

# Apuntes de la asignatura DAT

## Test diario

Test diario Miguel

~morduno/test

## Comandos para formatear el markdown y ayuda

Pagina para mirar como usar markdown Guia markdown

Pagina para validar los html Validador HTML

pandoc es pasar de un tipo de fichero a otro con variedades

```
pandoc -s ejemplo.md -o ejemplo.html //formato
```

formatear el texto para dejarlo bien

```
pandoc -s $1 -o ${tmp_name}
```

## Clase 4 de febrero

### Pregunta de clase

En HTML

- a. Un elemento siempre tiene etiqueta de apertura. Casi siempre (no siempre) tiene, además, contenido y cierre
- b. Un elemento está formado por etiqueta de apertura, contenido y etiqueta de cierre
- c. Una etiqueta contiene elemento de apertura y elemento de cierre.
- d. Una etiqueta se puede omitir, casi siempre.

### Respuesta correcta a

## Clase de 10 de febrero

- Esta mal meter css en html -> Penaliza
- Elementos bloques -> Tiene márgenes al redor (imagino que habla del div)
- Elementos inline -> Enlace, imagen, etiqueta
- Dentro de <p> no esta permitido ningun elemento de bloque, h1, table, form, etc
- Si permitidos los inline ( span, a em, label, img )

- Si a continuación de un párrafo hay un elemento no permitido, considera el párrafo como cerrado y luego nos queda un párrafo huérfano y no marca un error
- La etiqueta `<em>` define el elemento emphasized -> No da garantía de cómo va a salir en el navegador
- La etiqueta `<pre>` Muestra los espacios de línea respetando los espacios entre palabras y los saltos de línea
- Las etiquetas `<h1>` `<h2>` ... `<h6>` -> Títulos de mayor a menor
- La etiqueta `<a>` -> Para crear enlaces, no confundir con `<link>` (Historicamente viene de anchor, pero apunta (en teoría) a sitios en el documento. Para ello hay que poner un `#dirección`, pero puedo poner a una web con la url.
  - Para hacerlo bien
 

```
<h1 [......] <a href="#punto07">Volver al punto 7</a>
```
  - Añadiendo el atributo `target = "_blank"` abre una nueva pestaña
 

```
<a href="http://www.urjc.es" target="_blank">Abrir en nueva pestaña</a>
```
- La etiqueta `<div>` divide una división o sección dentro del documento
  - Hay varias cosas añadidas, pero vamos, team div
- La etiqueta `<span>` define una división o sección dentro del documento
- La etiqueta `<table>` crea una tabla
  - `<th>` Table header
  - `<tr>` Table row
  - `<td>` Table data
- La etiqueta `<ol>` -> Crea listas numeradas `<ul>` -> Crea listas no numeradas
  - `<li>` Lista item
- La etiqueta `<img>` -> Permite insertar imágenes, obligatorio poner el alt
 

```


```
- La etiqueta `<form>` Permite crear un formulario  
Ejemplos de formularios por miguel orduño

### **Pregunta de clase**

¿Cuál de estas respuestas es FALSA?

- a. Los documentos HTML 5 normalmente emplean codificación UTF-8. Pero también pueden emplear codificaciones más antiguas, con tal de que está correctamente indicado con el atributo charset del elemento meta.
- b. El elemento link sirve para hacer hiperenlaces HTML, que son la base del web al permitir al usuario navegar de página a página con un solo clic.
- c. El W3C es el organismo internacional que se encarga de normalizar los protocolos del web.
- d. El código HTML normalmente se escribe en minúsculas. Pero HTML es insensible a mayúsculas, así que puede escribirse en mayúsculas, no hay diferencia.

**Respuesta correcta b**

### **Clase 11 de febrero**

#### **Pregunta de clase**

¿Cuál es la respuesta correcta?

- a. En un atributo 'input' normalmente hay un elemento 'id' y un elemento 'name'. Es preferible que su valor sea el mismo, pero no tiene por qué serlo. El primero lo usa el cliente, el segundo, el servidor.
- b. En un atributo 'input' normalmente hay un elemento 'id' y un elemento 'name'. Es preferible que su valor sea el mismo, pero no tiene por qué serlo. El primero lo usa el servidor, el segundo, el cliente.
- c. En un elemento 'input' normalmente hay un atributo 'id' y un atributo 'name'. Es preferible que su valor sea el mismo, pero no tiene por qué serlo. El primero lo usa el servidor, el segundo, el cliente.
- d. En un elemento 'input' normalmente hay un atributo 'id' y un atributo 'name'. Es preferible que su valor sea el mismo, pero no tiene por qué serlo. El primero lo usa el cliente, el segundo, el servidor. Elige una opción (a,b,c,d)

**La respuesta correcta es la d** el input es un elemento y el id lo manejar el server

### **Clase 17 de feberero**

Supuestamente es la clase más difícil del año

- Ya no va dentro de los atributos, eso es del HTML4. En la norma del 5 esta un fallo.
- Se puede meter en HTML con las cabeceras style, pero la idea es tenerlo en fichero aparte

- Se enlaza el css con el elemento `link`
- Como HTML es inscosistente, CSS y javascript se insertan de forma distintan
- Selector -> Elementos Atributo -> Propiedad Valor -> Valor ejemplo `h1 { color: black;}`
- Hay herencias por elementos

- 
- \* -> *TODOS LOS ELEMENTOS*
  - Etiqueta(`h1,h2...`) -> *Todos los elementos de esa etiqueta*
  - , -> *ES UN OR*
  - (espacio) -> *El siguiente es descendiente del anterior* Ejemplos diapo 17
  - . -> *Para las clases* Se pueden concatenar y solo se aplica a los concatenados
  - # -> *Para los IDs*
- 

En caso de colision de estilos (no debería pasar si esta bien diseñado) prevalece la que sea más compleja/larga/específica.

### Pregunta de clase

- Las reglas CSS se pueden escribir como contenido de un elemento 'link'.
- Las reglas CSS se pueden escribir como contenido de un elemento 'style'.
- La misma regla CSS se puede aplicar a varios elementos. Un mismo elemento puede recibir el efecto de varias reglas.
- En una regla CSS el selector indica a qué elementos se aplica la regla, esto es, 'para estos elementos, este atributo tiene tal valor'.

Elige una opción (a,b,c,d)

**La respuesta correcta es la a** link tiene un atributo que indica donde esta el css

### Clase 18 de Febrero

No tienen que ser paginas muy bonitas, no somos diseñadores

No confundir los colores css con con los colores panthone ni con RAL

Conviene decidir usar una paleta

Margin con los bloques entre dos, no los suma, toma el mayor

**Display None** reposiciona **Display hidden** solo lo oculta

### **Pregunta de clase**

¿Cuál es la respuesta correcta?

- a. El número de colores que podemos usar en CSS es 140.
- b. En CSS podemos usar unos 16 millones de colores, aproximadamente.
- c. El número de colores que podemos usar en CSS es el establecido por el estandar Pantone. Son dos mil y pico, y van aumentando poco a poco con los años.
- d. El número de colores que podemos usar en CSS es el establecido por el estandar Pantone: unos 16 millones de colores, aproximadamente. Elige una opción (a,b,c,d)

**Respuesta correcta la b**

### **Clase 24 de Febrero**

Bootstrap es un framework libre para web, desarrollado por ingenieros de Twitter. Incluye plantillas HTML y CSS y más cosas. Es mobile first, osea centrado que se vea bien en el movil. Orientado a programadores noa diseñadores Alternativas como Foundation

- Ventajas:
  - Sencillo y rapido
  - Responsive
  - Diseño consistente
  - Compatibilidad con navegadores modernos

Todo el contenido debe estar en un container

El viewport es la parte que estoy viendo yo de un navegador Barra de desplazamiento horizontal es caca, la vertical guay

- Tecnicas de adaptacion
  - Tecnica inicial. El viewport es la parte que estoy viendo yo de un navegador. Barra de desplazamiento horizontal es caca, la vertical guay
  - Segunda Generación. El virwport virtual. Sobre una pantalla imaginaria normal se hace el render, se coloca ahi, y si la pantalla del movil es más pequeña cuando el usuario arrastra el dedo se mueve por la ventana
  - Diseño responsive. Basado en grid(regilla). El contenido se coloca en cajas. La pagina se adapta al tamaño de la ventana. En bootstrap se organiza en 12 cuadrículas(aunque tampoco es obligatorio), y se representan en distinta forma si es una patalla de tablet(horizontal) o de movil(vertical)

Diposicion Normal -> Cuando las cajas se ven una misma fila Diposicion Apilada  
-> Unas encima de otras porque no entran en la misma fila

Cada caja sera un div que tiene una clase. Ejemplo

```
<div class="col-md-4"> Bla, bla </div>  
<div class="col-md-8"> Bla, bla, blá </div>  
</div> ````
```

Pagina para ver pruebas

El div con la class container es obligatorio, esta prohibido meter un container dentro de otro. El validaor online no lo va a detectar pero esta mal

Hay dos tipos, el normal y el fluido, el normal usa espacios en blanco y el fluido usa todo el espacio y se maximiza

Ejemplos:

Container normal

Container fluido

### **Pregunta de clase**

¿Qué respuest es FALSA?

El viewport virtual es

- Una técnica de segunda generación para mostrar páginas web en dispositivos móviles.
- Una técnica obsoleta. Aunque sigue funcionando.
- Una pantalla imaginaria donde se compone una página web. Luego, en la pantalla de un dispositivo móvil, se muestra esta pantalla virtual, reducida.
- Una pantalla imaginaria donde se compone una página web. Luego, en la pantalla de un dispositivo móvil, se muestra una zona de esta pantalla virtual. Elige una opción (a,b,c,d)

**Respuesta correcta C**

### **Clase 25 de Febrero**

FreeFileSync es una aplicacion para sincronizar ficheros pero yo prefiero usar gitlab

En bootsrap la gama concreta de colores se decidira en CSS, y normalmente ya lo decidio github, yo solo le indico que tipo de asignacion le quiero dar

Se pueden hacer botones con button, un inputo con button y un a con button, se puede desabilitar un boton con el atributo **disable**

ejemplos botones

Se pueden modificar las imágenes con diferentes clases, como `rounded`, `rounded-circle` etc.. Vienen en las diapositivas las que son, también alinear etc

ejemplos imágenes

Las tablas tienen clases para dejarlas bonitas, mejor mirar los ejemplos

ejemplos tablas

Una tarjeta (card) es una caja redondeada dividida en cabecera, cuerpo y pie.

### **Pregunta de clase**

¿Cuál es la respuesta correcta?

- a. Una tabla en Bootstrap es una tabla HTML a la que le añadimos la clase 'table'.
- b. Una tabla en Bootstrap es una tabla HTML a la que le añadimos la clase 'table' y la clase 'table-striped'.
- c. Una tabla en Bootstrap es un elemento HTML 'bootstrap-table'.
- d. Cualquier tabla dentro de un 'container' de Bootstrap tendrá el formato indicado por Bootstrap.

### **Respuesta correcta A**

## **Clase 3 de Marzo**

En Bootstrap los formularios están obligados al uso de `<label>` y se le añade la clase `class="form-label"`

Los carruseles son fotos que van cambiando con el tiempo. Recordar que las fotos que usamos tienen que ser "legacy" o sea que te permitan los derechos de autor.

Bootstrap se va a encargar del tamaño de la foto. Nosotros tenemos que garantizar que la proporción sea la misma.

Usar mejor el ejemplo de clase

La clase `disabled` no deshabilita realmente, lo que lo hace es el atributo

```
<input type="text" name="lname" disabled><br>  
<button type="button" class="btn btn-lg" disabled>Botón</button>
```

Si vamos a usar Bootstrap no toquemos CSS a no ser que estemos muy seguros.

Plantillas para Bootstrap. Hay que tener cuidado porque suelen ser más avanzadas y pueden estar desfasadas, ir mal etc etc

### Pregunta de clase

- a. Podemos poner fotos de cualquier tamaño, que Bootstrap se encargará de adaptarlas para que se vean bien.
- b. Todas las fotos deberían tener mismo tamaño y relación de aspecto, de lo contrario el resultado será bastante pobre.
- c. Todas las fotos tienen que tener mismo tamaño y relación de aspecto, de lo contrario Bootstrap mostrará un error.
- d. Todas las fotos tienen que tener la misma relación de aspecto, eso Bootstrap no lo puede corregir. Pero si los tamaños cambian, Bootstrap los puede homogeneizar.

### Respuesta correcta D

---

## Apartir de aquí empezamos JavaScript

### Clase 3 de Marzo

El profe dice que es un lenguaje regular, hecho de prisa en 10 días y demás, algo ha mejorado con el tiempo pero sigue siendo bastante MEH.

Vamos a usar `node` en la terminal para programar.

JavaScript6 hace cambios muy relevantes, corrigiendo barbaridades.

### Clase 4 de Marzo

Los números en JavaScript son una basura, a veces falla y a veces acierta comparando los números. Hay que usar `epsilon` para comparar los números porque el comportamiento es aleatorio y a veces falla y a veces acierta si no.

Es un lenguaje que está pensando para gente que no es programadora profesional, por eso hay muchos programas de mierda.

El lenguaje no especifica si es interpretado o compilado, depende de la interpretación.

Igual que con el resto de cosas, la mejor idea es meter todos los `js` en un subdirectorio igual que las fotos, que es lo que vamos a hacer en clase. No usar los `path` absolutos lógicamente.

El `hola mundo` en `js` no aparece en el `html`, si no en la consola de `debug`.

En teoría pone solo el `;` pero es mejor ponerlo para evitar errores a futuro, aunque en principio se puede omitir.

Todo el código que se haga hay que usar el modo `strict`.

`use strict`

Normas de este modo son:

- Obligado a declarar todas las variables
- Solo un nivel de anidamiento
- No se puede repetir el nombre un parametro en la misma funcion
- El intento de modificar propiedades inmutables genera una excepcion
- No se permite el uso de with

Hay dos tipos de valores

- Primitive values: boolean, number, string, null, undefined
- Objetos: plain objects, arrays, regexp

Existe el concepto del numero **infinity**.

Paradojas extrañas, NaN se supone que no es un numero, y si haces type NaN devuelve que es un numero

null es un objeto que no tiene valor, se podria decir que es un objeto vacio, aunque eso realmente es {}

Otra tonteria, es que si estoy en estricto tengo que declarar las variables, pero si la uso y despues la declaro, funciona Y dice que es undefined

En las funciones si no metes el parametro, tampoco se queja

Si has creado un objeto vacio, asignamos cosas, y luego llamamos a algo que no existe, tambien devuelve el undefined

Una posible solucion, en vez de usar **var**, es usar **let**, y asi lo haremos en la asignatura, asi por lo menos da error

`Number()` -> Pasa una cadena a un numero

`String()` -> Pasa un numero a un string

`NaN === NaN` devuelve False. Mejor sin comentarios

Para detectar si es un NaN hay que usar `isNaN`, pero cuidado porque tambien devuelve True con otras cosas, es absurdo

La cosa es tener cuidado con el NaN, no van a ser preguntas del test porque si no, nos destruye la vida.

Redondeo -> `Math.round()` Truncado -> `Math.trunc()`

Da igual que escribamos las cosas en español, no es necesario hacerlo en ingles. Aunque deje usar tildes y Ñs, mejor evitarlo

Usamos el tripe `===` porque el doble `==` funciona raro.

### **Pregunta de clase**

¿JavaScript es un lenguaje compilado?

- a. Sí.
- b. No.
- c. Antes no, ahora sí.
- d. Depende.

**Respuesta Correcta D** Es una frontera difusa, en este caso es la D, porque en la actualidad todos hacen cosas de compilación y interpretación

## Clase 10 de marzo

JavaScript es un lenguaje de primera clase, además de los tipos visto, también puede ser otra función

Funciones anónimas, una función que no tiene nombre. A Miguel no le gustan nada,

Se pueden crear usando =>. Una función flecha es una forma más abreviada de escribir una función anónima. También se le puede quitar función. La idea es ahorrar tiempo al teclear.

Se le pueden pasar funciones a funciones y es lo ideal

Existen las funciones normales como todas

Se pueden usar antes de declarar (jaja mejor no hacerlo nunca)

No existe paso por referencia en JavaScript, hay una chapuza para “hacerlo”, es meter el valor dentro de un array.

Las funciones siempre devuelven un valor, un undefined

Cuidado con el número de parámetros como vimos el otro día porque JavaScript se lo come y devolverá cosas distintas en vez de dar errores. Por ejemplo

```
'use strict'  
function f(x,y){  
  return x+y;  
};  
console.log(f(1)); // NaN  
console.log(f(2,2)); // 4  
console.log(f(1,1,1)); // 2
```

En `arguments.length` vienen la cantidad de argumentos, y tendremos que comprobar nosotros a mano si están los argumentos que queremos y ejecutar errores en cuyo caso

Parámetros por omisión habrá que meterlos a mano también

```
'use strict'  
function f(x){  
  if (x===undefined) {
```

```

x=0};
return x + 1 ;
};
console.log(f()); //1

```

En javascript6, hasta hace mucho no se podía usar, aunque en teoría **Usaremos esta**

```

'use strict'
function f(x=10){
return x;
}
console.log(f(5)); // 5
console.log(f()); // 10
console.log(f(false)); // false

```

En javascript incrustado en HTML, todos los scripts incluidos en HTML comparten la variable global (porque HTML comparte la variable window)

**USAR SIEMPRE let.** No usar **var** porque puede dar problemas. Se pueden declarar variables “ultra-locales” metiéndola dentro de un `{}` que este dentro de un propio bloque.

```

'use strict'
function f() {
  var x = 1;
  if (true) {
    var x = 2; // La misma variable. Destruimos valor previo
    console.log(x); // 2
  }
  console.log(x); // 2
}
function g() {
  let x = 1;
  if (true) {
    let x = 2; // Variable diferente, local del bloque
    console.log(x); // 2
  }
  console.log(x); // 1. Mantiene el valor previo
}
f();
g();

```

### Pregunta de clase

¿Cuál de estas afirmaciones es FALSA?

a. Las funciones flecha y las funciones anónimas de JavaScript solo se pueden usar cuando una función es un argumento de otra función.

- b. Las funciones flecha y las funciones anónimas se pueden usar siempre, pero solo tienen alguna ventaja cuando una función es un argumento de otra función.
- c. Las funciones flecha también son funciones anónimas, pero con una sintaxis especial, más compacta.
- d. Las funciones flecha y las funciones anónimas se usan mucho en JavaScript, aunque sus ventajas son discutibles.

**Respuesta Correcta B** Esque solo puedo usarlas en ese caso. Habria que explicarlo un poco mejor

## Clase 11 de marzo

Usar const para no poder modificar una variable

A Miguel no le gustan las funciones muy largas, ni ifs dentro de otros muy anidados

El switch salta al siguiente si no hay break, hay gente que lo usa para un fall through. Si lo hacemos comentar que no se olvido break

El while igual que el de C

Los bucles son iguales que los de C, se pueden recorrer los strings como un array de chars. Curiosamente tambien se puede recorrer como en python de esta forma

```
'use strict'
let x="Lorem Ipsum";
for (let c of x){
console.log(c);
};
```

En javascript length es una propiedad `longtud_cadena = x.length`

Cuidado porque si se hace por ejemplo un `0<undefined` es falso y continua con el bucle porque patata en vez de dar un error

Manipulacion de cadenas

Diferenciar entre constructivo o destructivo, algunas borran la original y la modifican y otros no

- `toUpperCase()` - `toLowerCase()` -> Cadenas de mayusculas o minúsculas
- `trim()` -> Se carga los espacios a derecga y izquierda
- `indexOf()` -> Devuelve la posición de la primera aparición de una subcadena. O el valor -1 si no está incluida
- `lastIndexOf()` -> devuelve la última aparición de una subcadena. O el valor -1 si no está incluida

- `slice(x,y)` -> Devuelve la subcadena comprendida entre la posición `x` (incluida) y la `y` (excluida)
  - Si `x` o `y` exceden las dimensiones de la cadena, no es problema
  - Si `x` es mayor o igual que `y`, devuelve la cadena vacía
- `split(c)` -> trocea una cadena, usando el caracter `c` como separador. Devuelve un array
- `replace(x,y)` -> devuelve una cadena donde la subcadena `x` ha sido reemplazada por `y`

Listas (Arrays), son dinámicas, no hay que fijar su tamaño y los elementos no tienen porque ser del mismo tipo

```
'use strict'
let a,b,c,d;
a=[ ]; // array vacío
b=[7, 8, 9]; // array con números
c=['rojo', 3, 0]; // array con elementos heterogéneos
d=[ [0, 1], [1, 1]]; // array con arrays anidados
```

También hay propiedades para los arrays, igual más o menos que los de los strings, mejor mirar en las diapositivas, pero son como en python más o menos

Para recorrer arrays mejor usar `forEach` o `for-of`. aunque también se puede hacer como toda la vida de dios.

```
'use strict'
let l=["a",,"c"]
l.forEach(function(x){
  console.log(x);
});
// a c
```

Lo mejor según Miguel usar esta.

```
'use strict'
let l=["a",,"c"]
for(let x of l){
  console.log(x);
}
// a undefined c
```

Hay una cosa que es `for-in` pero es rarísimo, mejor no usar nunca

### Pregunta de clase

- `len(s)` sería una función.
- `s.len()` sería un método. Es como se hace en JavaScript.

- c) s.lengh() sería un método. Es como se hace en JavaScript.
- d) s.lengh sería una propiedad. Es como se hace en JavaScript.

Elige una opción (a,b,c,d)

**Opcion correcta la A** Pregunta cabrona porque la D le falta la t, la correcta es la a.

## Clase 17 de Marzo

Los diccionarios de otros lenguajes en js es un “plain object” Como todo diccionario tiene una clave y un valor.

```
use strict'
let x={
  unidades:2,
  color:'verde',
  tamaño:'grande',
};
console.log(x); // { unidades: 2, color: 'verde', 'tamaño': 'grande' }
console.log(x.unidades); // 2
console.log(x.precio); //undefined
```

Object.keys(MiObjeto) devuelve todas las claves de un diccionario

Array.isArray(MiArray) devuelve si es un array -> Esto viene porque js no sabe distinguir si es un array o un Objeto (ES NECESARIO E IMPORTANTE PARA MIGUEL)

Como en js solo se puede devolver una cosa por return, podemos devolverlo con un plain object

Excepciones, importante, esto no es realista para el ejemplo, pero sirve para ver la sintaxis.

```
'use strict'
// Atención, ejemplo no realista
try {
  throw 'xxx27';
} catch (e) {
  console.log('capturada excepción ' + e);
  // capturada excepción xxx27
}
```

```
'use strict'
// Atención, ejemplo no realista
try{
  console.log( no_definido);
} catch (e) {
```

```
console.log("capturada excepción ",e.name,e.message);
}
//capturada excepción ReferenceError no_definido is not defined
```

La idea es detectar el tipo de error que puede suceder para controlarlo

Hay 7 nombres de error

- Error
  - Unspecified Error
- EvalError
  - An error has occurred in the eval() function
- RangeError
  - A number out of range has occurred
- ReferenceError
  - An illegal reference has occurred
- SyntaxError
  - A syntax error has occurred
- TypeError
  - A type error has occurred
- URIError
  - An error in encodeURI() has occurred

Una excepcion deberia ser para un error especial, no un error del usuario

### Pregunta de clase

¿Cuál de estas declaraciones es errónea?

- a) let a = { "uno":1, "dos":2 }
- b) let b = { "uno":1, "dos":2 };
- c) let c = { "uno":"1"; "dos":"2" };
- d) let d = { "uno":["1"], "dos":["2"] };

**Opcion correcta la C**

### Clase 18 de Marzo

Todos los modulos en el navegador tiene que usar .js En node todos los modulos tiejn que usar .mjs

Para solucionrlo usamamos

- Enlaces simbolico en consola -> Se da en C pero no me acuerdo

```
ln -s fichero1.txt fichero2.txt
```

```
n -s modulo.mjs modulo.js
```

```
ln -s programa.mjs programa.js
```

Si queremos exportar una función usamos export

```
export function nombre(){ return }
```

Para importar

```
'use strict'  
import * as modulo from './modulo';  
console.log(modulo.f());  
console.log(modulo.g());
```

Uso en el navegador, en el elemento script añadimos el atributo type=module

```
!DOCTYPE html>  
<html lang="es-ES">  
<head>  
<meta charset="utf-8">  
<title>Probando modulos</title>  
</head>  
<body>  
<script type="module" src="programa.js"> </script>  
</body>  
</html>
```

Cuando tiene modulos en HTML no se puede leer desde el disco por motivos de seguridad -> Me suena que lo di en bachillearto

Hay que lanzarlo en servidor -> El que queramos, si no nginx, o incluso en vscode o incluso tambien en python con python3 server.html

TypeScript es un lenguaje alternativo a javascript de alto nivel, es de alto nivel, diseñado por Microsoft para hacer aplicaciones Web e el cliete y en el servidor

Lo que se hace es generar javascript apartit de typescript y así nos aseguramos que funciona en todas partes, es lo que se usa en el mundo real para hacer cosas en condiciones

### Pregunta de clase

¿Qué afirmación es FALSA?

- a) TypeScript es un lenguaje diferente de JavaScript
- b) Todo programa en JavaScript es compatible con TypeScript
- c) Todo programa en TypeScript es compatible con JavaScript
- d) Todo programa en TypeScript se puede convertir en JavaScript Elige una opción (a,b,c,d)

**Respuesta correcta C** Es porque TypeScript es JavaScript con cosas, entonces si intento “compilar” de TypeScript hacia JavaScript hay cosas que alomejor no se pueden hacer.

## Clase 24 de Marzo

Numeros aleatorios

```
/ Math.random() devuelve un número real
// aleatorio entre 0 (inc) y 1 (excl)
console.log(Math.random());
let k = 10;
let x;
//Número entero aleatorio entre 0(inc) y k (excl)
x = Math.trunc(k * Math.random()); //Cota superior K no incluido
console.log(x);
```

Fechas Internarmente ZULU pero realmente local, un poco extraño

```
'use strict'
// Si construimos un objeto Date sin pasar argumentos,
// obtenemos la fecha actual
let d1 = new Date();
console.log(d1); // 2026-03-17T16:01:09.858Z
// Hacemos un cálculo matemático cualquiera, complejo,
// para tardar cierto tiempo (en js no hay nada equivalente
// a sleep)
for (let i = 0; i<100000000; ++i){
  let x = Math.random();
  x**2.50;
}
let d2 = new Date();
console.log(d2); // 2026-03-17T16:01:10.687Z
// Tiempo transcurrido
console.log(d2-d1); // 829 (milisegundos)
```

Para que sea UTC

```
let intervalo3 = Date.UTC(2018, 8, 3, 9, 0, 0);
```

```
let d3 = new Date(intervalo3)
```

Resta de dos Dates es un numero (intervarlo), peor no puedo sumarle a un date un intervalo porque se jode.

### Pregunta de clase

¿Qué afirmación es FALSA?

- a) Un objeto Date siempre contiene la hora local.
- b) Un objeto Date siempre contiene la hora UTC.
- c) Un objeto Date siempre contiene la hora Zulu.

d) El constructor del objeto Date siempre entiende la hora como local.

**Respuesta correcta A** Tener en cuenta que puedes tener muchos usuarios y no tienen porque tener la misma hora todos. Ejemplo España y Canarias

## Clase 25 de Marzo

DOM -> Document object Model, es una API que permite procesar una página HTML usando JavaScript.

Cuando el navegador carga un HTML la procesa para convertirla en DM y de ahí se representa

El navegador cuando se lo entrega al usuario deja de utilizar el HTML

Antes se usaba jQuery pero se dejó de usar porque JavaScript vanilla se mejoró.

Eventos

En el contexto del desarrollo de software, un evento es una acción que sucede y que ha de ser tratada por el programa, por ejemplo eventos de ratón, pantalla táctil etc. . . A las respuestas se les llama **manejadores**. Tenemos que escribir las funciones manejador

En el DOM hay dos objetos globales muy importantes “Window” “document”, están predefinidos y están presentes en cualquier documento

Hay muchos eventos en JavaScript, trabajaremos principalmente con el click. Necesitamos saber el nombre del evento, donde pasa (event target), como responder (event handler) y más detalles (event object)

En JavaScript contemporáneo -> document.querySelector() su argumento será un selector CSS que especifique el elemento

Invocamos al método addEventListener de este objeto pasando como parámetros:

- El event type
- El manejador

Esto así mismo puede disparar varios elementos y basta solo con llamar a addEventListener

**MUY IMPORTANTE**, vamos a trabajar bajo esto a partir de ahora, saberlo bien

Esto es un ejemplo de Miguel de Hola mundo

En JavaScript antiguo existían elementos como métodos, no hay que usarlos. Tenemos que usar document.querySelector() y document.querySelectorAll()

## Pregunta de clase

El esquema básico de una aplicación en el DOM es

- a) Escribimos una función manejador, seleccionamos un evento (preferentemente por el ID del evento) e invocamos al manejador, pasándole el evento.
- b) Escribimos una función manejador, seleccionamos un elemento y le asignamos el manejador, indicando siempre el nombre del evento que disparará ese manejador.
- c) Preparamos un evento, seleccionamos un atributo del evento y vinculamos ese atributo con un atributo concreto de un elemento concreto del HTML.
- d) Escribimos una función manejador, y le pasamos como argumentos el id del objeto al que irá asociado y el nombre del evento.

**Respuesta correcta B** La A esta mal porque los eventos tienen nombre no ID y nunca invocamos al manejador

## Clase 7 de abril

El manejador no puede recibir parámetros, lo más cómodo es usar una variable global

Los eventos se propagan hacia atrás a los elementos antecesores.

En código, recetas y libros antiguos podremos encontrar selección de elementos con métodos como `document.getElementById` `document.getElementsByName` `document.getElementsByTagName` `document.getElementsByClassName` `document.images` `document.forms` `document.links` ... etc

Incluso ChatGPT, Deepseek, Gemini y Claude (en 2025) usan estas funciones anticuadas. Debemos evitarlas, en favor de **`document.querySelector()`** y **`document.querySelectorAll()`**

Trabajar siempre con trazas y consola de depuración para detectar errores

`.textContent` representa el contenido de texto de un nodo y sus descendientes.

## Pregunta de clase

Para escribir texto dentro de un elemento HTML mediante el DOM

- a. Empleamos el método `textContent`, al que pasamos un selector (para especificar dónde escribimos) y la cadena de texto a escribir.
- b. Empleamos el método `textContext`, al que pasamos un selector (para especificar dónde escribimos) y la cadena de texto a escribir.
- c. Primero seleccionamos el elemento con `getElement()`, luego, modificamos la propiedad `textContent` del elemento.
- d. Primero seleccionamos el elemento con `querySelector()`, luego, modificamos la propiedad `textContent` del elemento.

Elige una opción (a,b,c,d)

**Respuesta correcta D** básicamente es porque eso es lo que hacemos.

## Clase 8 de abril

Crear un elemento dinámico en el DOM hay que usar `document.createElement()`. Recibe como argumento una cadena con el nombre de elemento HTML (etiqueta). Devuelve un elemento de ese tipo

Con el método `append` de un elemento añadimos otros elementos en su interior

Ejemplo crear p

- `textContent` -> Macha lo que hubiera anteriormente **Es una propiedad**
- `append` -> Añade a lo que hubiera anteriormente **Es un método**

Una vez seleccionado un elemento o elementos mediante un selector, insertarlo en diferentes posiciones

- `append()` Inserta contenido al final de la selección, dentro de la selección
- `prepend()` Inserta contenido al principio de la selección, dentro de la selección
- `after()` Inserta contenido inmediatamente después de la selección, fuera de la selección
- `before()` Inserta contenido inmediatamente antes de la selección, fuera de la selección

También cambiarlo o eliminarlo

- `replacewith()` Reemplaza la selección
- `remove()` Borra la selección

Si se inspecciona el elemento, cambia el DOM nunca el HTML original.

## Pregunta de clase

Crear un elemento dinámicamente mediante el DOM significa

- a. Que el elemento lo creamos de la manera más rápida posible.
- b. Que el elemento no está originalmente en el documento HTML, lo creamos en JavaScript y lo insertamos en el HTML.
- c. Que el elemento no está originalmente en el documento HTML, lo creamos en JavaScript y lo insertamos en el DOM (aunque esto no modifica el HTML).
- d. Que el elemento lo creamos editando el código HTML en el servidor.

## Respuesta correcta C

## Clase 14 de abril

Vamos a ver con un manejador como manejar varios elementos

Como podemos por ejemplo modificar el css, podemos hacer cosas como ocultar imagenes, por ejemplo modificando su clase y ponerle que sea oculta

```
<div id="marco_foto" class="oculto">

</div>
```

Y modificamos el DOM asi para que sea visible

```
<div id="marco_foto">

</div>
```

Basicamente lo que heos hecho ha sido quitarle la clase oculto. Ejemplo

For no se puede usar ni tampoco class porque se confunde. Por eso por ejemplo para usar class hay que usar className

```
parrafo1.className = "noticia oculto deportes"
```

Pero es mas comodo usar Class.List.add o Class.List.remove

```
marco.classList.add("oculto");
marco.classList.remove("oculto");
```

### Pregunta del dia

- Un selector css que devuelve el elemento con ese identificador.
- Una propiedad del objeto con los detalles de un evento, que permite conocer el identificador del elemento que ha recibido el evento.
- Una propiedad del manejador de un evento, que permite conocer el identificador del elemento que ha recibido el evento.
- Un método que devuelve el identificador del elemento actual.

**Respuesta correcta B** La c no es porque los manejadores no tienen propiedades (son como las funciones, tienen parámetros)

## Clase 15 de abril

Eventos de raton

- mouseover -> Se recibe cuando el ratón se coloca sobre el elemento
- mouseout -> Se recibe cuando el ratón abandona el elemento o cualquiera de sus descendentes

mouseleave -> Se recibe cuando el ratón abandona el elemento seleccionado

Esto es un ejemplo Miguel no le gusta porque no estan las cosas separadas y esta todo en la interfez de usuario. Esto puede dar muchos problemas a futuro porque estas atado a esta interfaz.

Usar para editar las clases.

Otro esquema que se usa mucho es

Como se puede ver en este ejemplo, seleccionamos el DIV general y luego interactuamos con nada uno individualmente

### **Pregunta del dia**

Una de las siguientes afirmaciones falsa ¿cuál?

- a. Cuando un elemento recibe un evento, todos sus descendientes (hijos, nietos. . .) lo reciben tambien.
- b. Cuando un elemento recibe un evento, todos sus ascendentes (padre, abuelo. . .) lo reciben tambien.
- c. Cuando el ratón llega a un elemento, este recibe el evento mouseOver.
- d. Cuando el ratón abandona un elemento, o cualquiera de sus descendentes (hijos, nietos), este elemento recibe el evento mouseOut.

Elige una opción (a,b,c,d)

**Opcion correcta A** Tengo que buscar para desarrollarla

### **Clase 21 de abril**

Vamos a ver como manejar formularios -> Leer los valores

Añadimos un manejador para el evento *change*

Leemos el atributo value del objeto element correspondiente

La gracia esta en que cuando el usuario cambia el foco (click en otro sitio o lo que sea) se manda la informacion del formulario en este ejemplo

Ejemplo

Para consultar el estado de un radiobutton o un checkbox, usamos un selector como `input[name=figura]:checked`

El selector `:checked` captura todos los input activados

Ejemplo

Validar un formulario

Validar un formulario consiste en comprobar que el usuario ha rellenado los campos con valores adecuados. Podemos usar HTML con sus cosas, pero si queremos control absoluto es mejor procesarlo por javascript

Ejemplo

- Change: Fin de edición del input
- keyup: Pulsación de teclado (liberación de la tecla pulsada)
- paste: Pegado desde el portapapeles
- mouseup: Pulsación del botón del ratón (fin de la pulsación)

Ejemplo En este ejemplo a mi el que más me convence para una pass es keyup

### **Cambio de tema POO basada en herencias vs POO basada en protipos**

POO -> PROGRAMACIÓN ORIENTAZA A OBJETOS

La gran mayoría de lenguajes y herramientas diversas que emplean POO (programación orientada a objetos) aplican POO basada en herencia

Para solucionar una serie de problemas bien conocidos en la POO tradicional. No prefiere la composición frente a la herencia, sino que omite por completo la herencia y se basa solo en composición

Problemas de la erencia tradicional

- Código yo-yo:  
En jerarquías de cierta longitud, el programador se ve obligado a subir y bajar por las clases continuamente
- La herencia es una relación fuertemente acoplada:  
Los hijos heredan todo sobre sus padres, necesitan conocer todo sobre sus padres  
**PREGUNTA NORMALMENTE DE EXAMEN** Tiene que ver con la modulación, porque puedes añadir o quitar modulos a las cosas. -> Debilmente acoplado (que es facil)

Si cuesta mucho modificar/cambiar algo es Fuertemente acoplado

- Problema del gorila y el plátano: