**Repositorio del curso**

Aquí vas a poder encontrar el repositorio que vamos a esta utilizando durante el curso: <https://github.com/LeonidasEsteban/responsive-design-portafolio>

Recuerda que cualquier duda, estaremos pendientes en el sistema de discusiones.

¡Comencemos!

**Meta viewport**

En esta clase aprenderás qué es y cómo usar la etiqueta meta viwport, esta es una etiqueta de metadatos que te ayudará a configurar tu website para que sea visible en dispositivos de menor tamaño. Por lo tanto, escribirás tu primera línea de código y la alojarás en el mismo proyecto de código que ya tenías desde el **Curso de Desarrollo Web**.

Uno de los objetivos principales al usar esta etiqueta será que conserves la legibilidad de tu página web, al variar el escalado de tus contenidos.

**width=device-width** para que se adapte según la pantalla del dispositivo  
**initial-scale=1.0** para indicar el escalado según el dispositivo

**Medidas relativas útiles en Responsive Design**

En esta clase aprenderás las medidas que existen en el campo del responsive design.

Lo primero que debes tener en cuenta es que estas medidas son maleables, en la medida en que dependen de su fuente de origen o medida madre. Entre ellas se encuentran el porcentaje (longitud referente al tamaño de los elementos padre), los em (unidad relativa al tamaño de fuente especificada más cercano), los rem (unidad relativa al tamaño de fuente especificada en el ancestro más lejano, como html o body) y tamaños del viewport vw/vh (longitud relativa porcentual con respecto al viewport).

**Unidades de medida relativa:**

**Porcentaje**: Tamaño referente al tamaño de los elementos padre en porcentaje  
**em**: unidad relativa al tamaño de fuente especificada más cercano incluyendo a su propio elemento, siendo 1 la escala del 100%, en caso de no estar especificada en su propio elemento se irá a su padre, luego al padre de su padre, etc.  
**rem**: unidad relativa al tamaño de fuente especificada en el ancestro más lejano (normalmente html o body), siendo 1 la escala del 100%,  
**vw / vh**: ViewportWidth/ViewportHeight: unidad relativa porcentual con respecto al viewport.  
vh y vw se van a modificar con el resize de la pantalla

**Media queries**

Para que logres los resultados que deseas en tus proyectos, es necesario cambiar ciertas propiedades para modificar el tamaño de los textos, contenidos y hojas de estilo; la manera de hacer esto es el media queries.

El media queries es un módulo de css que hace posible al responsive design, éste existe desde el 2010 y se encarga de adaptar la representación del contenido a características del dispositivo.

En esta clase conocerás cómo funciona su estructura, cómo se construye y adiquirirás los conocimientos necesarios para trabajar con él, desde tu editor de código.

**media Queries**

gracias a este modulo exite responsive desing

mudulo CSS3S que permite adaptar la representacion del contenido a caracteristicas del dispositivo

@media media **type** and (condición) { }

se compone de un media type y una o mas condiciones

**ejemplo**

@media screen **type** and (max-width: 768px) { }

todas las pantallas con un ancho inferior o igual a 768px cumplen con esta funcion

@media screen type **and** (max-width: 768px) **and** (min-width: 480px) { }

todas las pantallas con un ancho de 480px hasta 768px cumplen con esta funcion

Mobile Firts

@media screen **type** and (max-width: 320px) { }

@media screen **type** and (max-width: 480px) { }

@media screen **type** and (max-width: 768px) { }

@media screen **type** and (max-width: 1024px) { }

usa min-width (min-width = desde)

Desktop Firts

@media screen **type** and (min-width: 1024px) { }

@media screen **type** and (min-width: 768px) { }

@media screen **type** and (min-width: 480px) { }

@media screen **type** and (min-width: 320px) { }

usa max-width (max-width = hasta)

**Formas de incluir media queries** En esta clase aprenderás a insertar un media querie en tu proyecto. Para ello, vas a trabajar sobre tu hoja de estilos, utilizando el tag style.

El primer paso para lograr esto será realizar una nueva hoja de estilos en tu proyecto, ésta debe contar, en primer lugar, con la etiqueta link; harás uso de la aplicación de medidas para la pantalla, bordes y colores, entre otras características.

**Breakpoints estandarizados que debemos de tener en cuenta:**

* Móviles: entre 320 y 480 píxeles.
* Tablets: entre 768 y 1024 píxeles.
* Pantallas grandes: más de 1024 píxeles o más.

**Diseño elástico con max-width y flex-wrap**

Como te habrás dado cuenta tu proyecto está desordenado aunque carga en el viewport. Por esta razón en esta clase aprenderás a ordenar tu proyecto por medio del uso de contenedores flexibles, que varíen su tamaño según la amplitud con la que cuentan en diversos dispositivos y permitan una óptima visualización.

Por otro lado, para hacer esto posible, aprenderás a aplicar las etiquetas de max-width y flex- wrap; éstas también te ayudarán a evitar que tu usuario necesite navegar la página de forma horizontal, pues la información se organizará en forma vertical para facilitar la experiencia.

**Ajustando la sección de contacto y footer**

Es hora de trabajar sobre la sección de contacto y el footer de tu proyecto, por lo que en esta clase aprenderás a modificar ambas secciones.

En el trabajo sobre estas secciones modificarás sus elementos de contacto, padding, el modelo de caja (aprenderás que en algunas ocasiones es necesario eliminarlo y rehacerlo) el container y su respectivo padding.

Y por último, también trabajarás sobre los elementos propios de los botones que se encuentran en estas secciones.

inspector de código los copiaba😂(Por si no les funciona esa serie de comandos)

<https://unicode.org/emoji/charts/full-emoji-list.html>

**CSS positions**

En tu proyecto, al probarlo en distintos dispositivos, te darás cuenta de que no solamente sus elementos y contenidos deben tener la propiedad de cambiar sus tamaños y condiciones para hacerse elementos de una buena visualización; sino que, además, hay unos ciertos contenidos que permanecen constantemente en el espacio de visualización, a pesar de que tu usuario realice scroll en la página.

Estos elementos te darán cuenta de uno, entre los varios tipos, de posiciones que permite CSS. Por esta razón, durante esta clase conocerás cuáles son las distintas posiciones (estática, relativa, absoluta, fixed y sticky) sobre las que podrás trabajar, aplicándolas a tu proyecto. Material de trabajo de esta clase: <https://codepen.io/LeonidasEsteban/pen/VGWzWK>

