**Como subir el proyecto a github fácilmente:**

**https://www.youtube.com/watch?v=H03E8\_HImAU**

**1) colocamos el cuerpo de la pagina web:**

! para el doctype html

**2) Escribimos el titulo de la pagina en el head:**

**Nota: Primero se escriben las librerías y luego los css nuestros.**

<link rel=”preload” href=”css/styles.css” as=”style”> // Primero que cargue lo mas rápido que pueda la hoja de estilos.

<link href=”css/styles.css” rel=”stylesheet”>

<title>

**3)En el body comenzamos a escribir el titulo de las posibles secciones de la pagina web:**

<h1>

<h2>

<p>

<h2>

<p>

<h2>

<p>

<h3>

<p>

**4) Comenzamos a estructurar en secciones el texto que escribimos**

1) Cabecera: <header> <a href=”index.html”>inicio </a>

2) Pie de pagina: <footer>

4) Navegador: <nav>

5) Section para agrupar elementos: <section>

6) Articulo para colocar historias o mucho texto: <article>

7) Colocar infromacion relaxionada a la section pero a un costado (lateral derecho): <aside>

8) Div : <div>

**Atajos de teclado:**

1. Con alt presionado y moviendo las flechas puedo mover un elemento de la línea en donde este parado.
2. Presionando alt y flecha de arriba y moviendo las flechas puedo copiar la línea de código si estoy posicionado en la misma y al final.
3. Presionando ctrl y cierre de corchete puedo comentar lo que marque

**Buscar iconos:**

<https://tablericons.com/> posee muchos iconos (busco map)

<https://heroicons.com/>

<https://fontawesome.com/icons?d=gallery&p=2>

**Utilizar siempre svg :**

Es importante utilizar este tipo de imágenes porque no se distorcionan porque son un vector, y también todos los navegadores lo soportan bien.

**En este proyecto utilzaermos los siguientes inconos:**

1. Map (posicionamiento del mapa)
2. Palette (es una paleta de pintura)
3. Android
4. Apple
5. Card (credi card)

**Colocar formularios:**

Utilizo <form> y todo lo que este dentro son los campos para completar, para luego conectarlos a una base de datos o autenticando una secion comparar con los registros actuales, para eso se utiliza php. Por otro lado, dentro de un form si quiero colocar como una pequeña sección o espacio para una leyenda utilizo la siguiente etiqueta: **<fieldset>**

Y dentro de esta etiqueta otra que se llama <leyend> para poder colocar el texto.

**Ahora para colocar los campos:** Se utiliza el indicador para el usuario con la etiqueta <label> telefono

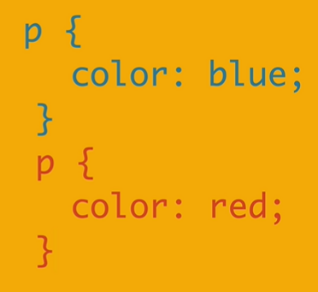
Luego colocamos el campo a rellenar con un <input type=”tel” pleiholder=”escribre tu telefono”>

**Ahora comenzamos con CSS (VERSION 2.1)**

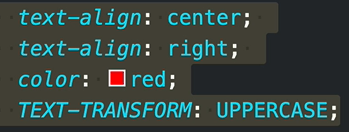


**https://www.udemy.com/course/desarrollo-web-completo-con-html5-css3-js-php-y-mysql/learn/lecture/24106068#overview**

**Porque es en cascada?** Porque si se tiene el mismo código escrito como se ve aquí, css solo va a tomar el ultimo que este escrito:

Lo mismo sucede con dos campos iguales pero con diferente valor, se quedara con el ultimo valor. Si por casualidad escribiste mal un campo de css, lo podemos saber en el navegador, realizando el inspeccionar, vemos el código css ahí, y podemos notar que esta tachado, esto quiere decir que esta mal escrito.

Ejemplo de estilos en el h1:



Nota:

Para que el navegador entienda que se esta trabajando con una hoja de estilos, tenemos que colocar en la parte del head el enlazado del css. Haciendo lo siguiente:

<link rel=”preload” href=”css/styles.css” as=”style”> // Primero que cargue lo mas rápido que pueda la hoja de estilos.

<link href=”css/styles.css” rel=”stylesheet”>

**Mas fácil de personalizar la pagina web: Uso de css matic**

[**https://www.cssmatic.com/gradient-generator#'\-moz\-linear\-gradient\%28top\%2C\%20rgba\%28248\%2C80\%2C50\%2C1\%29\%200\%25\%2C\%20rgba\%28241\%2C111\%2C92\%2C1\%29\%2050\%25\%2C\%20rgba\%28246\%2C41\%2C12\%2C1\%29\%2051\%25\%2C\%20rgba\%28240\%2C47\%2C23\%2C1\%29\%2071\%25\%2C\%20rgba\%28231\%2C56\%2C39\%2C1\%29\%20100\%25\%29\%3B**](https://www.cssmatic.com/gradient-generator#'\-moz\-linear\-gradient\%28top\%2C\%20rgba\%28248\%2C80\%2C50\%2C1\%29\%200\%25\%2C\%20rgba\%28241\%2C111\%2C92\%2C1\%29\%2050\%25\%2C\%20rgba\%28246\%2C41\%2C12\%2C1\%29\%2051\%25\%2C\%20rgba\%28240\%2C47\%2C23\%2C1\%29\%2071\%25\%2C\%20rgba\%28231\%2C5)**'**

Aquí puedes colocar sombrado (box shadow), degradado (gradient generator)

Tambien para saber si todos los navegadores soportan tales cosas de css se puede entrar en:

caniuse.com

**Selectores en css**

Pantalla: Desque que llegaron las tablets o los celulares ya no se habla de pixels, ahora se habla de rem, para calular el pasaje de unidades se utiliza este enlace:

<https://www.ninjaunits.com/converters/pixels/pixels-rem/>

Pero para no estar calculando continuamente escribimos en el css:

Html {

    font-size: 62.5%;

}

Body {

    font-size: 16px;

    /\*1rem=10px \*/

}

/\*html y body le cambie el tamaño para poder usar rem \*/

El span sirve como un indicador que actua como un selector.

**Selector de elemento (h1{)**

1. Si se quiere cambiar un único elemento y no todos, por ejemplo un único span.

Hacemos: h1 span{

1. Si se quiere cambiar de la misma manera todos los span.

Hacemos: span{

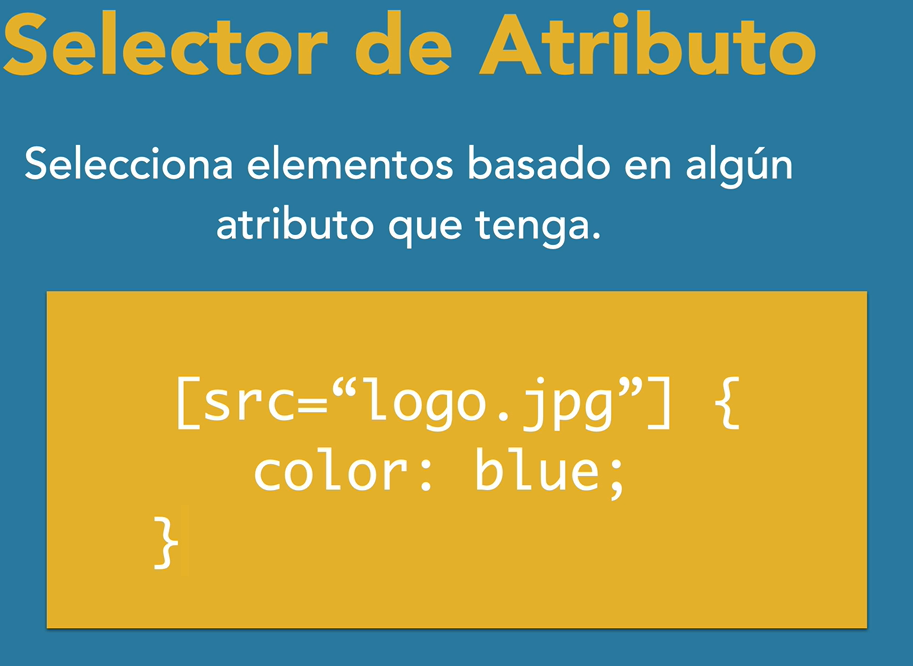
**Selector de clase: en html es class**



**Selector de ids: en html es id**



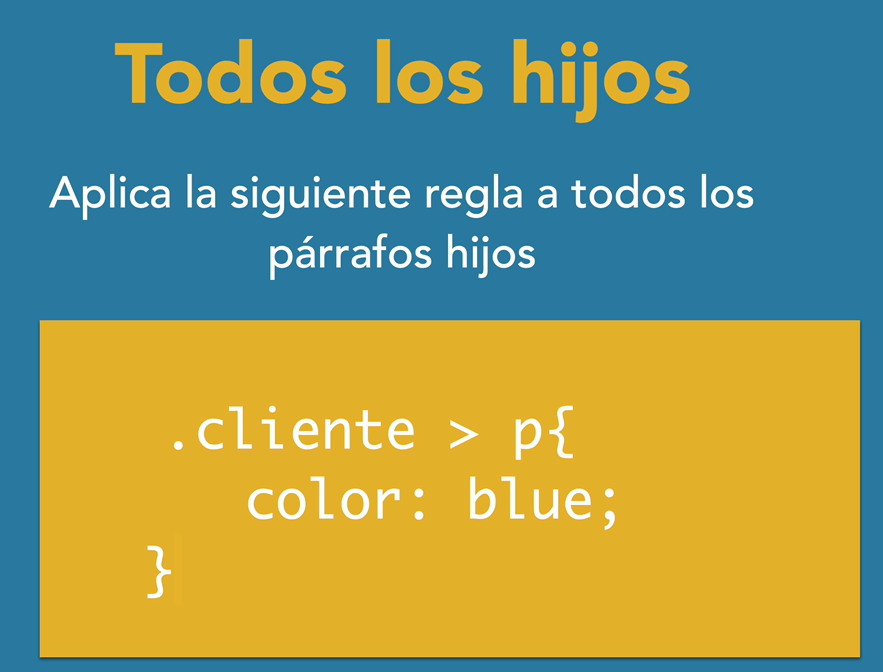
**Selector de atributo:**



**Selector de atributo con combinación descendente:**

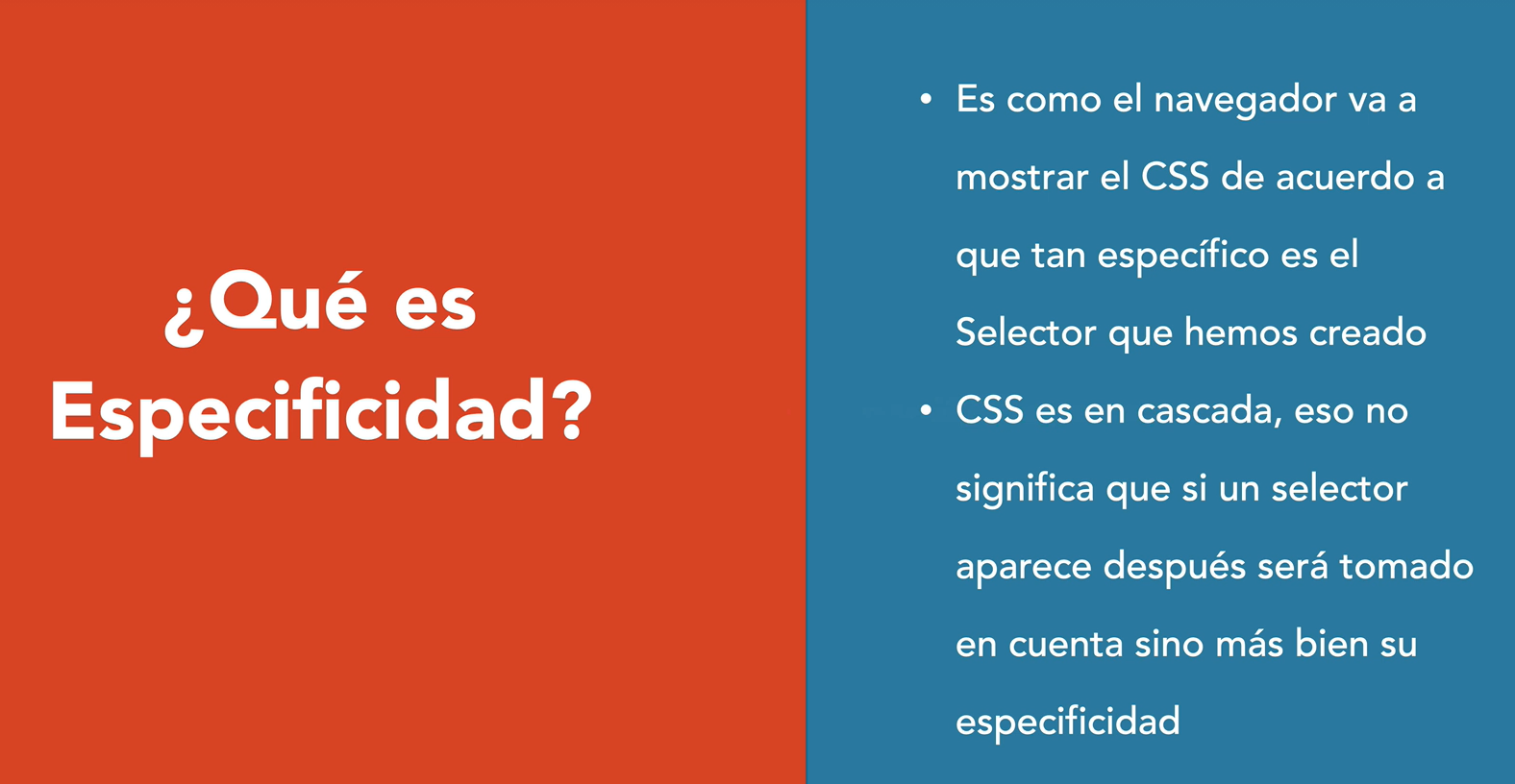


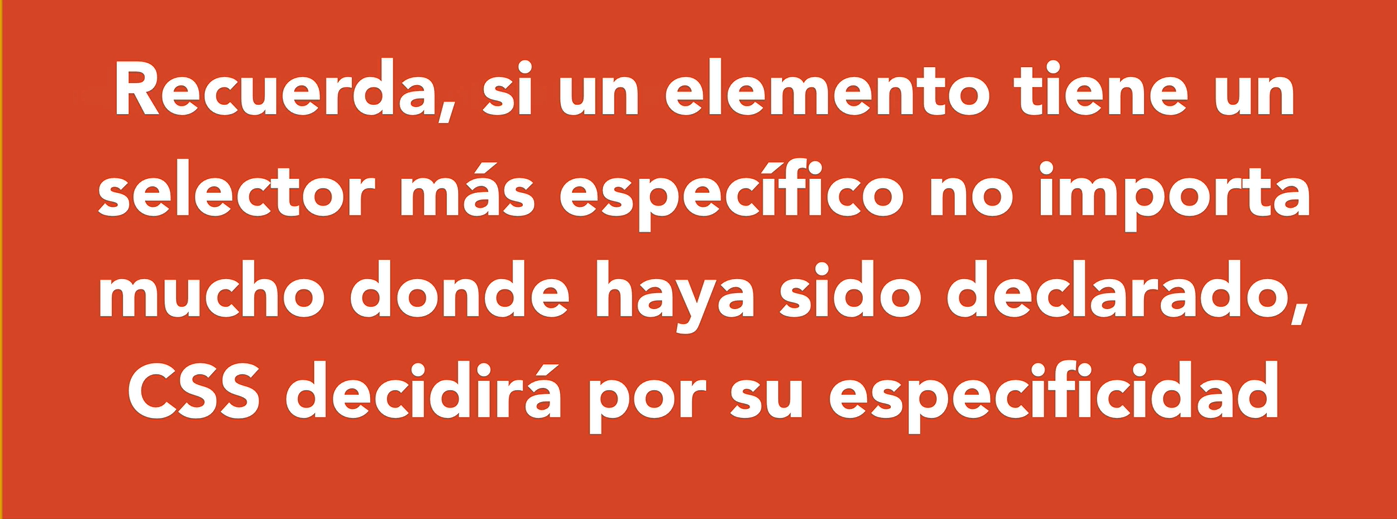
**Selector de atributo con todos los hijos:**



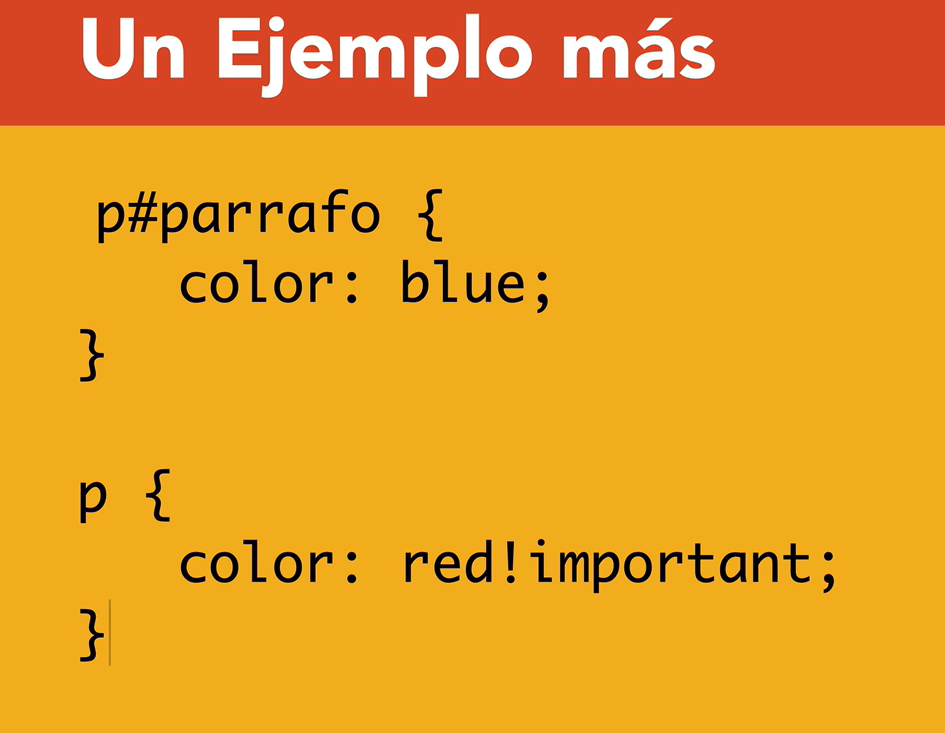
**Conclision: Es mejor utilizar las clases y lo id no.**

**Que es la especifidad:**





**Este código de important aplasta a todos lo selectores:**



**Paleta de colores:**

* **Escribo root en el css.**

**Fuente de texto:**

* **Visitar google fonts: Aquí se eligio Krub.**
* 

**Normalize:** Es una librería que hace que nuestra pagina se vea consistente sin importar el navegador que se este utilizando. Busco en googel, normalize.css y copio todo el código, me creo un css llamado normalize y lo pego ahí.

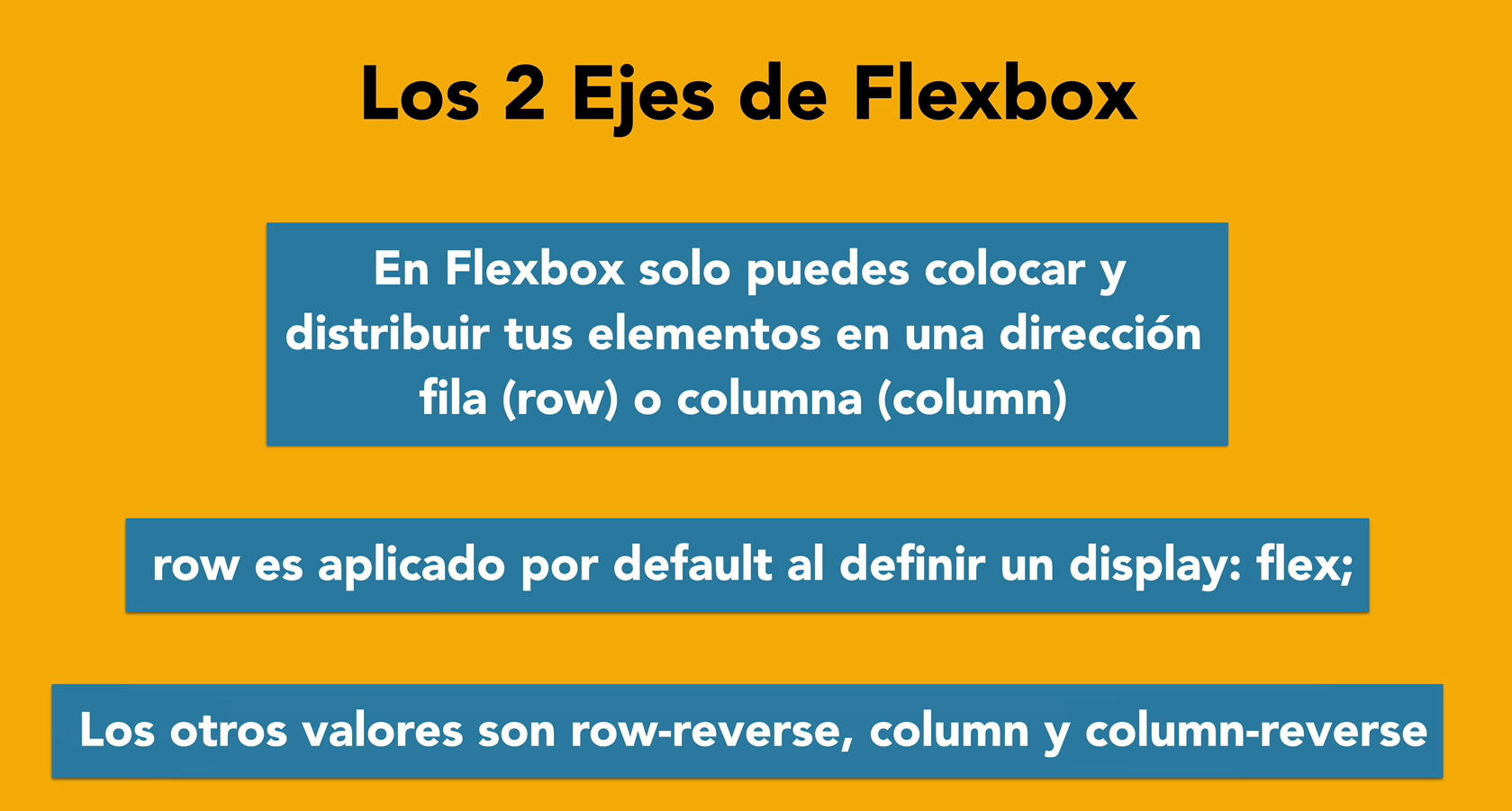
<https://necolas.github.io/normalize.css/>

**Uso de display:**

Es para saber si tus elementos estarán uno de bajo de otro (display:”Block”) o uno seguido de otro (display:”inline”)

**Margin y padin siguen la regla del reloj:**

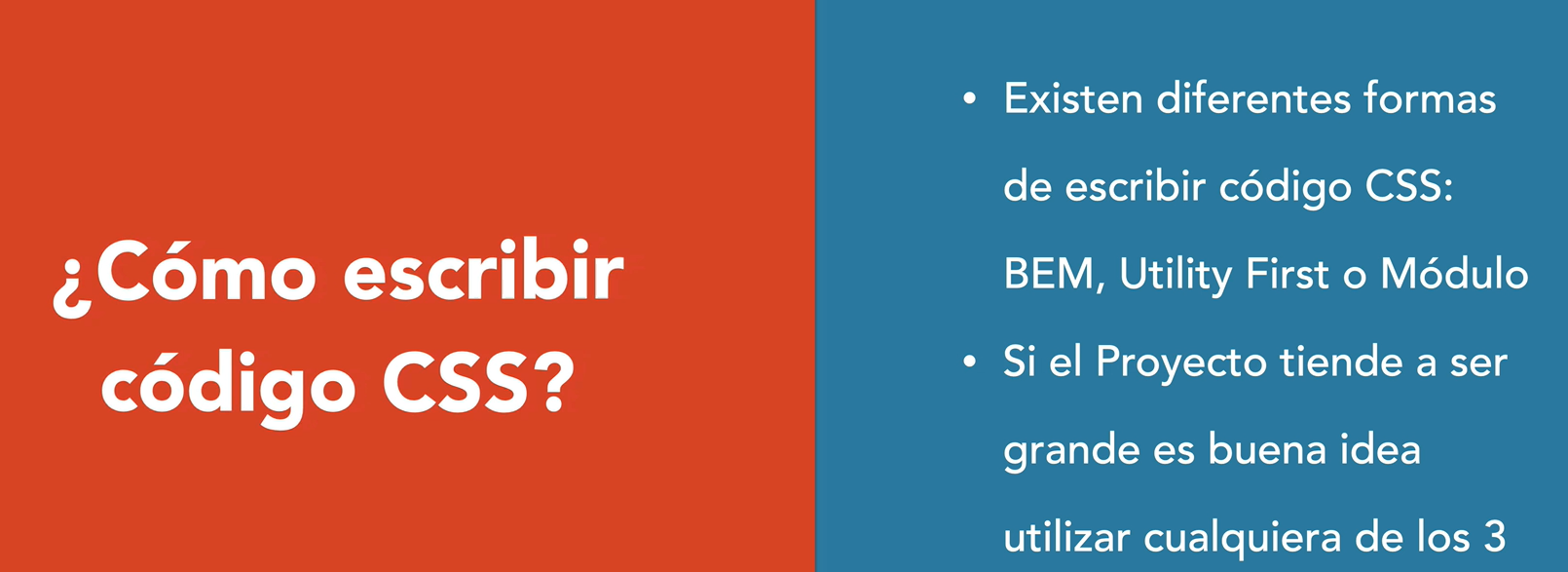
1. **Margin: Separa los elementos desde afuera.**
2. **Padin: Separa el contenido desde adentro.**
3. **Flex: Se usa para mejorar las separación de los elementos de una misma clase. PARA ELLO SE VA DIRECTO A LA CLASE PADRE Y AHÍ SE AGREGA EL flex.**



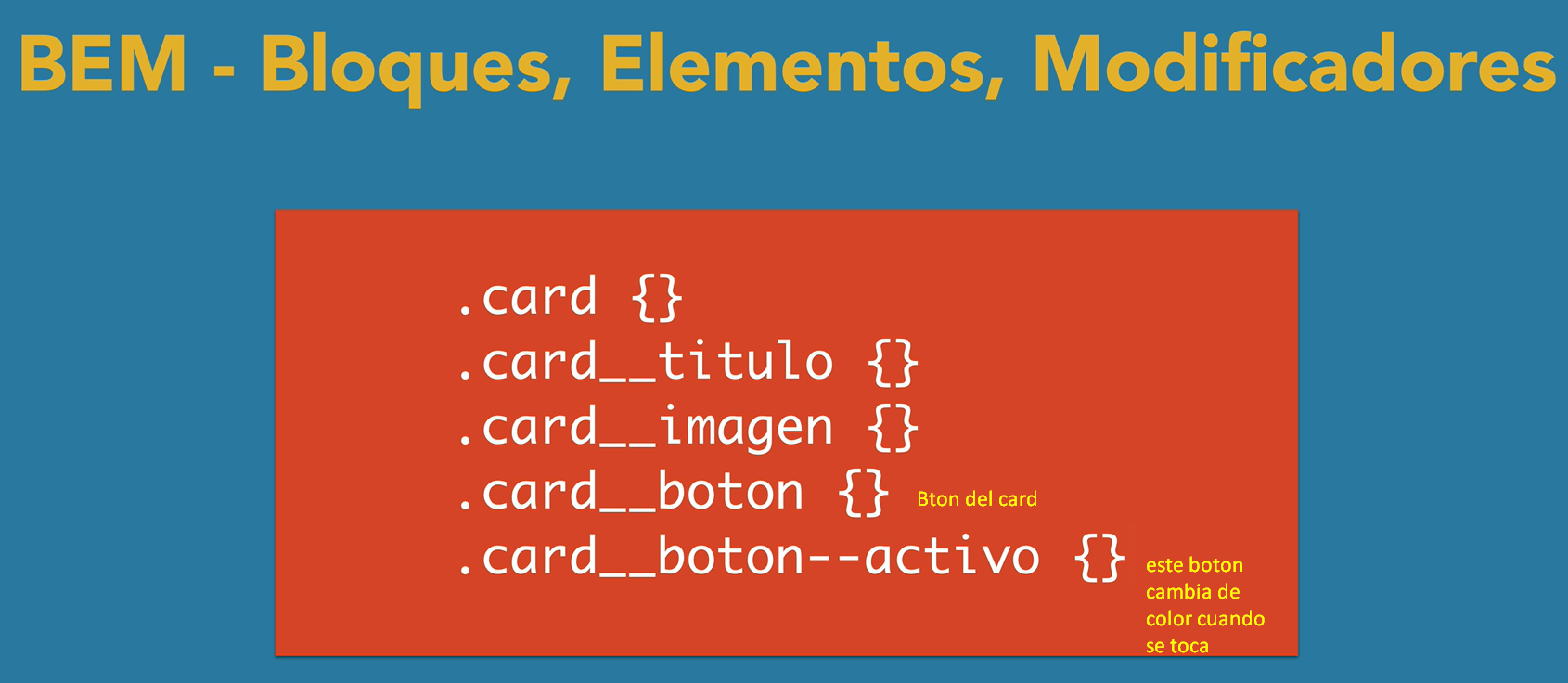
**En el caso de que se use display:flex luego se utiliza justify-content**

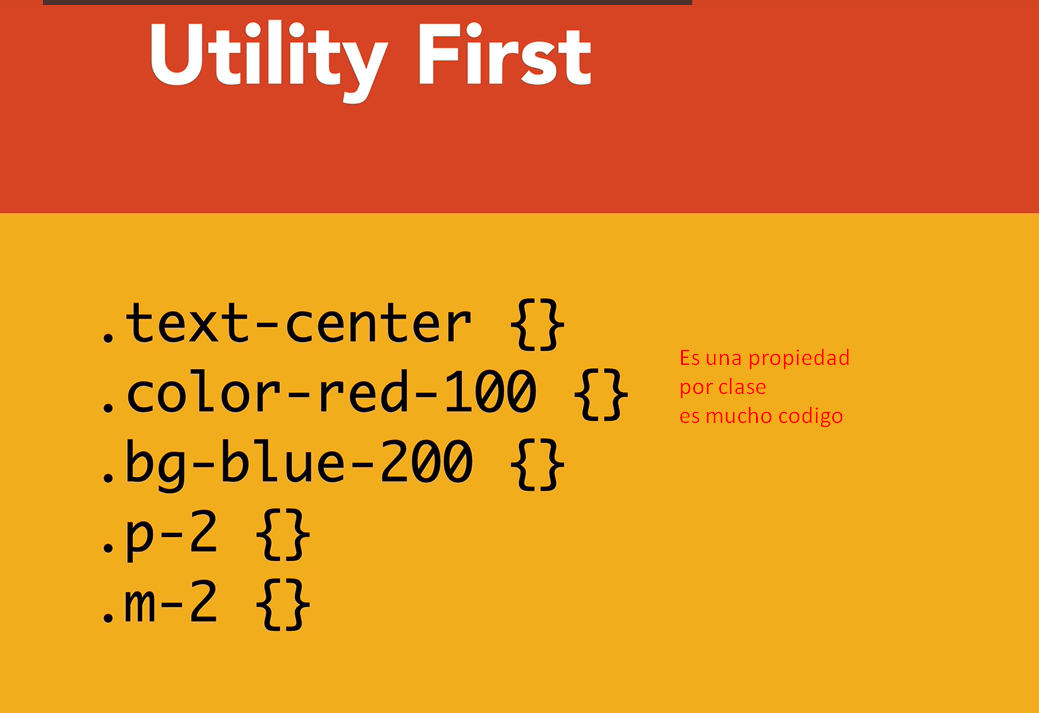
**En el caso de que se use display:column luego se utiliza text-aligne**

**Como escribir un código css: 3 formas:**



1. **Bem:**

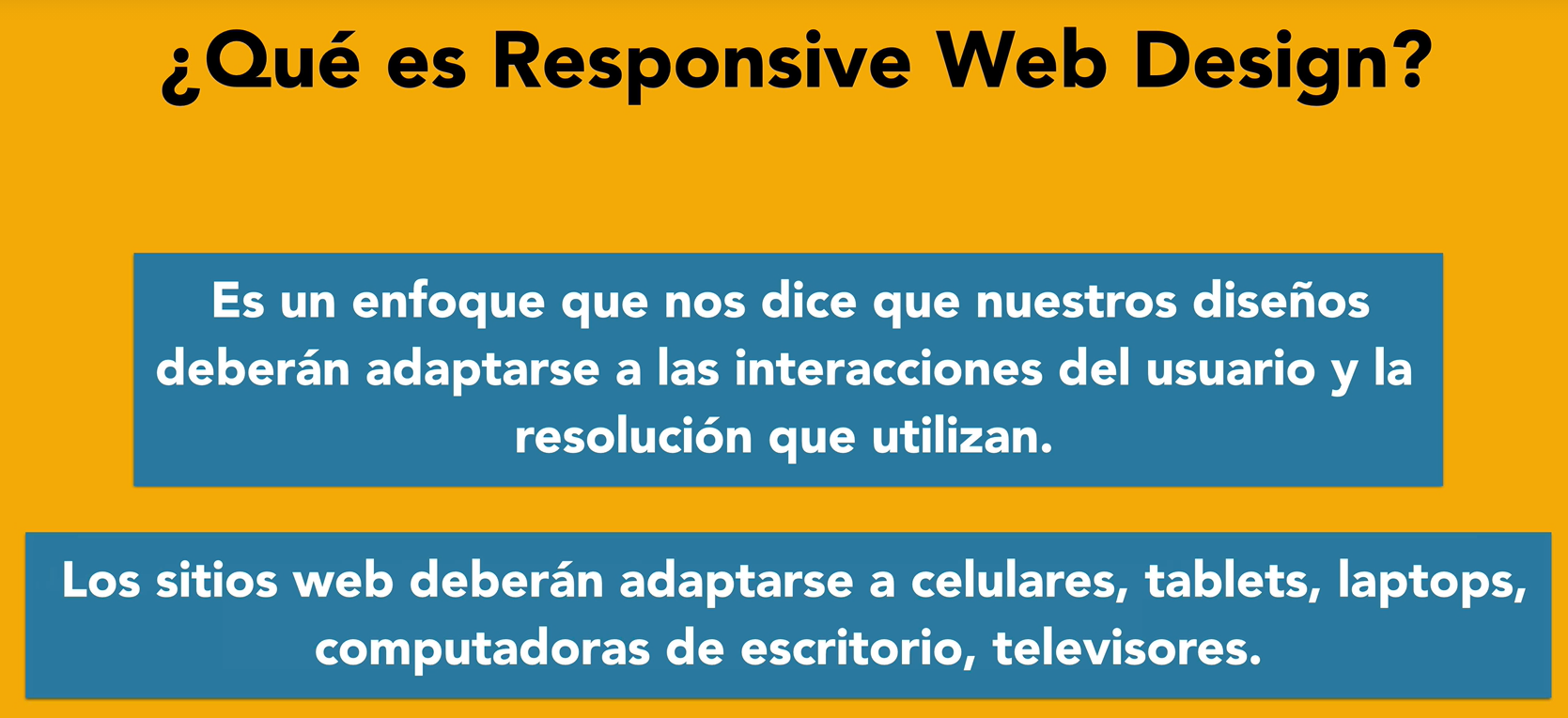




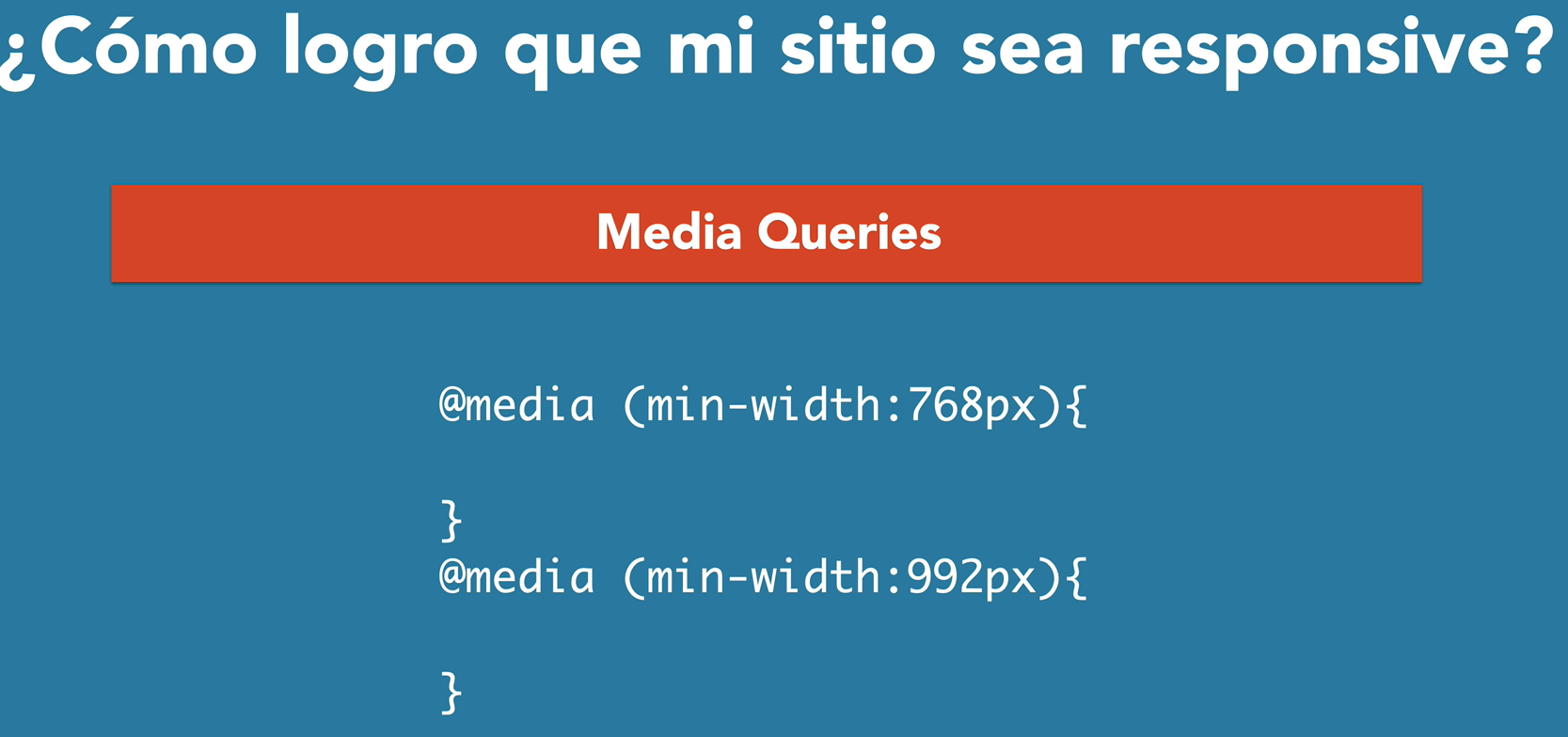


**Yo estoy usando modulos y utilidades.**

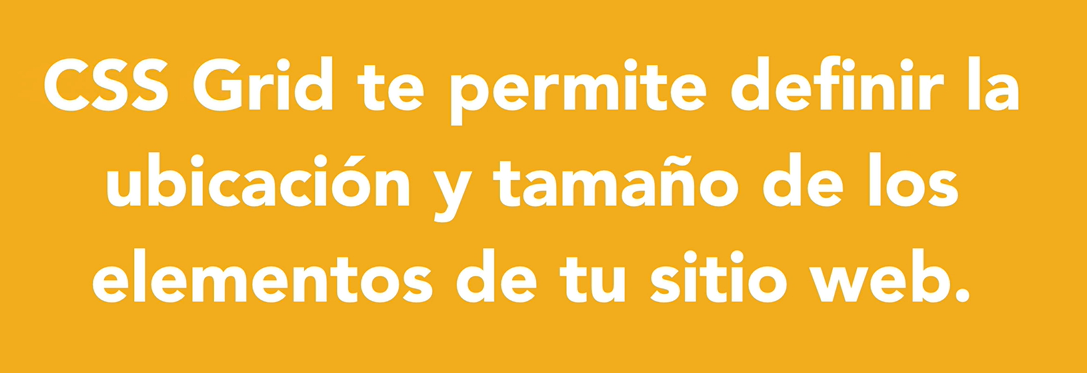
**Responsive:**

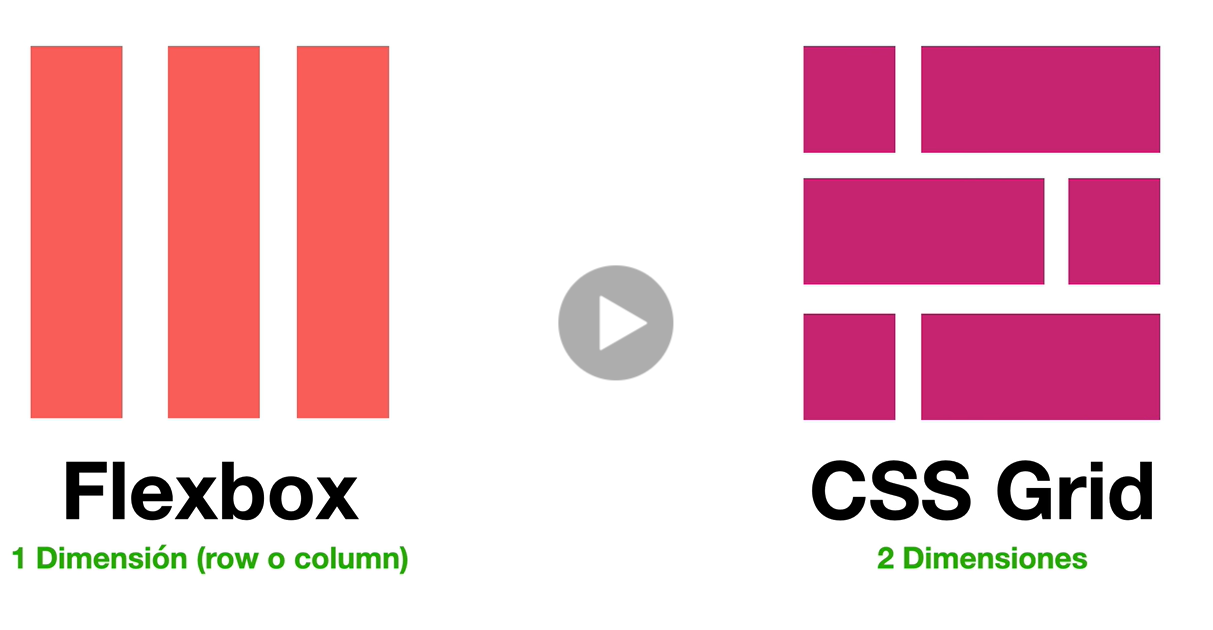


**La solución es media querys:** Se utilizan condiciones, según el tamaño de la pantalla en que se visualiza la web.

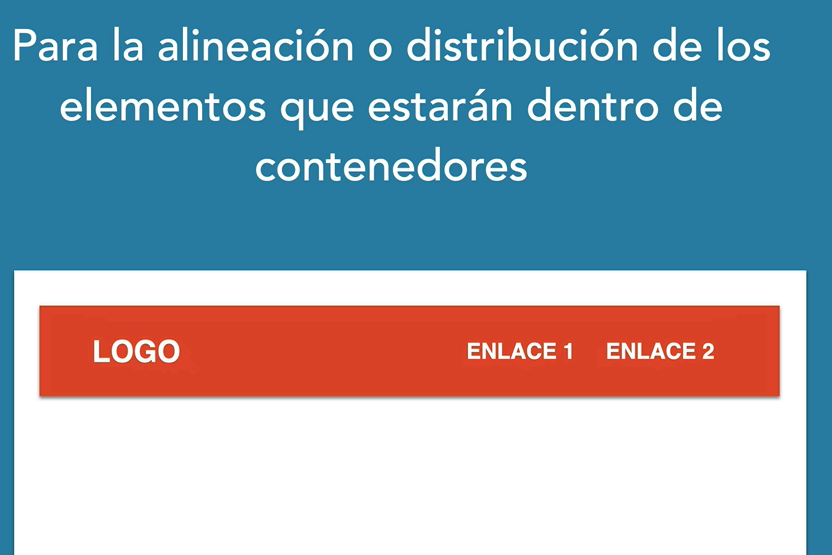


**CSS grid: para seprar en columnas la información**

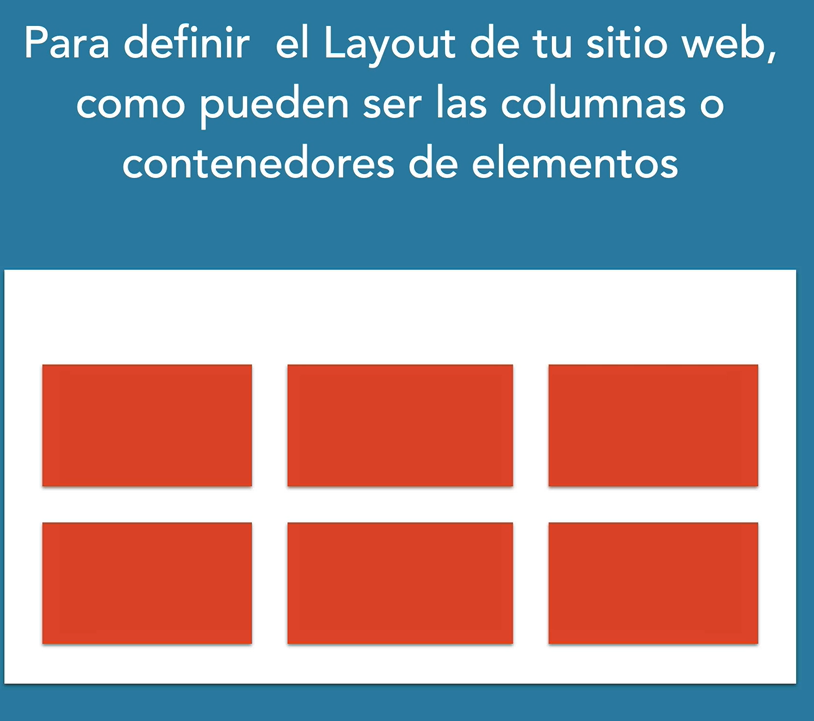




**Utiliza flexbox en la barra de navegación:**



**Utiliza css grid dentro del contenido del main:**



**Crear un atajo de código:**

**Ctrl+shift+p🡪 use snipers 🡪 css.json enter**

**En el css escribimos lo siguiente:**

   "crea un media query": {

        "prefix": "media",

        "body": "@media (min-width: $1) {\n $2 \n}"

    }

**Estandar para distintos dispositivos:**



**Para tener distintos dispositivos visuales al mismo tiempo descargar este programa: https://responsively.app/**

**Atajo para escribir lo mismo varias veces:**

Ctrl+d **(no me funciona)**