



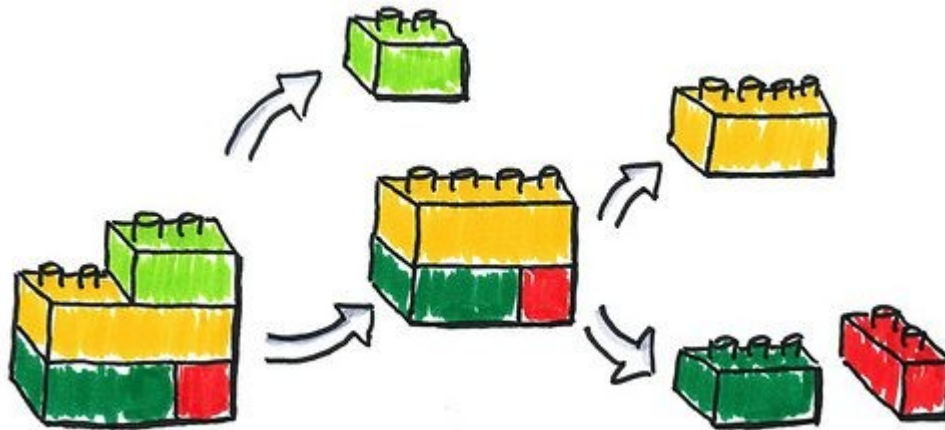
Funciones

JavaScript Fundamentals



Funciones

Son bloques de código o subrutinas que nos permiten ejecutar algunas tareas que pueden ser repetitivas a lo largo de un programa.



Funciones

```
function nombreFuncion(parámetros)  
{  
    // instrucción 1  
    // instrucción 2  
    // instrucción 3  
    // .....  
    // instrucción n  
}
```

```
function mensaje() {  
    console.log('Bienvenido');  
}  
// función sin parámetros
```

```
function saludar(nombre) {  
    console.log('Bienvenido ' + nombre);  
}  
// función con parámetros
```

Nota: Las variables creadas dentro de la función existen solamente en su ámbito y no son accesibles desde fuera de ella.

Reto 1

```
// Desarrollar una funcion que realice la multiplicacion de dos numeros enteros.  
// Se debe garantizar que los numeros sean enteros positivos.  
  
// Dado un numero por el usuario, realice una funcion que tome dicho numero  
// y nos indique si el numero es par o impar, el resultado se debe mostrar en consola.
```

Reto 2

```
let nombre = 'John';  
  
✓ function mostrarMensaje() {  
  |   let nombre = 'Jerry';  
  |   console.log('Bienvenido ' + nombre);  
  | }  
  
mostrarMensaje(nombre); // Que imprime esta linea de codigo?  
console.log(nombre);    // Que imprime esta linea de codigo?
```

Funciones

Valores por defecto

Son valores que podemos definir a una función en caso de que no se reciba ningún parámetro

```
function saludar (nombre = 'John Doe') {  
    console.log(nombre);  
}
```

```
saludar('Julio Cesar'); // Imprime el nombre como parámetro  
saludar();              // Imprime el nombre por defecto
```

Reto 3

```
// Crear una funcion que reciba dos parametros y realice la multiplicacion de ambos,  
// si algun parametro falta debe realizar la operacion con el valor de 0.
```


Funciones

Retorno de valores

Algunas funciones pueden retornar valores

```
function saludar (nombre = 'John Doe') {  
    return 'Bienvenido ' + nombre;  
}
```

```
let mensajeSaludo = saludar('Julio Cesar');  
console.log(mensajeSaludo);
```

Reto 4

```
// Desarrollar una funcion que realice la multiplicacion de dos numeros enteros.  
// Se debe garantizar que los numeros sean enteros positivos.  
  
// Dado un numero por el usuario, realice una funcion que tome dicho numero  
// y nos indique si el numero es par o impar, el resultado se debe mostrar en consola.  
|
```

Modificar estas funciones, para que nos retornen el valor solicitado, e imprimirlo en consola.

Funciones

Funciones como expresiones

Es una forma alternativa de definir una función.

```
let saludo = function (nombre) {  
    console.log('Bienvenido ' + nombre);  
}
```

```
saludo('John Doe');
```

Reto 5

```
// Utilizando la forma de funcion como expresion, cree una funcion cuadrado que, retorne  
// el cuadrado de un numero dado, este numero debe ser solicitado al usuario, e imprimir su  
// resultado en consola.
```

Reto 6

```
// Cree las siguientes funciones
// obtenerArea(largo, ancho), obtenerPerimetro(largo, ancho), para un rectangulo
// largo: numero que denota el largo del rectangulo
// ancho: numero que denota el ancho del rectangulo
//
// Restricciones
// 1 <= largo, ancho <= 1000
//
// formula
// El area del rectangulo = largo x ancho
// El perimetro del rectangulo es = 2 x (largo + ancho)
//
// datos de prueba
// largo = 3, ancho = 4.5
//
// salida
// area = (3 x 4.5) = 13.5
// perimetro = 2 x (3 + 4.5) = 15
```

Reto 7

```
// Cree una funcion que retorne el factorial de un numero
//
// factorial(n)
//
// El factorial se define como:  $4! = 4 \times 3 \times 2 \times 1$ 
// por lo tanto factorial(4) = 24
```

Reto 8

```
// Defina una constante PI y asignele el valor de la funcion Math.PI
// Solicite un numero al usuario que denotaremos por radio
// Cree una funcion que reciba el radio y calcule el area de un circulo
//
// Restriccion
// 0 <= radio <= 100
//
// Area de un circulo = PI * radio ** 2
```

Reto 9

```
// Cree una funcion que tome el string 'javascriptloops' y retorne el numero de vocales y el numero de  
// consonantes de la siguiente forma
```

```
/*
```

```
vocales:
```

```
a
```

```
a
```

```
i
```

```
o
```

```
o
```

```
consonantes:
```

```
j
```

```
v
```

```
s
```

```
c
```

```
r
```

```
p
```

```
t
```

```
l
```

```
p
```

```
s
```

```
*/
```


Reto 10

```
function validaEdad(edad) {  
  if (edad > 20) {  
    return 'true';  
  } else {  
    return 'false';  
  }  
}
```

```
// Reescriba la funcion anterior utilizando el operador ternario ?  
|
```

Funciones

Funciones arrow

Es una forma abreviada de expresar las funciones, fue introducido en la especificación ECMAScript6.

// función ordinaria

function nombreFuncion(parámetros) {

// instrucciones

}

// función arrow

(parametros) => { sentencias }

// llamado a la funcion

nombreFuncion(parametros);

Funciones

Ejemplo de funciones arrow

```
// función ordinaria  
function saludar(nombre) {  
    console.log('Bienvenido ' + nombre);  
}
```

```
saludo('John Doe');
```

```
// función arrow  
let saludar = (nombre) => { console.log('Bienvenido ' + nombre) }  
  
saludar('John Doe');
```

Funciones

Sintaxis alternativa de las arrow function.

(param1, param2, param3,.....paramn) => { instrucciones }; // normal

(param1, param2, param3,.....paramn) => instrucciones ; // Solo una instrucción

() => { instrucciones }; // sin parametros, paréntesis obligatorio

(param1) => { instrucciones }; // paréntesis opcionales con un solo parámetro

(param1) => { instrucciones };



Reto 11

Convertir: Reto1, Reto3, Reto4, Reto5 a **Arrow functions**.

Eso es todo, y ahora.....?



The big challenge.....

```
// Crear un programa, que muestre un menu de opciones al usuario, para realizar las siguientes
// actividades.

// suma(a, b) // realiza la suma de dos valores a y b
// resta(a, b) // realiza la resta de dos valores a y b
// producto(a, b) // realiza el producto de dos valores a y b
// division(a, b) // realiza la division de dos valores a y b
// modulo(a, b) // realiza el modulo de dos valores a y b
// cuadrado(a) // realiza el cuadrado de a
// raiz(c) // calcula la raiz cuadrada de c
// potencia(x, y) // calcula el valor de x a la potencia y
// maximo(a, b) // calcula el valor maximo entre a y b
// minimo(a, b) // calcula el valor minimo entre a y b

// el programa debe terminar, cuando el usuario presiona ingresa el numero 0.

// restricciones, los valores ingresados deben estar entre 0 <= argumento <= 100

// todos los valores resultados, deben ser escritos en la consola del navegador.
```