



El futuro digital
es de todos

MinTIC

Formador: Sergio Medina

» JavaScript
Funciones y Métodos

xxx

Ciclo No.3 – Semana No.4



Misión
TIC2022



Funciones

Una función en JavaScript es similar a un procedimiento, es un conjunto de instrucciones que realiza una tarea o calcula un valor, pero para que un procedimiento califique como función, debe tomar alguna entrada y devolver una salida donde hay alguna relación obvia entre la entrada y la salida. Para usar una función, se debe definir en algún lugar del ámbito desde el que se desea invocarla.



Declarar Funciones

- La palabra clave **function**.
- El nombre de la función.
- Una lista de parámetros de la función, entre paréntesis y separados por comas.
- Las declaraciones de JavaScript que definen la función, encerradas entre llaves, { ... }.

```
function nombre_funcion(parametros){  
  instrucciones;  
}
```



El futuro digital
es de todos

MinTIC



Declarar Funciones



Ejemplo:

```
function square(number) {  
  return number * number;  
}
```

Los parámetros pueden ser tipos de datos primitivos o tipo Array o tipo Object.



Declarar Funciones

```
function Saludar(nombre, edad){  
  alert("Hola mi nombre es: "+ nombre + " y  
  tengo: " + edad + " años.");  
}
```

```
Saludar("Claudia", 25);
```




Declarar Funciones

```
function Sumar(num1,num2){
```

```
    var resultado= num1 + num2;  
    alert("La suma es: "+ resultado);  
}
```

```
Sumar(10,15);
```



Funciones con la Instrucción return

Una función JavaScript puede tener la instrucción **return**. Esta es requerida si queremos que la función dada nos retorne un valor. La instrucción **return** debe ser la última línea en una función.

```
function Valores(a,b,c){  
    return a+b+c;  
}  
var suma = Valores(3,4,5);
```



Expresiones function

Las funciones también se pueden crear mediante una expresión **function**, se crea una función anónima y se le asigna a una variable

```
var cuadrado = function(numero) { return numero *  
numero }
```

```
var x = cuadrado(5) // x obtiene el valor 25
```




Llamado de funciones

Para invocar una función desde cualquier parte del programa, solo es necesario escribir el nombre de la función creada seguido de paréntesis(en caso de tener parámetros indicarlos dentro, separados por comas).

Ejemplo:

```
Sumar();
```

```
Sumar(5,3);
```



El objeto arguments

El **arguments** de una función se mantiene en un objeto similar a un arreglo.

Dentro de una función, se pueden abordar los argumentos que se le pasan de la siguiente manera:

arguments[i]



El objeto arguments

x
x
x

```
function Países(listado) {  
    var result = "";  
    var i;  
    for (i = 0; i < arguments.length; i++) {  
        result += arguments[i] + "-";  
    }  
    return result;  
}  
alert(Países('Bolivia', 'Ecuador', 'Francia', 'Brasil'));
```



» Funciones predefinidas

x
x
x

```
function Países(listado) {  
    var result = "";  
    var i;  
    for (i = 0; i < arguments.length; i++) {  
        result += arguments[i] + "-";  
    }  
    return result;  
}  
alert(Países('Bolivia', 'Ecuador', 'Francia', 'Brasil'));
```



Método

- Un método es una función la cual es propiedad de un **Objeto**. Existen dos tipos de métodos:
- **Métodos de Instancia** los cuales son tareas integradas realizadas por la instancia de un objeto.
- **Métodos Estáticos** que son tareas que pueden ser llamadas directamente en el constructor de un objeto.



Método

- Los strings, los números, los arrays y las variables de tipo fecha son en realidad objetos. Por lo tanto también tienen métodos.

Ejemplos:

MÉTODOS NUMÉRICOS:

- **parseInt()** transforma un string en número entero.
- **parseFloat()** hace lo mismo que parseInt pero en este caso con un número decimal.



Método

MÉTODOS PARA ARRAYS:

- **length()** Devuelve el número de elementos de una lista.
- **join()** junta todos los elementos de una lista separados por el carácter que indiquemos dentro del paréntesis (entre comillas porque será un string).



Método

MÉTODOS PARA ARRAYS:

- **pop()** elimina el último elemento de una lista.
- **sort()** ordena alfabéticamente o numéricamente los elementos de una lista.
- **slice()** obtienen una copia de la lista.



CREACIÓN MANUAL DE UN MÉTODO

```
var producto={  
    codigo: 5,  
    nombre: "Manzana",  
    precio: 3500,  
    incremento: function (valor){  
        this.precio+=valor  
    }  
}  
this.producto.incremento(500);  
alert(this.producto.precio);
```



El futuro digital
es de todos

MinTIC

Formador: Sergio Medina

» JavaScript
Funciones y Métodos

xxx

Ciclo No.3 – Semana No.4



Misión
TIC2022