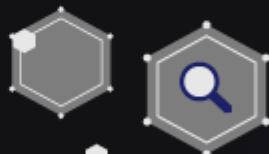


# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT





## Funciones en Javascript

Las funciones son un concepto fundamental en la programación y desempeñan un papel crucial en el desarrollo de aplicaciones. En Javascript, un lenguaje de programación ampliamente utilizado, las funciones nos permiten agrupar y reutilizar bloques de código, lo que facilita la organización y el mantenimiento de nuestro código. es esencial comprender los conceptos básicos de las funciones para poder crear programas más estructurados y modularizados.

En esta lección, exploraremos las funciones en Javascript. Comenzaremos por definir qué es una función y su importancia en la programación. Luego, nos sumergiremos en la sintaxis de las funciones, los diferentes tipos de funciones y cómo utilizarlas en nuestros programas.

# {.js}

# JavaScript



## Funciones en Javascript

### Definición de una Función

Una función en Javascript es un bloque de código que se puede invocar y reutilizar en diferentes partes de un programa. Una función puede recibir parámetros (datos de entrada), realizar ciertas operaciones y devolver un resultado (valor de retorno). Las funciones nos permiten modularizar nuestro código y dividirlo en partes más pequeñas y manejables.

### Sintaxis de una Función

La sintaxis básica de una función en Javascript es la siguiente:

```
function nombreFuncion(parametro1, parametro2) {  
    // Bloque de código  
    // que realiza operaciones  
    // con los parámetros  
    return resultado;  
}
```

**function:** Palabra clave que indica la declaración de una función.

**nombreFuncion:** Nombre de la función, que puede ser elegido por el programador.

**(parametro1, parametro2):** Parámetros que se utilizan para pasar datos a la función. Pueden ser opcionales.

**return:** Palabra clave que especifica el valor que la función devuelve al ser invocada.



## Tipos de Funciones en Javascript

### Funciones sin parámetros y sin valor de retorno:

```
function saludar() {  
    console.log("¡Hola, mundo!");  
}  
saludar(); // Llamada a la función
```

En este ejemplo, la función saludar no recibe ningún parámetro y no devuelve ningún valor. Simplemente imprime "¡Hola, mundo!" en la consola cuando es invocada.

### Funciones con parámetros y sin valor de retorno:

```
function sumar(num1, num2) {  
    var resultado = num1 + num2;  
    console.log("La suma es: " + resultado);  
}  
sumar(5, 3); // Llamada a la función
```

En este ejemplo, la función sumar recibe dos parámetros num1 y num2, realiza la suma de ambos y muestra el resultado en la consola.



## Funciones con parámetros y valor de retorno:

```
function multiplicar(num1, num2) {  
    var resultado = num1 * num2;  
    return resultado;  
}  
var resultadoMultiplicacion = multiplicar(4, 2);  
// Llamada a la función y almacenamiento del  
resultado  
console.log("El resultado de la multiplicación  
es: " + resultadoMultiplicacion);
```

En este ejemplo, la función `multiplicar` recibe dos parámetros `num1` y `num2`, realiza la multiplicación de ambos y devuelve el resultado. El valor de retorno es almacenado en la variable `resultadoMultiplicacion` y se muestra en la consola.



# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT

### Ejemplos y Demostraciones

Veamos algunos ejemplos y demostraciones adicionales relacionados con las funciones en Javascript:

#### Ejemplo de una función para calcular el área de un círculo:

```
function calcularAreaCirculo(radio) {  
    var area = Math.PI * radio * radio;  
    return area;  
}  
var radioCirculo = 5;  
var areaCirculo = calcularAreaCirculo(radioCirculo);  
console.log("El área del círculo es: " + areaCirculo);
```

En este ejemplo, la función `calcularAreaCirculo` recibe el radio de un círculo como parámetro, calcula el área utilizando la fórmula adecuada y devuelve el resultado.

# LENGUAJES DE PROGRAMACIÓN

## JAVA SCRIPT



### Ejemplo de una función para verificar si un número es par:

```
function esNumeroPar(numero) {  
  if (numero % 2 === 0) {  
    return true;  
  } else {  
    return false;  
  }  
}  
  
var numeroEjemplo = 4;  
var esPar = esNumeroPar(numeroEjemplo);  
console.log("¿El número es par? " + esPar);
```

En este ejemplo, la función `esNumeroPar` recibe un número como parámetro, verifica si es divisible por 2 y devuelve `true` si es par, y `false` si no lo es.





Es fundamental comprender estos conceptos básicos de las funciones en Javascript para poder escribir programas más estructurados, modulares y reutilizables. A medida que avancemos en nuestro aprendizaje, podremos utilizar funciones más complejas y aprovechar las capacidades avanzadas de Javascript para crear programas más poderosos y flexibles.

An illustration of a person with purple hair, wearing a light blue t-shirt and blue pants, sitting in a pink office chair at a white desk. They are using a keyboard and mouse. On the desk is also a white mug. A large computer monitor displays a code editor with various colored lines of code. Floating around the monitor are several icons: a blue box with 'Java', a pink box with 'PHP', a blue box with a Wi-Fi symbol, a purple box with 'C++', and a pink box with 'SQL'. The background is a light blue gradient with some faint, abstract shapes.