

Introducción a la Programación Web - Maquetador Web

Clase 7

NEORIS



Repaso

¿Qué vimos en la clase anterior?

- Arrays
- Objetos
- Dom



¿Cómo se integra con HTML?

Aunque Javascript se puede **incluir directamente** en un documento **HTML**, ocurre como las hojas de estilo (**CSS**): suele ser más útil como un archivo independiente vinculado a cada documento que necesite de los comportamientos definidos en él; así sólo hay que mantener unos pocos archivos .js para actualizar los comportamiento de todo un sitio.

Para ello habría que incluir en el head del documento una línea como ésta:

```
<script type="text/javascript" src="URL_de_archivo.js"></script>
```

Eventos: EventListeners

Podemos utilizar Javascript para que realice ciertas funciones respondiendo a eventos determinados.

Algunos de los eventos más utilizados son:

- onclick
- onchange
- onload
- onkeyup
- onkeydown
- onkeypress





EventListeners - ¿Cómo los utilizamos?

Una de las formas es definir un atributo en nuestro elemento html que indica a qué evento queremos que javascript responda y qué función queremos que ejecute.

```
<button type="button" name="button" onclick="cargarDatosEnTabla()">  
  Cargar datos  
</button>
```



Eventos: EventListeners

Ejercicios:

1. Generar un elemento `<p>` que cuando lo toquemos cambie su color de fondo a rojo
2. Generar un select, y que cada vez que se seleccione alguna opción mostrarlo en un párrafo



Eventos: EventListeners

Ejercicios:

- 1-

```
<p id="prueba" onclick="cambiarColor()">Si hacemos click, cambia de color.</p>
function cambiarColor() {
    document.getElementById("prueba").style.color = "red";
}
```
- 2-

```
<select id="miSelect" onchange="cambiarTexto()">
  <option value="Audi">Audi</option>
  <option value="BMW">BMW</option>
  <option value="Mercedes">Mercedes</option>
  <option value="Volvo">Volvo</option>
</select>
<p id="prueba2"></p>
function cambiarTexto() {
    var x = document.getElementById("miSelect").value;
    document.getElementById("prueba2").innerHTML = "You selected: " + x;
}
```



Javascript - setTimeout()

El **método** setTimeout() **llama una función** o evalúa una expresión después de un número específico de **milisegundos**. La función solo **se ejecuta una vez**.

```
function miFuncion () {  
    setTimeout (otraFuncion, 3000)  
};  
  
function otraFuncion () {  
    alert ("Somos necesarios, un fragmento de la fatalidad; formamos parte  
del todo, somos en el todo; no hay nada que pueda juzgar, medir, comparar  
y condenar nuestra existencia, pues ello equivaldría a juzgar, medir,  
comparar y condenar el todo. Friedrich Nietzsche" );  
};  
  
miFuncion (); //llamamos a la función
```




Javascript - setTimeout()

Parámetros

```
setTimeout(a,b);
```

a = la función que se va a ejecutar pasado el tiempo definido.


b = el tiempo que debe “esperar” antes de ejecutar la función.

En milisegundos.

1000 ms = 1 seg

El parámetro “a” es una función previamente definida que podemos llamar o una función anónima definida dentro del mismo setTimeout

```
setTimeout(function(){ alert("Hello") }, 3000);
```



Javascript - setInterval()

El método setInterval() llama a una función o evalúa una expresión cada determinados intervalos de tiempo.

```
var myVar = setInterval(consoleLogFunc, 3000);
```

```
function consoleLogFunc() {  
    console.log("Hola!");  
};
```

Detener función

```
function detenerIntervalo() {  
    clearInterval(myVar);  
};
```



Javascript - setInterval()

Parámetros

```
setInterval(a,b);
```

a = la función que se va a ejecutar después del intervalo de tiempo definido.

b = el tiempo que debe “esperar” antes de ejecutar la función. En milisegundos.

El parámetro “a” es una función previamente definida que podemos llamar o una función anónima definida dentro del mismo setTimeout

```
setInterval(function(){ alert("Hello") }, 3000);
```



Practiquemos un poco...

- Crear un documento .html y linkearlo con un .js
- Poner dos imágenes en el documento.
- Utilizando css, superponer una imagen sobre otra.
- *Utilizando setInterval(), lograr que vayan alternando las imágenes que se visualizan.*

Resolución

html

```


<button onclick="pararIntervalo()"
class="boton1">Parar el intervalo</button>
<button onclick="iniciarIntervalo()"
class="boton2">Iniciar el intervalo</button>
```

css

```
.imagen {
  margin-top: 50px;
  width: 200px;
  height: 200px;
  position: absolute;
}
```

javascript

```
let intervalo = setInterval(ocultarFoto, 1000);
let miImagen = document.getElementById("foto1");
let miOtraImagen =
document.getElementById("foto2");
function ocultarFoto(){
  if(miOtraImagen.style.display == ""){
    miOtraImagen.style.display = "none";
  } else {
    miOtraImagen.style.display = "";
  }
}
function pararIntervalo(){
  clearInterval(intervalo);
}
function iniciarIntervalo(){
  intervalo = setInterval(ocultarFoto, 1000)
}
```



jQuery - ¿Qué es?

Según jquery.com, jQuery es una **biblioteca** de JavaScript rápida, pequeña y rica en funciones.

Permite que cosas como el recorrido y la *manipulación* de documentos HTML, el manejo de *eventos*, la *animación* y *Ajax* sean mucho más simples.



jQuery - ¿Cómo se usa?

Podemos empezar a usar jQuery en nuestra página web [descargando](#) la librería linkeandola localmente o incluyendola desde un CDN (Content Delivery Network).

Ejemplo CDN

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
</head>
```



jQuery - Sintaxis

La sintaxis de jQuery está creada para seleccionar elementos HTML y realizar acciones con ellos.

La sintaxis básica es:

```
$(selector).acción()
```

Ejemplos:

```
$('p').hide() //Ocultar todos los elementos <p>
```

```
$('.test').hide() //Ocultar todos los elementos con la clase 'test'
```

```
$('#test').hide() //Ocultar el elemento con id igual a 'test'
```




Manipulación de elementos: Selectores

Sintaxis	Descripción	Sintaxis	Descripción
\$("#*")	Todos los elementos	\$(this)	El elemento actual
\$("#p.intro")	Todos los <p> con la clase 'intro'	\$("#p:first")	El primer elemento <p>
\$("#ul li:first")	El primer del primer 	\$("#tr:even")	Todos los elementos <tr> par
\$("#tr:odd")	Todos los elementos <tr> impares	Etc	Etc



jQuery - Manipulación de elementos

Generalmente se escribe el código jQuery dentro de un evento document ready.

```
$(document).ready(function() {  
    // jQuery aquí...  
});
```

Esto se hace para **prevenir** que el código jQuery corra antes de que el documento esté **completamente cargado**.

Algunos ejemplos de acciones que pueden fallar si los métodos corren antes de que la página esté completamente cargada:

- Esconder un elemento que no ha sido creado aún.
- Obtener el tamaño de una imagen que aún no cargó.

Esto también permite colocar nuestros `<script>` en el `<head>` de la página. (Ya que nuestro código se va a ejecutar una vez que el documento esté cargado).



jQuery - Manipulación de elementos

Algunas funciones que nos brinda jQuery para manipular elementos:

```
.width()      .height()      .addClass()  
.attr()       .remove()       .append()
```

Muchas de estos métodos sirven tanto para *obtener* como para *definir* propiedades.

Por ejemplo:

```
$("#logo").width()      //Nos devuelve el ancho de el elemento con id = "logo"  
$("#logo").width(600)   //Setea el ancho del elemento "logo" a 600px;
```

Más funciones para manipular elementos en [la página oficial de jQuery](#).

jQuery - Manejo de eventos

Así cómo jQuery nos ofrece funciones para solucionar de forma más rápida y sencilla la manipulación de elementos, también nos brinda métodos para el manejo de eventos.

Algunos de ellos son:

```
.click()      -> Se ejecuta al hacer click  
.change()    -> Se ejecuta al cambiar una opcion  
.dblclick()  -> Se ejecuta al hacer doble click  
.focus()    -> Pone el foco en un elemento  
.focusout() -> Le saca el foco al elemento  
.hover()     -> Cuando pasas por un elemento lo atenúa  
..on("evento", función(){}))
```



Para ver la lista completa de eventos que podemos manejar con jQuery, podemos acceder a api.jquery.com/category/events/



¿Cómo usamos los métodos del slide anterior?

```
$(document).ready(function(){  
    $("#achicarImagen").click(function(){  
        var anchoImg = $("#nuestraImagen").width();  
        if (anchoImg >= 1200) {  
            $("#nuestraImagen").width(300);  
        };  
    });  
});
```



jQuery - Animaciones y otras funciones

<u>hide()</u>	Hides the selected elements
<u>queue()</u>	Shows the queued functions on the selected elements
<u>show()</u>	Shows the selected elements
<u>slideDown()</u>	Slides-down (shows) the selected elements
<u>slideToggle()</u>	Toggles between the slideUp() and slideDown() methods
<u>slideUp()</u>	Slides-up (hides) the selected elements
<u>stop()</u>	Stops the currently running animation for the selected elements
<u>toggle()</u>	Toggles between the hide() and show() methods



Ejercicio: Validar Formularios con jQuery

Usando jQuery, y alguno de los métodos para manejar eventos mencionados anteriormente, crear un formulario que pida ciertos datos personales y, si el usuario no completa los campos, que se muestre debajo del input correspondiente una leyenda en rojo que diga 'Completar este campo'.