

Módulo Javascript

Tipos de datos, variables y constantes

Objetivos: Conocer los distintos tipos de datos y el uso de variables/constantes

Tipos de Datos

JavaScript es un lenguaje de programación dinámico y flexible, lo que significa que no es necesario definir los tipos de datos al declarar una variable.

A continuación se describen los tipos de datos básicos en JavaScript y algunos ejemplos de uso:

Números

Los números en JavaScript se representan mediante el tipo de dato "**number**". Los números pueden ser enteros o decimales. Algunos ejemplos de uso son:

```
let numeroEntero = 10;  
let numeroDecimal = 3.14;  
let resultado = numeroEntero + numeroDecimal; // 13.14
```

Cadena de Texto (String)

Las cadenas de texto en JavaScript se representan mediante el tipo de dato "**string**". Las cadenas de texto pueden ser declaradas utilizando comillas simples o dobles. Algunos ejemplos de uso son

```
let nombre = "Juan";  
let apellido = 'Pérez';  
let saludo = "Hola " + nombre + " " + apellido; // "Hola Juan Pérez"
```

Booleanos

Los **booleanos** en JavaScript se representan mediante el tipo de dato "boolean". Los booleanos pueden ser "true" (verdadero) o "false" (falso). Algunos ejemplos de uso son:

```
let esMayorDeEdad = true;  
let tieneLicencia = false;  
let puedeConducir = esMayorDeEdad && tieneLicencia; // false
```

Undefined

El valor **undefined** se utiliza para representar una variable que no ha sido definida o que no tiene valor asignado. Algunos ejemplos de uso son:

```
let variableSinValor; console.log(variableSinValor); // undefined
```

Null

El valor **null** se utiliza para representar la ausencia de un valor. Algunos ejemplos de uso son:

```
let objeto = null;  
console.log(objeto); // null
```

Además de los tipos de datos básicos que ya hemos mencionado, JavaScript también tiene otros tipos de datos más complejos. A continuación, se describen algunos de estos tipos de datos y su uso en JavaScript:

Arreglos (array)

Los arreglos en JavaScript se utilizan para almacenar colecciones de datos del mismo tipo o de diferentes tipos. Los arreglos se representan mediante corchetes []. Algunos ejemplos de uso son:

```
let numeros = [1, 2, 3, 4, 5];  
let colores = ["rojo", "verde", "azul"];  
let datos = [1, "Juan", true];  
console.log(numeros[2]); // 3
```

Objetos

Los objetos en JavaScript se utilizan para almacenar colecciones de datos y funciones. Los objetos se representan mediante llaves {}. Algunos ejemplos de uso son:

```
let persona = {  
  nombre: "Juan",  
  apellido: "Pérez",  
  edad: 30,  
  saludar: function() {  
    console.log("Hola, mi nombre es " + this.nombre + " " + this.apellido);  
  }  
};  
  
console.log(persona.nombre); // "Juan"  
persona.saludar(); // "Hola, mi nombre es Juan Pérez"
```

Funciones

Las funciones en JavaScript se utilizan para encapsular una pieza de código que se puede reutilizar y llamar en diferentes partes de un programa. Las funciones se pueden declarar utilizando la palabra clave "function". Algunos ejemplos de uso son:

```
function sumar(numero1, numero2) {  
  return numero1 + numero2;  
}  
  
let resultado = sumar(3, 5); // 8
```

Variables y Constantes

En JavaScript, las variables y constantes se utilizan para almacenar y hacer referencia a valores. Las variables se pueden actualizar con nuevos valores a lo largo del tiempo, mientras que las constantes tienen un valor inmutable. A continuación se describen las diferencias entre las palabras clave `let/const` y `var`:

LET

La palabra clave **let** se utiliza para declarar variables que pueden cambiar su valor en el tiempo. Las variables declaradas con `let` pueden ser reasignadas y solo están disponibles dentro del bloque de código en el que se declaran. Por ejemplo:

```
let mensaje = "Hola";
mensaje = "Hola mundo";
console.log(mensaje); // "Hola mundo"

if(true) {
    let variableDentroDelIf = 2;
}
console.log(variableDentroDelIf); // error: variableDentroDelIf no está accesible
```

CONST

La palabra clave `const` se utiliza para declarar constantes que no pueden cambiar su valor. Las constantes también solo están disponibles dentro del bloque de código en el que se declaran. Por ejemplo:

```
const PI = 3.14159;
PI = 3.14; // Esto causará un error
```

VAR

La palabra clave `var` se utilizaba antes de la introducción de `let` y `const`. Las variables declaradas con `var` también pueden cambiar su valor y están disponibles dentro de la función en la que se declaran, o en el ámbito global si se declaran fuera de cualquier función. A diferencia de `let` y `const`, las variables declaradas con `var` pueden ser redeclaradas en el mismo ámbito. Por ejemplo:

```
var mensaje = "Hola";  
var mensaje = "Hola mundo"; // Esto es posible con var  
console.log(mensaje); // "Hola mundo"  
  
if(true) {  
    var variableDentroDelIf = 2;  
}  
console.log(variableDentroDelIf); // 2. Esta vez si funciona
```

En resumen, la principal diferencia entre `let/const` y `var` es su alcance. Las variables y constantes declaradas con `let` y `const` solo están disponibles dentro del bloque de código en el que se declaran, mientras que las variables declaradas con `var` están disponibles dentro de la función en la que se declaran o en el ámbito global. Además, las variables declaradas con `var` pueden ser redeclaradas en el mismo ámbito, mientras que esto no es posible con `let` y `const`.

Nomenclatura de variables y constantes

En JavaScript, los nombres de variables válidos son aquellos que cumplen ciertas reglas sintácticas y semánticas. Aquí hay algunas pautas a seguir para nombrar variables en JavaScript:

- Los nombres de variables deben comenzar con una letra (a-z o A-Z), un signo de dólar (\$) o un guión bajo (_). No pueden comenzar con un número.
- Los nombres de variables pueden contener letras, números, signos de dólar y guiones bajos. No se permiten otros caracteres especiales.
- Los nombres de variables distinguen entre mayúsculas y minúsculas. Por ejemplo, "nombre" y "Nombre" se consideran nombres de variables diferentes.
- Los nombres de variables no pueden ser palabras reservadas en JavaScript. Por ejemplo, "if", "else", "for", "function", "var", "let", "const" y "while" son palabras reservadas y no se pueden utilizar como nombres de variables.
- Los nombres de variables pueden estar en camelCase o snake_case. La convención de camelCase es escribir palabras juntas, pero la primera letra de cada palabra, excepto la primera, en mayúscula (por ejemplo, miVariableCamelCase). La convención snake_case es escribir palabras juntas, separadas por guiones bajos (por ejemplo, mi_variable_snake_case).

// Nombre válidos ✓

```
let usuario;  
let usuario1;  
let usuario_1;  
let millonario$;  
let v4r$4bl3;  
let _;  
let $;
```

// Nombre inválidos ✗

```
let let;  
let 1usuario;  
let 1_usuario;  
let 2_2;  
let +amor;  
let -odio;
```