

M4.250 - Introducción a la programación en JavaScript I aula 1

PEC 1

Alumno: Ivan Schartun

1.- Analizad el siguiente programa en JavaScript y calculad qué valores acabarán teniendo las variables. Debéis considerar las ocho líneas como un único programa: cada expresión puede contener variables cuyos valores se han calculado en líneas anteriores. Explicad el porqué de cada resultado:

```
let x01=0;
let x02=(x01=="0");
let x03=(x01===0);
let x04=(x01="0");
let x05=(x04=="0");
let x06=(x04===0);
let x07=(x04===0);
let x08=(x04==0);
```

Respuesta

```
let x01=0;
>>let x01 = 0 // El valor 0 es asignado a la variable let x01
```

```
let x02=(x01=="0");
>>let x02 = true // Siendo x01 = 0. JavaScript devuelve true al ser considerados ambos operandos iguales. En este caso tenemos un operador de igualdad débil. Este tipo de operador compara la igualdad de dos valores después de convertir ambos valores a un tipo de datos común. Recordemos que JS no tiene una fuerte verificación de tipos.
```

```
let x03=(x01===0);
>>let x03 = true // Siendo x01 = 0, y al estar en presencia de un operador de comparación estricto JS devuelve true si los operandos son iguales y del mismo tipo. En ambos casos los tipos son iguales.
```

```
let x04=(x01="0");
>>let x04 = "0" // Siendo que se asigna el string "0" a la variable x01, al realizar nuevamente la asignación a x04 estaremos apuntando al mismo valor de string "0"
```

```
let x05=(x04=="0");
>>let x05 = true // Siendo que se realiza una comparación entre el valor de x04="0" (valor asignado en la línea anterior) y luego una comparación con un string igual "0" obtendremos como resultado el valor "true" que a su vez será asignado a la variable x05
```

```
let x06=(x04==="0");  
>>let x06 = true // De líneas anteriores témenos que la variable x04="0", si realizamos  
una comparación estricta de tipos, al tener ambos lados un string "0"  
tendremos como resultado un valor "true"
```

```
let x07=(x04===0);  
>>let x07 = false // De líneas anteriores témenos que la variable x04="0", si realizamos  
una comparación estricta de tipos, al tener ambos lados tipos  
diferentes tendremos como resultado un valor "false"
```

```
let x08=(x04==0);  
>>let x08 = true // De líneas anteriores témenos que la variable x04="0", si realizamos  
la una comparación simple de tipos, JS por ser un lenguaje de baja  
verificación de tipos nos dará como resultado "true"
```