

Fundamentos de Programación con JavaScript

Funciones

06

JS



Funciones

JS

function()

Son fragmentos de código que escribimos para ejecutar una tarea y no volver a escribir el mismo código más de una vez. Las funciones son la base de la modularidad en JavaScript.

Las funciones en JavaScript se definen mediante la palabra reservada **function**, seguida del nombre de la función.



El nombre de la función se utiliza para llamar a esa función cuando sea necesario. El concepto es el mismo que con las variables, a las que se les asigna un nombre único para poder utilizarlas dentro del código. Después del nombre de la función, se incluyen dos paréntesis donde indicaremos los parámetros de la función. Por último, los símbolos `{` y `}` se utilizan para encerrar todas las instrucciones que pertenecen a la función.

Argumentos, parámetros y valores de retorno

Las funciones más sencillas no necesitan ninguna información para producir sus resultados, aunque lo normal es que necesiten de datos para producir resultados.

Las variables **que reciben** las funciones se llaman **parámetros**. Antes de que pueda utilizarlos, la función debe indicar cuántos parámetros necesita y cuál es el nombre de cada parámetro.

Además, al llamar a la función, se deben incluir los valores (expresiones o argumentos) que se le van a pasar a la función. Los **argumentos que se envían** se indican dentro de los paréntesis que van detrás del nombre de la función y se separan con una coma (,).

Las funciones pueden devolver los valores que han calculado. Para devolver valores dentro de una función, se utiliza la palabra reservada **return**.

Aunque las funciones pueden devolver valores de cualquier tipo, solamente pueden devolver un valor cada vez que se ejecutan.

Funciones anónimas

Un patrón común en JavaScript son las funciones anónimas autoejecutables. Este patrón consiste en crear una expresión de función e inmediatamente ejecutarla.

```
(function() {  
    var foo = 'Hola mundo';  
})();  
  
console.log(foo);    // indefinido (undefined)
```

Llamar a una función suspende la ejecución de la función actual, pasando el control y los parámetros a la nueva función.

Además de los argumentos declarados, todas las funciones reciben dos argumentos extra: **this** y **arguments**.

Ejemplo

```
let num1 = 2;  
let num2 = 4;  
let resultado;
```

```
✓ function sumar(a, b) {  
  let suma;  
  suma = a + b;  
  return suma;  
}
```

```
resultado = sumar(num1, num2);  
console.log(`Resultado: ${num1}+${num2} = ${resultado}`);
```

Fundamentos de Programación con JavaScript



JS