

Bootcamp Front End, ReactJs:

Lectura 4

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Febrero-2022

Comparaciones booleanas

Existen un número de comparaciones booleanas dentro de JavaScript que se usan para comprobar la veracidad de una proposición.

En JS existen 8 operadores comparativos. Por ejemplo `==` significa valor igual, `===` quiere decir igual valor e igual tipo.

Así mismo los valor de desigualdad `!=` y `!==` que expresan la desigualdad entre valor y valor y tipo de las proposiciones respectivamente.

Por su parte, las comparaciones de mayor o menor, que son típicas en los lenguajes de programación. `>`, `<`, `>=`, `<=`.

Los operador lógicos para conjunción de proposiciones son: `&&` (and), `|||` (or), (not).

Condicionales

La sintaxis básica para el condicional **if** es:

```
if (condition){  
    // código  
}
```

Para el caso de tener más de una condición en un mismo flujo, la sintaxis para 2 condiciones será:

```
if (condition){  
    // código  
} else {  
    // código  
}
```

Else if

En otra situación, pueden existir más de 2 posibles situaciones para la condicional, por lo que se usa:

```
if (condition){  
    // código  
} else if (condition 2) {  
    // código  
} else  
    // código  
}
```

Switch

Si existen varios casos de análisis, JS presenta también una posibilidad de escribir estos casos con **switch**

```
switch (expression){  
  case x:  
    código para este caso  
    break;  
  case y:  
    código para este caso  
    break;  
  case z:  
    código para este caso  
    break;  
  default:  
    código para este caso  
}
```

Arrays

Un array es una colección de elementos en JS. Contiene múltiples elementos unidos por comas. Su sintaxis básica:

```
const arrayname = [item1, item2, ...]
```

A diferencia de otros lenguajes de programación, los items pueden ser de distinto valor.

En el caso del DOM, los métodos del API de JS para recibir arrays de distinto valor son:

```
grupo = document.getElementsByTagName(name)
```

```
grupo = document.getElementsByClassName(name)
```

Arrays

Para acceder a un valor del array tenemos su indexación, la sintaxis para cada elemento es:

```
objeto1 = grupo[x];
```

Donde x es el índice del elemento, que comenzará por 0.

Existen algunos métodos definidos para arrays, particularmente buenos: **length** es una propiedad para saber la cantidad de elementos, **sort()** para ordenar los elementos, **push(element)** para añadir elementos

For Loops

Normalmente queremos repetir lo mismo para varios elementos dentro de un array o hasta que se cumpla un valor determinado, por lo que es útil conocer acerca de loops.

For es un bucle que permite la repetición de algunos métodos para 2 o más elementos, de acuerdo con límite. La sintaxis de for será:

```
for (let i = 0; i < 5; i++) {  
  comandos  
}
```