedan cosas en nuestras páginas, utilizando como un atributo más de nuestras etiquetas html el conocido "onclick" tal como vimos en el vídeo.

Pero resulta ser que no es necesario invocar funciones de javascript en nuestro código html.

Imaginemos que tenemos una etiqueta cualquiera con id="x" en nuestro código html y queremos que se invoque una función llamada funcionX cuando se haga click sobre el elemento que muestre tal etiqueta en la página.

Una instrucción válida en nuestro código javascript para lograr tal objetivo sería la siguiente:

```
document.getElementById("x").onclick=function(){ funcionX(); }
```

Luego del signo igual, la invocación de la función debe ir dentro del bloque de código de una función anónima sin argumentos como se muestra en la imagen. Si en el código html ya tenemos definida la propiedad onclick, luego de hacer esto explicado se sobrescribirá el contenido de dicha propiedad.

Otra forma de hacer esto mismo de definir las propiedades “onclick”, “onmouseover”, “onmouseout”, etc dentro del código javascript sería usando el método del objeto document llamado addEventListener().

A continuación un ejemplo de cómo podríamos utilizar este método en el caso expuesto anteriormente.

```
const elemento = document.getElementById("x");

elemento.addEventListener("click", funcionX);
```

Lo de dejar en una constante la referencia de la etiqueta que se quiere añadir el evento es opcional.

El primer argumento debe ser un string donde se indique el evento a añadirse a la etiqueta. Al usar este método, se debe obviar el prefijo “on” en los eventos antes mencionados. De segundo argumento va el nombre de la función a invocarse o también se puede definir ahí mismo.

Método querySelector()

Resulta ser que el objeto document tiene un método llamado querySelector(), retorna la referencia de un elemento html, buscandolo por ya sea su id, clase o tipo de etiqueta.

Para buscar mediante su id , el argumento debería ser entre comillas, un # y seguido el id del elemento a buscar.

Ejemplo, acceder a un elemento que tiene id=”x”:

```
document.querySelector("#x")
```

Para buscar mediante su clase , el argumento debería ser entre comillas, un . (punto) y seguido la clase del elemento a buscar.

Ejemplo, acceder a un elemento que tiene class=”x”:

```
document.querySelector(".x")
```

Para buscar mediante su nombre de etiqueta, el argumento debería ser entre comillas el nombre de la etiqueta.

Ejemplo, acceder a un elemento que tiene la etiqueta “img”:

```
document.querySelector("img")
```

ANEXADO A ESTA PÍLDORA HAY CÓDIGOS FUENTES DONDE SE EXPONEN MEDIANTE EJEMPLOS EL FUNCIONAMIENTO DE addEventListener Y querySelector