
SENPAI
ACADEMY

Repaso 🤦

SENPAI
ACADEMY

Repaso - Variables y Constantes

Las **variables** y **constantes** nos permiten **guardar información**.

```
//Declaración y asignación de una variable
let nombre = 'Senpai';

//Declaración y asignación de una constante
const ciudad = 'Montevideo';

//Reasignación de variable
nombre = 'Senpai Academy';

//Reasignación de una constante DA ERROR
ciudad = 'Maldonado';
```

SENPAI
ACADEMY

Repaso - Tipos primitivos y Objetos

Los valores **primitivos** no están compuestos por otros **tipos de datos**, mientras que los objetos pueden **alojar varios**.

```
//Primitivos
//String
let nombre = 'Senpai';

//Number
let edad = 30;

//Boolean
let esEstudiante = true;

//null
let apellido = null;

//undefined
let hijos;
```

SENPAI
ACADEMY

Repaso - Tipos primitivos y Objetos

```
//Objetos
const persona = {
    nombre: 'Senpai',
    apellido: 'Academy',
    edad: 30,
    esEstudiante: true
};

//Arrays
const personas = ['Gustavo', 'Matías', 'Bruno'];
const edades = [31, 22, 25];
const personasObj = [
    {
        nombre: 'Gustavo',
        apellido: 'Rodríguez',
        edad: 31,
    },
    {
        nombre: 'Matías',
        apellido: 'Bais',
        edad: 24,
    }
];
```

SENPAI
ACADEMY

Repaso - Operadores algebraicos

Podemos utilizar operadores algebraicos para calcular valores.

```
// Teniendo dos variables:
const primerValor = 10;
const segundoValor = 6;

// Suma
const suma = primerValor + segundoValor; // 16

// Resta
const resta = primerValor - segundoValor; // 4

// Multiplicación
const multi = primerValor * segundoValor; // 60

// División
const div = primerValor / segundoValor; // 1.66..

// Módulo
const mod = primerValor % segundoValor; // 4
```

SENPAI ACADEMY

Repaso - Unir strings

Podemos utilizar operadores algebraicos para unir cadenas de texto, o utilizar template strings.

```
//Variable a concatenar
let nombre = 'Senpai';

//Resultado de la concatenación
let saludo1 = 'Hola ' + nombre + '!';

//Igual pero usando template strings
let saludo2 = `Hola ${nombre}!`;
```

SENPAI ACADEMY

Repaso - Operadores comparativos y lógicos

Los operadores comparativos siempre devuelve **true** o **false**, y nos permiten comparar valores.

```
// Teniendo la siguiente variable
const miValor = 10;

const mayorQueDiez = miValor > 10 // False
const menorQueDiez = miValor < 10 // False
const menorOIgualQueDiez = miValor <= 10 // True
const esDiez = primerValor === 10; // True


// NOT ! (negación)
const cuatro = 4; // 4
const esCuatro = cuatro === 4; // True
const noEsCuatro = !esCuatro; // False

// AND &&
const totalCompra = 1500;
const balanceDeCuenta = 12000;
const cuentaBloqueada = false;
const puedeComprar = !cuentaBloqueada && (total < balanceDeCuenta)

// OR ||
const mostrarAyuda = cuentaBloqueada || (balanceDeCuenta < total)
```

SENPAI ACADEMY

Repaso - Control de flujo

Podemos modificar el comportamiento de nuestro código utilizando **condicionales** o **switches**.

Se utiliza **if/elseif/else** en la mayoría de los casos de control de flujo.

```
// Condicionales
let mensaje;

if (itemsCarrito > 0) {
    mensaje = 'Tienes ' + itemsCarrito + ' en tu carrito';
} else {
    mensaje = 'Tu carrito está vacío 😢'
}

// Switch
const estado = 'Sin stock';

switch(estado) {
    case 'Sin stock':
        // Deshabilitar botón de comprar
        // Mostrar mensaje de re-stock
    break;
    case 'En stock':
        // Habilitar botón de comprar
        // Mostrar mensaje de unidades disponibles
    break;
    case 'Descontinuado':
        // Ocultar botón de comprar
        // Mostrar mensaje de producto descontinuado
    break;
}
```

SENPAI ACADEMY

Repaso - Iteraciones o bucles

Los **bucles** de programación están relacionados con todo lo referente a hacer una misma cosa una y otra vez — que se denomina como **iteración** en el idioma de programación.

```
const personas = ['Gustavo', 'Matías', 'Bruno'];

//Iterar con for
for (let i = 0; i < personas.length; i++) {
    console.log('Hola ' + personas[i] + '!');
}

//Iterar con for..in
for (const key in personas) {
    console.log('Hola ' + personas[key] + '!');
}

//Iterar con for..of
for (const persona of personas) {
    console.log('Hola ' + persona + '!');
}
```

SENPAI ACADEMY

Repaso - Funciones

Una función en JavaScript es similar a un procedimiento (**un conjunto de instrucciones que realiza una tarea**).

Hay varias formas de declararlas.

```
//Function declaration
function saludar(){
    alert('Hola!');
}

//Function expression
const saludar2 = function(){
    alert('Hola!');
}

//Arrow function
const saludar3 = () => {
    alert('Hola!');
}

//Parámetros
const saludar4 = (nombre) => {
    alert('Hola' + nombre);
}

//Ejecutar las funciones
saludar();
saludar2();
saludar3();
saludar4('Senpai');
```

SENPAI ACADEMY

Repaso - Funciones (Ejemplo)

```
//Pedir nota
const nota = prompt('Ingrese la nota:');
let notaMensaje;

//Pregunto si la nota es mayor a 100
if(nota > 100){
    notaMensaje = "La nota no es valida!";
}
//Si ese valor es igual 100 le muestro "Excelente!"
else if(nota == 100){
    notaMensaje = "Excelente!";
}
//Sino si ese valor está entre 99 y 70 le muestro "Muy buen
trabajo!"
else if((nota <= 99) && (nota >= 70)){
    notaMensaje = "Muy buen trabajo!";
}
//Sino le muestro "Lo lamento, vuelve a intentarlo luego!"
else {
    notaMensaje = "Lo lamento, vuelve a intentarlo luego!";
}

//Mostrar mensaje
alert(notaMensaje);
```

SENPAI ACADEMY

Repaso - Funciones (Ejemplo)

```
//Movemos a una función que se encarga de darnos el mensaje
function obtenerMensajeNota(nota) {
    let notaMensaje;

    //Pregunto si la nota es mayor a 100
    if(nota > 100){
        notaMensaje = "La nota no es valida!";
    }
    //Si ese valor es igual 100 le muestro "Excelente!"
    else if(nota == 100){
        notaMensaje = "Excelente!";
    }
    //Sino si ese valor está entre 99 y 70 le muestro "Muy buen trabajo!"
    else if((nota <= 99) && (nota >= 70)){
        notaMensaje = "Muy buen trabajo!";
    }
    //Sino le muestro "Lo lamento, vuelve a intentarlo luego!"
    else {
        notaMensaje = "Lo lamento, vuelve a intentarlo luego!";
    }
    return notaMensaje;
}
```

Repaso - Funciones (Ejemplo)

```
//Pedir nota
const nota = prompt('Ingrese la nota:');

//Obtengo el mensaje
let notaMensaje = obtenerMensajeNota(nota);

//Mostrar mensaje
alert(notaMensaje);
```

SENPAI ACADEMY

Repaso - Métodos Arrays

Los arrays son objetos similares a una **lista** cuyo prototipo proporciona **métodos para efectuar operaciones** de recorrido y de mutación. Tanto la **longitud** como el tipo de los elementos de un array son variables.

```
const personas = ['Gustavo', 'Matías', 'Bruno'];

//Obtener el largo del array
console.log(personas.length);
// 3

//Agregar al array
personas.push('Rodrigo');
//['Gustavo', 'Matías', 'Bruno', '']

//Index del item en el array
personas.indexOf('Matías');
// 1

//Recorrer el array (similar for..in + for..of)
personas.forEach((persona) => {
    console.log(persona);
});
//Gustavo
//Matías
//Bruno
//Rodrigo
```

SENPAI ACADEMY

Repaso - Métodos Strings

Las **cadenas** son útiles para almacenar datos que se pueden representar en forma de texto. Algunas de las operaciones más utilizadas en cadenas son verificar su **length** o extraer subcadenas con el método **substring()**

```
const fullName = 'Senpai Academy';

//Obtener el largo del string
console.log(fullName.length);
// 14

//Obtener el carácter en el index
console.log(fullName.charAt(1));
//e

//Obtener el carácter en el index
console.log(fullName[0]);
//S

//Incluye el texto
console.log(fullName.includes('Academy'));
//true

//Divide el texto en un array con las partes
console.log(fullName.split(' '));
//['Senpai', 'Academy']

//Obtiene una sub cadena
console.log(fullName.substring(0, 3));
//Sen
```

SENPAI ACADEMY

SENPAI
ACADEMY

gustavguez

gustavguez

gustavguez

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE