

[Área personal](#) / [Mis cursos](#) / [Web1-2020-1C](#) / [Exámenes](#) / [Final 25/03/2021: Parte Teorica](#)

Comenzado el	jueves, 25 de marzo de 2021, 14:15
Estado	Finalizado
Finalizado en	jueves, 25 de marzo de 2021, 15:15
Tiempo empleado	59 minutos 57 segundos
Calificación	10,00 de 10,00 (100%)

Información

Pautas Acordadas de Final

Los alumnos se conectarán a un Google Meet grupal donde podrán hacer consultas y demás.

Cada alumno deberá además conectarse a una llamada individual. En la llamada individual compartirá su pantalla y deberá (en caso que posea) tener la cámara encendida y micrófono. No habrá docentes conectados a dicha llamada. Esta llamada dejará grabado el examen en su totalidad para consulta posterior en caso de considerarse necesario. En caso de no tener cámara/micrófono pueden conectarse dos veces y usar la cámara del celular.

No se permite el uso del celular. Los alumnos deberán durante el tiempo de cada actividad estar solos, sin personas a su alrededor. Se permite utilizar internet para consultas puntuales, no para copiar ejemplos enteros (ni de trabajos realizados previamente).

En caso de no poseer cámara en la computadora ni celular, se hará una defensa de 10 minutos de lo presentado al finalizar el examen o día siguiente a coordinar. Dicha defensa podrá ser por llamada telefónica estándar.

Los alumnos que rinden libre deberán además presentar el Trabajo Práctico Especial que se trabaja durante la cursada finalizado completamente y correctamente y defenderlo oralmente en la videollamada individual.

IMPORTANTE: Las respuestas con copia textual descontarán el puntaje de la pregunta, describa los conceptos con sus palabras.

Las dudas de consigna deben pedirse permiso por chat.

Solo se pueden hacer preguntas de consigna.

Prestar atención al tiempo, una vez cerrado el examen en Moodle no se aceptarán excepciones. Cada uno es responsable de mirar su reloj y asegurarse de enviar el examen antes del horario límite.

Sugerimos tener capturas de pantalla / documento de lo que hagan por cualquier inconveniente técnico.

Pregunta **1**

Finalizado

Puntúa 2,00 sobre 2,00

¿Qué debe incluir un proyecto para poder implementar un diseño web [responsive](#)? Escriba un ejemplo simple.

Para poder utilizar un diseño [responsive](#) se debe incluir en el HTML el

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

y luego en el CSS se deben utilizar las media queries que son la base de todo diseño [responsive](#).

Las media queries son las encargadas de establecer diversos breakpoints (puntos de quiebre) que permiten según las dimensiones del dispositivo donde mostrara la pagina web , cambiar la ubicacion, tamaño y diseño de los elementos.

Combene siempre realizar los diseños [responsive](#) orientados a mobile first ya que perminte priorizar qué es lo importante.!

EJEMPLO en CSS:

```
.right-box{  
background-color: green;  
}
```

@media only screen and (min-width :600px){. -> propiedades en css que seran cambiadas para la version desktop

```
.right-box{  
background-color:blue;  
}
```

Comentario:

Pregunta **2**

Finalizado

Puntúa 2,00 sobre 2,00

¿Qué es el DOM? ¿Para que se utiliza? Explique y dé ejemplos de los diferentes usos del DOM.

El DOM ,Document Object Model, es la base de toda pagina web y especifica todos los elementos que van a estar dentro de nuestra pagina web.

Mediante el DOM podemos interactuar con el usuario, permitiendole por ejemplo ingresar datos, realizar eventos (click, draw..) y demás que luego utilizaremos con JS para realizar diversas tareas..

Un ejemplo claro es asignarle a un

```
<button id="sumarDatos"> sumar </button>
```

de HTML una funcionalidad en JS:

```
let btn-sumar = document.getElementById(sumarDatos).addEventListener("click", sumarDatos);
```

donde sumarDatos es una función que luego especificaremos..

Otro ejemplo podria ser leer datos ingresados por el usuario en un input

Es utilizado tambien en CSS para editar estilos segun orden en los que se escriben los elementos.

Comentario:

Pregunta **3**

Finalizado

Puntúa 2,00 sobre 2,00

En Javascript, que diferencia hay entre una variable simple, un arreglo y un objeto.

Escriba un ejemplo de como declararía una variable simple, un arreglo y un objeto.

En JS en una variable simple se puede almacenar únicamente 1 dato a la vez, este puede ser string, integer, boolean..

Ejemplos:

```
let nombre = "Nina";
```

```
let edad = 23;
```

Un arreglo es una estructura que mantiene un orden especifico separado por comas (se lo conoce como indice/index, arranca en 0) y en el cual se pueden almacenar diversos datos. Permite tambien la funcionalidad de recorrerlos con un for, vaciarlos, editarlos..

Ejemplo:

```
let arr = [ ];      -> lo inicializo
```

```
arr.push("Nina");  -> le cargo un dato
```

```
let otroArr = [1,25,49] -> creo un arreglo y lo inicializo con 3 valores
```

Dentro de un Objeto se pueden almacenar tanto datos como comportamiento, un claro ejemplo de esto son los Objetos JSON que tienen una estructura de este estilo:

```
let estudiante = {  
  'nombre' : "Nina" ,  
  'edad' : 23,  
  'porcentaje-academico' : calcularPorcentaje();  
};
```

Comentario:

Pregunta **4**

Finalizado

Puntúa 2,00 sobre 2,00

¿Qué es y que ventajas tienen las aplicaciones o sitios web SPA? ¿Qué tecnología se utiliza para hacerlas y como se relacionan entre si?

Con las aplicaciones o sitios web SPA (Single Page Applications), nunca se recarga completamente el navegador, esto es posible gracias a la tecnología AJAX. (Asynchronous HTML y XML JS).

Existen 2 maneras de utilizar AJAX, mediante el uso de PARTIAL RENDER o mediante el uso de REST.

Partial render es el encargado de permitir cargar fragmentos de código HTML en nuestro código de la página SPA, permitiendo una navegación fluida sin necesidad de recargar la página con cada interacción.

Con el uso de REST podemos consultar un objeto JSON y procesarlo del lado del cliente.

Una gran ventaja de las SPA, es que al no recargarse con cada interacción del cliente a menos que presione F5, no se pierden los datos y al cargar fragmentos de HTML de manera asincrónica, permite un mayor y más fácil mantenimiento.

Comentario:

Pregunta **5**

Finalizado

Puntúa 2,00 sobre 2,00

Explique que es el uso de async / await en el manejo de promesas. Que ventajas vé en su uso y que desventajas ve?

El async / await es la nueva manera de realizar peticiones REST que incorpora ES6, son funciones asíncronas que permiten realizar un pedido HTTP, ya sea POST, PUT, GET o DELETE al servidor y esperar su respuesta permitiendo que la pagina web continúe su ejecución sin quedar "bloqueada esperando" hasta que esta termine.

Cuando realizamos una funcion async / await, esta nos devuelve una promesa la cual nos retorna un valor de respuesta ya sea response.ok ó en caso de entrar en el catch equivale a decir que el pedido realizado no pudo llevarse a cabo.

Estas funciones tienen una estructura base :

async funcionAsincronica(){ -> palabra reservada async indica que sera asincronica

try{ -> indicamos que intente realizar un pedido HTTP

let r = await fetch (url , { -> guardamos en una variable la peticion

//pedido HTTP

};

let json = await r.json(); ->una vez realizada la peticion quedamos a la espera de que el pedido HTTP que trajo un json en formatp string, vuelva a ser pasado a json para manipularlo como deseemos

//realizamos cosas con el json de respuesta

}

catch(e) { --> Fallo el pedido

console.log(e); -> mostramos cual fue el error que surgio durante el catch

}

}

Como ventajas, este tipo de funciones es mucho mas simple de realizar, de una lectura en un lenguaje mas natural para el programador. El hecho de que sea mas simple equivale tambien a decir que es mas facil de mantener (menos costo de mantenimiento), permite escalabilidad.

Como desventaja hay que incluir el costo de aprendizaje como toda incorporacion de nuevos conceptos, conlleva a tener una curva de aprendizaje.

Comentario:

[◀ Prefinal: Examen de Código Tandil](#)

Ir a...

[Final 25/03/2021: Parte Práctica ▶](#)