

Comenzado el	Thursday, 17 de September de 2020, 15:43
Estado	Finalizado
Finalizado en	Thursday, 17 de September de 2020, 17:54
Tiempo empleado	2 horas 11 minutos
Puntos	7,99/8,50
Calificación	9,40 de 10,00 (94%)

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.

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 0,75 sobre
1,00

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.
- Tenga en cuenta los **colores que cambian** en el diagrama.
- Si lo desea puede utilizar Bootstrap (ningún otro framework).



ACLARACIÓN: en la consigna 3 adjunté un archivo rar con el nombre "Final de Web 1- Marina Caseres" es una carpeta con el final completo todo montado como una pagina. Pensé esta presentación para hacer más completa la entrega y facilitar la visualización en la corrección.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Consigna 1</title>
  <link rel="stylesheet" href="style.css">
</head>
```

```
<body>
  <a href="../index.html">Volver</a>

  <article class="head">
    <section class="logo">LOGO</section>
    <section class="encabezado"></section>
  </article>
  <article class="footer">
    <section class="informacion"></section>
    <section class="informacion"></section>
    <section class="informacion"></section>
  </article>

</body>
</html>
```

CSS

```
*{
  text-align: center;
  font-family: Arial, Helvetica, sans-serif;
}
```

```
article {
  height: 283px;
}
```

```
.informacion{
  background-color: gray;
  height: 30%;
  margin: 2%;
}
```

```
.logo{
  background-color: gray;
  margin: 2%;
  height: 20%;
}
```

```
color: red;
padding-top: 10%;
font-size: 30px;
}
.encabezado{
background-color: gray;
margin: 2%;
height: 68%;
}

@media only screen and (min-width: 600px){

    *{
        box-sizing: border-box;
    }

    article.head{
        width: 100%;
        height: 370px;
        margin: 0 0 30px 0;
    }

    article.footer{
        width: 100%;
        height: 240px;
        display: flex;
        margin: 0 0 30px 0;
    }

    .logo{
        padding-top: 5%;
        height: 30%;
    }

    .encabezado{
        height: 70%;
    }
}
```

```
.informacion{  
  height: 100%;  
  width: 30%;  
}  
}
```

```
@media only screen and (min-width: 800px){
```

```
  *{  
    box-sizing: border-box;  
  }
```

```
  article{  
    width: 100%;  
    height: 305px;  
    display: flex;  
    margin: 0 0 30px 0;  
  }
```

```
  .logo{  
    padding-top: 10%;  
    font-size: 80px;  
    color: black;  
    width: 32%;  
    height: 100%;  
  }
```

```
  .encabezado{  
    width: 68%;  
    height: 100%;  
  }
```

```
  .informacion{  
    height: 100%;  
    width: 30%;  
  }
```

}



consigna1.rar

Comentario:

Centrar todo con `* { text-align: center; }` no es buena idea, la mayoría de cosas no van centradas, además rompe la herencia de `text-align`, de hecho en general el `*` no se usa salvo para igualar algunos detalles entre navegadores.

No entiendo pq cambia `article.footer` en tablet, o el `height` entre dispositivos

Entiendo que anda aunque excesivamente complicado el código y algunas malas prácticas como usar `*`, pisar propiedades innecesariamente, etc.

Pregunta 2

Finalizado

Puntúa 0,84 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:

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

1. La tipografía del sitio debe ser “Verdana” y los titulos h1 a h4 deben ser de color gris.
2. El pie de pagina debe tener el texto en azul, los hiperlinks contenidos alli deben verse en rojo y en cursiva, el borde superior del bloque debe ser de 2px y negro.
3. Los items del menu principal deben ser con texto color blanco, fondo negro y el texto debe cambiar a gris cuando el mouse paso por encima.

```
body{  
    font-family: Verdana;  
}
```

```
h1, h2, h3, h4 {  
    color: grey;  
}
```

```
nav a{  
    color: white;  
    background-color: black;  
}
```

```
nav a:hover{  
    color: rgb(196, 196, 196);  
}
```

```
footer {  
    color: blue;  
    border-top: solid black 2px;  
}
```

```
footer a{
```

```
color: red;  
font-weight: lighter;  
}
```

 style.css

Comentario:

1. ok
2. ok
3. ok- , no es sobre items, sino sobre "a"

Pregunta 3

Finalizado

Puntúa 3,40 sobre 3,50

Censo Nacional - JS + HTML

Implementar una pagina donde se puedan incluir las poblaciones por provincia y hacer algunas consultas.

En la implementación se deben cumplir con los siguientes requerimientos / funcionalidades:

- **Se deben poder ingresar los habitantes de cada provincia.**
- **Se debe poder consultar y mostrar el total de habitantes y cual sería el promedio entre todas las provincias.**
- **Se debe poder consultar y mostrar en la página la provincia menos habitada.**

NOTA: Para este ejercicio todo debe realizarse localmente sin uso de AJAX/REST.

ACLARACIÓN: adjunté un archivo rar más, con el nombre "Final de Web 1- Marina Caseres" es una carpeta con el final completo todo montado como una pagina. Pensé esta presentación para hacer más completa la entrega y facilitar la visualización en la corrección.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Consigna 3</title>
  <link rel="stylesheet" href="../style.css">
  <script src="add.js" defer></script>
</head>
<body>
  <form action="">
    <input id="habitantes" type="number" placeholder="Habitantes">
    <input id="provincia" type="text" placeholder="Provincia">
    <button id="cargar">Cargar</button>
  </form>
  <a href="../index.html">Volver</a>
</body>
</html>
```

JS

```
"use strict";
```

```
let poblaciones= [];
```

```
document.querySelector("#cargar").addEventListener("click", function(e){  
    e.preventDefault();
```

```
    let provincia = document.querySelector("#provincia").value;  
    let habitantes = document.querySelector("#habitantes").value;
```

```
    cargarPoblacion(provincia, habitantes, poblaciones);
```

```
});
```

```
function cargarPoblacion(provincia, habitantes, array){  
    let cargar= {  
        "provincia": provincia,  
        "habitantes": habitantes  
    }  
    array.push(cargar);  
    console.log(array);  
    let promediofinal = promedio(array);  
    console.log("Promedio de habitantes: " + promediofinal);  
    menosHabitada(array);  
}
```

```
function promedio(array){  
    let total = 0;  
    let contador = 0;  
  
    for(let i = 0; i<array.length; i++){  
        total += parseInt(array[i].habitantes);  
        contador ++;  
    }  
    console.log("total: " + total)
```

- Final de Web 1- Marina Caseres.rar
- consigna3.rar

1. 100%
2. 95% se muestra por consola
3. 95% se muestra por consola

Pregunta 4

Finalizado

Puntúa 3,00 sobre
3,00

Un servicio de clima donde se registran las precipitaciones por provincia y año. Tiene almacenados los datos y puede obtenerse en la URL www.climanacional.com/api/lluvias, y devuelve JSONs como el siguiente:

```
{
  "lluvias" : [
    {
      "provincia": "Buenos Aires",
      "año": 2019,
      "mm": 1000
    },
    {
      "provincia": "Tucumán",
      "año": 2018,
      "mm": 1300
    },
    ...
  ]
}
```

Escriba el código JS para:

1. Desarrollar una función que imprima por consola las lluvias del año 2018.
2. Escriba el código para agregar un registro de lluvia de una provincia y de un año en particular. Deben ser leídos desde inputs (puede suponer su existencia).

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 TAMBIÉN poner dentro del texto el código delimitando claramente cada una de las partes.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Consigna 4</title>
  <link rel="stylesheet" href="../style.css">
  <script src="add.js" defer></script>
</head>
<body>

  <form action="">
    <input id="provincia" type="text" placeholder="Provincica" >
    <input id="año" type="numero" placeholder="Año">
    <input id="mm" type="number" placeholder="Milímetros de lluvia">
    <button id="agregar">Agregar</button>
  </form>

  <a href="../index.html">Volver</a>
</body>
</html>
```

JS

```
"use strict";
```

```
let lluvias = [];
```

```
let url = 'http://www.climanacional.com/api/lluvias';
```

```
document.querySelector("#agregar").addEventListener("click", function(e){
  e.preventDefault();
```

```
  let provincia = document.querySelector("#provincia").value;
```

```
  let año = document.querySelector("#año").value;
```

```
  let mm = document.querySelector("#mm").value;
```

```
    obtenerLLuvia(provincia, año, mm);  
  });
```

```
function obtenerLLuvia(provincia, año, mm){  
  let nuevaLluvia = {  
    "thing": {  
      "provincia": provincia,  
      "año": año,  
      "milímetros": mm  
    }  
  }  
  surbirLluvia(nuevaLluvia);  
}
```

```
function surbirLluvia(dato){  
  
  fetch(url, {  
    "method": "POST",  
    "headers": {  
      "Content-Type": "application/json",  
    },  
    "body": JSON.stringify(dato)  
  })  
  .catch (error=> console.log(error));  
}
```

```
function traerRegistroDeLLuvia(){  
  
  fetch(url)  
  .then(response=>response.json())  
  .then(json=>{  
    insertarRegistrto(lluvias,json);  
  })  
  .catch (error=>console.log(error));  
}
```

```
function insertarRegistrto(array,json){
```

```
for(let i = 0; i<json.lluvias.lenght; i++){  
    let lluvia = json.lluvias[i].thing;  
    lluvias.push(lluvia);  
}  
lluvias2018(array)  
}  
  
function lluvias2018(array){  
    for (let i = 0; i<array.lenght; i++){  
        if(array[i].año === 2018){  
            console.log(array[i]);  
        }  
    }  
}  
}
```

 consigna4.rar

Comentario:

1. 100%
2. 100%

Facultad de Ciencias Exactas – UNICEN

Contacto administradores plataforma: E-mail moodle@exa.unicen.edu.ar – Tel. +54 0249 4385650 int. 2098