

Lenguaje de Marcas y Sistemas de Gestión

UA 2.15 - Introducción CSS: Responsive Web Design



Centro Profesional
**Universidad
Europea**

Raúl Rodríguez Mercado

raul.rodriguez@universidadeuropea.es / @raulrodriguezue

Dpto. Ciencias y Tecnología de la Informática y Comunicación

UA 2.15: Introducción CSS: Responsive Web Design



Objetivos

- Definir qué es CSS
- Conocer la estructura de las hojas de estilo y como se aplican a los documentos HTML



UA 2.15: Introducción CSS: Responsive Web Design



Contenidos

- Definir Responsive Web Design
- Características del RWD
- El “viewport” en HTML5 y CSS3
- Breakpoints y tipos de diseño
- Media Queries
- Patrones Responsives
- Imágenes, Tablas y Texto Responsive

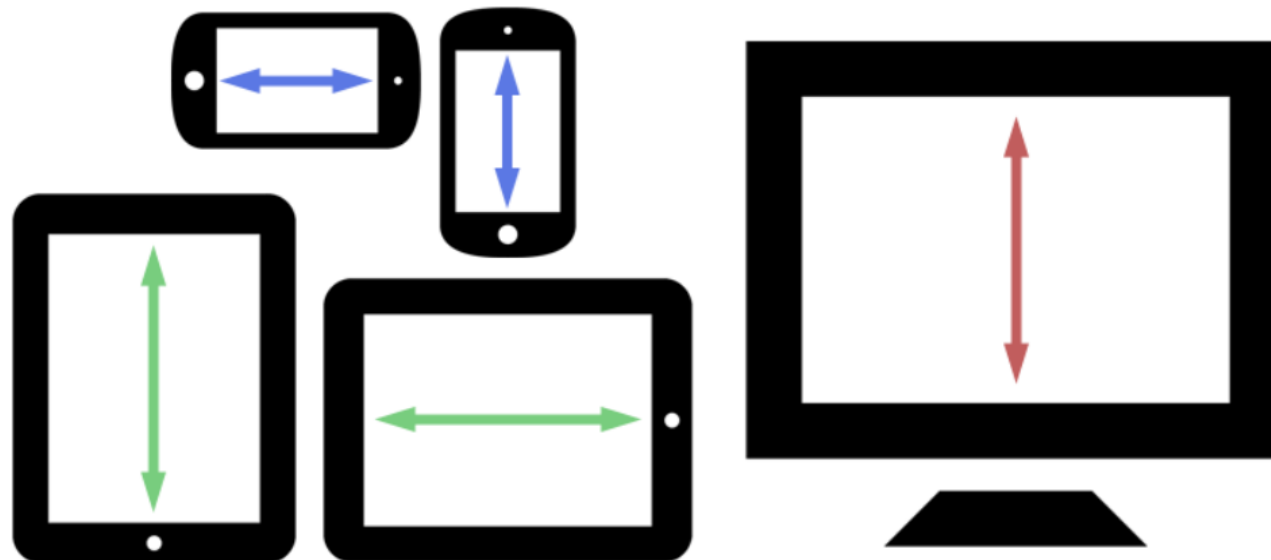


UA 2.15: Introducción CSS: Responsive Web Design



Definición de Responsive Web Design

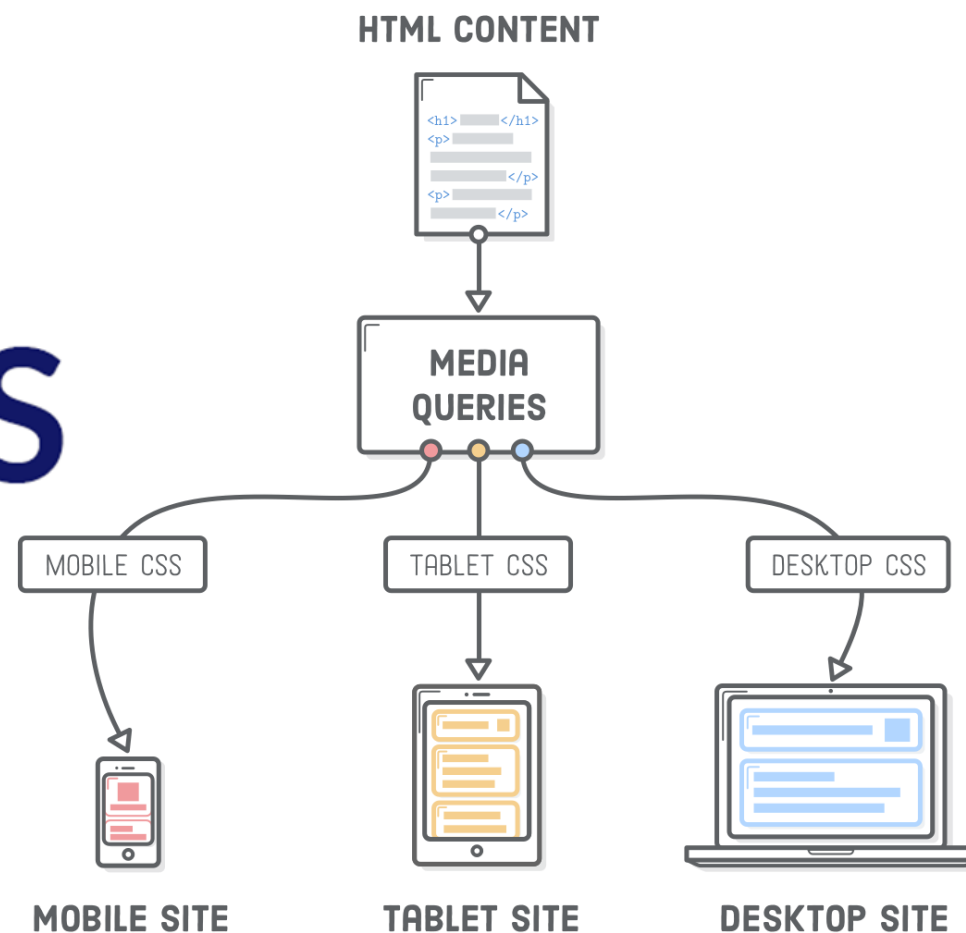
- Definición Wikipedia: “...una aproximación al **diseño web** que permite que nuestras página web se vean **correctamente** en una variedad de dispositivos y de tamaño de pantalla...adaptando el layout al entorno de visualización...”
- Se le denomina **diseño adaptativo**(Responsive Web Design o RWD) a los diseños web que tienen la capacidad de adaptarse al tamaño y formato de la pantalla en la que se visualiza el contenido, respecto a los diseños tradicionales.



¿Cómo se hacen estas adaptaciones?

CSS+ ETIQUETAS+ MEDIA QUERIES

(CSS3)



¿Por qué es necesario utilizarlo?

...**demoted** in rankings or
displayed with a **warning**
in mobile search results...



<https://developers.google.com/search/mobile-sites/mobile-seo>



UA 2.15: Introducción CSS: Responsive Web Design

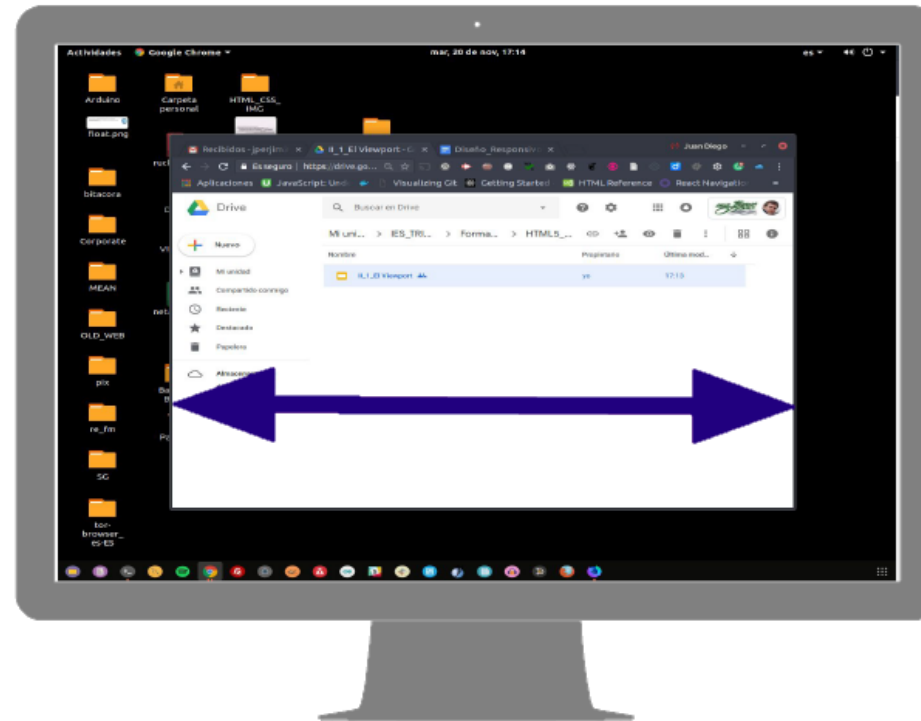


Definición de Viewport

- Área de pantalla en la que el navegador puede renderizar contenido, es decir, e espacio disponible para mostrar mi página web.

VIEWPORT EN PC / LAPTOPS

- En DESKTOP es fácil, el viewport es la pantalla de nuestro navegador



Definición de Viewport

VIEWPORT EN TABLETS / MÓVILES

- Cuando los móviles/tablets irrumpieron la páginas se maquetaban en un longitud fija en pixels que variaba entre 800px y 1000px y que, obviamente, era mucho mayor que las pantallas de los móviles.
- Ante esta circunstancia los fabricantes hicieron que este tipo de dispositivos tuvieran dos Viewports:
 - ✓ El *Layout-viewport* que es el viewport que es tenido en cuenta para la aplicación de estilos. Suele ser de aproximadamente 900px. → Para CSS
 - ✓ El *Visual-viewport* que es el viewport que realmente ve el usuario cuando está navegando. → Para el usuario, lo que realmente verá.

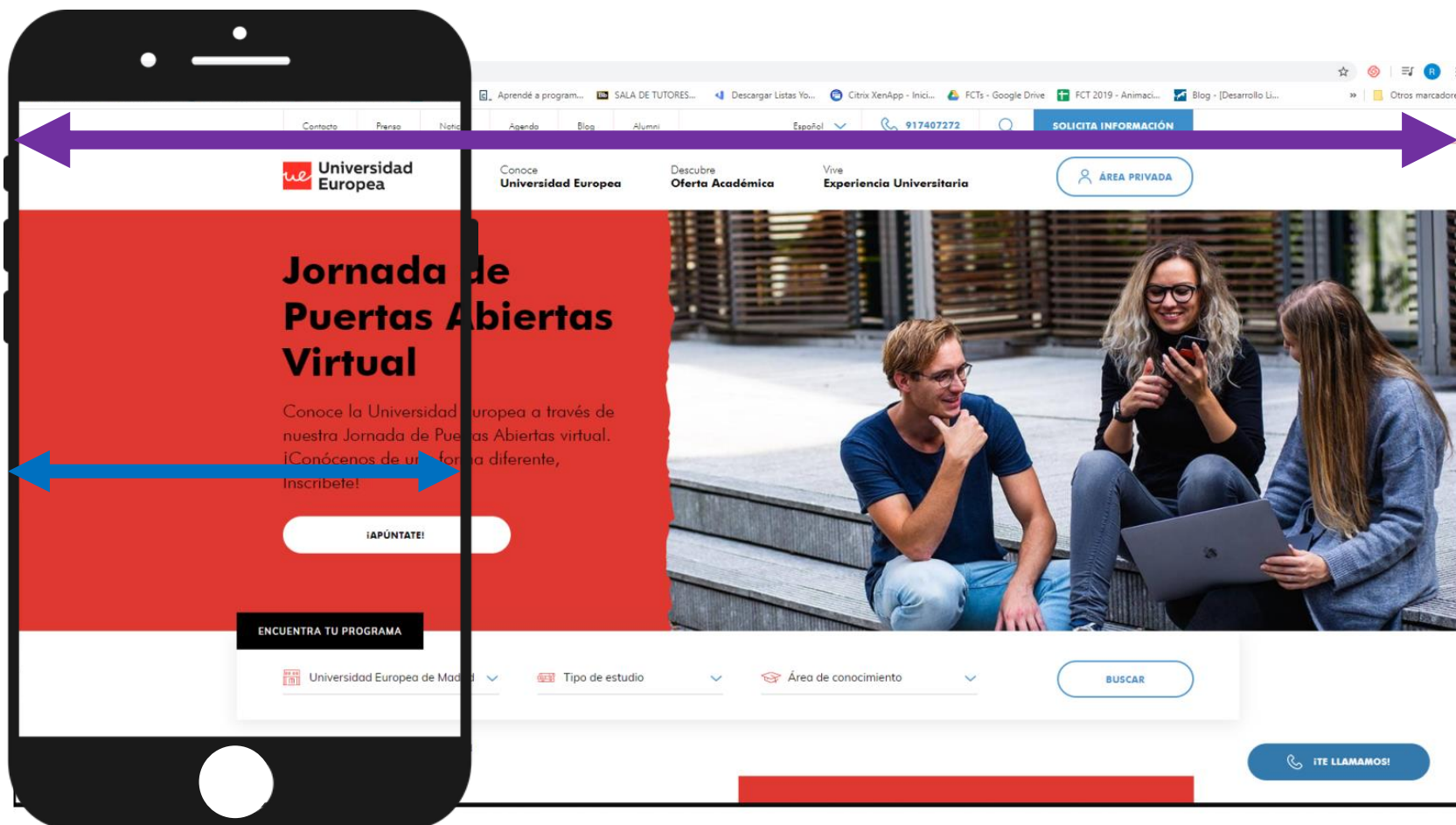
UA 2.15: Introducción CSS: Responsive Web Design



Definición de Viewport

VIEWPORT EN TABLETS / MÓVILES

Visual



Layout

UA 2.15: Introducción CSS: Responsive Web Design



Definición de Viewport

VIEWPORT EN TABLETS / MÓVILES

- Además en los dispositivos móviles puedo hacer **Zoom**. Esto provoca que:
 - ✓ No cambia el layout-viewport
 - ✓ Cambia el visual-viewport

Todo esto es un poco “locura”, a la que además hay que sumar que “un pixel no es un pixel”...



UA 2.15: Introducción CSS: Responsive Web Design



Viewport: Un pixel no es un pixel

Resolución en las Pantallas

- **Hardware Pixels:** Son los pixels de resolución que nos da la tarjeta gráfica
- **Device Independent Pixels (DPI):** Es la unidad de medida del navegador. Se relaciona con la distancia real y ocupan lo mismo independientemente de la densidad de pixels de la pantalla.
- **Device Pixel Ratio (DPR):** $\text{Hw Pixels} / \text{DPI}$ (en una dimensión).



+Info

Viewport: ¿Cómo solucionar este jaleo?

DISEÑO RESPONSIVO



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

Se debe de poner en la cabecera del HTML, donde los META, aunque no tenga reglas Responsive establecidas.

Content: Anchura del dispositivo y escalo el contenido al layout-viewport

UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

- Una vez hemos definido el importante concepto de **Viewport**, antes de hacer **diseño responsivo** debemos conocer qué **tipos de diseños o layouts** podemos tener.

TIPOS DE DISEÑOS O LAYOUT

- **FIXED:** Anchura de página fijo expresado en pixels.
- **ELASTIC:** Anchura de página fijo expresado en ems (múltiplos del tamaño de letra).
- **FLUID / LIQUID / RELATIVE:** Anchura de página depende del tamaño del viewport del usuario y se expresa en porcentaje (%)



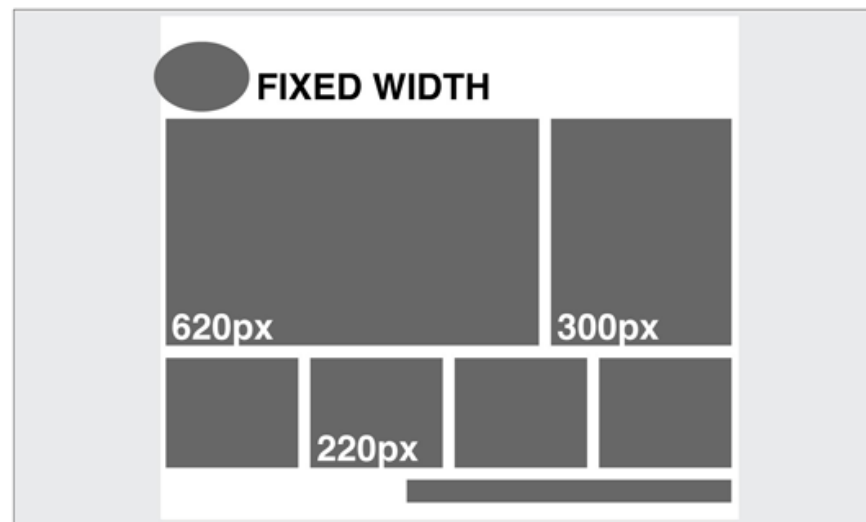
UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

FIXED LAYOUT

- **VENTAJA:** Control total
- **DESVENTAJA:** Scroll horizontal (si no escalo)
- **ÚTIL:** Tiene sentido si no quiero cambiar el layout y sus proporciones en ningún dispositivo (no suele ser el caso para los las pantallas de los móviles).



UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

Ejemplo Fixed Layout

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>FIXED LAYOUT</title>
    <link rel="stylesheet" type="text/css" href="fixed.css" />
  </head>
  <body>
    <div class="container">
      Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod natus illum
      id assumenda possimus perferendis, esse explicabo! Illum facilis dolores,
      quis repudiandae dolorem, ratione itaque eveniet repellendus voluptas esse
      eligendi. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Atque
      ipsum nulla odit labore, autem maiores perspiciatis a dolorem id accusamus
      unde aut sapiente deleniti ut debitis nobis saepe quaerat nostrum.
      
    </div>
  </body>
</html>
```



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

.container {
  border: 10px solid black;
  margin: 20px auto;
  overflow-y: auto;
  width: 1000px;
}

img {
  width: 100%;
}
```



Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod natus illum id assumenda possimus perferendis, esse explicabo! Illum facilis dolores, quis repudiandae dolorem, ratione itaque eveniet repellendus voluptas esse eligendi. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Atque ipsum nulla odit labore, autem maiores perspiciatis a dolorem id accusamus unde aut sapiente deleniti ut debitis nobis saepe quaerat nostrum.



UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

ELASCTIC LAYOUT

- **VENTAJA:** Control al hacer zoom-in y zoom-out. Se mantienen las proporciones
- **DESVENTAJAS:**
 - Al final son fixed.
 - Dificultad de calcular dimensiones reales
 - Ems se calculan en relación al padre (rem)
 - No hay fluidez cuando cambia el tamaño de la pantalla. No se adapta bien.



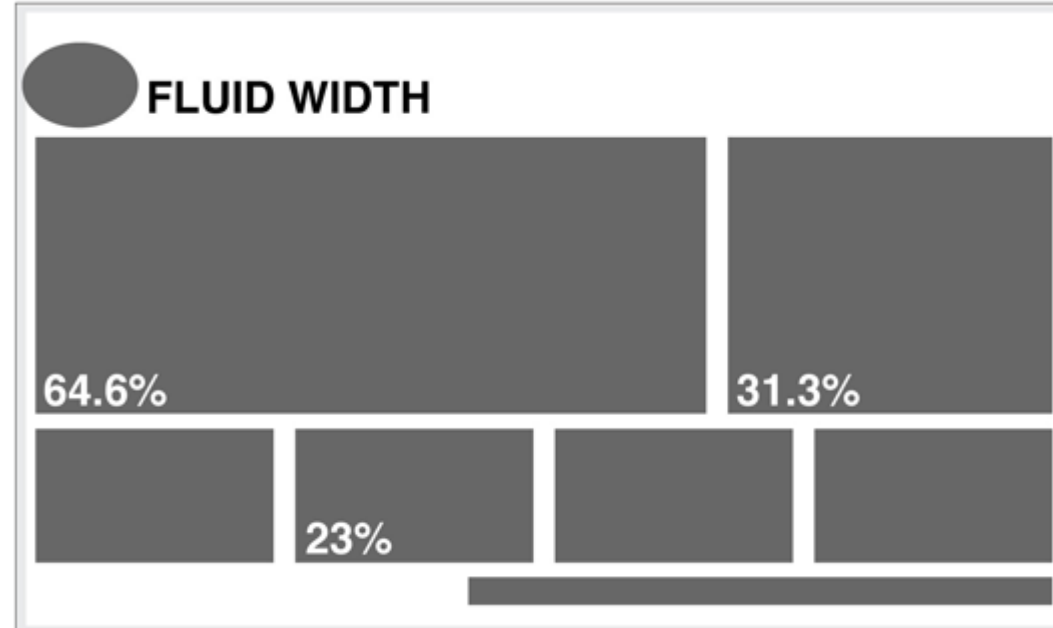
UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

FLUID LAYOUT

- **VENTAJA:** Se adaptan al viewport, al tamaño de lo que está viendo el usuario.
- Es el tipo de Layout que se usa para el **DISEÑO RESPONSIVO**.




UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

Ejemplo de FLUID LAYOUT

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>FLUID LAYOUT</title>
    <link rel="stylesheet" type="text/css" href="fluid.css" />
  </head>
  <body>
    <div class="container">
      <div class="header"></div>
      <div class="menu"></div>
      <div class="main"></div>
    </div>
  </body>
</html>
```



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

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

.header {
  background-color: red;
  height: 100px;
}

.menu,
.main {
  float: left;
  height: 400px;
}

.main {
  background-color: blue;
  width: 70%;
}

.menu {
  background-color: yellow;
  width: 30%;
}
```

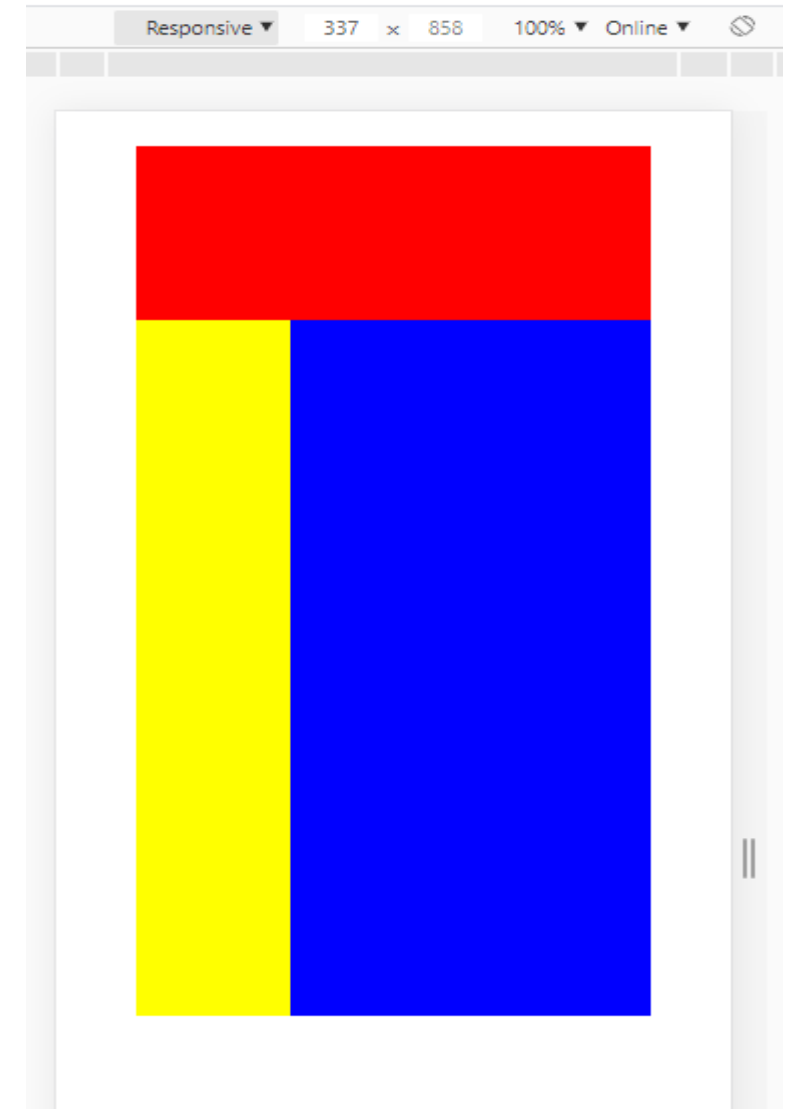


UA 2.15: Introducción CSS: Responsive Web Design



Tipos de Diseños o Layout

Ejemplo de FLUID LAYOUT



UA 2.15: Introducción CSS: Responsive Web Design



Breakpoints

- Una vez hemos establecido que usaremos diseños de tipo *Fluid* debemos recordar que el concepto de diseño responsivo, además de *adaptación* al tamaño implicaba el *cambio* en el diseño atendiendo al tamaño u otras características del dispositivo.
- La anchura de la pantalla en la que se produce el cambio (layout-viewport), es lo que se conoce como *Breakpoint* en respuesta a ciertas condiciones del dispositivo (media queries).
- La elección de unos *breakpoints* adecuados no es una tarea fácil pero tenemos la suerte de que ya ha habido empresas que han hecho estudios muy serios al respecto.
- Destacar los Breakpoints elegidos por [Twitter Bootstrap](#)

UA 2.15: Introducción CSS: Responsive Web Design



Breakpoints

Típicos Breakpoints

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



0-480

Smaller
smartphones



481-768

Tablets & larger
smartphones



769-1279

Laptops, larger tablets
in landscape, and small
desktops



1280+

Larger desktops
and monitors

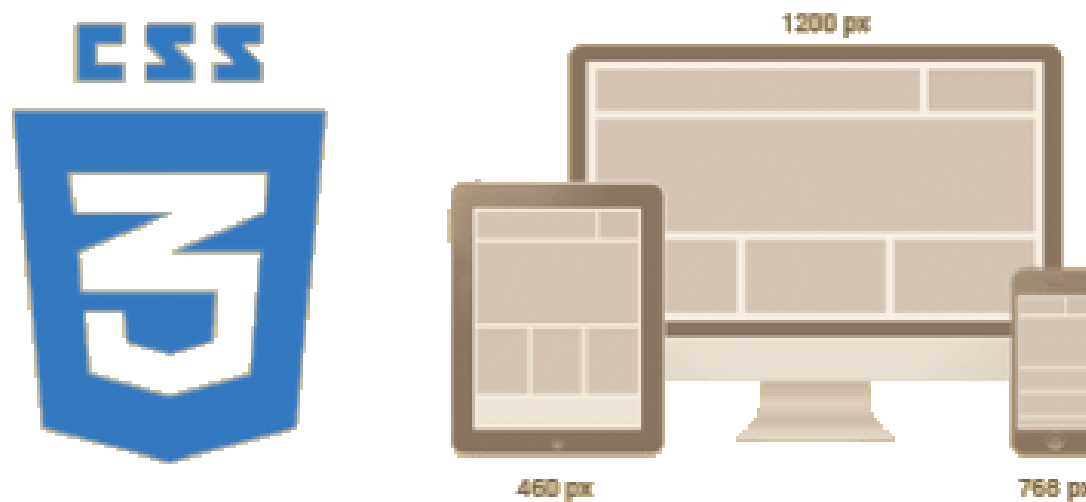
UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Definición

- Es un módulo de CSS3 que permite adaptar la representación del contenido a las características del dispositivo
- Por lo tanto son una parte Principal del Diseño Responsivo



Medias Queries

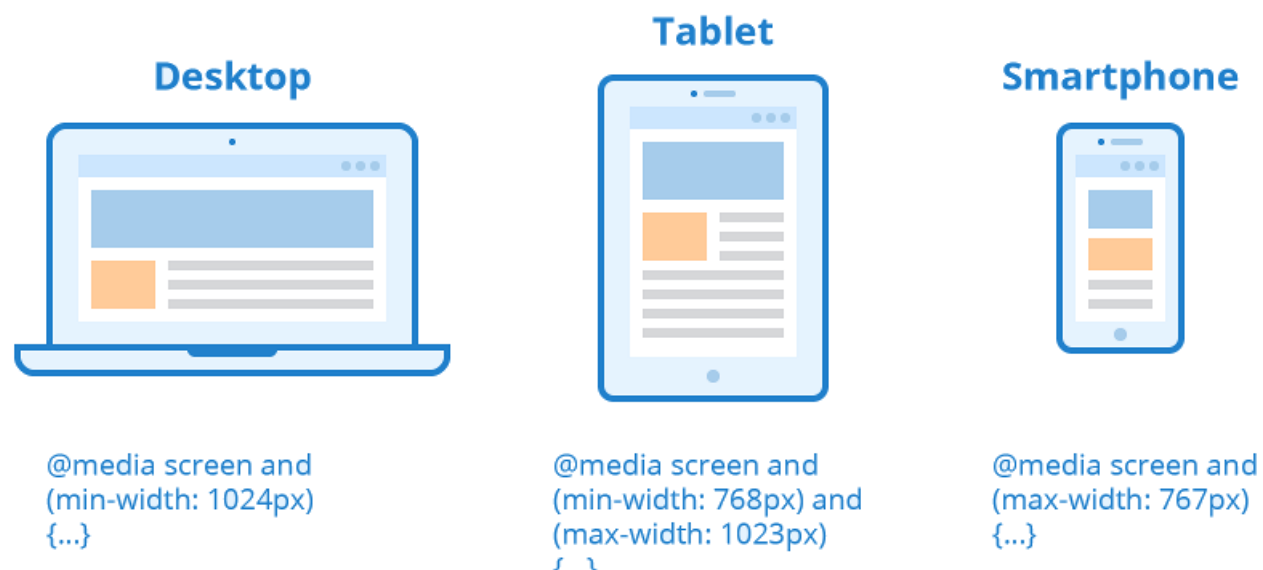
Sintaxis

@media mediatype [condiciones]

Son expresiones en las que indicamos un tipo de medio y una consulta en relación a las características del dispositivo como alto, ancho e incluso el color

TIPOS DE MEDIA(MEDIA TYPES)

- ✓ all → Todos los dispositivos
- ✓ screen → Pantallas
- ✓ tty → Terminales antiguos
- ✓ print → para impresoras
- ✓ tv → Para Televisión
- ✓ projected → Proyector
- ✓ braille → Braille
- ✓ ...



UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Sintaxis: ¿Qué puedo consultar?

- ✓ width | min-width | max-width
- ✓ height | min-height | max-height
- ✓ orientation (landscape / portrait)
- ✓ aspect-ratio | min-aspect-ratio | max-aspect-ratio
- ✓ color | min-color | max-color
- ✓ ...

Condiciones

- ✓ AND
- ✓ NOT
- ✓ ALL
- ✓ ONLY

```
@media (min-width:...) { ... }
```



Mobile First Desktop last

UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Ejemplos de Uso

```
@media (max-width: 576px) {  
  /*  
  CSS STYLES  
  */  
}
```

```
@media screen and (min-width: 992px)  
and (orientation: landscape) {  
  /*  
  CSS STYLES  
  */  
}
```

```
@media only screen and (min-width: 768px) {  
  /*  
  CSS STYLES  
  */  
}
```

UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Ejemplos de Uso en Hojas de Estilos Distintos

```
<link rel="stylesheet" media="(max-width: 576px)" href="small.css" />
```

```
<link rel="stylesheet" media="(max-width: 768px)" href="medium.css" />
```

```
<link rel="stylesheet" media="(max-width: 992px)" href="large.css" />
```

UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Ejemplo Medias Queries

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Ejemplo Media Query</title>
    <link rel="stylesheet" type="text/css" href="mediaQueries.css" />
  </head>
  <body>
    <div class="container">
      <header><h1>Ejemplo con Media Queries</h1></header>
      <div class="red"></div>
      <div class="blue"></div>
      <div class="green"></div>
    </div>
  </body>
</html>
```



UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Ejemplo Medias Queries

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

.container {
  margin: 0px auto;
  width: 90%;
}

.container > * {
  height: 150px;
}

header {
  background-color: yellow;
  padding-top: 2em;
  text-align: center;
}

.red {
  background-color: red;
}

.blue {
  background-color: blue;
}

.green {
  background-color: green;
}
```

```
@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;
  }
}
```

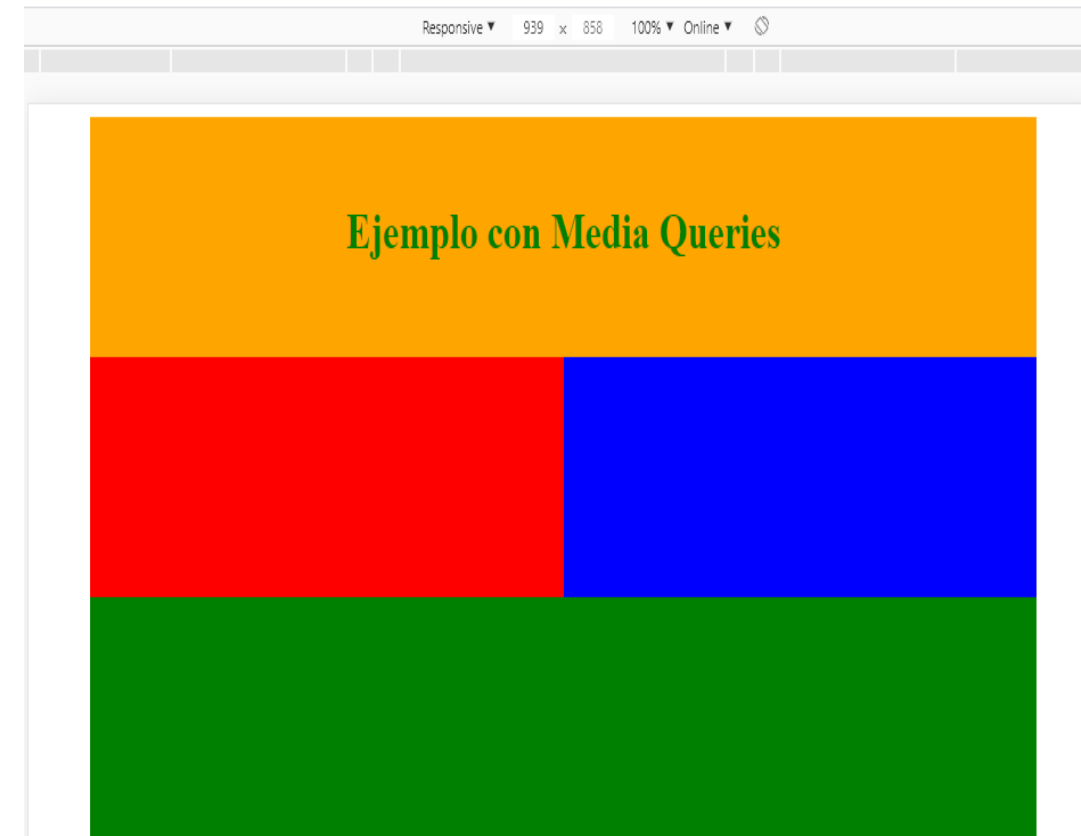


UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Ejemplo Medias Queries



UA 2.15: Introducción CSS: Responsive Web Design



Medias Queries

Ejemplo Medias Queries



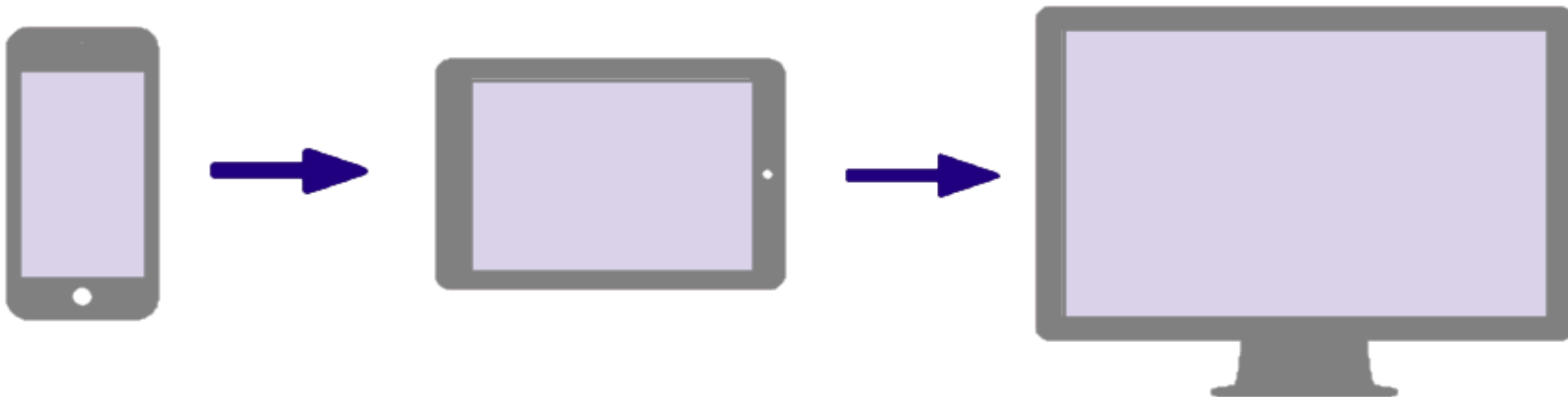
UA 2.15: Introducción CSS: Responsive Web Design



Patrones Responsive

¿Cómo empezamos?

- Lo primero que debo de hacer, es hacer mi diseño del dispositivo más pequeño al más grande → **Start Small!!!**
- Priorizo siempre lo que es importante: Al revés es muy fácil quitar cosas importantes.
- Me pregunto en cada diseño si es necesario un nuevo diseño para pantallas más grandes y elijo los breakpoints necesarios.



UA 2.15: Introducción CSS: Responsive Web Design



Patrones Responsive

Definición

Los **patrones** son **soluciones** que se han dado por buenas para cierto tipo de **problemas**. En nuestro caso, el diseño de páginas **web responsivas**.

PATRONES DE DISEÑO COMUNES

- ✓ COLUMN DROP
- ✓ MOSTLY FLUID
- ✓ LAYOUT SHIFTER
- ✓ OFF CANVAS
- ✓
- ✓ MEZCLA DE VARIOS...
- ✓ PEQUEÑOS AJUSTES (TINY TWEAKS..)

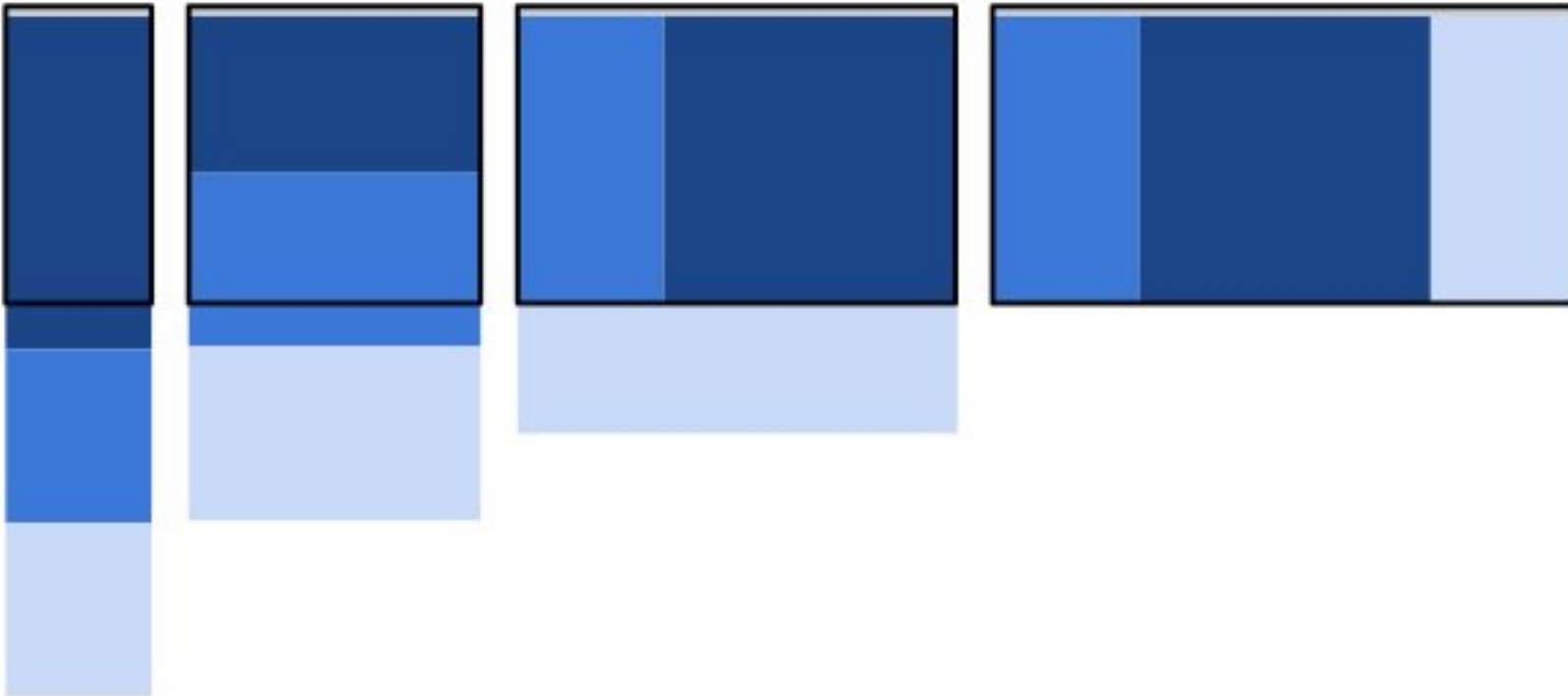
UA 2.15: Introducción CSS: Responsive Web Design



Patrones Responsive

COLUMN DROP

- ✓ Es el patrón más básico y básicamente consiste en que en cada breakpoint se va apilando un elemento.



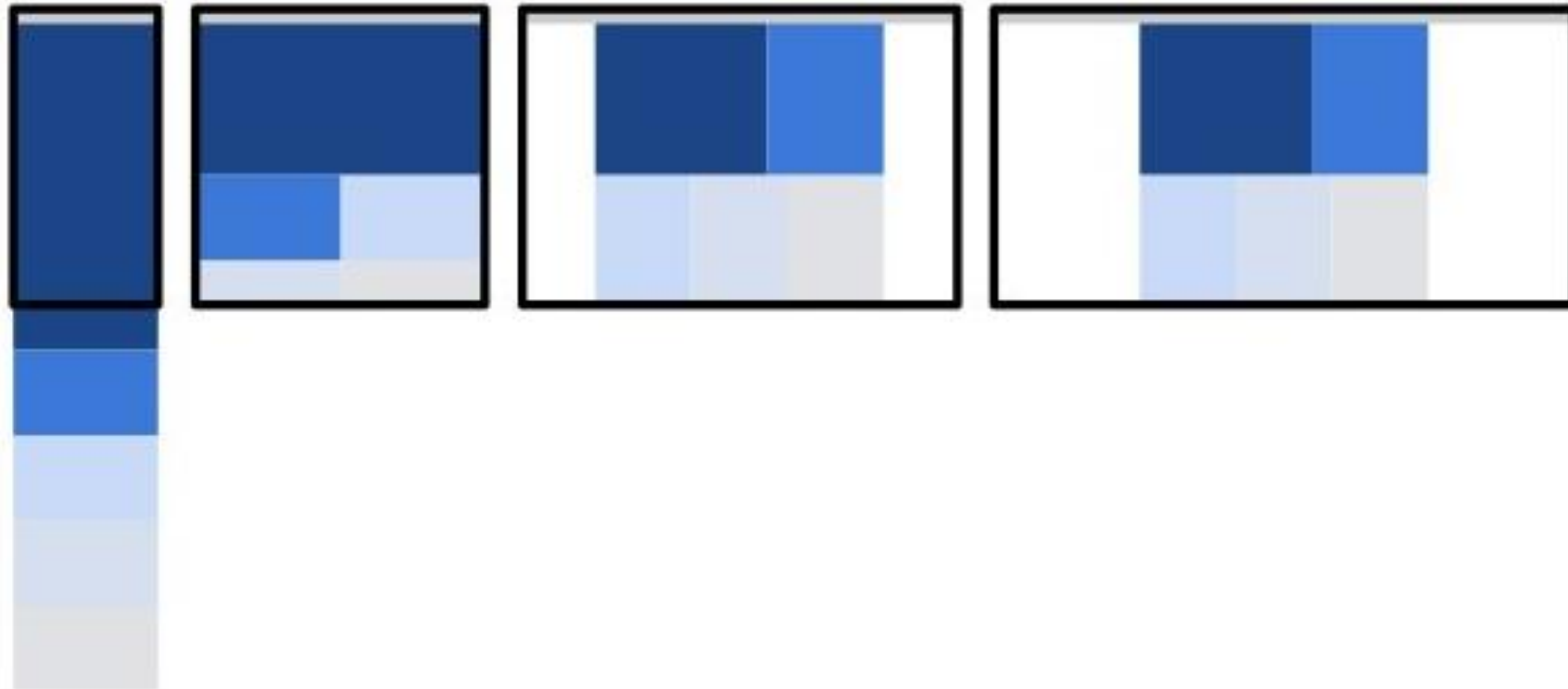
UA 2.15: Introducción CSS: Responsive Web Design



Patrones Responsive

MOSTLY FLUID

- ✓ Parecido a Column Drop. Es una cuadrícula fluida y en cada Breakpoint hay un redimensionamiento de varias columnas.



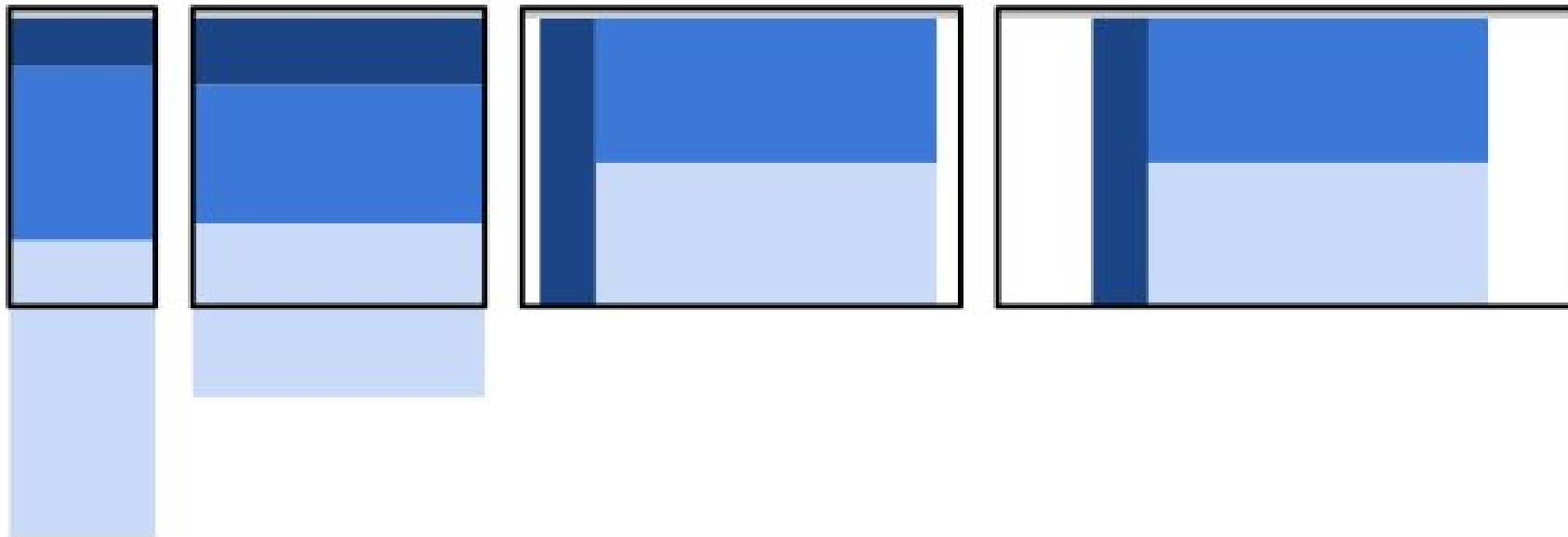
UA 2.15: Introducción CSS: Responsive Web Design



Patrones Responsive

LAYOUT SHIFTER

- ✓ Es el patrón más “responsivo”, cambio el diseño en cada Breakpoint en vez de únicamente organizar el flujo y la anchura de los elementos.



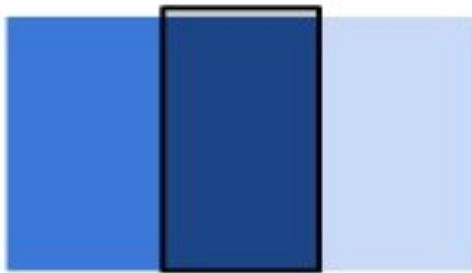
UA 2.15: Introducción CSS: Responsive Web Design



Patrones Responsive

OFF CANVAS

- ✓ En vez de apilar contenidos éstos se colocan fuera de la pantalla cuando el tamaño de pantalla no es lo suficientemente grande.



UA 2.15: Introducción CSS: Responsive Web Design

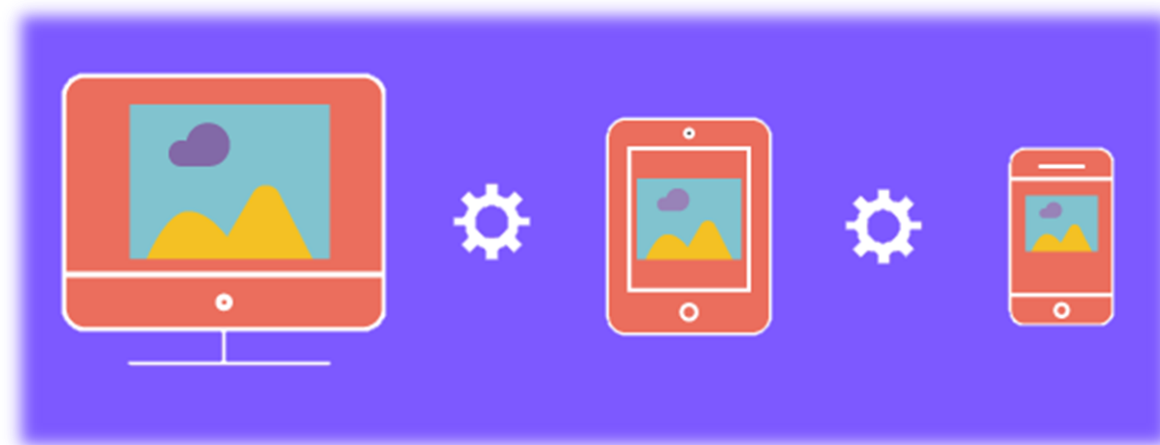


Imágenes Responsive

- Las **imágenes** son un elemento **fundamental** de todas las páginas y representan una **gran parte** del “**peso**” de la misma.
- Esta situación plantea ciertos **retos** a la hora de hacer diseño responsive.

Retos Responsivos con Imágenes

- A nivel de *optimización*
- A nivel de *diseño*



UA 2.15: Introducción CSS: Responsive Web Design



Imágenes Responsive

Optimización de Imágenes

En diseño responsivo **optimizar el uso de imágenes** consiste en:

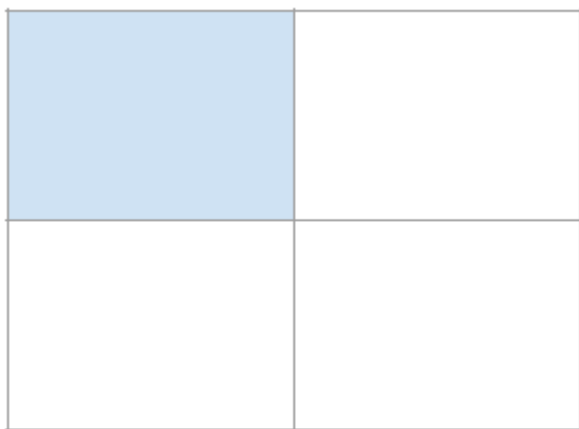
- ✓ Consumir el menor ancho de banda
- ✓ Elegir la versión de una misma imagen más adecuada para la resolución.

En este proceso de optimización debemos tener en cuenta:

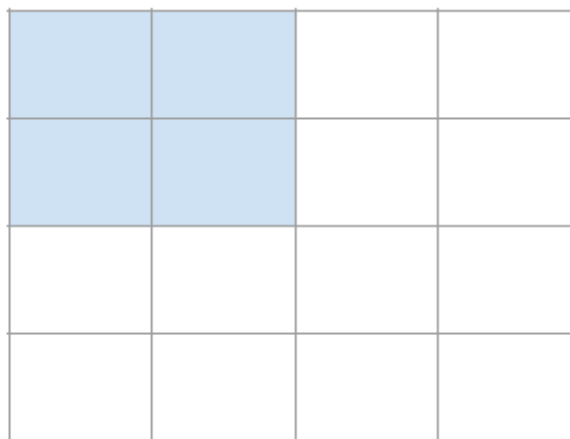
- ✓ Ancho del dispositivo.
- ✓ Dimensiones de la imagen.
- ✓ Resolución de la imagen (en especial en los dispositivos RETINA DISPLAY)

Imágenes Responsive

RETINA DISPLAY



**RESOLUCIÓN
NORMAL**



**RETINA
DISPLAY**

(necesito imágenes de
tamaño doble...)

Importante

Retina Display: Este tipo de dispositivos tiene una densidad de pixels superior a la normal. En la imagen se muestra la explicación para dispositivos con densidad 2x. En estos casos para que las imágenes se muestren bien tendrán que ser de tamaño doble para que a la hora de ampliarlos, no se distorsionen.

Imágenes Responsive

Optimización. Imágenes SVG

- ✓ Para solucionar este tipo de problemas lo más fácil es usar imágenes **SVG** que son gráficos vectoriales que escalan y encogen sin perder resolución.
- ✓ El problema es que no siempre disponemos de gráficos SVG.



Imágenes Responsive

Optimización. Imágenes PNG/GIF/JPEG

- ✓ Si no nos importa la optimización es suficiente con usar una imagen de gran resolución y dimensiones y acotarla dentro de un contenedor

HTML

```
<div>  
      
</div>
```

CSS

```
div{  
    /* dimensiones deseadas */  
}  
  
img{  
    max-width: XXXXXpx;  
    width: 100%;  
}
```

Imágenes Responsive

Optimización. Imágenes PNG/GIF/JPEG

- ✓ Si nos importa la optimización utilizaremos los atributos srcset y/o sizes de la imagen que queremos mostrar.

Ejemplo:

Considerando resolución:

```
<div class='container'>  
  <img src='img/small.jpg'  
    srcset='img/big.jpg 2x,img/small.jpg 1x' />  
</div>
```

Coge una imagen u otra, si el navegador no soporta el srcset, coge la primera foto.

Considerando dimensiones:

```
<div class='container'>  
  <img src='img/small.jpg'  
    srcset='img/big.jpg 2000w, /* W → Tamaño real , unidad especial */  
          img/small.jpg 1000w'  
    sizes='(min-width: 960px) 960px,100vw' />  
</div>
```

W indica que son 2000px reales, de la pantalla.

Imágenes Responsive

Diseño Art-Director

- ✓ La técnica de diseño responsive Art Director consiste en elegir una u otra imagen utilizando la etiqueta source dentro la etiqueta picture y sus atributos srcset para indicar la imagen y media que funciona de manera similar a una media query.

Ejemplo:

```
<div class='art'>
  <picture>
    <source media='(min-width: 576px)' srcset='img/big-art.jpg'/>
    <source media='(max-width: 575px)' srcset='img/small-art.jpg'/>
    <img src='img/small-art.jpg'/> /* En caso de no soportar picture */
  </picture>
</div>
```

Tablas Responsivas

Tablas

- ✓ Las **tablas** son un elemento **problemático en el diseño responsivo** ya que en cuanto tienen más de unas pocas columnas van a provocar un **scroll horizontal** en pantallas pequeñas.
- ✓ Para afrontar este tipo de problemas hay tres soluciones que son aceptadas como buenas:
 - Esconder columnas.
 - Convertir las filas en listas
 - Crear un scroll horizontal que solo se aplique a la tabla.

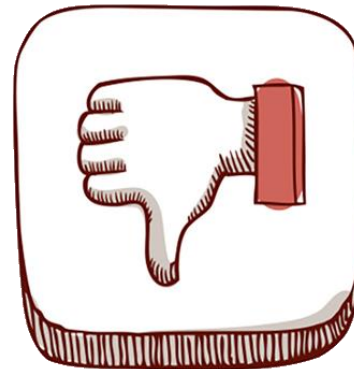
No hay una solución universal. Debemos experimentar según el Layout que tengamos.



Tablas Responsivas

Esconder Columnas

- ✓ Esta técnica consiste básicamente en esconder ciertas columnas, que debe de ser las menos importantes, cuando el tamaño de la pantalla es menor que un breakpoint establecido.
- ✓ Tiene las siguiente **ventajas**:
 - Conseguimos un diseño responsivo.
 - Priorizo el contenido que quiero mostrar.
- ✓ Aunque también tiene **desventajas**:
 - Pierdo información en determinadas pantallas.






UA 2.15: Introducción CSS: Responsive Web Design



Tablas Responsivas

Esconder Columnas

Ejemplo

Responsive ▾ 968 × 784 100% ▾ Online ▾												
Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	MPP
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	20
Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	
Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	MPP
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	20

UA 2.15: Introducción CSS: Responsive Web Design



Tablas Responsivas

Esconder Columnas

Ejemplo

```
<table class="hidden">
  <thead>
    <tr>
      <th>Player</th>
      <th>PPP</th>
      <th>RPP</th>
      <th>APP</th>
      <th class="primero segundo">EFI</th>
      <th class="primero">%TC</th>
      <th class="primero">%3P</th>
      <th class="primero">%TL</th>
      <th class="primero segundo">ROF</th>
      <th class="primero segundo">RDEF</th>
      <th class="primero">FPP</th>
      <th class="primero">PER</th>
      <th class="primero segundo">MPP</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
      <td>10</td>
      <td>6</td>
      <td>3</td>
      <td class="primero segundo">7.5</td>
      <td class="primero">33%</td>
      <td class="primero">15%</td>
      <td class="primero">90%</td>
      <td class="primero segundo">5</td>
      <td class="primero segundo">1</td>
      <td class="primero">3.3</td>
      <td class="primero">2</td>
      <td class="primero segundo">20</td>
    </tr>
  </tbody>
</table>
```



```
/* ESTILOS PARA OCULTAR CIERTAS COLUMNAS */
table.hidden thead {
  background-color: #555555;
  color: white;
}

@media screen and (max-width: 576px) {
  .hidden td.primero,
  .hidden th.primero {
    display: none;
  }
}

@media screen and (max-width: 768px) {
  .hidden td.segundo,
  .hidden th.segundo {
    display: none;
  }
}
```



UA 2.15: Introducción CSS: Responsive Web Design





Tablas Responsivas


Esconder Columnas

Ejemplo

Responsive ▾ 766 × 768 100% ▾ Online ▾

Player	PPP	RPP	APP	%TC	%3P	%TL	FPP	PER
	10	6	3	33%	15%	90%	3.3	2


Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RD
	10	6	3	7.5	33%	15%	90%	5	1


Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	MPP
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	20

```
@media screen and (max-width: 768px) {  
  .hidden td.segundo,  
  .hidden th.segundo {  
    display: none;  
  }  
}
```

Responsive ▾ 574 × 768 100% ▾ Online ▾

Player	PPP	RPP	APP
	10	6	3

Player	PPP	RPP	APP	EFI	%TC	%3P
	10	6	3	7.5	33%	15%

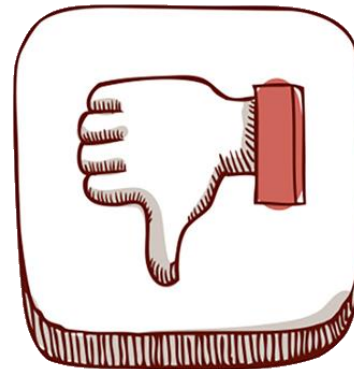
profile	
PPP	10
RPP	6
ASP	3
EFI	7.5

```
@media screen and (max-width: 576px) {  
  .hidden td.primerio,  
  .hidden th.primerio {  
    display: none;  
  }  
}
```


Tablas Responsivas

Convertir Tablas a Filas

- ✓ Esta técnica consiste en hacer desaparecer las cabeceras de la tabla cuando la pantalla es menor que una determinada cantidad y hacer que todas las celdas se conviertan en elementos de bloque para que se muestren una debajo de otra y no al lado.
- ✓ Tiene las siguiente ventajas:
 - Conseguimos un diseño responsivo.
 - No pierdo información.
- ✓ Aunque también tiene desventajas:
 - No priorizo la información.
 - “Desplazo” mucho el resto del layout hacia abajo.



UA 2.15: Introducción CSS: Responsive Web Design



Tablas Responsivas

Convertir Tablas a Filas

Ejemplo

```
<table class="listas">
  <thead>
    <tr>
      <th>Player</th>
      <th>PPP</th>
      <th>RPP</th>
      <th>APP</th>
      <th>EFI</th>
      <th>%TC</th>
      <th>%3P</th>
      <th>%TL</th>
      <th>ROF</th>
      <th>RDEF</th>
      <th>FPP</th>
      <th>PER</th>
      <th>MPP</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td data-th="profile"></td>
      <td data-th="PPP">10</td>
      <td data-th="RPP">6</td>
      <td data-th="ASP">3</td>
      <td data-th="EFI">7.5</td>
      <td data-th="%TC">33</td>
      <td data-th="%3P">15</td>
      <td data-th="%TL">90</td>
      <td data-th="ROF">5</td>
      <td data-th="RDEF">1</td>
      <td data-th="FPP">3.3</td>
      <td data-th="PER">2</td>
      <td data-th="MPP">20</td>
    </tr>
  </tbody>
</table>
```



```
/* ESTILOS PARA LA CONVERSIÓN A LISTAS */
table.listas {
  margin-top: 50px;
}

table.listas thead {
  background-color: red;
  color: white;
}

@media screen and (max-width: 700px) {
  table.listas,
  table.listas thead,
  table.listas tbody,
  table.listas tr,
  table.listas th,
  table.listas td {
    display: block;
  }
}
```



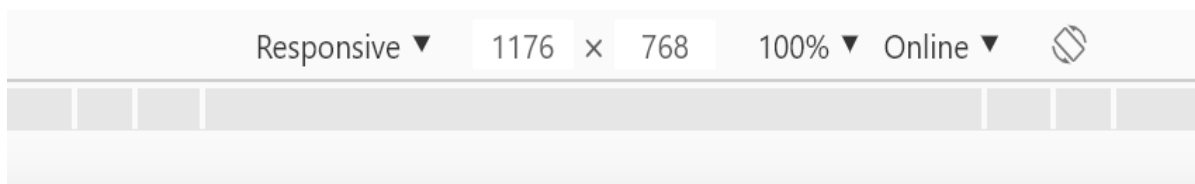
UA 2.15: Introducción CSS: Responsive Web Design





Tablas Responsivas


Convertir Tablas a Filas

Ejemplo




Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	MPP
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	20


Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	MPP
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	20

Player	PPP	RPP	APP	EFI	%TC	%3P	%TL	ROF	RDEF	FPP	PER	MPP
	10	6	3	7.5	33%	15%	90%	5	1	3.3	2	20



Player	PPP	RPP	APP	%TC	%3P	%TL	FPP	PER
	10	6	3	33%	15%	90%	3.3	2

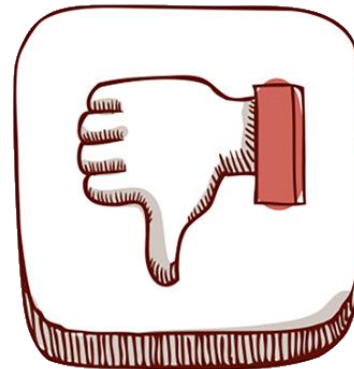
Player	PPP	RPP	APP	EFI	%TC	%3P	%TL
	10	6	3	7.5	33%	15%	90%

profile	
PPP	10
RPP	6
ASP	3
EFI	7.5
%TC	33%
%3P	15%
%TL	90%
ROF	5
RDEF	1

Tablas Responsivas

Scrollo Controlado

- ✓ Esta técnica consiste en acotar el scroll horizontal para que si ha de aparecer solo afecte a la tabla y no a la página entera.
- ✓ Tiene las siguiente ventajas:
 - Conseguimos un diseño responsivo.
 - No pierdo información.
- ✓ Aunque también tiene desventajas:
 - No priorizo la información.
 - Aunque local sigue habiendo un scroll horizontal.



Tablas Responsivas

Scrollo Controlado

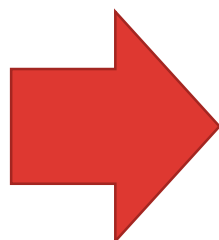
- ✓ Para conseguir esto debemos “envolver” la tabla en un contenedor y darle las siguientes propiedades:

HTML

```
<div class="scroll">  
  <table>  
  </table>  
</div>
```

CSS

```
div.scroll {  
  overflow-x: auto;  
  width: 100%;  
}
```



```
<div class="scroll">  
  <table>  
    <thead>  
      <tr>  
        <th>Player</th>  
        <th>PPP</th>  
        <th>RPP</th>  
        <th>APP</th>  
        <th>EFI</th>  
        <th>%TC</th>  
        <th>%3P</th>  
        <th>%TL</th>  
        <th>ROF</th>  
        <th>RDEF</th>  
        <th>FPP</th>  
        <th>PER</th>  
        <th>MPP</th>  
      </tr>  
    </thead>  
    <tbody>  
      <tr>  
        <td></td>  
        <td>10</td>  
        <td>6</td>  
        <td>3</td>  
        <td>7.5</td>  
        <td>33%</td>  
        <td>15%</td>  
        <td>90%</td>  
        <td>5</td>  
        <td>1</td>  
        <td>3.3</td>  
        <td>2</td>  
        <td>20</td>  
      </tr>  
    </tbody>  
  </table>  
</div>
```

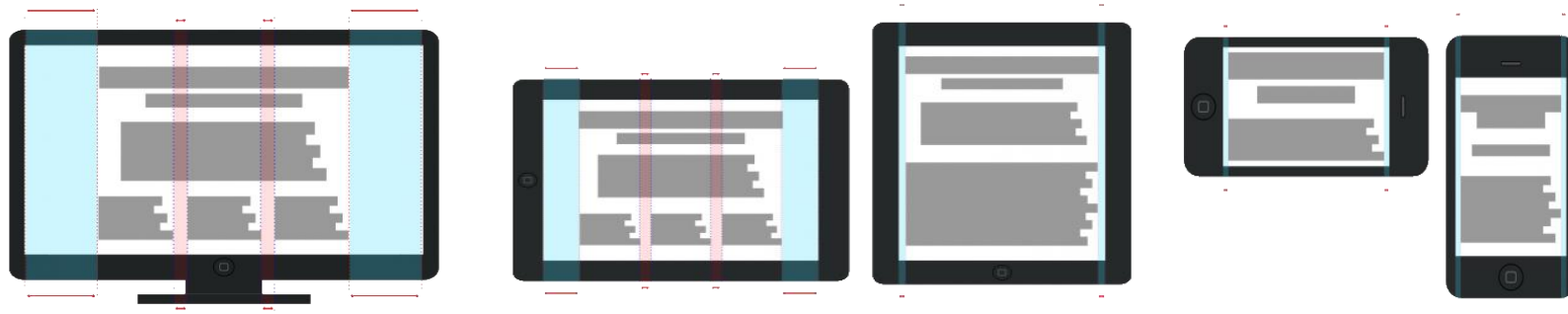


```
/* ESTILOS PARA EL SCROLL CONTROLADO */  
div.scroll table {  
  margin-top: 50px;  
}  
  
div.scroll table thead {  
  background-color: blue;  
  color: white;  
}  
div.scroll {  
  overflow-x: auto;  
  width: 100%;  
}
```



Texto Responsivo

- ✓ Cuando hablamos de diseño responsivo solemos dar más importancia al layout, pero el texto que vamos a presentar es igual de importante para conseguir un buen diseño.
- ✓ Si no tenemos cuidado nos podemos encontrar con varios problemas:
 - Líneas cortas con pocos caracteres, lo que dificultan la lectura.
 - Líneas largas con muchos caracteres, lo que también dificulta la lectura.
 - Caracteres muy pequeños.
- ✓ Hay diversos estudios que indican que la **longitud de una línea** para facilitar la lectura sea entre **60-80 caracteres**.



Texto Responsivo

Solución

- ✓ La solución más recomendada es utilizar para el texto **unidades de tamaño** relacionadas con el **viewport** → **Unidades nuevas incorporadas en HTML5.**

Unidades Relativas al Viewport

- ✓ vw: anchura viewport
- ✓ vh: altura del viewport
- ✓ vmin: el valor mínimo en relación a la dimensión pequeña del viewport (anchura o altura)
- ✓ vmax: el valor máximo en relación a la dimensión más grande del viewport (anchura o altura)

Debemos de tener en cuenta que **1vw** es un **1%** de la **anchura del viewport** (anchura real)

UA 2.15: Introducción CSS: Responsive Web Design




Texto Responsivo

Ejemplo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Texto Responsivo</title>

  <link rel="stylesheet" type="text/css" href="css/estilos.css" />
</head>
<body>
  <div class="container">
    <header>
      <h1>Cabecera de la página</h1>
    </header>
    <main>
      <section>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Maxime, non. Accusamus corrupti fugiat itaque nulla, voluptates perspiciatis dolor quis nesciunt. Ipsum magni quae vitae minima quas perferendis officiis, aliquid voluptates?</p>
      </section>
      <section>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Maxime, non. Accusamus corrupti fugiat itaque nulla, voluptates perspiciatis dolor quis nesciunt. Ipsum magni quae vitae minima quas perferendis officiis, aliquid voluptates?</p>
      </section>
      <section>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Maxime, non. Accusamus corrupti fugiat itaque nulla, voluptates perspiciatis dolor quis nesciunt. Ipsum magni quae vitae minima quas perferendis officiis, aliquid voluptates?</p>
      </section>
    </main>
  </div>
</body>
</html>
```



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

/* Cuidado porque cambio luego em o rem */
html {
  font-size: 3.5vw;
}

.container {
  margin: 0px auto;
  width: 80%;
}

header {
  text-align: center;
}

header h1 {
  font-size: 8vw;
}

main {
  display: flex;
  justify-content: space-evenly;
}

main section {
  width: 30%;
}

@media screen and (min-width: 576px) {
  html {
    font-size: 2.5vw;
  }
}
```

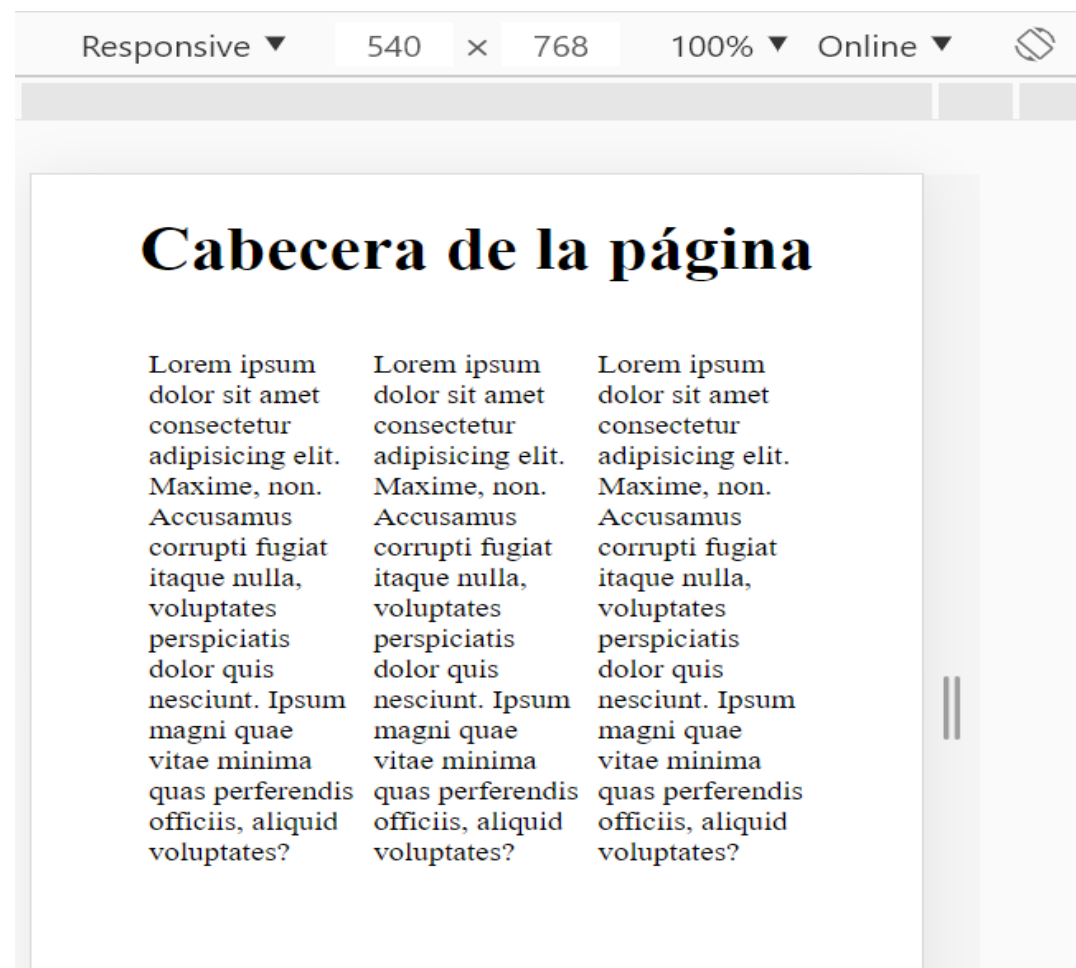
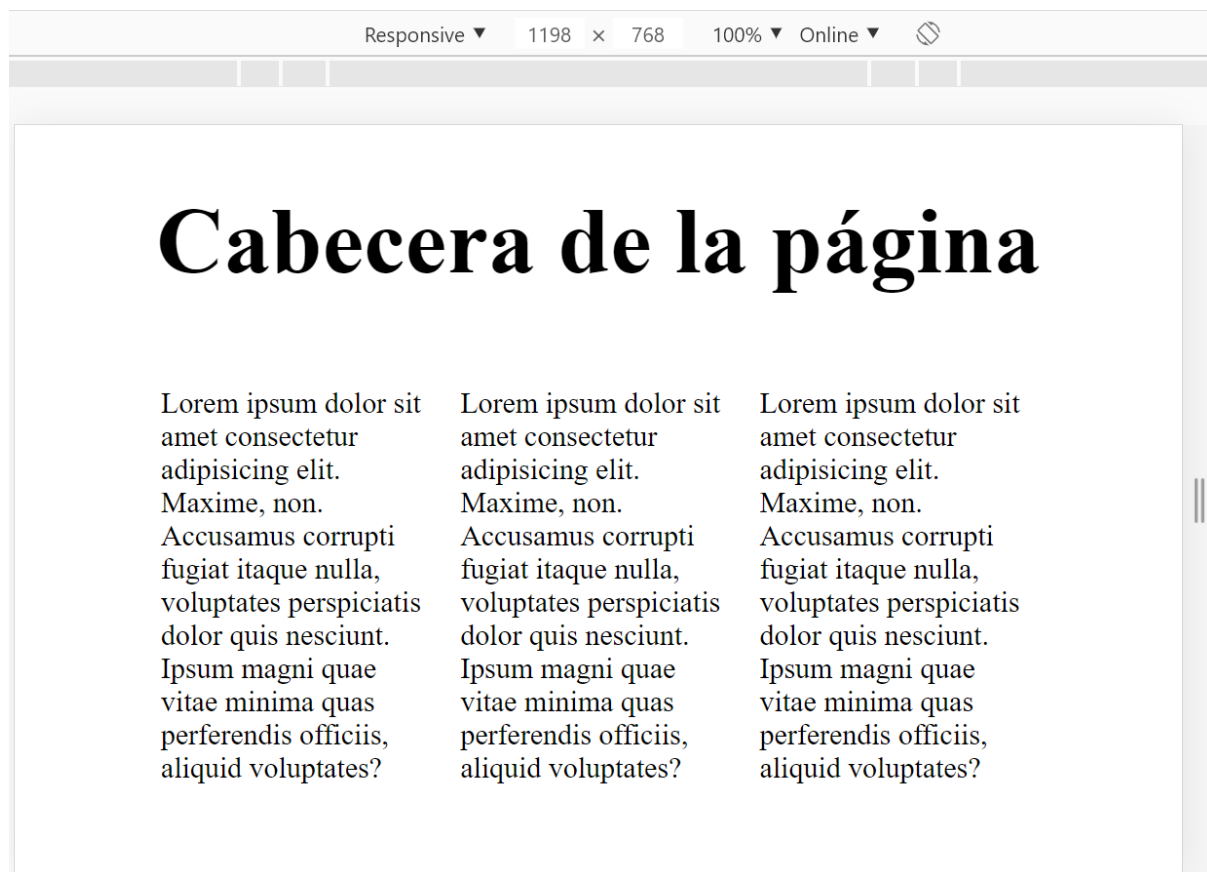


UA 2.15: Introducción CSS: Responsive Web Design



Texto Responsivo

Ejemplo



Texto Responsivo

ES COMPLEJO, Y debemos hacer pruebas con multitud de pantallas, usar funciones como calc() etc...

PAUTAS PARA TEXTO RESPONSIVO

- ▶ Calcula el tamaño mínimo y máximo que me puedo permitir. (Usa calc())
- ▶ No importa si hay texto que hace “reflow” (pasar a la siguiente línea)
- ▶ Mantener el tamaño “perfecto” de línea puede que sea imposible.
- ▶ Prioriza móviles y tablets.
- ▶ En algunos elementos (como navegación) puede tener sentido usar tamaño fijos.



**Universidad
Europea**