

DISEÑO ADAPTABLE

Web 2.0

- Con la aparición de la web 2.0 Internet cambió y se transformó para dar acogida a todas las necesidades de los usuarios. Conforme se avanza hacia la web 3.0 los cambios continúan.
- Los sitios web en general han tenido que cambiar.

Web 2.0

- Uso de técnicas CSS avanzadas.
- Uso de frameworks → Bootstrap.
- Ya en 2011 se empezó a hablar de los sitios web **adaptables** a todo tipo de pantallas y dispositivos fuese cual fuese su tamaño.

Tipos de diseño web

- En el diseño web existen fundamentalmente 4 tipos de diseños.
 - El diseño web fijo
 - El diseño web fluido o diseño web líquido
 - El diseño web elástico, elastic web design
 - Por último tenemos el diseño web adaptable, sensible, o responsive web design

Tipos de diseño web - fijo

- El **diseño web fijo**, o fixed web design en inglés, tiene la ventaja de su sencillez.
- Es aquel que permanece inalterable sea cual sea el dispositivo donde se visualiza el sitio web. Es indiferente al tamaño de la pantalla y a la anchura del navegador web que la visualiza. Sean cuales sean las características del dispositivo, pantalla o navegador, siempre se visualiza el mismo tamaño preestablecido por el diseñador web en píxeles.

Tipos de diseño web - fluido

- El **diseño web fluido** o diseño web líquido, fluid web design o liquid web design, es aquel que tiende a ocupar todo el ancho de la pantalla, sea cual sea el tamaño de ésta, sin necesidad de una barra de desplazamiento horizontal.
- Requiere de mucho más trabajo por parte del diseñador web ya que si no se realiza correctamente su resultado puede resultar bastante atractivo para tamaños de pantalla “normales” y pequeñas, pero cuando se emplean pantallas de muchas pulgadas su estética resulta, cuanto menos, horrible, a no ser que se utilicen técnicas como el uso de max-width para limitar el máximo ancho aceptado por nuestro diseño.

Tipos de diseño web - fluido

- (...) El diseñador web utiliza porcentajes en lugar de píxeles para establecer los anchos de sus diseños, y también se pueden emplear píxeles y medidas máximas y mínimas con min-width y max-width para adaptarse a pantallas grandes y pequeñas.

Tipos de diseño web - elástico

- El **diseño web elástico**, **elastic web design**, es un tipo de diseño muy parecido al fluido, sólo que en este caso el contenido también crece junto al ancho del diseño para rellenar la pantalla. Es, con diferencia, un tipo de diseño prácticamente residual y muy poco usable. Como con el diseño fluido, puede resultar más o menos agradable en tamaños de pantalla más o menos pequeños, aunque la letra podría volverse ilegible si es muy pequeña. Cuando se utilizan pantallas muy grandes el contenido se vuelve irracionalmente grande, por lo que salvo contadas excepciones, no tiene razón de ser.

Tipos de diseño web - elástico

- (...)En este caso el diseñador web suele utilizar em para fijar los anchos de su diseño web y del resto del contenido que alberga.

Tipos de diseño web - adaptable

- El **diseño web sensible, adaptable, o responsive web design** en inglés, es un tipo de diseño que se transforma en función del tipo de dispositivo o del ancho del navegador web que lo visualiza. De esta forma se muestra un diseño web en dispositivos smartphones, otro diferente en dispositivos tipo tabletas, y otro distinto en ordenadores.
- Aunque la información mostrada es exactamente igual, sea cual sea el dispositivo que la visualiza, el diseño, que es único pero adaptable, varía para optimizar al máximo el acceso a la misma.

Tipos de diseño web - adaptable

- (...) Se emplean diferentes hojas de estilo que se cargarán en función de las necesidades que se establezcan, principalmente ancho del navegador web y tipo de dispositivo. Este es el diseño que más en auge está teniendo gracias al uso masivo de dispositivos smartphones y tabletas en el uso diario de la Internet. Plantea algunos problemas, como su complejidad, peso y tráfico generado.

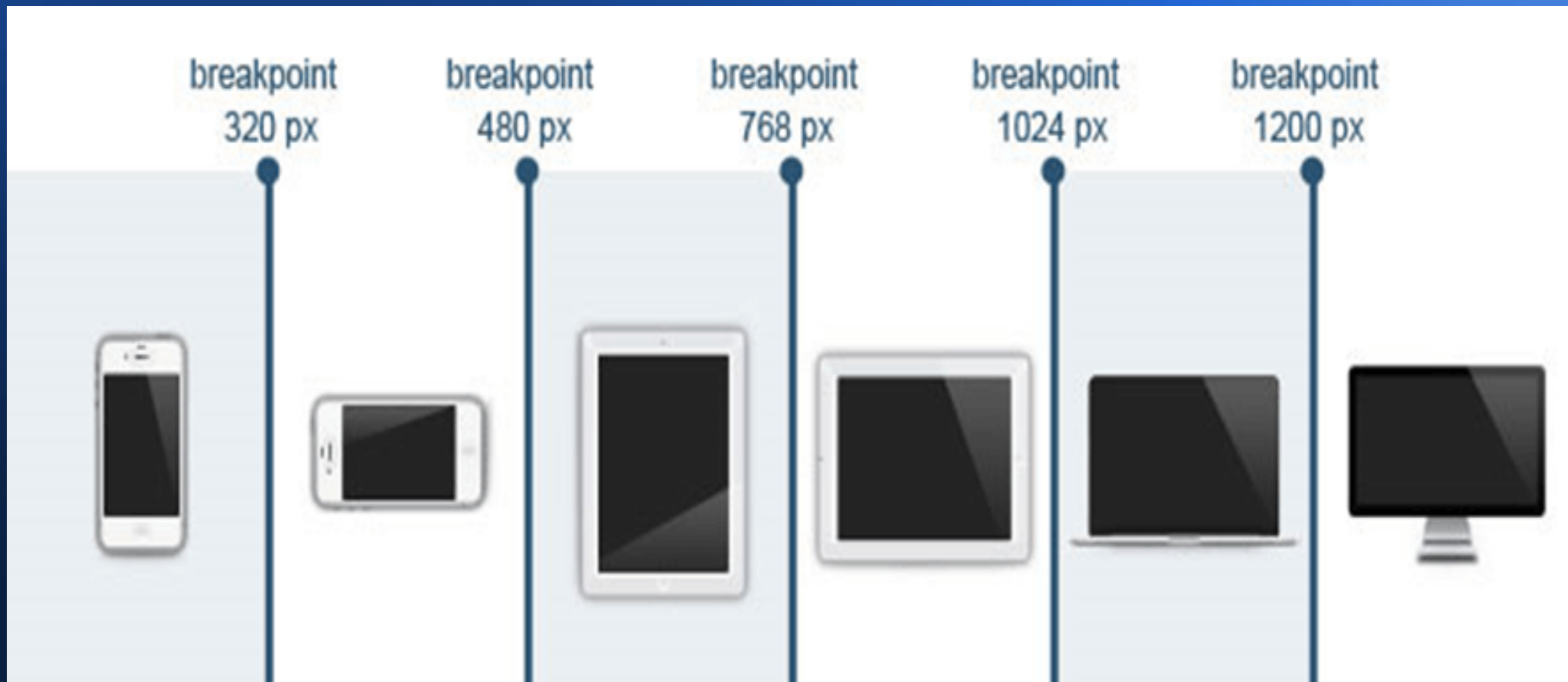
Diseño adaptable

- Dada una web, podemos hacer una maqueta adaptable o responsive mediante el uso de media queries CSS.
- Media Queries es un módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo como la resolución de pantalla.

Diseño adaptable



Diseño adaptable



Diseño adaptable

- Ejemplos:

@media print

```
body { font-size: 10pt }  
}
```

@media screen {

```
body { font-size: 13px }  
}
```

@media screen, print {

```
body { line-height: 1.2 }  
}
```

Diseño adaptable

- Más ejemplos:

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

```
...
```

```
}
```

- @media (orientation: landscape) {

```
...
```

```
}
```

- @media screen and (min-width:500px) { ... }

Diseño adaptable

- Aunque el método anterior es el mejor, también tenemos la opción de indicar las características del dispositivo con el atributo media dentro de la etiqueta link (y cargar un css distinto):

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="theme.css">
```

```
<link rel="stylesheet" type="text/css" href="print.css" media="print">
```

```
</head>
```

Diseño adaptable

- Más ejemplos:

```
<link rel="stylesheet" media="only screen and (max-width: 768px)" href="estilos.css">
```

Diseño adaptable

- Operadores lógicos para las media queries
- Operador and: las dos condiciones deben cumplirse para que se apliquen los estilos.
- Operador not: es una negación de una condición. Cuando esta condición no se cumpla, se aplicarán las media queries definidas.
- Operador only: se aplican las reglas solo en el caso de que se cumpla.
- Operador or: se pueden poner varias condiciones separadas por comas y en el momento que se cumpla cualquiera de ellas, se aplicarán los estilos.

Diseño adaptable

- Ejemplo:

```
@media only screen and (min-width:320px) and (max-width:480px){
```

```
/*Estilos CSS*/
```

```
}
```

Diseño adaptable

- Algunos de los parámetros generales que se pueden emplear son:
 - width: anchura de la ventana del navegador.
 - height: altura de la ventana del navegador.
 - device-width: anchura de la resolución de pantalla.
 - device-height: altura de la resolución de pantalla.
 - orientation (portrait/landscape): dispositivo en horizontal o en vertical.
 - resolution: densidad de píxeles.

Diseño adaptable

- Excepto la orientación, el resto de parámetros admiten los valores “max” y “min”.
- max-width: La anchura será menor que la indicada.
- min-width: La anchura será mayor que la indicada.
- Ejemplo:
- @media only screen and (min-device-width: 960px){
- /* Aquí van todos los estilos CSS */
- }

Diseño adaptable

- Ejemplo con una página en concreto: podemos tener la siguiente página, con un diseño adaptable con tres puntos de ruptura:
 - Hasta 600 px: un diseño a una columna.
 - A partir de 600 px y hasta 1200 px: un diseño a dos columnas.
 - A partir de 1200 px: un diseño a tres columnas.
- Ver ejemplo [Ejemplo1_2_columnas.html](#)

Curriculum Vitae de Bruce Wayne

Datos personales

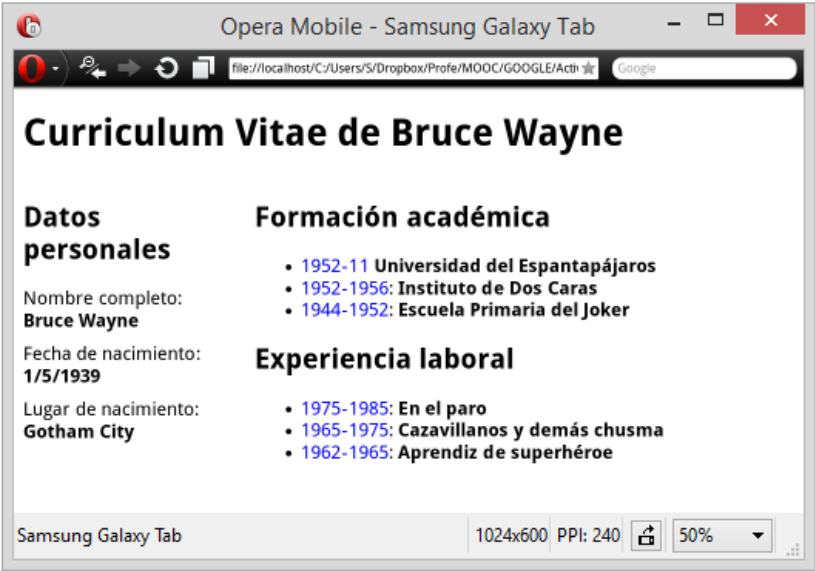
Formación académica

Experiencia laboral

- Nombre completo: **Bruce Wayne**
- Fecha de nacimiento: **1/5/1939**
- Lugar de nacimiento: **Gotham City**

- 1956-1961: **Universidad del Espantapájaros**
- 1952-1956: **Instituto de Dos Caras**
- 1944-1952: **Escuela Primaria del Joker**

- 1975-1985: **En el paro**
- 1965-1975: **Cazavillanos y demás chusma**
- 1962-1965: **Aprendiz de superhéroe**



Diseño adaptable

- Google ofrece la siguiente web, para comprobar si una página está adaptada para móviles:
- https://search.google.com/test/mobile-friendly?utm_source=mft&utm_medium=redirect&utm_campaign=mft-redirect&hl=es

Técnicas de diseño

- Podemos intentar abordar el diseño adaptable desde:
 - CSS puro y tal vez javascript.
 - Framework.

CSS

- Flexbox: unidimensionalmente permite ordenar cajas, en filas o en columnas.
- CSSGrid: muy reciente. Utiliza dos dimensiones para el posicionamiento sobre una grilla o rejilla.

Flexbox - repaso

- El módulo Flexbox Layout es un modelo de layout implementado en la especificación CSS3 para mejorar el alineamiento, dirección y orden de los elemento en el documento web, incluso cuando estos tienen algún tamaño aplicado dinámicamente o incluso desconocido.
- Define el contenedor flexible, cuya principal característica es la capacidad de modificar la posición, la anchura y la altura de los elementos hijos para así llenar los espacios disponibles de la mejor manera posible adaptándose a cualquier dispositivo o a cualquier resolución.

Flexbox - repaso

- Para usarlo, tendremos un flex-container, o padre, y unos flex-items, o elementos hijos (aquellas etiquetas contenidas dentro del contenedor definido como flex).
- Definimos así el contenedor:

```
.flex-container {  
    display: flex; /* ó inline-flex */  
}
```

Flexbox - repaso

- Algunas propiedades del contenedor flex son:
 - flex-direction: row (por defecto) | row-reverse | column | column-reverse
 - flex-wrap: nowrap (por defecto) | wrap | wrap-reverse
 - justify-content: flex-start (por defecto) | flex-end | center | space-between | space-around

- Ejemplo:


```
.flex-container {  
    display: flex;  
    flex-direction: row-reverse;  
}
```


Flexbox - flex-direction



Flexbox - flex-wrap

flex-wrap: nowrap					
1	2	3	4	5	6

flex-wrap: wrap			
1	2	3	
4	5	6	

flex-wrap: wrap-reverse			
4	5	6	
1	2	3	

Flexbox - justify-content

`justify-content: flex-start`



`justify-content: flex-end`



`justify-content: center`



`justify-content: space-between`



Flexbox – Patrones de diseño adaptable

- Existen los siguientes patrones de diseño adaptable con flexbox:
- Tiny Tweaks: pequeños retoques
- Mostly Fluid : principalmente fluido
- Column Drop: colocación de columnas
- Layout Shifter: mover los bloques del diseño
- Off Canvas: desplazamiento del lienzo

Flexbox – Patrón tiny tweaks

- Una sola columna para el contenido



- Básicamente cambiaremos el tipo de letra y el espaciado según el tamaño de la pantalla.
- Útil para sitios con mucho texto, con el fin único de facilitar la lectura.

Flexbox – Patrón tiny tweaks

- Ejemplo:

```
<!DOCTYPE html>

<html>

<head>

<title>Tiny Tweaks</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<nav>Navbar</nav>

<section class="columna1">Sección 1</section>

</body>

</html>
```

```
(CSS)

/* Tiny Tweaks */

.columna1 {

padding: 10px;

width: 100%;

}

@media (min-width: 600px) {

.columna1 {

padding: 20px;

font-size: 1.5em;

}

}

@media (min-width: 800px) {

.columna1 {

padding: 40px;

font-size: 2em;

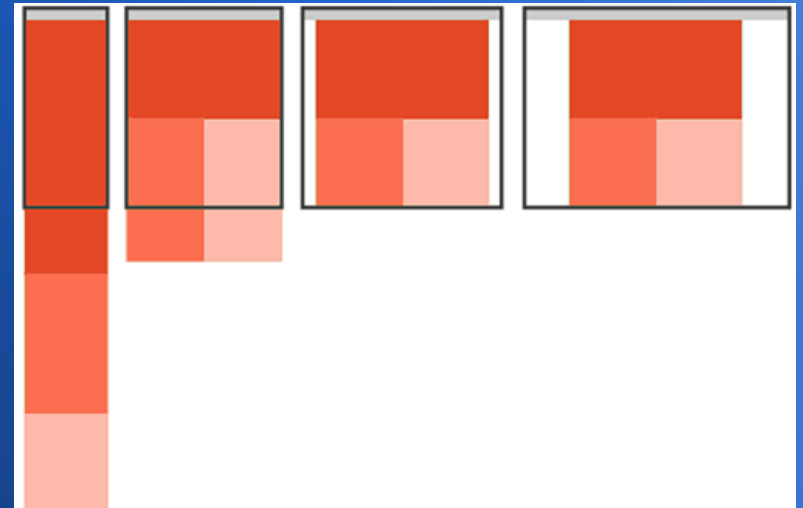
}
```

Flexbox – Patrón tiny tweaks

- Ejemplos:
- <http://gingerwhale.com>
- <http://futurefriendlyweb.com>

Flexbox – Patrón Mostly Fluid

- Tener un “grid” o grilla de tamaño flexible y fluida
- Según vaya creciendo la pantalla, los distintos bloques se agrupan ocupando toda la pantalla disponible.
- En pantallas más grandes, el diseño es el mismo pero queda agrupado dentro de un contenedor que queda centrado en la página con un tamaño fijo de ancho.



Flexbox – Patrón Mostly Fluid

- `<!DOCTYPE html>`
- `<html>`
- `<head>`
- `<title>Mostly Fluid</title>`
- `<link rel="stylesheet" href="styles.css">`
- `</head>`
- `<body>`
- `<nav>Navbar</nav>`
- `<div class="container">`
- `<section class="columna1">Sección 1</section>`
- `<section class="columna2">Sección 2</section>`
- `<section class="columna3">Sección 3</section>`
- `<section class="columna4">Sección 4</section>`
- `<section class="columna5">Sección 5</section>`
- `</div>`
- `</body>`
- `</html>`

- `/* Mostly Fluid */`
- `.container {`
- `display: -webkit-flex;`
- `display: flex;`
- `-webkit-flex-flow: row wrap;`
- `flex-flow: row wrap; * En filas, y que puedan ocupar varias filas (wrap) */`
- `}`
-
- `.columna1,`
- `.columna2,`
- `.columna3,`
- `.columna4,`
- `.columna5 {`
- `width: 100%;`
- `}`
- `/*Continúa...*/`

Flexbox – Patrón Mostly Fluid

```
• @media (min-width: 600px) { /*Simula tablet*/  
• .columna2,  
• .columna3,  
• .columna4,  
• .columna5 {  
• width: 50%;  
• }  
• }  
•  
• @media (min-width: 800px) { /*Simula laptop y desktop*/  
• .columna1 { width: 60%; }  
• .columna2 { width: 40%; }  
• .columna3,  
• .columna4,  
• .columna5 {  
• width: 33.3%;  
• }  
• }  
• .container {  
• width: 800px;  
• margin-left: auto;  
• margin-right: auto;  
• }  
• }
```

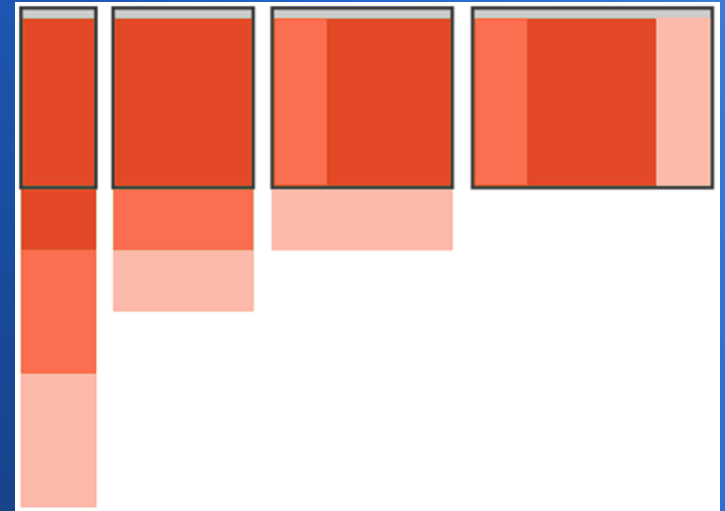

Flexbox – Patrón Mostly Fluid

- Ejemplos:

<https://mediaqueri.es/ala/>

Flexbox – Patrón Column-Drop

- Este patrón consiste en que cada bloque de contenido, que en un smartphone vemos en filas, vaya formando columnas según vaya siendo más grande la pantalla del dispositivo.
- Tendremos un primer breakpoint donde la segunda fila pasa a ser columna, pero ocupando la primera posición, habitualmente para un menú de navegación.
- El contenido que aparecía en primer lugar en la versión móvil, pasa a ocupar la segunda columna en el primer corte.
- Tendremos un segundo punto de corte donde la última fila se convierte en columna también.



Flexbox – Patrón Column Drop

- <!DOCTYPE html>
- <html>
- <head>
- <title>Mostly Fluid</title>
- <link rel="stylesheet" href="styles.css">
- </head>
- <body>
- <nav>Navbar</nav>
- <div class="container">
- <section class="columna1">Sección 1</section>
- <section class="columna2">Sección 2</section>
- <section class="columna3">Sección 3</section>
- <section class="columna4">Sección 4</section>
- <section class="columna5">Sección 5</section>
- </div>
- </body>
- </html>

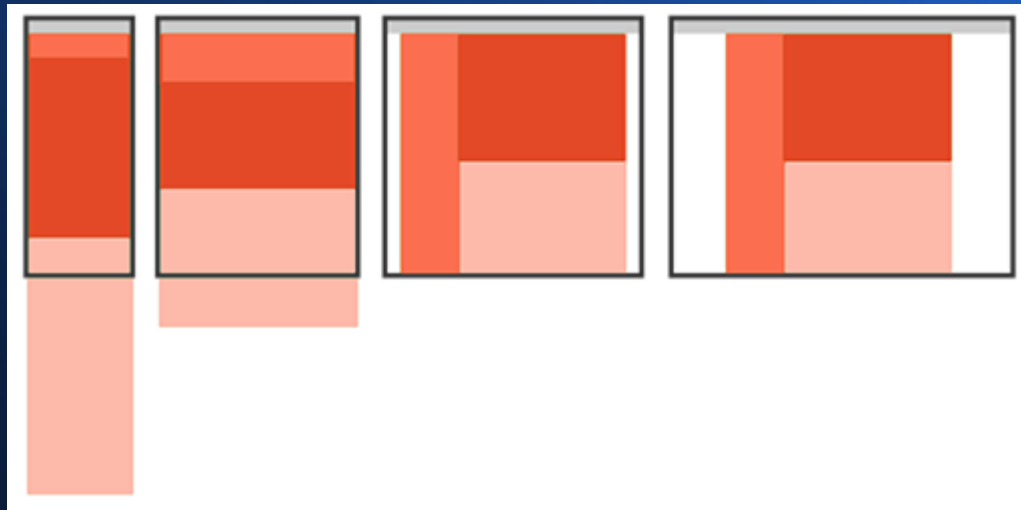
- /* Column Drop */
- .container {
- display: -webkit-flex;
- display: flex;
- -webkit-flex-flow: row wrap;
- flex-flow: row wrap;
- }
- .columna1,
- .columna2,
- .columna3 {
- width: 100%;
- }
- @media (min-width: 600px) {
- .columna1 {
- width: 60%;
- -webkit-order: 2;
- Order: 2; /*Con order indicamos la posición. Quedará como segunda columna*/
- }
- .columna2 {
- width: 40%;
- -webkit-order: 1;
- order: 1;
- }
- .columna3 {
- width: 100%;
- -webkit-order: 3;
- order: 3;
- }
- }
- @media (min-width: 800px) {
- .columna2,
- .columna3 {
- width: 20%;
- }
- }

Flexbox – Patrón Column-Drop

- Ejemplos:
- <https://modernizr.com>
- <http://weenudge.com>

Flexbox – Patrón Layout Shifter

- Este es uno de los patrones más complejos. Consiste en mover los bloques de contenido cambiando totalmente el Layout, de ahí el nombre del patrón.

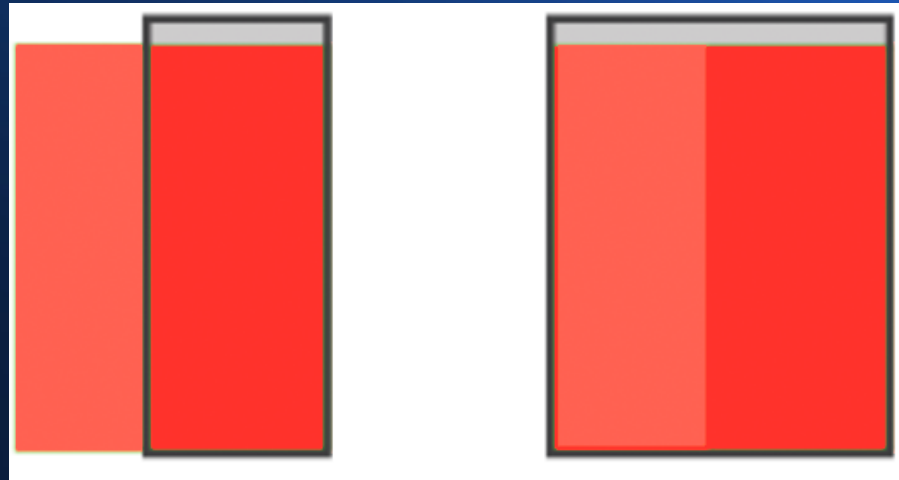


Flexbox – Patrón Layout Shifter

- Ejemplos:
- <http://foodsense.is>
- <http://alistapart.com/d/responsive-web-design/ex/ex-site-FINAL.html>
- <http://www.anderssonwise.com>

Flexbox – Patrón Off Canvas

- Utilizado sobre todo en aplicaciones móviles, este patrón esconde contenido en la pantalla y únicamente es visible si realizamos un determinado gesto. Este contenido oculto normalmente es un menú de navegación. Cuando la pantalla es más ancha, este contenido se hace visible.



- Esto se realizaría con CSS (flexbox, transiciones...) y JavaScript (o mediante un framework de JavaScript).

CSS Grid

- Seguiríamos el esquema de poner puntos de ruptura y distintas configuraciones según los mismos. Ejemplo:

<https://codepen.io/tutsplus/pen/mAOzVq/>

Ejercicio 1

- Crea una página web en el que apliques el patrón Mostly Fluid, a partir de la web del ejercicio anterior.