

[Área personal](#) / [Cursos](#) / [Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas](#) / [2021](#) / [Web1-2021-1C](#) / [Exámenes](#)
/ [Parcial 17/06/2021: Exámen de Código - TANDIL](#)

Comenzado el jueves, 17 de junio de 2021, 19:06

Estado Finalizado

Finalizado en jueves, 17 de junio de 2021, 21:06

Tiempo 2 horas

empleado

Calificación 7,70 de 10,00 (77%)

Información

1. Deben entrar a la sala que le corresponde de su sede y horario durante el examen y escribir nombre y apellido en el chat. **En caso de tener problemas de conexión enviar mail a web1@alumnos.exa.unicen.edu.ar.**
2. Las dudas de consigna deben pedirse permiso por chat. Se los derivará a otra sala para hacer la consulta. Solo se pueden hacer preguntas de consigna.
3. 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.
4. Sugerimos tener capturas de pantalla / documento de lo que hagan por cualquier inconveniente técnico.
5. La detección de copia anula el examen e implica la pérdida de la cursada para todas la personas involucradas en la copia. Es decir tanto aquellos que soliciten la copia como aquellos que la suministren.

Los ejercicios están en páginas diferentes para que se guarde cada una. Pueden resolverlos en cualquier orden. Pueden ir subiendo una versión y luego volver a modificarla.

Entrega: Se debe subir un archivo comprimido y TAMBIEN poner dentro del texto el código delimitando claramente cada una de las partes.

Pregunta 1

Finalizado

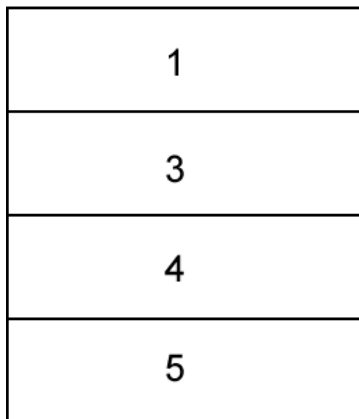
Se puntúa 2,00 sobre 2,00

Diseño Responsive

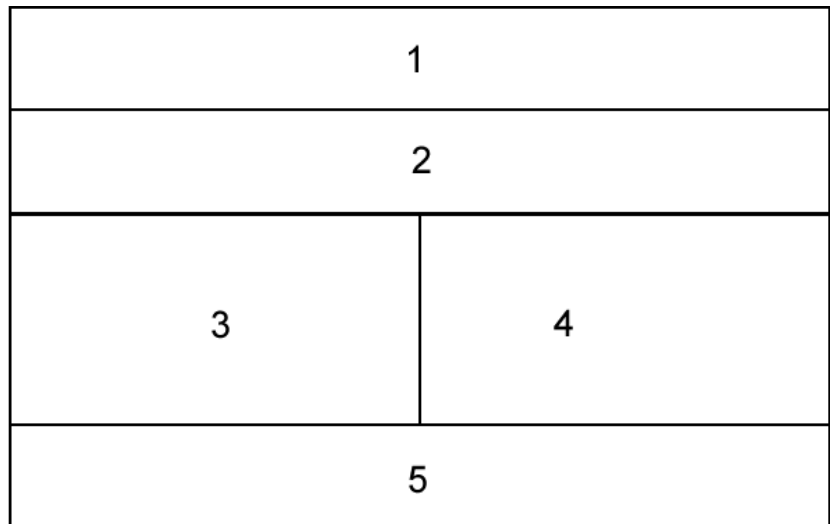
Realice un [diseño responsive](#) mobile first que se comporte según el siguiente diagrama.

NOTA

- **No se preocupe por las dimensiones exactas de las cajas**, solo se toma en cuenta la posición de las mismas.
- Si lo desea puede utilizar Bootstrap (ningún otro framework).



Mobile



Desktop

HTML:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio 1 - Parcial</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div class="box">1</div>
  <div class="box visibilidad">2</div>
  <div class="box-agrupado">
    <div class="box">3</div>
    <div class="box">4</div>
  </div>
  <div class="box">5</div>
</body>
</html>

```

CSS:

```

.box{
  text-align: center;
  font-size: 4vh;
  border: solid red 5px;
}

```

```
width: 100%;  
height: 15vh;  
  
}  
  
.visibilidad{  
  display: none;  
}  
  
@media only screen and (min-width:660px) {  
  
  .visibilidad{  
    display: block;  
  }  
  
  .box-agrupado{  
    display: flex;  
  }  
  
}
```

 [Ejercicio1-Parcial.rar](#)

Comentario:

respeto mobile first, oculta el div 2 en version mobile. todo bien

Pregunta 2

Finalizado

Se puntúa 1,00 sobre 1,00

CSS

Para cada una de las siguientes consignas escriba el código CSS correspondiente para que el sitio se visualice de la siguiente manera (No es necesario escribir código HTML pero debe suponer un correcto uso de los tags contemplando semántica):

NOTA: se evalúa el uso de buenas prácticas css. Utilice selectores anidados, combinados, reutilice clases, etc.

1. El menú debe tener texto en mayúscula con color verde.
2. Los hiperlinks dentro de los divs con clase "social" deben ser de color naranja.
3. La fuente del pie de pagina debe ser "RawFont". Los links dentro del pie de pagina deben estar en cursiva.

El HTML no lo voy a insertar porque no ingresé nada. Igualmente el .html estará dentro del proyecto de este ejercicio.

CSS:

/*El menú debe tener texto en mayúscula con color verde.

Pongo nav ya que es un tag semántico, y un menú va encerrado dentro de este tag.

Además cada una de las opciones se hace con una lista */

```
nav li {  
  text-transform: uppercase; /*Pone el texto en mayúsculas*/  
  color: green; /*Pone el texto de color verde*/  
}
```

/*Los hiperlinks dentro de los divs con clase "social" deben ser de color naranja.

En principio, se tiene que los links se identifican con el tag a. Luego, se que deben estar dentro de un div con la clase social. Aquí no hay semántica de elementos.

```
*/  
div.social a {  
  color: orange; /*Pone el texto de color naranja*/  
}
```

/*La fuente del pie de pagina debe ser "RawFont". Los links dentro del pie de pagina deben estar en cursiva.

El primer selector solo se refiere al footer en general, con lo que el segundo corresponden a los links dentro del footer.

```
*/  
  
footer {  
  font-size: "RawFont"; /*Pone el texto con la fuente RawFont*/  
}  
  
footer a {  
  font-style: italic; /*Pone el texto con el estilo itálico (cursiva) */  
}
```

Comentario:

Pregunta 3

Finalizado

Se puntúa 3,00 sobre 3,50

Se desea hacer un sistema sencillo para poder llevar los ingresos y egresos de un negocio durante el año.

Requerimientos

- 1. Se deben poder cargar los ingresos y egresos mensuales mediante un formulario. Por lo tanto se debe poder almacenar cada mes con el respectivo importe de ingreso y egreso. Se debe agregar la restricción en la carga que los datos que se carguen no puedan ser más de 12.**
- 2. Se debe poder listar el resumen anual mostrando mes y el saldo (ingreso - egreso). Los meses en los existen ganancias (saldo positivo) se deben resaltar con color verde.**
- 3. Con un botón se debe mostrar la cantidad de meses en los que hubo pérdidas (saldo negativo).**

Implementar el código HTML, CSS y JS necesario.

Adjuntar un archivo comprimido y TAMBIEN copiar los tres archivos en el cuadro de texto.

ACLARACIÓN: Los datos deben manejarse a nivel local (no se requiere de un servicio)

HTML : aclaración: debido a que el primer botón lo definí con un evento frente al submit del campo, para que no tirara error, el botón de ver los saldos negativos, lo puse por fuera del formulario.

Código:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejercicio 3 - Parcial</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <form id="formulario">
    <div><label for="mes"></label> Ingrese el mes<input type="text" name="mes"></div>
    <div><label for="ingreso"> Ingrese el ingreso de su mes</label> <input type="number" name="ingreso"></div>
    <div><label for="egreso"> Ingrese el egreso de su mes</label> <input type="number" name="egreso"></div>
    <button id="btn-agregar-balance-mensual"> Agregar Balance mensual </button>
  </form>

  <button id="btn-ver-cant-saldos-negativos"> Ver meses con saldo negativo </button>
  <!--Sección Lista del balance-->
  <ul id="lista-balance"></ul>

  <!--Sección Meses con Saldo negativo-->

  <p class="ocultar"> La cantidad de meses con saldo negativo es de : <span id="meses-saldo-negativo"></span></p>
  <script src="js/main.js" type="text/javascript"></script>

</body>
</html>
```

CSS:

```
.ocultar{
  display: none;
}

.saldo-positivo{
```

```
color:green;
}
```

JS:

```
"use strict"
```

```
const CANT_MESES = 12;
let balanceMensual = [];
let form = document.querySelector("#formulario");
form.addEventListener('submit', (e) => {
    e.preventDefault();
    if ( balanceMensual.length != CANT_MESES){
        let formData = new FormData(form);
        balanceMensual.push( {
            "mes": formData.get("mes"),
            "ingreso": formData.get("ingreso"),
            "egreso": formData.get("egreso")
        } )
    }
    actualizarLista();
})
```

```
document.querySelector("#btn-ver-cant-saldos-negativos").addEventListener('click', (e) => {
    document.querySelector("p").classList.remove("ocultar");
    let mesesSaldoNegativo = 0;
    for ( let elem of balanceMensual)
        if ( elem.ingreso - elem.egreso < 0)
            mesesSaldoNegativo++;
    document.querySelector("#meses-saldo-negativo").innerHTML = mesesSaldoNegativo;
})
```

```
function actualizarLista() {
    let lista = document.querySelector("#lista-balance");
    lista.innerHTML = "";
    for ( let elemento of balanceMensual){
        lista.innerHTML += `<li> Mes : ${elemento.mes} ; Saldo: ${elemento.ingreso - elemento.egreso} </li> `
        if ( elemento.ingreso - elemento.egreso > 0 )
            lista.classList.add("saldo-positivo");
    }
}
```

 [Ejercicio3-Parcial.rar](#)

Comentario:

- Si hay un mes positivo, se marcan todos

Pregunta **4**

Finalizado

Se puntúa 1,70 sobre 3,50

El sitio WebStats posee un servicio donde se almacenan los accesos a nuestra web cada año. Para almacenar la información nos brindan el acceso un servicio REST. El servicio REST en la URL `www.webstats.com/api/cliente/456` devuelve objetos con la siguiente estructura:

```
{ "accesos": [ { "id": "1e32a34", "anio": 2019 "mobile": 2901, "tablet": 1234, "desktop": 2211, }, {  
"id": "1e62a34", "anio": 2020 "mobile": 3514, "tablet": 1932, "desktop": 2989, } ] }
```

1- Escriba una función de JavaScript que nos permita mostrar por consola un listado con la cantidad de total de visitas para cada año.

2- Insertar un nuevo registro de accesos (Por ejemplo para 2021).

Aclaración: el objeto JSON es de ejemplo, no suponga que las posiciones de los arreglos o los valores de los datos se respetan. Debe funcionar con cualquier objeto de este tipo.

Entrega: Se debe subir un archivo comprimido y TAMBIEN poner dentro del texto el código delimitando claramente cada una de las partes.

Aclaración: solamente hecho el 1

JS:

"use strict"

```
async function totalVisitasPorAnio () {  
  try {  
    let res = await fetch('www.webstats.com/api/cliente/456');  
    let json = await res.json();  
    for ( let elemento of json )  
      console.log ( `Anio: ${elemento.anio} ; Total visitas: ${elemento.mobile + elemento.tablet + elemento.desktop } ` );  
  }  
  
  catch (error) {  
    console.log(error);  
  }  
}  
  
totalVisitasPorAnio();
```

 [Ejercicio4-Parcial.rar](#)

Comentario:

1) * No accede a la propiedad "accesos" correctamente

2) Sin implementar

[◀ Parcial 17/06/2021: Teórico - TANDIL](#)

Ir a...

[Parcial 17/06/2021: Teórico - OLAVARRIA ▶](#)