

Introducción a la programación en JavaScript

Módulo 4 - Funciones

Funciones

Funciones - Sintaxis

Las funciones son bloques de código que tienen la característica **de ser llamados o invocados**. La siguiente sería una sintaxis de una **función que, por ejemplo, contiene una alerta**:

```
<script>

function miFuncion(){

    alert('Hola estoy dentro de una función')

}

</script>
```

Funciones

Si levantamos esto en **nuestro navegador**, básicamente no obtendremos nada, porque esta función no ha sido invocada.

Para **invocar debemos hacer lo siguiente**:

```
<script>

function miFuncion(){

    alert('Hola estoy dentro de una función')

}

miFuncion()

</script>
```

Funciones

De esta manera, estamos llamando a nuestra función. Si queremos hacer un llamado dos veces, haremos lo siguiente:

```
<script>

function miFuncion(){

    alert('Hola estoy dentro de una función')

}

miFuncion()
miFuncion()

</script>
```

El resultado será el siguiente:



Funciones, return y parámetros

Por ejemplo, en el siguiente caso hemos generado una función que tomará parámetros, es decir, pasamos a esta **función determinados datos**.

```
<p id="caja"></p>

<script>
function miOperacion(p1, p2) {
  return p1 * p2;
}
document.getElementById("caja").innerHTML = miOperacion(4, 3);
</script>
```

Funciones, return y parámetros

En este caso vemos cómo return nos permite retornar un valor, y de esta forma también podemos ir variando los datos que pasamos en la misma función. **Es decir, una misma función nos sirve para todo.**

Por ejemplo, **podríamos volver a utilizarla de la siguiente forma.**

```
<p id="cajaA"></p>
<p id="cajaB"></p>

<script>
function miOperacion(p1, p2) {
  return p1 * p2;
}
document.getElementById("cajaA").innerHTML = miOperacion(4, 3);
document.getElementById("cajaB").innerHTML = miOperacion(2, 18);
</script>
```

Funciones, return y parámetros

La misma función nos dará dos resultados diferentes, sin necesidad de crear una función nueva:

12

36

Es decir,

```
function ejemplo(parametro1, parametro2, parametro3)
{
    //lo que hace a función
}
```


Ámbito de la variable

Entendiendo el ámbito de una variable

Algo importante a tener en cuenta es que las variables pueden **ser locales o globales**. Si, por ejemplo, yo hago esto:

```
<script>
function miOperacion() {
  var resultado = 30;

}
alert(resultado)
</script>
```

Nos dará el **siguiente error**, es que básicamente la variable resultado está dentro del ámbito **de miOperación()** y no puedo utilizarla por fuera.

```
✖ 18:05:33.604 Uncaught ReferenceError: resultado is not defined
    at ejemplo.html:17
```

Revisión

- Repase el concepto de **Función**.
- Aprenda cómo funcionan los **módulos desde el código**.
- Estudie cómo **trabajar con parámetros** desde una **Función**.
- Entienda cómo definir correctamente el **ámbito de una variable**.
- Trabaje en el **Proyecto Integrador**.
- Realice las preguntas necesarias al/la docente antes de continuar.



¡Muchas gracias!

¡Sigamos trabajando!