

Introducción a la programación en JavaScript

Módulo 4 - Eventos

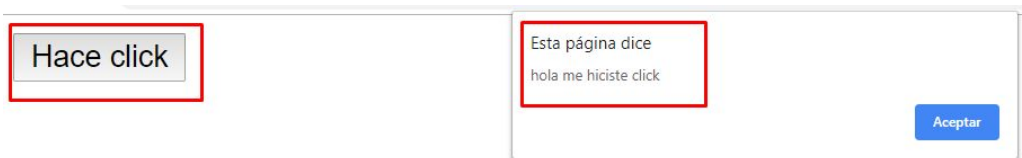
Eventos

Entendiendo los eventos...

Los eventos son algo qué les pasa a los **elementos del HTML**,

```
<button onclick="alert('hola me hiciste click')"> Hace click </button>
```

El resultado en el navegador será el siguiente,



De esta forma, hemos dicho que al hacerle clic a ese **botón disparará una alerta**.

```
<button onclick="alert('hola me hiciste click')"> Hace click </button>
<button onmouseover="alert('hola me pasaste el mouse por encima')"> Acerca el cursor </button>
<button onmouseout="alert('Alejá el cursor')"> Aleja el cursor </button>
<button onmousemove="alert('Mové el cursor del mouse sobre mí')"> Mové el cursor </button>
```

Eventos desde el HTML

Por otro lado, también podemos usar eventos relacionados con el teclado. En la imagen de abajo, generamos una **caja de texto (input)**.

```
<input type="text" onchange="alert('haz hecho un cambio en la caja')">  
<input type="text" onblur="alert('haz dejado de hacer foco en la caja')">  
<input type="text" onfocus="alert('haz hecho foco en la caja')">  
<input type="text" onkeypress="alert('haz presionado una tecla dentro de la caja')">  
<input type="text" onkeyup="alert('haz dejado de presionar una tecla dentro la caja')">
```

Como vemos, hay diferentes eventos, pero también hay diferentes formas de trabajarlos, en el caso anterior **lo hacemos desde el propio HTML**.

Eventos desde el código

También a los eventos se los puede llamar **directamente desde el código**.

```
<button> Haceme click </button>

<script>

document.querySelector('button').onclick = alerta;
function alerta(){

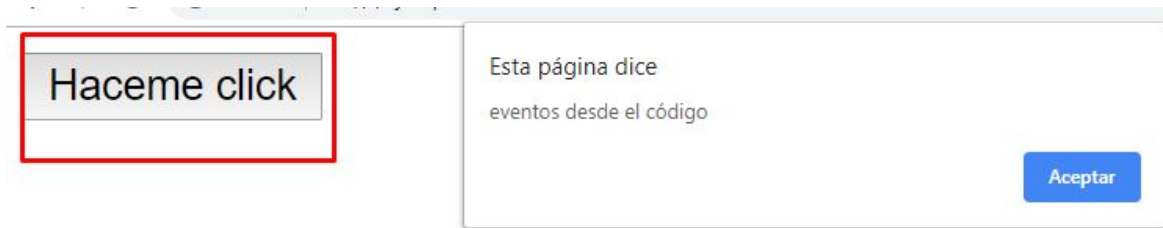
  alert('eventos desde el código')

}

</script>
```

Eventos desde el código

Para entenderlo mejor, mostramos una imagen donde el **usuario al hacerle clic al botón** dispara la **siguiente alerta**.



Eventos con parámetros

Ahora toma especial interés lo antes visto, porque podemos **pasar parámetros** y por ejemplo hacer lo siguiente.

```
<button onclick="saludarA('Ana')"> Ana </button>

<script>

function saludarA(nombre){

  alert('Hola ' + nombre)

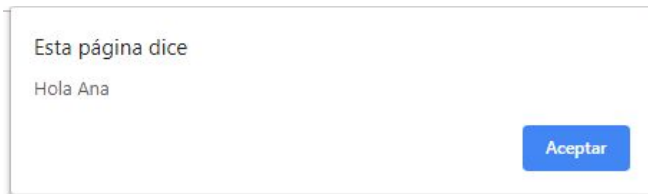
}

</script>
```

Eventos con parámetros

El resultado al hacerle click al botón será el siguiente:

Ana

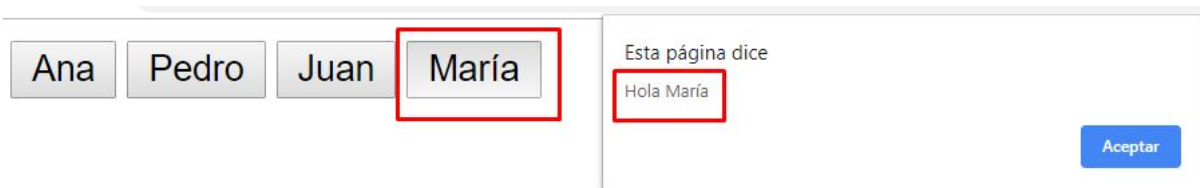


Lo interesante, **es qué podemos reutilizar esa misma función para diferentes botones y pasarle** a la misma distintos datos según lo necesitemos.

```
<button onclick="saludarA('Ana')"> Ana </button>
<button onclick="saludarA('Pedro')"> Pedro </button>
<button onclick="saludarA('Juan')"> Juan </button>
<button onclick="saludarA('María')"> María </button>
```


Eventos con parámetros

El código anterior, sigue intacto porque la función es una sola, no la hemos cambiado, el resultado será que por ejemplo al botón de Juan al hacerle **click nos otorga el siguiente resultado:**



De esta forma no tenemos que generar **múltiples funciones, sino una sólo y reutilizable.**

Eventos con parámetros

También con el mismo propósito podemos hacer lo siguiente:

```
<button> Ana </button>
<button> Pedro </button>
<button> Juan </button>
<button> María </button>
<script>

var boton = document.getElementsByTagName('button')

for(var i = 0; i <2; i++){

  boton[i].onclick = saludar

}

function saludar(){

  alert('hola')

}
```

Eventos con parámetros

No se preocupen, pasaremos a explicar lo que hemos realizado, vamos por partes.

Primero generamos **cuatro(4) botones**.

Como sería muy tedioso en caso de **una interfaz donde estos vayan variando**, modificar el HTML, es decir no queremos trabajar desde los botones, **lo haremos desde el código**.

En JS, estos botones son partes del DOM, y como ponerle a cada uno un ID o CLASS es poco práctico si es que estos botones serán más o menos de forma dinámica.

```
<button> Ana </button>  
<button> Pedro </button>  
<button> Juan </button>  
<button> María </button>
```

Eventos con parámetros

La mejor **forma de acceso** no utilizada aún es **document.getElementsByTagName**, esto es, tomamos a los elementos **por su nombre dentro del HTML**:

```
var boton = document.getElementsByTagName('button')
```

Es decir, me sirve porque simplifica la forma de acceso, pero ahora necesito **individualizarlos**. Es así que quiero que al **hacerle clic a cada uno**, estos hagan individualmente algo.

```
for(var i = 0; i < 2; i++){  
  
    boton[i].onclick = saludar  
  
}
```

Eventos con parámetros

Ahora que hemos entendido esto, vemos que al hacerle clic a cada botón el resultado será el siguiente:



Más allá de que hemos avanzado muchísimo, en realidad **no queremos esto**. La idea es que a su vez cada botón pase el valor de su contenido. **Lo vamos a modificar un poco para lograr nuestro cometido.**

Eventos con parámetros

```
<button> Ana </button>
<button> Pedro </button>
<button> Juan </button>
<button> María </button>
<script>

var boton = document.getElementsByTagName('button')

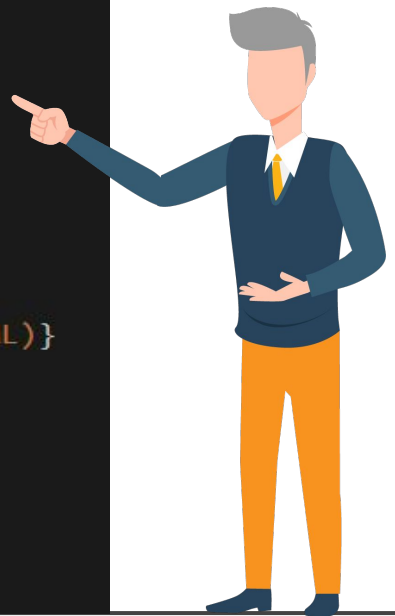
for(var i = 0; i < boton.length; i++){

  boton[i].onclick = function(){ saludarA(this.innerHTML)}
}

function saludarA(nombre){

  alert(nombre)

}
```

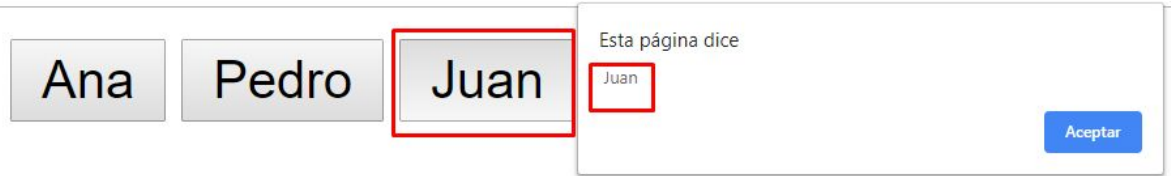


Eventos con parámetros

La idea es que **pasemos el texto contenido en cada botón**. Por esa razón lo hacemos con **innerHTML**, que de esta manera trabaja **obteniendo el texto del botón**. Como lo hacemos dentro del for, **el innerHTML será diferente según sobre qué botón hagamos clic**. Por otro lado, debemos pasar esto como **parámetro**. Por lo tanto, lo hacemos con la siguiente sintaxis:

```
function(){ saludarA(this.innerHTML)}
```

El resultado será el siguiente:



Revisión

- Repase el concepto de **Evento**.
- Investigue todos los **eventos posibles**.
- Combine **funciones con eventos**.
- Comprenda cómo es **document.getElementsByTagName**.
- Integre un **bucle para reutilizar** una función desde el código.
- Trabaje en el **Proyecto Integrador**.
- Realice las preguntas necesarias al/la docente antes de continuar.



¡Muchas gracias!

¡Sigamos trabajando!