



RESPONSIVE DESIGN

© 2017, ACTIBYTI PROJECT SLU, Barcelona
Autor: Ricardo Ahumada

ÍNDICE DE CONTENIDOS

1. Caso práctico
2. Introducción
3. Diseño Flexible
4. Imágenes y Videos Flexibles
5. Media Queries

1

CASO PRÁCTICO

Caso Práctico: BananaTube Responsive



“BananaTube” es el proyecto estrella de Banana Apps.

BananaTube será el próximo boom! de las redes sociales; permitirá a sus usuarios gestionar videos, exponerlos en su muro, comentar videos propios y de sus amigos, calificarlos y compartirlos en varios canales.

Uno de los objetivos primordiales del proyecto es que BananaTube sea multicanal, esto significa que la interface deberá adaptarse a múltiples medios y poder ser consumido en diversos dispositivos, sin perder la excelente experiencia de usuario que ya proporciona. En esta etapa queremos implementar los mecanismos necesarios que garanticen que BananaTube se adapta a los distintos canales de consumo.



Discutamos

- En qué medios se puede consumir BananaTube?
- Qué implicaciones tendrá esto en nuestra implementación de interface?
- Qué impacto tendrá en las funcionalidades?
- Qué tecnologías tenemos para lograr el objetivo planteado?

1

Introducción

Responsive Web Design.

- Cada vez más los usuarios se vuelven exigentes con respecto al diseño de las páginas web. En la actualidad, se hace imprescindible que el diseño se ajuste a cada tipo de pantalla, y por supuesto el desarrollo de diferentes versiones para cada resolución se vuelve casi imposible de conseguir, lo cual hace la experiencia del usuario desagradable.
- El Responsive Web Design (Diseño web adaptable) determina que el diseño se pueda adaptar a todos los comportamientos que realiza el usuario, orientado principalmente al tamaño de su pantalla, la plataforma y la orientación.
- La practica se consigue mediante la mezcla de diseños flexibles, imágenes, y principalmente el uso inteligente de CSS, lo cual logra que el usuario pueda disfrutar del uso de una página web desde su ordenador portátil, un smartphone, una tableta o televisor; eliminando de esta manera, la necesidad de diferentes diseños para cada dispositivo del mercado.



Beneficios

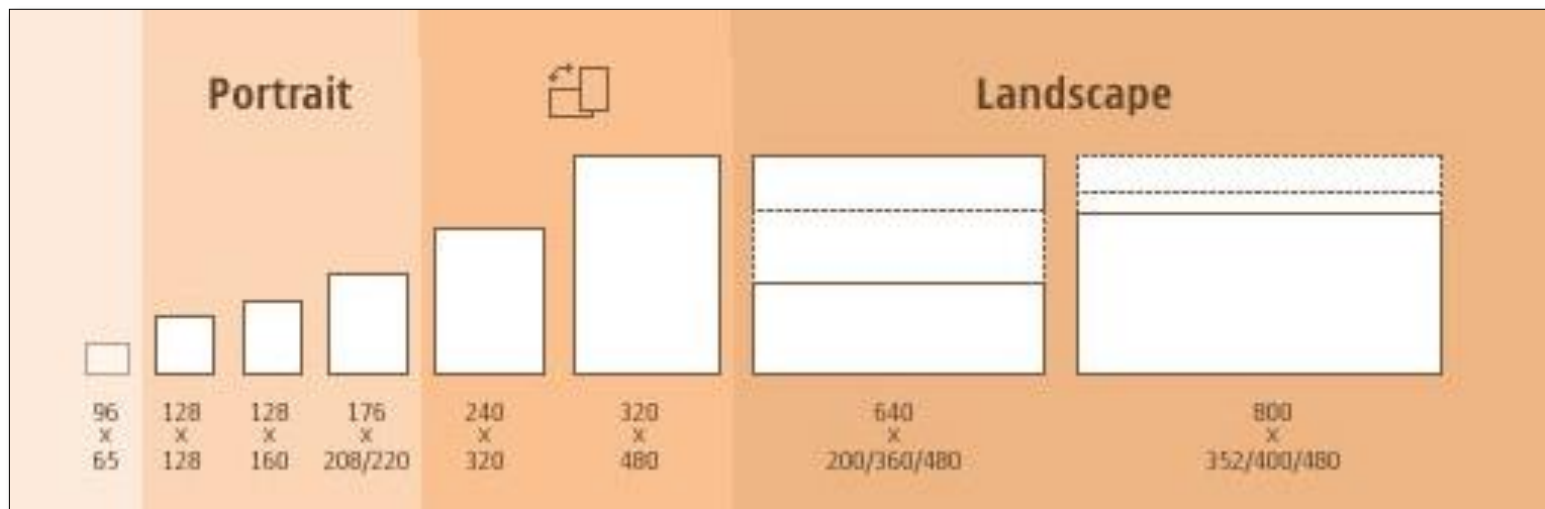
- **Reducción de costos.** Se reducen los costos ya que hasta hoy se debe hacer un portal para la Web y otro para dispositivos móviles. Esto origina mayores costos de creación y mantenimiento de la información.
- **Eficiencia en la actualización.** El sitio solo se debe actualizar una vez y se ve reflejada en todas las plataformas. Cuando tenemos los portales independientes para Web y Mobile se debe realizar la actualización dos veces lo que crea la necesidad de mayor cantidad de recursos y posibilidad de error.
- **Mejora la usabilidad y conversión.** Esta tecnología aporta a la usabilidad y la conversión. Se han hecho estudios que demuestran que cuando se aprovecha toda la pantalla del visitante aumenta la conversión.
- **Impacto en el visitante.** Esta tecnología por ser nueva genera impacto en las personas que la vean en acción, lo que permitirá asociar a la marca con creatividad e innovación.

Antecedentes

- **Ethan Marcotte** utilizó el termino ***Responsive Web Design*** en su artículo publicado en «***A List Apart***». En él, describe la teoría y la práctica de diseño de páginas web sensibles. (25 Mayo 2010)
 - Fuente: <http://www.alistapart.com/articles/responsive-web-design/>
- Tres elementos claves.
 - Una cuadrícula flexible
(<http://www.alistapart.com/articles/fluidgrids/>)
 - ***Responsive Media*** (imágenes, video, etc.)
(<http://www.alistapart.com/articles/fluid-images/>)
 - ***Media queries*** (<http://www.w3.org/TR/css3-mediaqueries/>)

Ajuste de resolución de pantalla.

- Nuevos equipos son desarrollados, con distintas resoluciones de pantalla, definiciones y orientaciones; muchos son capaces de cambiar de vertical a horizontal.
- ¿Como podemos diseñar para todos estos diferentes escenarios?,
- Es posible tomar en cuenta todos los diferentes tamaños disponibles, agruparlos en categorías principales, y hacer cada diseño tan flexible como sea necesario. Pero puede ser abrumador.
- Morten Hjerde y colegas identificaron estadísticas sobre unos 400 dispositivos vendidos entre 2005 y 2008. A
- Algunos de los más comunes



2

Diseño Flexible

Diseño Flexible

- Es una técnica que combina grids fluidos, imágenes fluidas y marcado inteligente donde sea necesario.
- La creación de grids fluidos es práctica bastante común, y hay una serie de técnicas para crear imágenes fluidas:
 - › Ocultando y revelando porciones de imágenes
 - › Creación de imágenes compuestas deslizantes
 - › Imágenes de primer plano que escalan con el diseño
- Zoe Mickley Gillenwater propone un método para crear sitios fluidos en su libro “Flexible Web Design: Creating Liquid and Elastic Layouts with CSS”.
- En su blog se pueden encontrar varios recursos para generar sitios flexibles:
 - › <http://zomigi.com/blog/essential-resources-for-creating-liquid-and-elastic-layouts/>

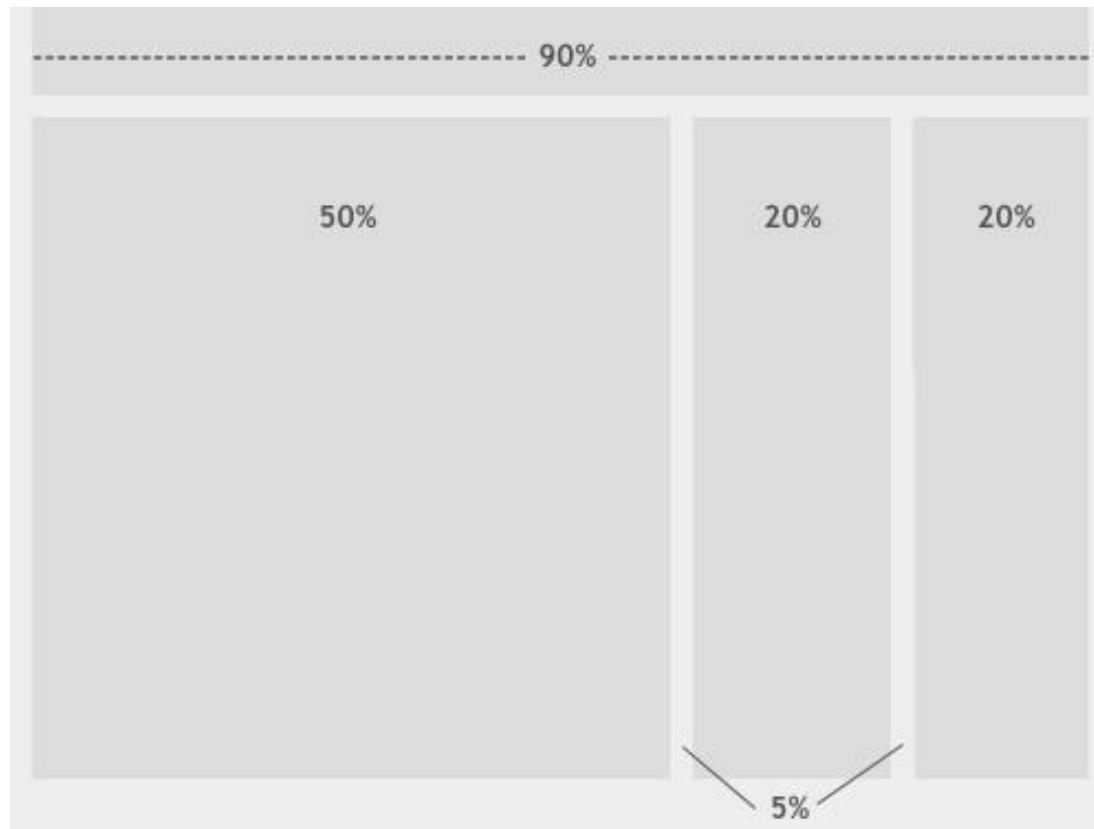
Diseño Flexible

- Una página web sin un diseño flexible provoca que las imágenes se rompan fácilmente (diseños e incluso estructuras);
- Aunque no es una solución completa, la solución nos da muchas más opciones.
- Es perfecto para dispositivos que cambian de orientación vertical a horizontal en un instante o cuando los usuarios cambian de una gran pantalla de computadora a una tableta.



Layouts de sitios web de fluidos

- En un diseño de página fluido (layout líquida), la mayoría de los componentes internos tienen **anchos en porcentaje** y, por lo tanto, se ajustan a la resolución de pantalla del usuario.



Ventajas y desventajas

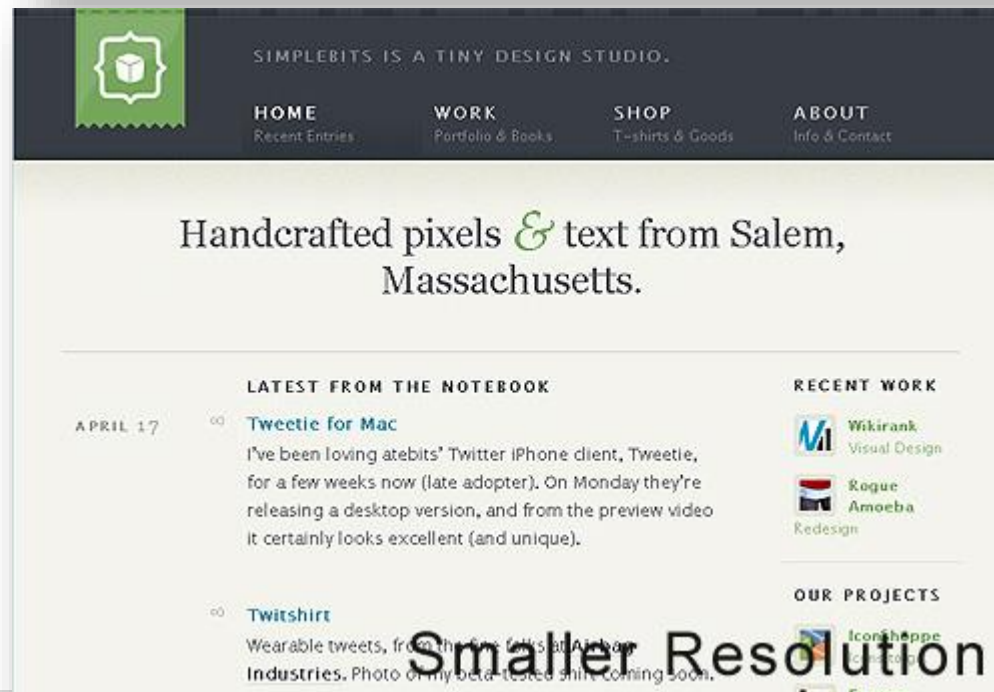
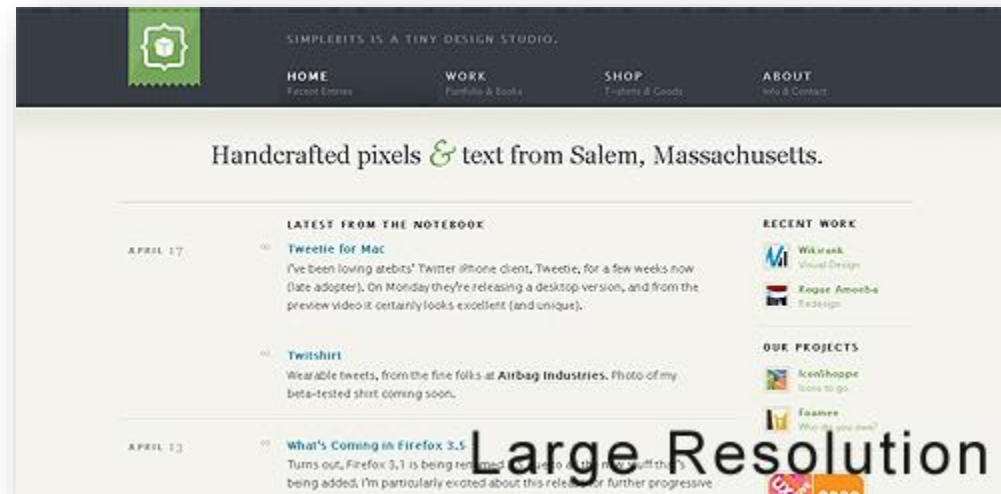
➤ PROS

- Las páginas web fluido pueden ser más amigable, ya que se ajusta a la configuración del usuario.
- La cantidad de espacio extra en blanco es similar entre todos los navegadores y las resoluciones de pantalla, lo que pueden ser más atractivo visualmente.
- Si está diseñado bien, un diseño fluido puede eliminar las barras de desplazamiento horizontal en resoluciones de pantalla más pequeñas.

➤ CONTRAS

- El diseñador tiene menos control sobre lo que el usuario ve y puede pasar por alto los problemas debido a que el diseño se ve bien en su resolución de pantalla específica.
- Las imágenes, videos y otros tipos de contenido con anchos definidos, pueden necesitar ser configurados en varios anchos para acomodarse a diferentes resoluciones de pantalla.
- Con resoluciones de pantalla increíblemente grandes, la falta de contenido puede crear un exceso de espacio en blanco que puede disminuir el atractivo estético.

Ejemplo





Pongámoslo en práctica

Analizando un site fluido



- Accede al sitio <http://andreagandino.com/>
- Redimensiona la pantalla para ver el comportamiento de la página
- Abre el código fuente (CTRL+U) y localiza su archivo de css
 - › <http://andreagandino.com/wp-content/themes/twentyseventeen/style.css?ver=4.7.5>
 - › Analiza el archivo
- Abre el panel de desarrollador del navegador y observa los bloques de la página con el inspector de elementos

Genera un sitio fluído



- Genera un HTML y CSS para el siguiente esquema



Consejos

➤ Usar un diseño simple

- Cuanto menos un diseño fluido dependa de gráficos y técnicas difíciles, más fácil será de crear y mantener.
- También será más compatible con resoluciones de pantalla alternativas.
- Con un código y un diseño más limpios, los problemas de compatibilidad son más fácilmente prevenibles, encontrables y tratables.

➤ min-width y max-width

- Dos propiedades CSS, min-width y max-width, se pueden usar para crear un ancho fijo si la pantalla del usuario es demasiado pequeña o demasiado grande para que el diseño sea utilizable.
- En este caso, el diseño obtiene una barra de desplazamiento y funciona esencialmente como un diseño de ancho fijo

Diseño elástico

- 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.
- Funciona clasificando todos los elementos con **em's**.
- La siguiente cita explica exactamente lo que es un em y por qué puede ser beneficioso.

"Un pixel es un punto no escalable en una pantalla, mientras que un **em** es un cuadrado de su tamaño de fuente. Debido a que los tamaños de fuente varían, el em es una unidad relativa que responde a las preferencias de tamaño de texto de los usuarios. "

– [*Patrick Griffiths, A List Apart*](#)

Ventajas y desventajas

> PROS

- > Si se implementa correctamente, este estilo de diseño puede ser muy fácil de usar.
- > El objetivo es que todo crezca más o menos en proporción con la preferencia del usuario.
- > Los diseños elásticos son perfectos para los diseñadores que aman diseños fluidos y fijos, porque los pros de cada uno se encuentran en diseños elásticos.

> CONTRAS

- > Este tipo de diseño puede crear un gran problema con la usabilidad. Se necesita mucha experiencia y pruebas para obtener el diseño correcto para todos los usuarios.
- > Este tipo de disposición es mucho más difícil de crear que las otras dos, y el poco de utilidad adicional que añade puede no parecer que valga la pena el esfuerzo.
- > Dependiendo de los detalles del diseño, algunos diseños elásticos pueden requerir hojas de estilo suplementarias y trucos para IE6.

Ejemplo





Pongámoslo en práctica

Analizando un site elástico



- Accede al sitio <http://www.elliottjaystocks.com/>
- Redimensiona la pantalla para ver el comportamiento de la página
- Abre el código fuente (CTRL+U) y localiza su archivo de css
 - › <http://www.elliottjaystocks.com/stylesheets/style.css>
 - › Analiza el archivo
- Abre el panel de desarrollador del navegador y observa los bloques de la página con el inspector de elementos

Genera un sitio elástico



- Genera un HTML y CSS para el siguiente esquema



3

Imágenes y Videos Flexibles

El problema de las imágenes

- El problema surge cuando una imagen proporcionada supera las medidas del bloque que la contiene.
- La estructura no se ve afectada, y las proporciones de la columna siguen intactas, pero la imagen excede el tamaño del contenedor, produciéndose un efecto no deseado.

```
<figure>
  <p>
    
    <b class="figcaption">Lo, the robot walks</b>
  </p>
</figure>
```

```
figure {
  float: right;
  margin-bottom: 0.5em;
  margin-left: 2.53164557%; /* 12px / 474px */
  width: 48.7341772%; /* 231px / 474px */
}
```

Post With Threaded Comments

Jardner — 6 Comments



a WordPress post, you could edit this to put information
site so readers know where you are coming from. You can

Imágenes y videos fluidos

- La solución al problema anterior, la vimos en el capítulo anterior y es tan sencillo como aplicar la siguiente restricción a las imágenes:

```
img {
  max-width: 100%;
}
```

- Las imágenes del documento ocuparán como máximo el mismo tamaño que su contenedor, independientemente de su tamaño original.
- El navegador redimensionará la imagen, manteniendo las proporciones, en la medida que su contenedor cambie de tamaño
- Además, esta regla puede ser aplicada al resto de elementos multimedia:

Sample Post With Threaded Comments

April 1, 2013 by Brian Gardner — 6 Comments



This is an example of a WordPress post, you could edit this to put information about yourself or your site so readers know where you are coming from. You can create as many posts as you like in order to share with your readers what is on your mind.

```
img,
embed,
object,
video {
  max-width: 100%;
}
```

Imágenes de fondo

- En 2004, Dan Cederholm <http://bkaprt.com/rwd/18/>, publicó un magnífico artículo para conseguir un efecto de dos columnas a partir de imágenes de fondo.
- La técnica es muy sencilla, y crea una imagen con colores diferenciados que simulan crear columnas de un mismo alto.
- Este efecto se consigue con una simple regla de CSS:

```
.blog {  
    background: #F8F5F2 url("blog-bg.png") repeat-y 50% 0;  
}
```

- Esta técnica funciona perfectamente, pero está pensada para estructuras de ancho fijo, por lo tanto debemos adaptarla para que funcione correctamente en nuestros diseños flexibles.
- La manera para conseguirlo es aplicar la fórmula:

$$\text{target} \div \text{context} = \text{result}$$



Generando un fondo con imagen flexible

Busca el punto de transición



- Primero el pixel exacto donde se simula la transición entre las columnas.
- Vamos a suponer que el ancho inicial de nuestra columna es de 900px, y que la separación entre las columnas se produce en el pixel 568px.



- Aplicamos la fórmula que conocemos para calcular el nuevo punto de manera relativa:

$$568 \div 900 = 0.631111111111111 \rightarrow 63.1111111111111\%$$

- Y tendríamos el siguiente estilo:

```
.blog {  
    background: #F8F5F2 url("blog-bg.png") repeat-y 63.1111111111111% 0;  
}
```


Imágenes de fondo totalmente flexibles

- La solución anterior no es del todo flexible, ya que no estamos redimensionando la imagen sino trasladándola para mostrarla correctamente.
- Esta solución no es válida si necesitamos una imagen de fondo que deba redimensionarse, como un logo o sprites que hemos utilizado en nuestros enlaces.
- La solución a este problema viene dada por la propiedad `background-size` de CSS 3, la cual nos permite realmente redimensionar las imágenes de fondo.

```
background-size: auto|length|percentage|cover|contain|initial|inherit;
```

```
div {  
    background-size: 80px 60px;  
}
```

4

Media Queries

Mejoras para consultas relacionadas con multimedia

- Consisten en una serie de mejoras relacionadas con la regla (directiva) **@media**
- Mediante la regla @media y las palabras reservadas disponibles, podemos establecer indicaciones especiales para muchos dispositivos de salida
 - Podemos indicar un cierto estilo aplicable a las salidas por impresora, o la disposición visual en un teléfono o TV
 - En la versión anterior, la lista completa de media types incluía 'aural', 'braille', 'handheld', 'print', 'projection', 'screen', 'tty' y 'tv'
 - En la nueva "aural", se considera obsoleta, debido al nuevo soporte de atributos WAI-ARIA
 - Se añaden los tipos "embossed", y "speech"

Mejoras para consultas relacionadas con multimedia (II)

- Por ejemplo, para indicar una regla para los dispositivos móviles, podemos utilizar los operadores **only** y **and**
- Una indicación de orientación para definir cómo queremos que se muestren los elementos de una lista

```
ul { overflow: hidden; }  
li { float: left; }
```

```
@media only screen and (orientation: portrait#landscape) {  
  li { float: none; }  
}
```

CSS para distintos medios

- También podemos usar otros tipos de definición condicional en el código.
- Usando la etiqueta `<link>` para enlazar los archivos CSS externos, se puede utilizar el atributo `media` para indicar los medios donde hay que aplicar los estilos de cada archivo:

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

- Si la etiqueta `<link>` no indica el medio CSS, se sobreentiende que los estilos se deben aplicar a todos los medios, por lo que es equivalente a indicar `media="all"`.

Queries en función del tamaño visual

- También podemos
 - Indicar condiciones para el tamaño visual del dispositivo de salida
 - Exigir unos tamaños mínimos o máximos antes de aplicar una regla
- Los prototipos de código adoptarían una sintaxis genérica similar a los siguientes:

```
@media [media] and (device-height:value) { ... }  
@media [media] and (max-device-height:value) { ... }  
@media [media] and (min-device-height:value) { ... }
```

Queries en función de aspecto

- Es posible establecer condiciones más específicas, como la relación de aspecto (aspect ratio) pero que se dan en conjuntos de dispositivos similares (móviles)

```
@media [medio] and (aspect-ratio: horizontal/vertical) {...}  
@media [medio] and (device-aspect-ratio: horizontal/vertical) {...}
```

- Lo mismo cabe decir de otros aspectos visuales como el “pixel ratio” que podríamos manejar comprobándolo mediante indicaciones que tendrán un código genérico del tipo:

```
@media [medio] and (-moz-device-pixel-ratio: [número]) {...}
```

Diseños según puntos de ruptura

- Para hacer un diseño flexible se toma en cuenta un punto de ruptura, de manera que cuando el tamaño de pantalla alcance dicho límite, se aplicarán unas reglas de diseño específicas.
- De esta manera, se puede diseñar una pagina web para varias dimensiones distintas.



- Marca
- Navegación global
- Navegación local
- Búsqueda
- Contenido principal
- Contenido secundario

Distintos archivos por breakpoint

- CSS3 nos permite construir diseños flexibles, definiendo reglas de ajuste a pantallas distintas mediante **media queries**.
- Se pueden crear múltiples hojas de estilos, donde se puedan alterar el diseño dependiendo de las condiciones de tamaño.

```
<link rel="stylesheet" type="text/css"  
media="screen and (max-device-width: 480px)" href="style1.css" />
```

```
<link rel="stylesheet" type="text/css"  
media="screen and (max-device-width: 960px)" href="style2.css" />
```

- El código anterior indica que si se muestra la pagina en una pantalla y el ancho es de 480 pixeles o menos, se aplicará el estilo definido en el archivo style1.css.
 - En el caso de 960px, el style2.css
- De esta manera se puede diseñar en base a un numero máximo de pixeles en la pantalla.

Breakpoints

- Los puntos o medidas de anchura donde se pueden crear saltos en el diseño Responsive, llamados comúnmente breakpoints, a partir de donde aplicar las media queries para Responsive Web Desing.
- Existen un conjunto de breakpoints para distintos dispositivos

```
@media (min-width:320px) { /* smartphones, iPhone, portrait 480x320 phones */ }  
@media (min-width:481px) { /* portrait e-readers (Nook/Kindle), smaller tablets @ 600 or @ 640  
wide. */ }  
@media (min-width:641px) { /* portrait tablets, portrait iPad, landscape e-readers, landscape  
800x480 or 854x480 phones */ }  
@media (min-width:961px) { /* tablet, landscape iPad, lo-res laptops ands desktops */ }  
@media (min-width:1025px) { /* big landscape tablets, laptops, and desktops */ }  
@media (min-width:1281px) { /* hi-res laptops and desktops */ }
```

- <https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

Ejemplo

```
@media only screen  
and (min-device-width : 320px)  
and (max-device-width : 480px) {  
/* Styles */  
}
```

```
@media only screen  
and (min-width : 640px) {  
/* Styles */  
}
```

```
@media only screen  
and (max-width : 960px) {  
/* Styles */  
}
```

- La primera condición indica que solo se aplicara el estilo definido a una pantalla con un ancho entre 320 y 480 pixeles
- La segundo indica que el estilo solo se aplicara a pantallas con una pantalla de mínimo 640 pixeles de ancho
- La tercero solo a un máximo de 960 pixeles de ancho.
- De esta manera se definen tres diseños diferentes para una pantalla pequeña, mediana y grande.

Ejemplos

```
@media screen and (max-width: 1000px) {
  #content { width: 100% }
}
```

```
@media screen and (max-width: 800px) {
  #nav { float: none }
}
```

```
@media screen and (max-width: 600px) {
  #content aside {
    float: none;
    display: block;
  }
}
```

```
@media screen and (max-width: 1000px) {
  #content { width: 100% }
}
```

```
@media screen and (max-width: 800px) {
  #nav { float: none }
}
```

```
@media screen and (max-width: 600px) {
  #content aside {
    float: none;
    display: block;
  }
}
```

Orientación

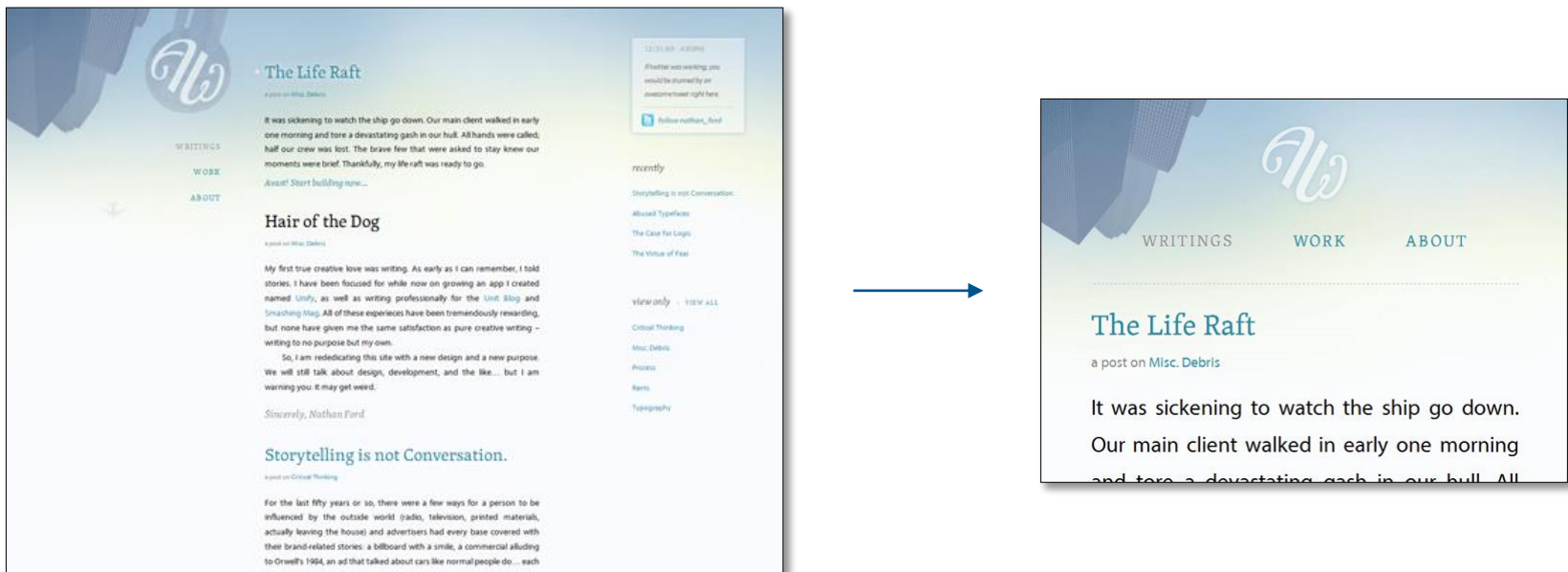
- También se pueden aplicar condiciones según la orientación:

```
@media screen and (orientation: landscape) {  
  .iPadLandscape {  
    width: 30%;  
    float: right;  
  }  
}  
  
@media screen and (orientation: portrait) {  
  .iPadPortrait {  
    clear: both;  
  }  
}
```

- La primera condición se aplica para orientación horizontal y la segunda para vertical.

Mostrar y Ocultar Contenido

- Es posible que se desee organizar o reducir los elementos de una pagina web para hacer que todo encaje a medida que una pantalla va cambiando de tamaño u orientación.
- El diseño no solamente debe ser flexible, también debe ser capaz de escoger y elegir el contenido adecuado a mostrar.



Mostrar y Ocultar Contenido

- Podemos comenzar a mostrar u ocultar elementos que nos puede ayudar a que nuestra pagina web se vea de mejor manera en diferentes situaciones

➤ Versión ordenador

```
@media (min-width:1025px) {
    #content{
        width: 100%;}

    #sidebar-left{
        display: none;}

    #sidebar-right{display: none;}
    .sidebar-nav{display: inline;}
}
```

➤ Versión mobile

```
@media (min-width:320px) {
    #content{
        width: 54%;
        float: left;
        margin-right: 3%;
    }

    #sidebar-left{width: 20%;
        float: left; margin-right: 3%;
    }

    #sidebar-right{width: 20%; float: left;}
    .sidebar-nav{display: none;}
}
```



Creando BananaTube Responsivo

- Usa las técnicas propuestas para lograr una versión responsiva de BananaTube
- Haz que sea visible correctamente en dispositivos pequeños (smartphones), y en pantallas grandes (portátiles y ordenador)



 **netmind**

WeKnowIT

Barcelona

C. Almogàvers, 123
08018 Barcelona
Tel. 93 304.17.20
Fax. 93 304.17.22

Madrid

Plaza Carlos Trías Bertrán, 7
28020 Madrid
Tel. 91 442.77.03
Fax. 91 442.77.07

www.netmind.es



MINISTERIO
DE ENERGÍA, TURISMO
Y AGENDA DIGITAL

red.es



UNIÓN EUROPEA

Fondo Social Europeo
"El FSE invierte en tu futuro"