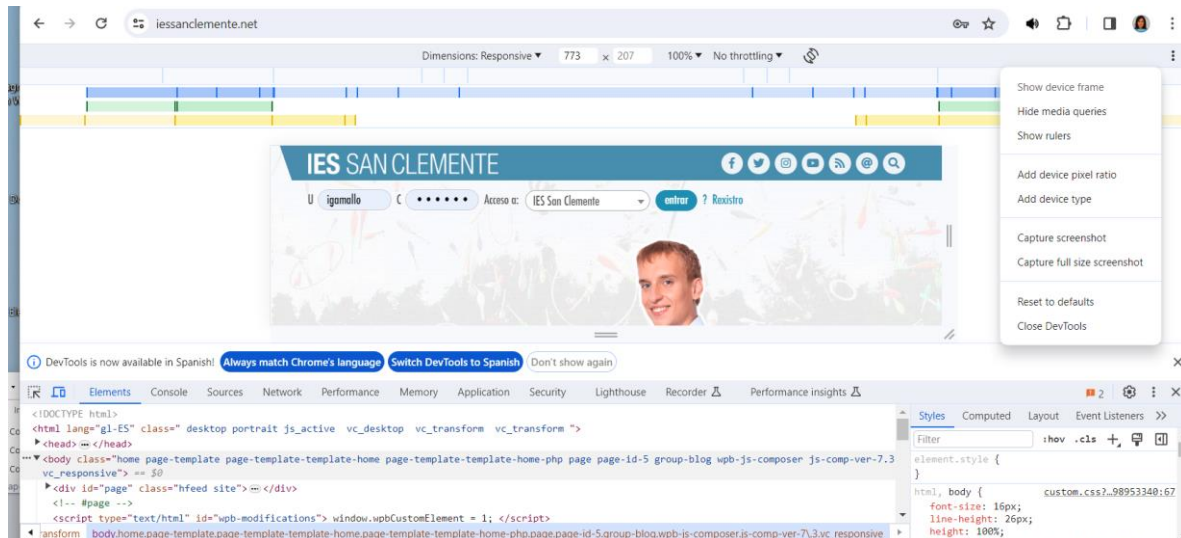


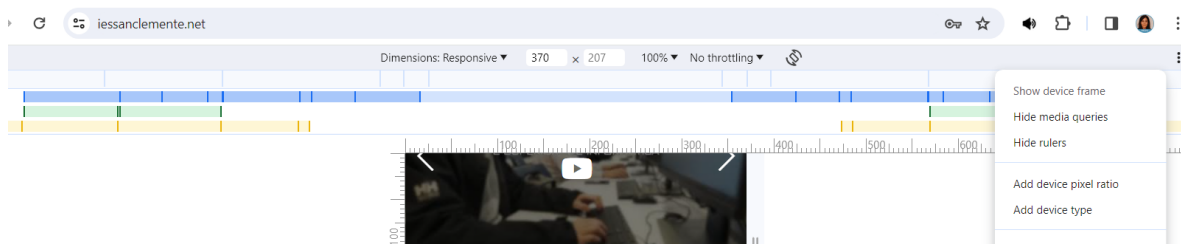
DESEÑO RESPONSIVO

Web vexa ben en todos os dispositivos, sen scrolls horizontais nen dificultar a navegación.

Ferramentas de desenvolvedor (desarrollador) Firefox ou Crhome



Distintas opcións no menú dos 3 puntos



Extensión para axudarnos ao escribir html e css en VSCODE

- Auto CloseTag (cerro etiquetas)
- Auto Rename Tag (correxir nome das etiquetas en caso de erro)
- GitLens (integrar Visual Studio Code con Git e GitHub)
- Intellisense for CSS (autocompleta as regras css no html)
- Prettier-Code Formatter (axuda a formatar)

Info de referencia

1. Mozilla developers: https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design
2. W3schools: https://www.w3schools.com/html/html_responsive.asp
3. Firefox <https://web.dev/articles/responsive-web-design-basics?hl=es>

Diseño Responsive

Viewport

Área de pantalla onde o navegador pode amosar contido.

En desktop viewport coincide coa pantalla do navegador. Os móbiles e tablets teñen ancho de pantalla menor, implementáronse dous viewport

- layout-viewport (para css)
- visual-viewport (para o usuario)

Solución inicial (aínda que non teñamos regras responsivas), engadimos a etiqueta

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

- Distintos tipos de pantallas
- Anchura igual á anchura do dispositivo
- Escalamos o layout para o tamaño (tería que facer zoom-in ou zoom-out)

Investigar máis de viewports e pantallas <https://vimeo.com/52851511>

Con esa liña, non descuadra nin hai scroll horizontal

Exemplo: imaxe con tamaño 100% e liña viewport

Adaptable


250 × 480

DPR: 1....

Sin limitación

UA: Agente de usuario personalizado


EL VIEWPORT



Lorem ipsum dolor sit amet
consectetur adipisicing elit.
Blanditiis tempore aut debitis
consectetur rerum, distinctio ab qui
aliquam magni quis quo unde
cupiditate facilis. Nemo
exercitationem enim maiores ad
obcaecati! Lorem, ipsum dolor sit
amet consectetur adipisicing elit.
Quibusdam, quis recusandae sequi
est asperiores reiciendis fuga culpa

Los cambios en la simulación de dispositivos requieren una recarga para aplicarlos por completo. Las recargas automáticas están deshabilitadas de forma predeterminada para evitar perder cambios en DevTools. Puede habilitar la recarga a través del menú Configuración.

EL VIEWPORT



Lorem ipsum dolor sit amet consectetur adipisicing elit. Blanditiis tempore aut debitis consectetur rerum, distinctio ab qui aliquam magni quis quo unde cupiditate facilis. Nemo exercitationem enim maiores ad obcaecati! Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quibusdam, quis recusandae sequi est asperiores reiciendis fuga culpa harum sapiente, a amet facilis nam explicabo. Ullam voluptate placeat praesentium. Neque, tempore.

Breakpoints e Tipos de deseño

Tipos de deseños ou layouts:

- Fixed: indicamos o ancho de páxina fixado en píxeles.
- Elastic: Aos elementos do meu site asignolle ancho fixo expresado en ~~px~~ ^{rem o em} (múltiplos do tamaño por defecto da letra)
- Fluid/liquido /relativo: o ancho de páxina depende do tamaño do viewport do usuario. Exprésase en porcentaxe.

Deseño fixed:

- Ventaxa: control total
- Desventaxa: scroll horizontal se non escalamos
- Útil se non quero cambiar proporcións en ningún dispositivo

Deseño Elastic:

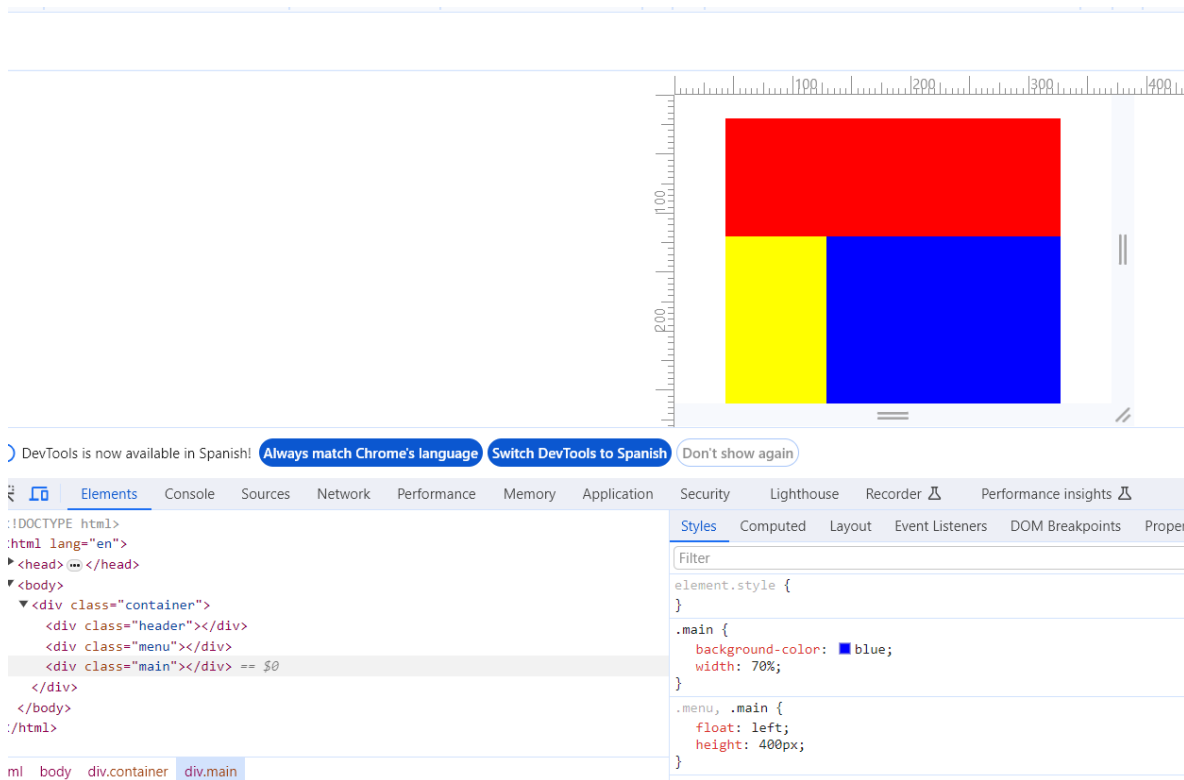
- Ventaxa: mantéñense as proporcións
- Desventaxa:
 - o Dificultar dimensións reais
 - o Eml calculanse en función do pai (rem) Se teño distintos tamaños de letras, teño que calcular como se ven os tamaños calculados .
 - o Non hai fluidez cando cambia tamaño pantalla. Non se adapta nen

Deseño Elastic:

- Ventaxa: adaptanse ao viewport, ao tamaño que está a ver o usuario.
- É o que se usa para o deseño responsivo. Expresamos en porcentaxes.

Menú ten: width:30% e height:400px

Main width: 70% e height:400px



```
.container {
  margin: 20px auto;
  width: 80%;
}
```

O container tem ancho 80% do viewport

Breakpoint é a anchura na que cambiará o layout.

TÍPICOS BREAKPOINT

- ▶ <576px (pantallas pequeñas)
- ▶ 576px-768px (móviles apaisados)
- ▶ 768px-992px (tablets)
- ▶ 992px-1200px (desktops)
- ▶ >1200px (pantallas grandes)

Media Querys

As media-queries son:

...módulo de CSS3 que nos permite adaptar a representación do contido ás características do dispositivo...

Síntaxis de las Media-Queries

As media queries son expresións nas que indicamos un tipo de medio e unha consulta en relación ás características do dispositivo como alto, ancho e mesmo cor:

Algo similar ao seguinte:

```
@media mediatype [condicions] {  
    .... /* Estilos para esas condiciones */ .....  
}
```

Os distintos tipos de valores que podo ter para **mediatype** son:

- all
- screen (o mais usado)
- tty (terminal)
- print (para impresoras)
- tv (para pantallas de televisión)
- projected (para proxectar contidos)
- braille (para dispositivos para invidentes)
- etc...

En canto ás **condicions** que podo consultar:

- width | min-width | max-width
- height | min-height | max-height
- orientation (landscape / portrait) (apaisado-retrato)
- aspect-ratio | min-aspect-ratio | max-aspect-ratio
- color | min-color | max-color

Estas condicions pódense combinar e modificar utilizando cláusulas como:

- And
- Not
- All
- Only

Se xuntamos todo podemos ver algúns exemplos:

```
/* Estilos para todo tipo de pantallas cunha anchura máxima de 576px */  
@media all and (max-width: 576px) {  
    .....  
}
```

```

/* Estilos para pantallas con alomenos 992px de anchura e que estén apaisadas (máis ancho que alto)*/
@media screen and (min-width: 992px) and (orientation: landscape) {
    .....;
}

/* Estilos sólo para pantallas que teñan cando menos 768px de ancho*/
@media only screen and (min-width: 768px) {
    .....;
}

```

Follas de estilos diferentes

Podemos usar media-queries na etiqueta link para seleccionar follas de estilos diferentes atendendo ás características do dispositivo.

Por exemplo:

```

<!--Para pantallas de ata 576px de ancho -->
<link rel="stylesheet" media="(max-width: 576px)" href="small.css" />

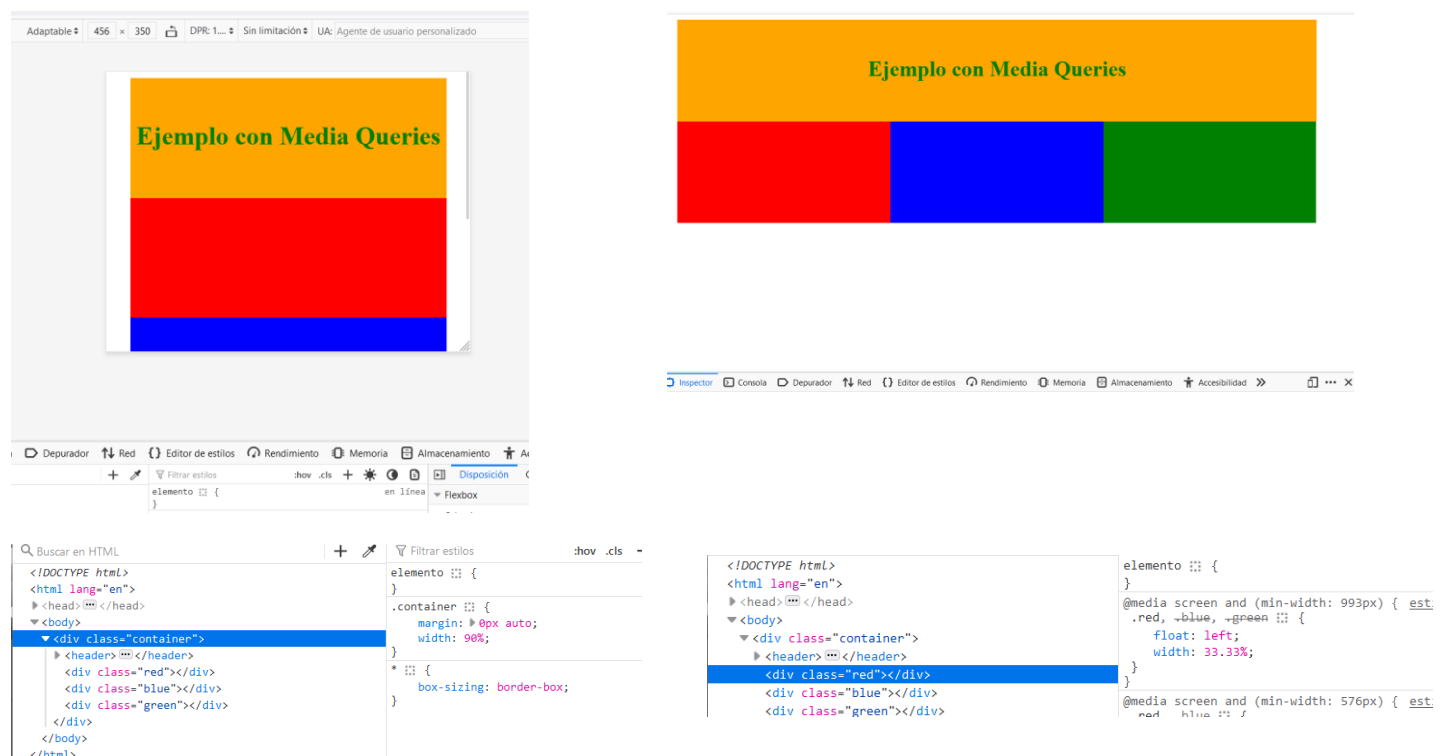
<!--Para pantallas entre 576px e 768px de ancho -->
<link rel="stylesheet" media="(max-width: 768px)" href="medium.css" />

<!--Para pantallas entre 768px e 992px de ancho -->
<link rel="stylesheet" media="(max-width: 992px)" href="large.css" />

```

Exemplo

Para pantallas pequenas



The screenshot shows a web browser window with a yellow header containing the text "Ejemplo con Media Queries". Below the header, there are three colored boxes: red, blue, and green. The browser's developer tools are open, showing the HTML structure and the CSS media queries being applied.

The HTML structure shown in the developer tools is:

```

<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
<div class="container">
<div class="red"></div>
<div class="blue"></div>
<div class="green"></div>
</div>
</body>
</html>

```

The CSS media queries shown in the developer tools are:

```

@media screen and (min-width: 993px) {
  .red, .blue, .green {
    float: left;
    width: 33.33%;
  }
}

@media screen and (min-width: 576px) {
  .red, .blue, .green {
    float: left;
    width: 33.33%;
  }
}

```

```

@media screen and (min-width: 576px) {
  .red,
  .blue {
    float: left;
    width: 50%;
  }

  .green {
    float: left;
    width: 100%;
  }
}

@media screen and (min-width: 993px) {
  .red,
  .blue,
  .green {
    float: left;
    width: 33.33%;
  }
}

@media screen and (orientation: landscape) {
  header {
    background-color: orange;
    color: green;
  }
}

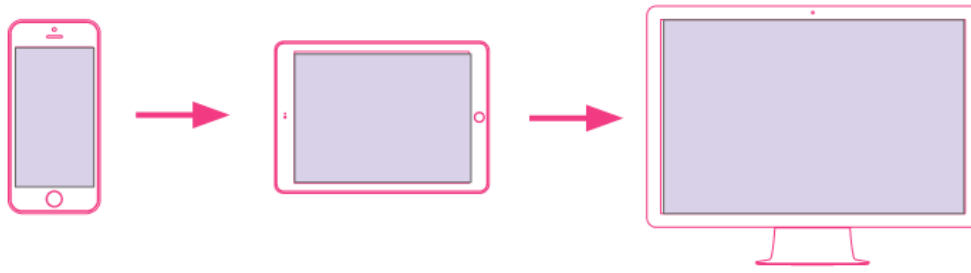
```



Patrons Responsivos

Afrontar un proxecto que requira o desenvolvemento dunha páxina web

responsiva non é fácil e hai varias estratexias posibles para afrontalo. No entanto xa é comunmente aceptado que o máis recomendable é o seguinte:



É dicir, que o proceso de deseño da nosa páxina web debe empezar facendo que todo quede correctamente **en pantallas pequenas**. Desta maneira:

- Priorizo sempre o que é importante. Se o proceso fóra ao revés é moi fácil que caíamos no erro de quitar cousas que sexan realmente importantes.
- Pregúntome en cada deseño se é necesario un deseño novo para pantallas máis grandes.
- Elixo os breakpoints máis adecuados.

No noso proceso de deseñar web responsivas debemos coñecer o que se coñece como patróns responsivos.

Os patróns responsivos son solucións que se deron por boas para o problema de deseñar páxinas web responsivas. Hai moito pero os máis comúns son o seguintes:

- Column Drop
- Mostly Flúide
- Layout Shifter
- Off Canvas
- ...
- Mesturar varios deles
- Pequenos axustes (Tiny Tweaks)

A continuación pasamos a comentar os principais, o catro primeiros para os que hai un exemplo nesta mesmo #cartafol deste repositorio:

Column Drop

É o patrón máis básico e consiste en que en cada breakpoint vaise a apilando un elemento

Mostly Flúide

Parecido a Column Drop. É unha cuadrícula flúida e en cada breakpoint hai redimensionamiento de varias columnas.

Layout Shifter

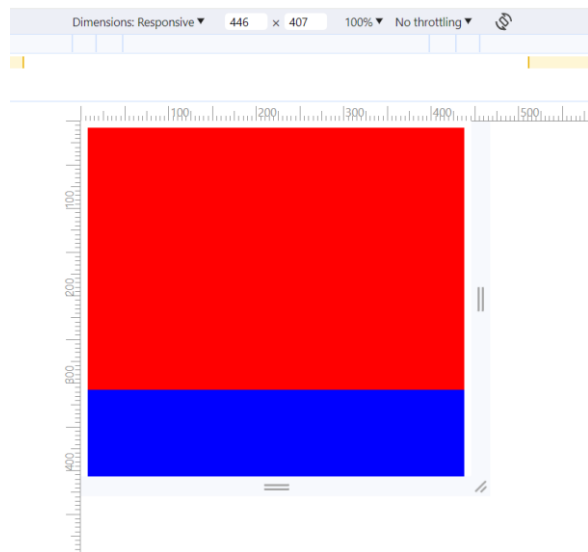
É o patrón máis responsivo, en cada breakpoints cambia o deseño do layout, non unicamente o fluxo e a anchura dos elementos.

Off Canvas

No canto de apilar contidos estes colócanse fóra da pantalla cando o tamaño de pantalla non é o suficientemente grande.

Columnndrops

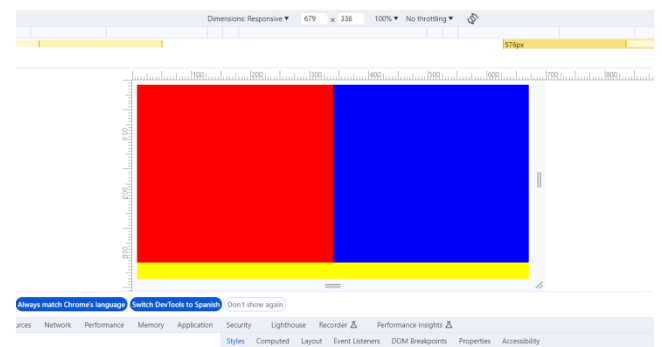
Inicialmente



```
@media screen and (min-width: 576px) {  
  .red {  
    float: left;  
    width: 50%;  
  }  
  
  .blue {  
    float: left;  
    width: 50%;  
  }  
  
  .yellow {  
    float: left;  
    width: 100%;  
  }  
}
```

```
@media screen and (min-width: 992px) {  
  .red {  
    float: left;  
    width: 20%;  
  }  
  
  .blue {  
    float: left;  
    width: 50%;  
  }  
  
  .yellow {  
    float: left;  
    width: 30%;  
  }  
}
```

Definimos 2 breakpoints





Mostly Fluíde

Parecido ao anterior, só que en cada breakpoint hai un redimensionamento non só dunha senón de varias columnas.

```
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    <div class="container">
      <div class="box red"></div>
      <div class="box green"></div>
      <div class="box yellow"></div>
      <div class="box blue"></div>
      <div class="box orange"></div>
    </div>
  </body>
```

Inicial



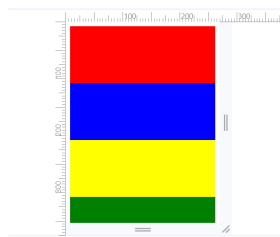
```
@media screen and (min-width: 576px) {
  .green,
  .yellow {
    width: 50%;
  }
}

@media screen and (min-width: 992px) {
  .green,
  .yellow,
  .blue {
    width: 33.3333%;
  }
}
```

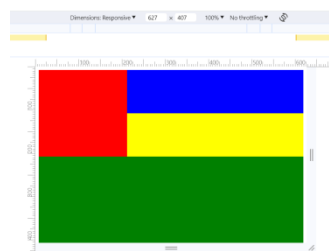


Layout Shifter

Es el más responsivo, cambia posición además de tamaño , de orden...



```
<div class="container">
  <div class="box red"></div>
  <div class="box blue"></div>
  <div class="box yellow"></div>
  <div class="box green"></div>
</div>
```



Vermello
:

inicia en columna 1 acaba en columna 2

inicia fila 1 acaba en fila 3

azul:

inicia en columna 2 acaba en columna 4

inicia fila 1 acaba en fila 2

amarillo:

inicia en columna 2 acaba en columna 4

inicia fila 2 acaba en fila 3

Verde:

inicia en columna 1 acaba en columna 4

inicia fila 5 acaba en fila 5

```
@media screen and (min-width: 576px) {
  .red {
    grid-column: 1 / 2;
    grid-row: 1 / 3;
  }
}
```

```
.blue {
  grid-column: 2 / 4;
  grid-row: 1 / 2;
}
```

```
.yellow {
  grid-column: 2 / 4;
  grid-row: 2 / 3;
}
```

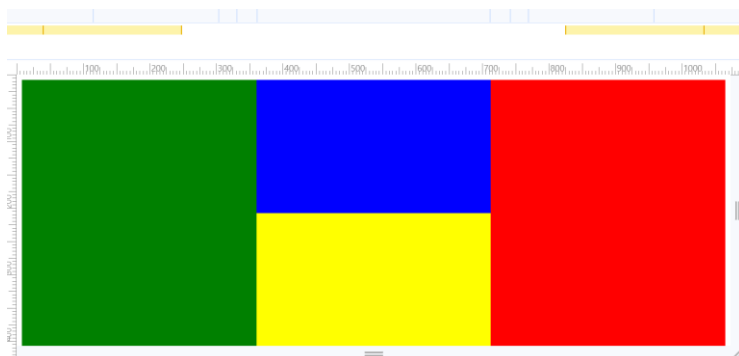
```
.green {
  grid-column: 1 / 4;
  grid-row: 3 / 5;
}
```

```
@media screen and (min-width: 992px) {
  .red {
    grid-column: 3 / 4;
    grid-row: 1 / 5;
  }
}
```

```
.blue {
  grid-column: 2 / 3;
  grid-row: 1 / 3;
}
```

```
.yellow {
  grid-column: 2 / 3;
  grid-row: 3 / 5;
}
```

```
.green {
  grid-column: 1 / 2;
  grid-row: 1 / 5;
}
```



```

.container {
  display: grid;
  margin: 0px auto;
  max-width: 1200px;
  grid-template-columns: 33.333% 33.333% 33.333%;
  grid-template-rows: 100px 100px 100px 100px;
}

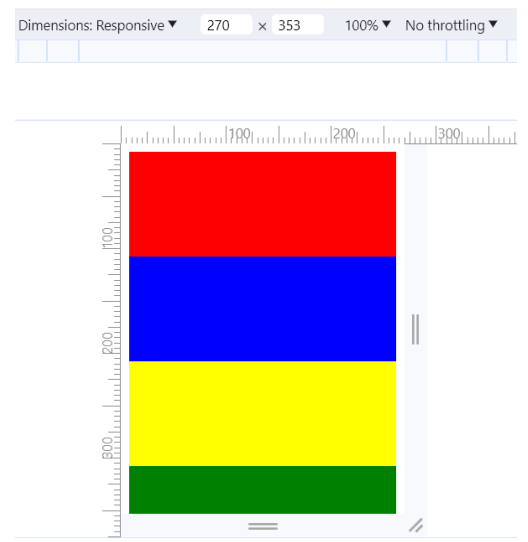
.red {
  background-color: red;
  grid-column: 1 / 4;
  grid-row: 1 / 2;
}

.blue {
  background-color: blue;
  grid-column: 1 / 4;
  grid-row: 2 / 3;
}

.yellow {
  background-color: yellow;
  grid-column: 1 / 4;
  grid-row: 3 / 4;
}

.green {
  background-color: green;
  grid-column: 1 / 4;
  grid-row: 4 / 5;
}

```



Off Canvas

Off Canvas

En vez de apilar contenido, saca contenido fuera de pantalla si no es suficientemente ancho.

```
<div class="container">

  <div class="red"></div>

  <div class="blue"></div>

</div>
```

Red inicialmente está oculto (transición 0.5 segundos)

```
.red {

  background-color: red;

  height: 100%;

  position: absolute;

  transform: translate(-300px, 0);

  transition: transform 0.5s ease;

  width: 300px;

}
```

Cando pantalla maior de 768px, amosamos

```
@media screen and (min-width: 768px) {

  .red {

    transform: translate(0px);

  }

}
```

