

Funciones en JS. Con parámetros

funciones.js

```
function suma1(a, b){  
    let resul = a+b;  
    alert(resul);  
}
```

funciones.js

llamada desde el archivo .js

```
window.onclick=suma1(10,90);
```

index.html llamada desde html a la function

```
<script src="funciones.js"></script>  
  
<script>  
    suma1(10,90);  
</script>  
  
<div onclick="suma1(10,90);">Suma</div>
```

Funciones en JS. Sin parámetros

funciones.js

```
function salir(){  
    location.assign("http://www.w3schools.com");  
}  
  
let salir=function salir(){  
    location.assign("http://www.w3schools.com");  
}
```

index.html llamada desde html

```
<script src="funciones.js"></script>  
<script>  
    salir();  
</script>  
<div onclick="salir();">Salir</div>
```

funciones.js

llamada desde el archivo .js

```
window.onclick=salir;
```

Funciones en JS. Devolviendo un valor

funciones.js

```
let recarga = function recarga(){  
    if(confirm('Atención! Los datos se borrarán.')){  
        location.reload(true);  
        return true;  
    }  
    return false;  
};
```

funciones.js llamada desde el archivo .js

`window.onclick=recargar;`

index.html llamada desde html

```
<script src="funciones.js"></script>  
  
<script>  
  
</script>  
<div onclick="recarga();">Recargar</div>
```

Funciones en JS. Arrow functions

(declaración tradicional con o sin parámetros, devuelva o no resultado)

```
let recarga = function recarga(){  
    if(confirm('Atención! BORRAMOS.')){  
        location.reload(true);  
        return true;  
    }return  
    false; };
```

(declaración directa arrow function)

```
function suma1(a, b){  
    let resul = a+b;  
    alert(resul); }
```

```
let recarga = () =>{... }  
let suma1 = (a,b)=>{... }
```

(declaración tradicional función anónima)

```
window.setTimeout(function(){ alert("Wake Up!");},5000);
```

Eventos capturadores en JS

EVENTOS

- Definen las posibles las interacciones con los elementos de la web.

MOUSE EVENTS

onclick, ondblclick,
onmousedown, onmousemove,
onmouseover, onmouseout,
onmouseup

KEYBOARD EVENTS

- onkeydown, onkeypress, onkeyup

OBJECT EVENTS

- onload, onresize, onscroll, onunload

FORM EVENTS

- onblur, onfocus, onselect, onchange , onsubmit, onreset

Capturar un evento en el HTML

ESTABLECER CAPTURADOR EN CÓDIGO

```
<div id="divAzul" onclick="ejecutaEsto();" > </div>  
<body onload="ejecutaEsto()">
```

AÑADIR FUNCIONES A UN EVENTO EN CÓDIGO

```
<body onload="ejecutaEsto(); estoTambien();" >
```

Capturar un Evento por JS

- Utilizando una función anónima, o de manera directa

```
window.onload = function(){ };
```

- Definiendo la función en una variable de tipo constante

```
const miFuncion = function ejecutaEsto(){ //código a ejecutar };
```

```
const miFuncion2 = function estoTambien(){ //código a ejecutar2 };
```

```
window.onload = miFuncion, ;
```

```
elem.onclick = miFuncion2 ;
```

AÑADIR CAPTURADOR POR JAVASCRIPT

```
window.addEventListener("load", miFuncion);
```

```
window.addEventListener("load", miFuncion2);
```

```
elem.addEventListener("click", ()=>{...}); ...
```