**Funciones**

* **Tabla de contenido**
* **Declaración de funciones**
* **Variables Locales**
* **Variables Externas**
* **Tutorial de Funciones**
* **AprendeJavascript.dev**
* **30 días de Javascript**
* **Javascript.info**

Muy a menudo necesitamos realizar acciones similares en muchos lugares del script.

Por ejemplo, debemos mostrar un mensaje atractivo cuando un visitante inicia sesión, cierra sesión y tal vez en otros momentos.

Las funciones son los principales “bloques de construcción” del programa. Permiten que el código se llame muchas veces sin repetición.

Ya hemos visto ejemplos de funciones integradas, como alert(message), prompt(message, default) y confirm(question). Pero también podemos crear funciones propias.

Declaración de funciones#

Para crear una función podemos usar una declaración de función.

Se ve como aquí:

function showMessage() {

alert( '¡Hola a todos!' );

}

La palabra clave function va primero, luego va el nombre de función, luego una lista de parámetros entre paréntesis (separados por comas, vacía en el ejemplo anterior) y finalmente el código de la función entre llaves, también llamado “el cuerpo de la función”.

function name(parameter1, parameter2, ... parameterN) {

// body

}

Nuestra nueva función puede ser llamada por su nombre: showMessage().

Por ejemplo:

function showMessage() {

alert( '¡Hola a todos!' );

}

showMessage(); // llamando a la funcion showMessage()

showMessage(); // llamando a la funcion showMessage()

La llamada showMessage() ejecuta el código de la función. Aquí veremos el mensaje dos veces.

Este ejemplo demuestra claramente uno de los propósitos principales de las funciones: evitar la duplicación de código..

Si alguna vez necesitamos cambiar el mensaje o la forma en que se muestra, es suficiente modificar el código en un lugar: la función que lo genera.

**Variables Locales**

Una variable declarada dentro de una función solo es visible dentro de esa función.

Por ejemplo:

function showMessage() {

let message = "Hola, ¡Soy JavaScript!"; // variable local

alert( message );

}

showMessage(); // Hola, ¡Soy JavaScript!

alert( message ); // <-- ¡Error! La variable es local para esta función

**Variables Externas**

Una función también puede acceder a una variable externa, por ejemplo:

1

2

3

4

5

6

let userName = 'Juan';

function showMessage() {

let message = 'Hola, ' + userName;

alert(message);

}

showMessage(); // Hola, Juan

La función tiene acceso completo a la variable externa. Puede modificarlo también.

Por ejemplo:

1

2

3

4

5

6

7

8

9

let userName = 'Juan';

function showMessage() {

userName = "Bob"; // (1) Cambió la variable externa

let message = 'Hola, ' + userName;

alert(message);

}

alert( userName ); // Juan antes de llamar la función

showMessage();

alert( userName ); // Bob, el valor fué modificado por la función

La variable externa solo se usa si no hay una local.

Si una variable con el mismo nombre se declara dentro de la función, le hace sombra a la externa. Por ejemplo, en el siguiente código, la función usa la variable userName local. La exterior se ignora:

1

2

3

4

5

6

7

8

9

10

11

let userName = 'John';

function showMessage() {

let userName = "Bob"; // declara variable local

let message = 'Hello, ' + userName; // Bob

alert(message);

}

// la función crea y utiliza su propia variable local userName

showMessage();

alert( userName ); // John, se mantiene, la función no accedió a la variable externa

Variables globales

Variables declaradas fuera de cualquier función, como la variable externa userName en el código anterior, se llaman globales.

Las variables globales son visibles desde cualquier función (a menos que se les superpongan variables locales con el mismo nombre).

Es una buena práctica reducir el uso de variables globales. El código moderno tiene pocas o ninguna variable global. La mayoría de las variables residen en sus funciones. Aunque a veces puede justificarse almacenar algunos datos a nivel de proyecto.

Tutorial de Funciones#

En los siguientes tutoriales aprenderás más sobre como puedes usar esta estructura, sin embargo sientete libre a investigar un poco mas por tu cuenta 😄.

AprendeJavascript.dev#

Tu primera funcion

Parametros

Function Expression

30 días de Javascript#

Funciones

Javascript.info#

Funciones