

Recorte las tarjetas. ✂

**FUNCIÓN
MÉTODO DOM**

FUNCIÓN ANONIMA

**FUNCIÓN SIN
PARÁMETROS**

FUNCIÓN ESCAPE

FUNCIÓN STRING

FUNCIÓN NUMBER

FUNCIÓN ISNAN

FUNCIÓN ISFINITE

FUNCIÓN EVAL

FUNCIÓN FLECHA



**FUNCIÓN
PARÁMETROS REST**



**FUNCIÓN CON
PARÁMETROS POR
DEFECTO**



**FUNCIÓN CON
CLAUSURAS**



**FUNCIÓN
MULTI ANIDADA**



FUNCIÓN ANIDADA



**FUNCIÓN
MULTIPLICACIÓN**



**FUNCIÓN CON
ARGUMENTOS**



**FUNCIÓN CON
CONFLICTO DE
NOMBRES**



**FUNCIÓN
EXTERNA/INTERNA**



FUNCIÓN CON NODOS



**FUNCIÓN DE
LLAMADA RECURSIVA**

FUNCIÓN BUCLE

**FUNCIÓN RECURSIVA
FACTORIAL**

FUNCIÓN RECURSIVA

**LLAMADO DE
FUNCIÓN**

FUNCIÓN MYFUNC

FUNCIÓN MAP

FUNCIÓN FACTORIAL

FUNCIÓN SQUARE

**FUNCIÓN CON
RETURN**



**FUNCIÓN CON
PARÁMETROS**



**EXPRESIÓN DE LA
FUNCIÓN**



SUMA



**LLAMADO DE
FUNCIONES**



**DECLARACIÓN DE LA
FUNCIÓN**



SINTAXIS



Recorte las plantillas. ✂

```
var myFunc;  
if (num == 0){  
  myFunc = function(theObject) {  
    theObject.make = "Toyota"  
  }  
}
```

```
var cuadrado = function(number) {  
  return number * number;  
}  
  
//Llamada a la función  
  
var resultado = cuadrado(4);  
  
//resultado vale ahora 16
```

```
var objRef;  
objRef = Number(objRef);  
objRef = String(objRef);
```

```
function A(x) {  
  function B(y) {  
    function C(z) {  
      console.log(x + y + z);  
    }  
    C(3);  
  }  
  B(2);  
}  
A(1); // registra 6 (1 + 2 + 3)
```

```
var factorial = function fac(n) {return n<2 ? 1 : n*fac(n-1)};  
  
print(factorial(3));
```

```
isNaN(testValue);
```

```
function multiply(multiplier, ...theArgs) {  
  return theArgs.map(x => multiplier * x);  
}  
  
var arr = multiply(2, 1, 2, 3);  
console.log(arr); // [2, 4, 6]
```

```
function myConcat(separator) {  
  var result = "", // initialize list  
      i;  
  // iterate through arguments  
  for (i = 1; i < arguments.length; i++) {  
    result += arguments[i] + separator;  
  }  
  return result;  
}
```

```
var createPet = function(name) {  
  return {  
    setName: function(name) {  
      name = name;  
    }  
  }  
}
```

```
function factorial(n){
  if ((n == 0) || (n == 1))
    return 1;
  else
    return (n * factorial(n - 1));
}
```

```
function map(f,a) {
  var result = [], // Crea un nuevo Array
  i;
  for (i = 0; i != a.length; i++)
    result[i] = f(a[i]);
  return result;
}
```

```
function mostrarDia() {
  var fecha = new Date();
  alert("Hoy es día " + fecha.getDate());
}

//Llamada a la función.
mostrarDia();
```

```
var x = 0;
while (x < 10) { // "x < 10" es la condicion del bucle
  // hacer cosas
  x++;
}
```

```
var Cantar = function(){
  alert("Yo canto muy bien");
}
```

```
function Saludar(){
  alert("Hola a todos");
}
```

```
function outside(x) {
  function inside(y) {
    return x + y;
  }
  return inside;
}

fn_inside = outside(3); // Pensar en esto como: dar una funcion que suma 3 a lo que sea que des
result = fn_inside(5); // retorna 8

result1 = outside(3)(5); // retorna 8
```

```
var str = "12",
    num;
num = Number(str);
```

```
function Sumar(num1,num2){

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

Sumar(8,9);
```



```
escape(string);  
unescape(string);
```

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

```
var a, b, c, d, e;  
a = factorial(1); // a obtiene el valor 1  
b = factorial(2); // b obtiene el valor 2  
c = factorial(3); // c obtiene el valor 6  
d = factorial(4); // d obtiene el valor 24  
e = factorial(5); // e obtiene el valor 120
```

```
if(isFinite(ClientInput)){  
    /* tomar pasos esepficos */  
}
```

```
Sumar();  
Saludar();  
Cantar();
```

```
function walkTree(node) {  
    if (node == null) //  
        return;  
    // hacer alguna cosa con el nodo  
    for (var i = 0; i < node.childNodes.length; i++) {  
        walkTree(node.childNodes[i]);  
    }  
}
```

```
var a = [  
    "Hydrogen",  
    "Helium",  
    "Lithium",  
    "Beryllium"  
];  
  
var a2 = a.map(function(s){ return s.length });  
  
var a3 = a.map( s => s.length );
```

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

```
function multiply(a, b = 1) {  
    return a*b;  
}  
  
multiply(5); // 5
```

```
function sumar(){

    var num1 = 5;
    var num2 = 8;
    suma = num1 + num2;
    alert("La suma es: "+suma);
}
```

```
function outside() {
    var x = 10;
    function inside(x) {
        return x;
    }
    return inside;
}
result = outside()(20); // retorna 20 en lugar de 10
```

```
function addSquares(a,b) {
    function square(x) {
        return x * x;
    }
    return square(a) + square(b);
}
a = addSquares(2,3); // retorna 13
b = addSquares(3,4); // retorna 25
c = addSquares(4,5); // retorna 41
```

```
function multiply(a, b) {
    b = typeof b !== 'undefined' ? b : 1;

    return a*b;
}

multiply(5); // 5
```

Ejemplo:

```
function Valores(a,b,c){

    return a+b+c;
}

var suma = Valores(3,4,5); // esta función nos retorna el valor de 12
```

```
console.log(square(5));
/* ... */
function square(n) { return n*n }
```

`eval(cadena de caracteres o string);`

```
function loop(x) {
    if (x >= 10) // "x >= 10" es la condicion de salida (equivalente a "!(x < 10)")
        return;
    // hacer cosas
    loop(x + 1); // la llamada recursiva
}
loop(0);
```

```
var str = "12",
    num;
document.write(typeof str);
document.write("<br/>");
num = Number(str);
document.write(typeof num);
```