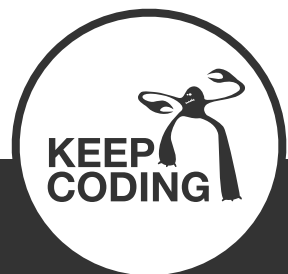




■ Introducción



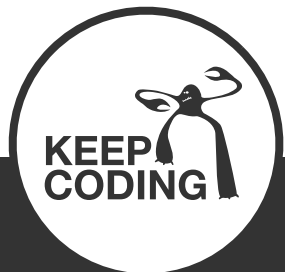
■ Introducción

Tras la aparición de smartphones y tablets, nacieron las “versiones para móviles” de los sitios web.

https://youtu.be/Shu6_IO1PW8?t=1h12m50s

Estas “versiones móviles”, acababan siendo páginas web abandonadas que nadie mantenía.

La solución pasaba por tener una única página que se adaptara al dispositivo en el que fuera a ser visionada.



■ Introducción

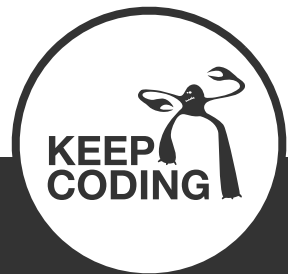
El diseño web responsive pretende:

- Adaptar la apariencia de una página o aplicación web al dispositivo donde se muestra
- Es decir, tener un solo diseño y que éste pueda adaptarse
- Evitar desarrollos ad-hoc para cada dispositivo
- Evitar la complejidad de tener varias versiones de una misma página o aplicación



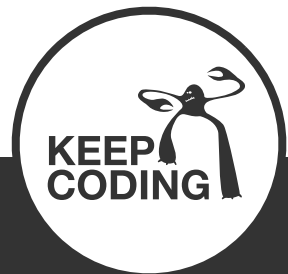
■ Introducción

RWD es un arte, no una ciencia. Hay que practicar para mejorar nuestra capacidad de “pensar en responsive”.





Mobile first



■ Mobile first



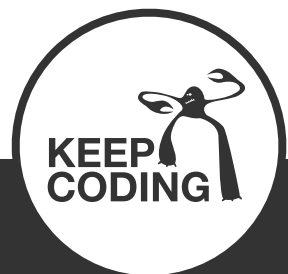
Siempre debemos empezar a desarrollar los componentes de la web para el dispositivo que más restricciones de tamaño tenga.
Dicho de otra manera: debemos empezar por el más pequeño.





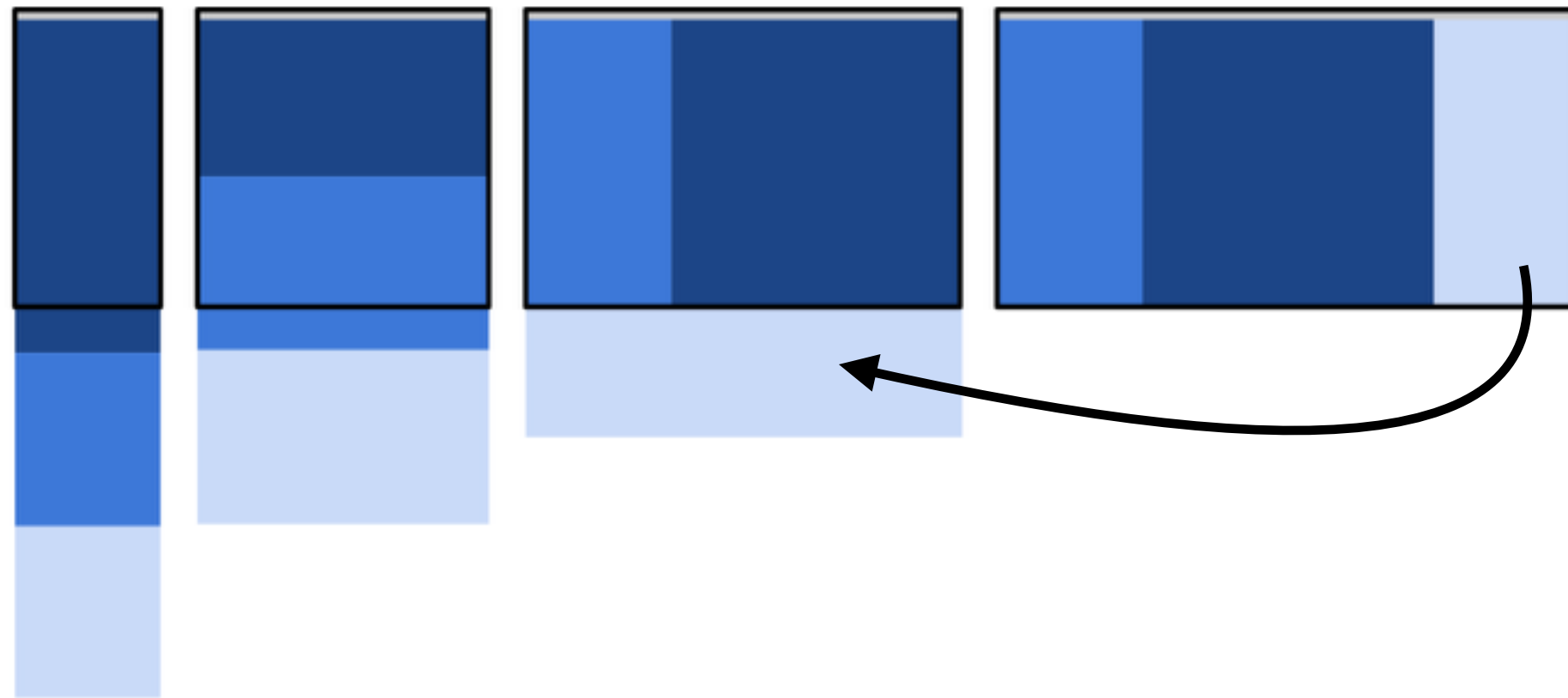
■ Patrones de layout

Column drop, Mostly fluid, Layout Shifter, Off canvas, Tiny Tweaks

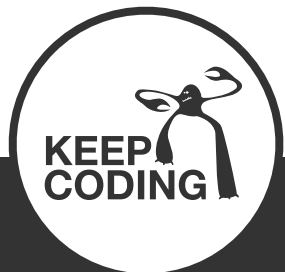


Column drop

Cuando la pantalla se hace más pequeña, las columnas se apilan verticalmente cuando no entran en pantalla.

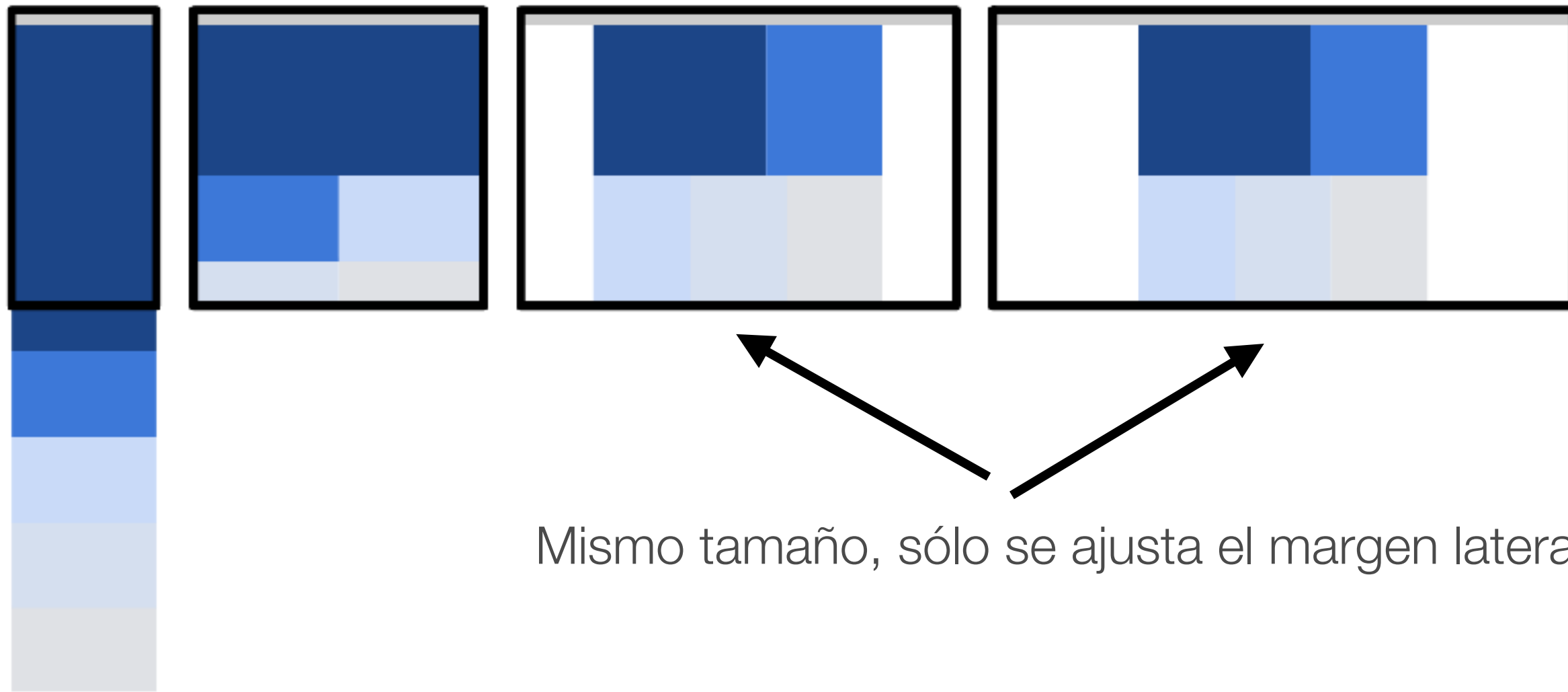


<https://modernizr.com>



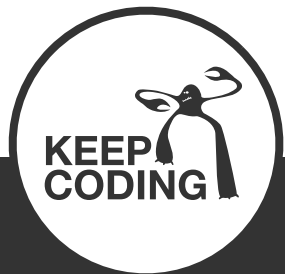
■ Mostly fluid

En las pantallas grandes o medianas, se mantiene el mismo tamaño; simplemente se ajustan los márgenes en las pantallas más anchas.



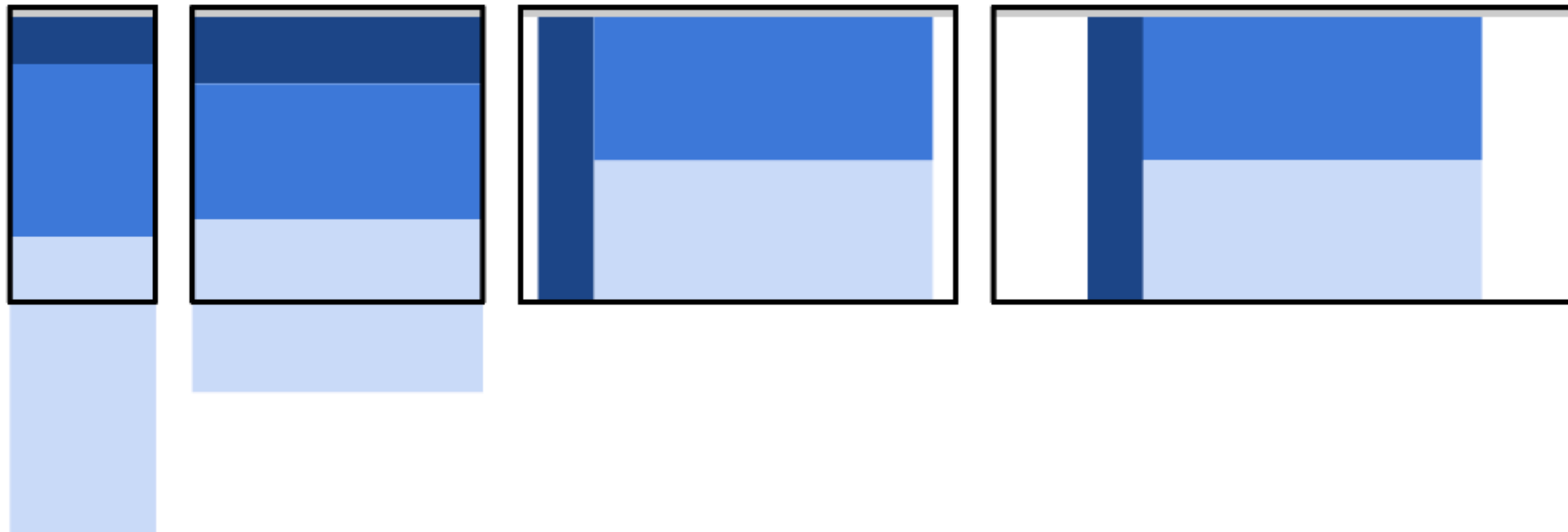
Mismo tamaño, sólo se ajusta el margen lateral.

<http://mediaqueri.es>

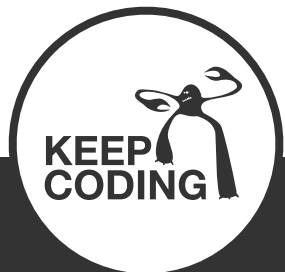


Layout shifter

El más receptivo y difícil de mantener ya que en lugar de apilar columnas verticalmente, es posible que tengamos que realizar cambios en los elementos.

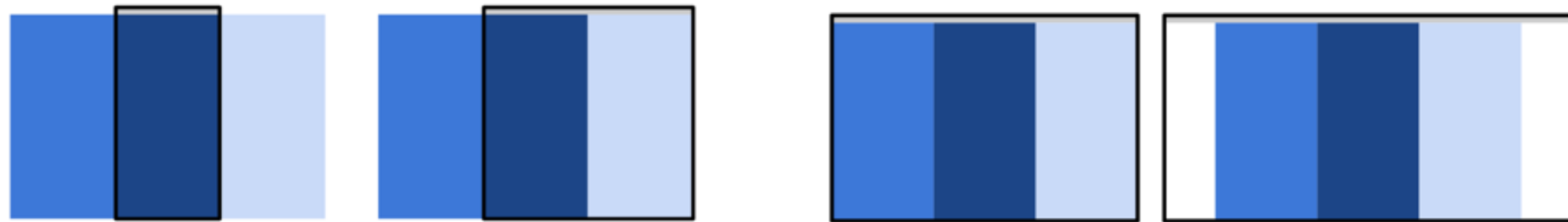


<http://foodsense.is>

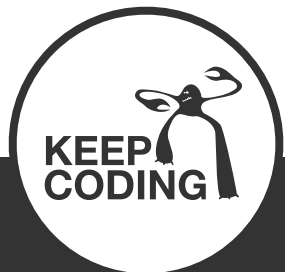


■ Off canvas

En lugar de apilar verticalmente contenido, los elementos menos usados (como menús de navegación) se colocan fuera de la pantalla y se muestra mediante alguna interacción del usuario (por ejemplo pulsar un botón)



<http://www.html5rocks.com/en/tutorials/developertools/async-call-stack/>



■ Tiny tweaks

El patrón Tiny tweaks permite realizar pequeños cambios en el diseño, como ajustar el tamaño de la fuente, cambiar el tamaño de las imágenes o desplazar el contenido de diferentes maneras.

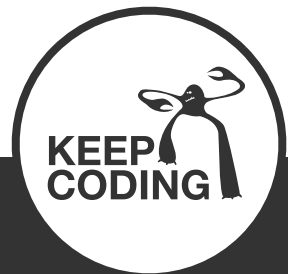


<http://futurefriendlyweb.com>



■ Patrones de layout

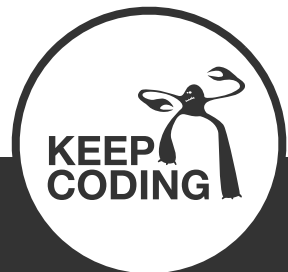
No tienes por qué usar un único patrón, los patrones se pueden combinar entre los diferentes elementos que tiene una web.





■ La meta-etiqueta viewport

Define el comportamiento del área visible de una página web (viewport) en un dispositivo.

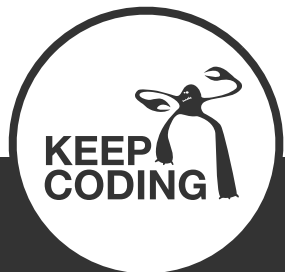


■ La meta-etiqueta viewport

```
<meta name="viewport"
      content="width=device-width, initial-scale=1.0,
              maximum-scale=1.0, user-scalable=no">
```

Explicación:

- width=device-width: el ancho de la web es el ancho del dispositivo
- initial-scale=1.0: el zoom inicial es 1
- maximum-scale=1.0: el máximo zoom es 1 (no hay más zoom)
- user-scalable=no: el usuario no puede hacer zoom





■ Media queries

Las media queries nos permiten establecer puntos de ruptura donde se modifican propiedades de estilos CSS en función de condiciones del dispositivo (como el ancho de la pantalla).

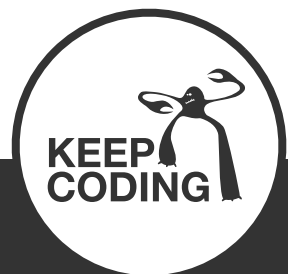


Comprendiendo las **@media** queries

```
@media tipo_de_dispositivo and|not|only (atributo) {  
    /* Estilos */  
}
```

Tipos de dispositivo:

- **all**, para cualquier tipo de dispositivo
- **print**, para impresoras
- **screen**, para pantallas de ordenador, tablet o smartphone
- **speech**, para dispositivos que leen la pantalla

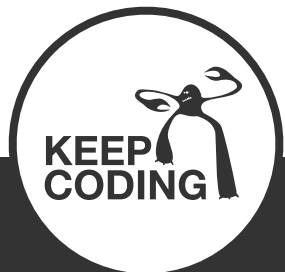


Comprendiendo las **@media** queries

```
@media tipo_de_dispositivo and|not|only (atributo) {  
    /* Estilos */  
}
```

Atributos:

- **max-width**, indica el ancho máximo de área disponible
- **max-device-width**, indica el ancho máximo del dispositivo
- **orientation**, indica la disposición del dispositivo
- ...

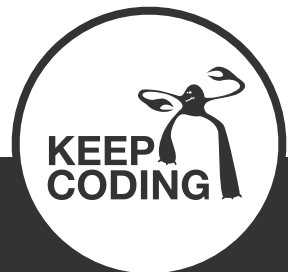


Ejemplos de @media queries

```
/* Pantalla de ordenador, tablet o smartphone dispuesta horizontalmente */  
@media screen and (orientation: landscape) { ... }
```

```
/* Impresión dispuesta verticalmente, o bien cualquier dispositivo cuyo  
ancho máximo sea 800px */  
@media print and (orientation: portrait), (max-width: 800px) { ... }
```

```
/* Solo pantallas de ordenador, tablet o smartphone en color */  
@media only screen and (color) { ... }
```

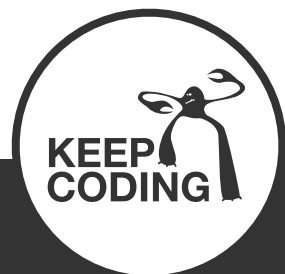


@media queries como referencias a hojas de estilo

```
<!-- Pantalla de ordenador, tablet o smartphone dispuesta horizontalmente -->  
<link rel="stylesheet" media="screen and (orientation: landscape)"  
href="my_stylesheet_01.css">
```

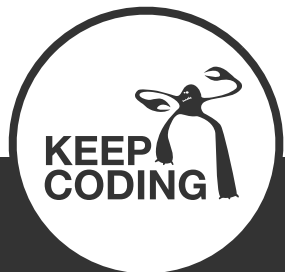
```
<!-- Impresión dispuesta verticalmente, o bien cualquier dispositivo cuyo ancho  
máximo sea 800px -->  
<link rel="stylesheet" media="print and (orientation: portrait), (max-width:  
800px)" href="my_stylesheet_02.css">
```

```
<!-- Solo pantallas de ordenador, tablet o smartphone en color -->  
<link rel="stylesheet" media="only screen and (color)"  
href="my_stylesheet_03.css">
```





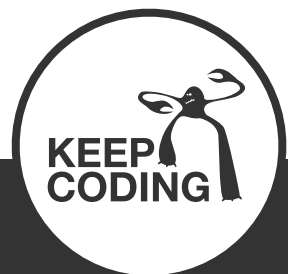
No hagas trabajo de más. Hay muchos frameworks CSS3 que aportan diseño adaptativo a tus páginas y aplicaciones web, como **Bootstrap**, **Foundation** o **Skeleton**. Haz uso de ellos y adapta solo aquellos aspectos que no encajen con los requerimientos de tus diseños.



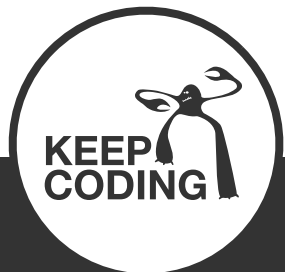
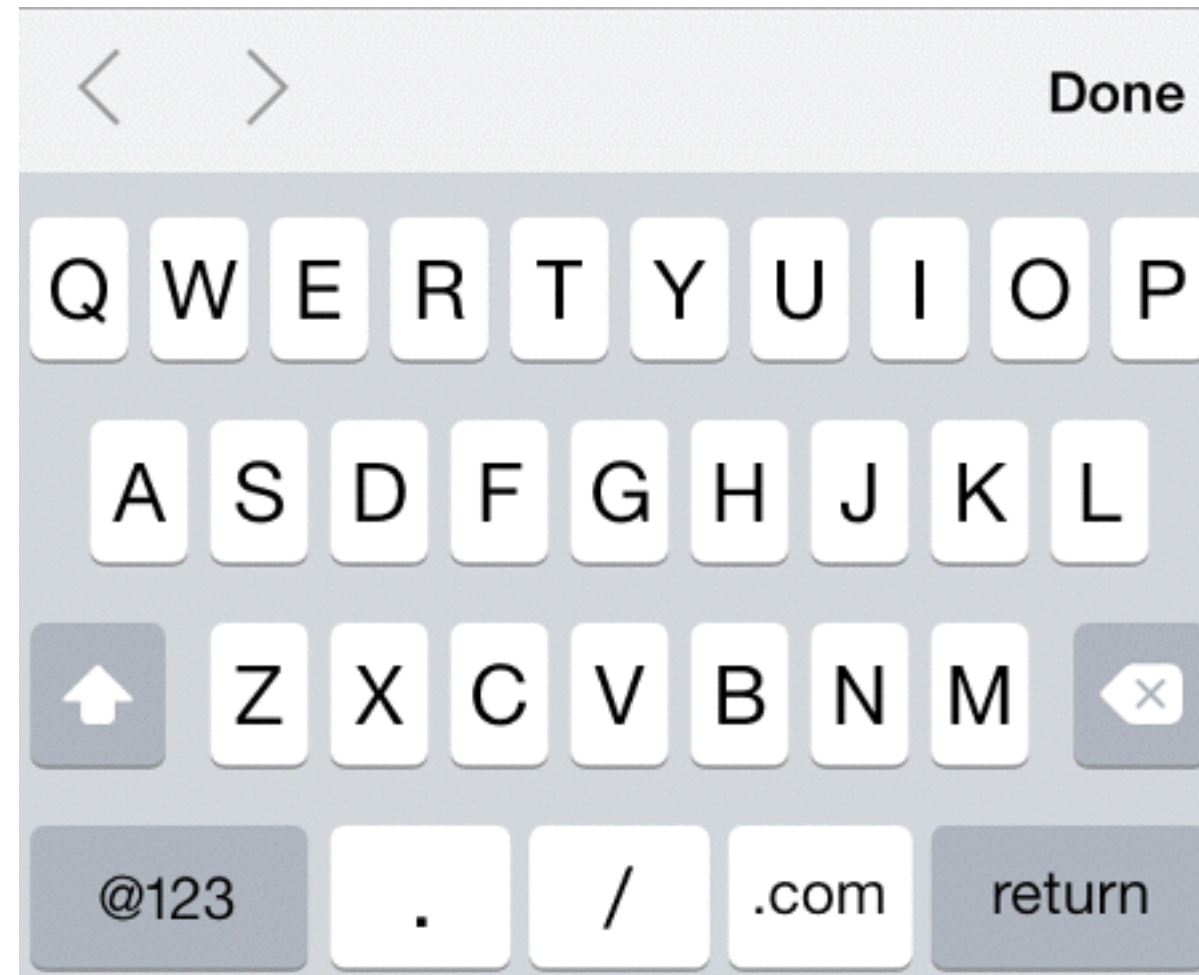


■ User input

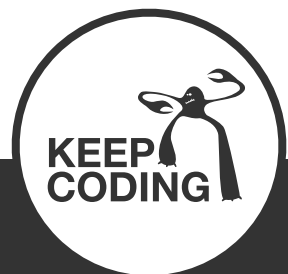
Hay que optimizar los inputs de usuario que utilizamos pensando en las posibilidades de los usuarios a través de cada dispositivo.



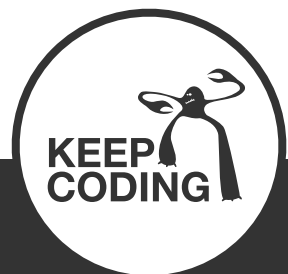
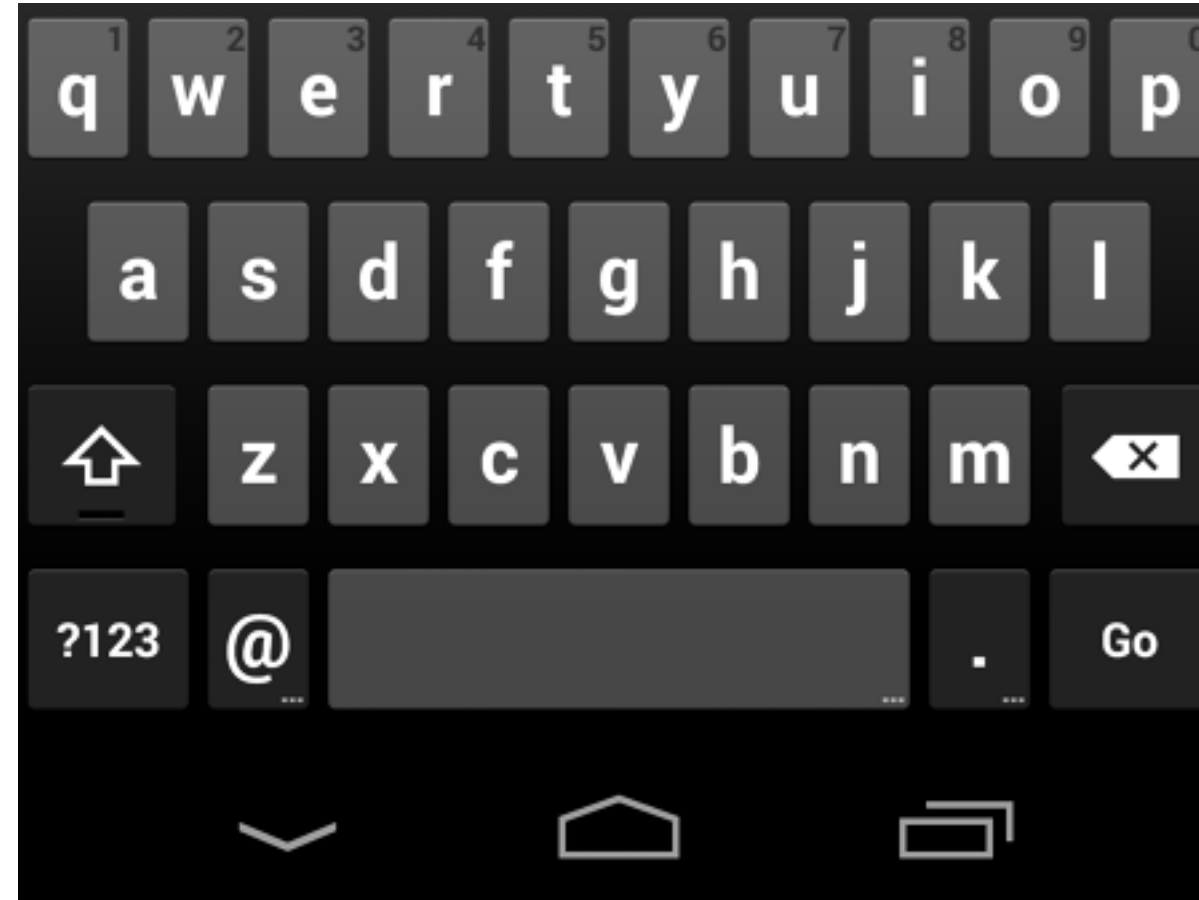
■ <input type="url">



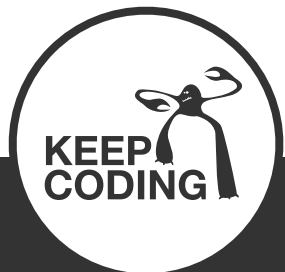
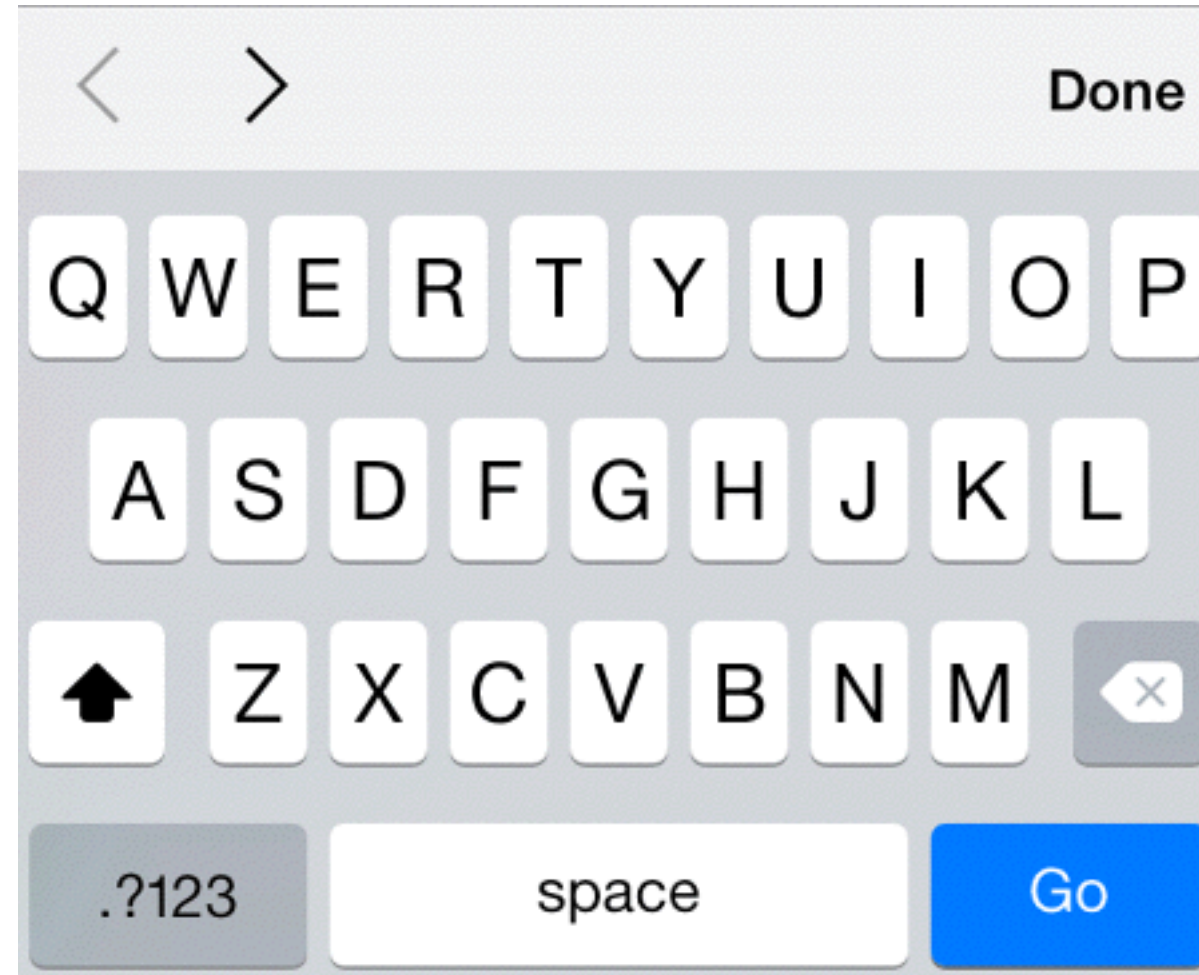
■ <input type="tel">



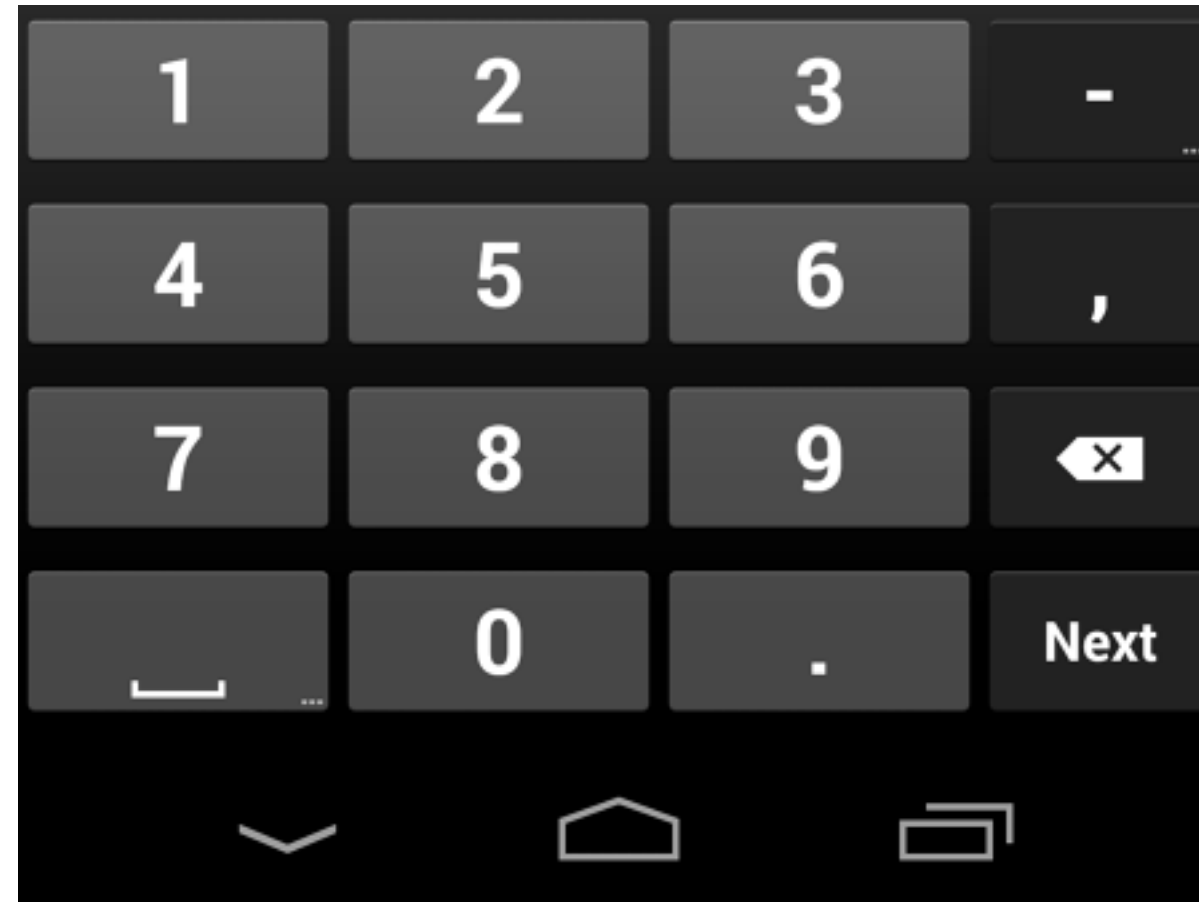
■ `<input type="email">`



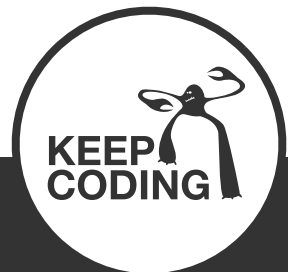
■ <input type="search">



■ <input type="number">



■ <input type="range">



■ <input type="datetime-local">

<

>

Clear

Done

Sat Apr 12	5	42	
Sun Apr 13	6	43	
Mon Apr 14	7	44	
Tue Apr 15	8	45	AM
Today	9	46	PM
Thu Apr 17	10	47	
Fri Apr 18	11	48	
Sat Apr 19	12	49	



■ <input type="date">

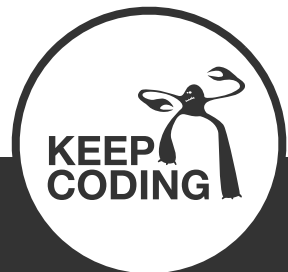
Set date

Mar	15	2013
Apr	16	2014
May	17	2015

Cancel

Clear

Set



■ <input type="time">

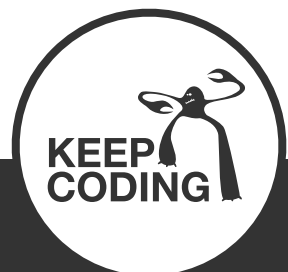
<

>

Clear

Done

6	44	
7	45	
8	46	AM
9	47	PM
10	48	
11	49	
12	50	

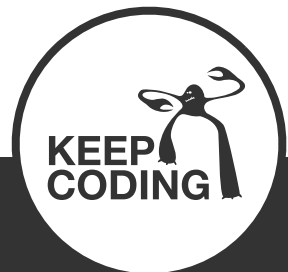


■ <input type="week">

Set week

15	2013
16	2014
17	2015

Cancel	Clear	Set
--------	-------	-----



■ <input type="month">

<

>

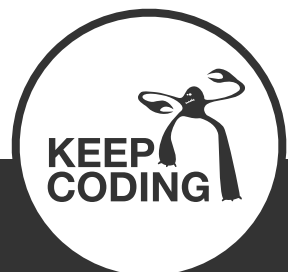
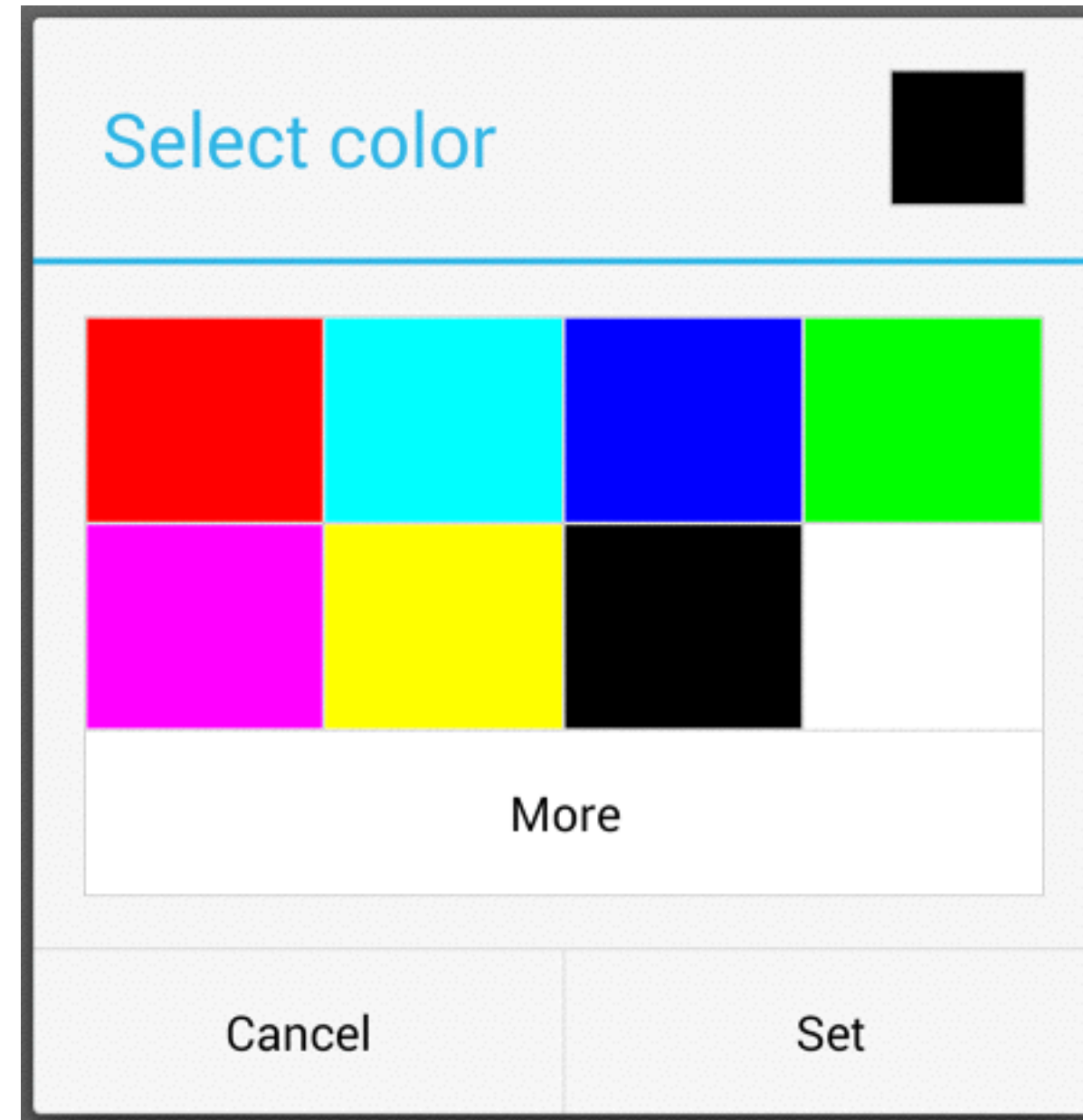
Clear

Done

December	2010
January	2011
February	2012
March	2013
April	2014
May	2015
June	2016
July	2017
August	2018



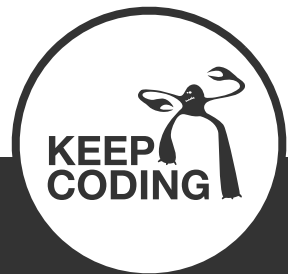
■ <input type="color">





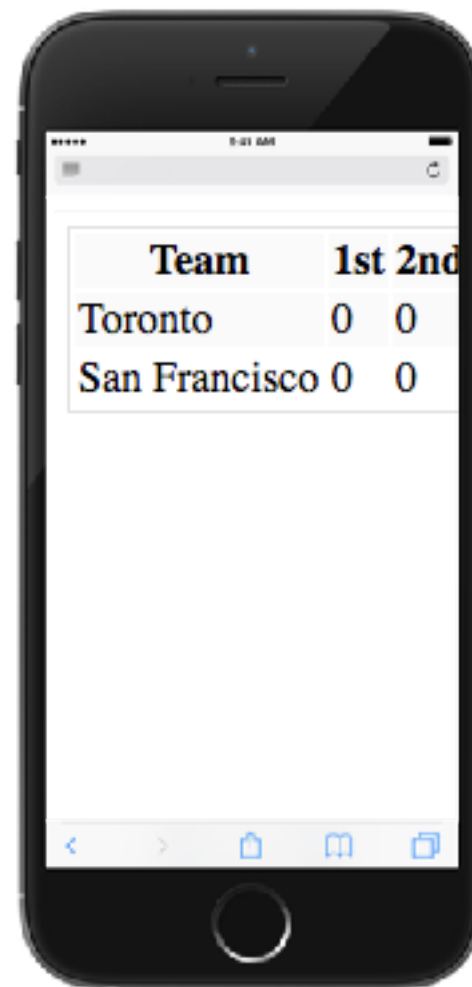
■ Tablas

Otro de los quebradero de cabeza

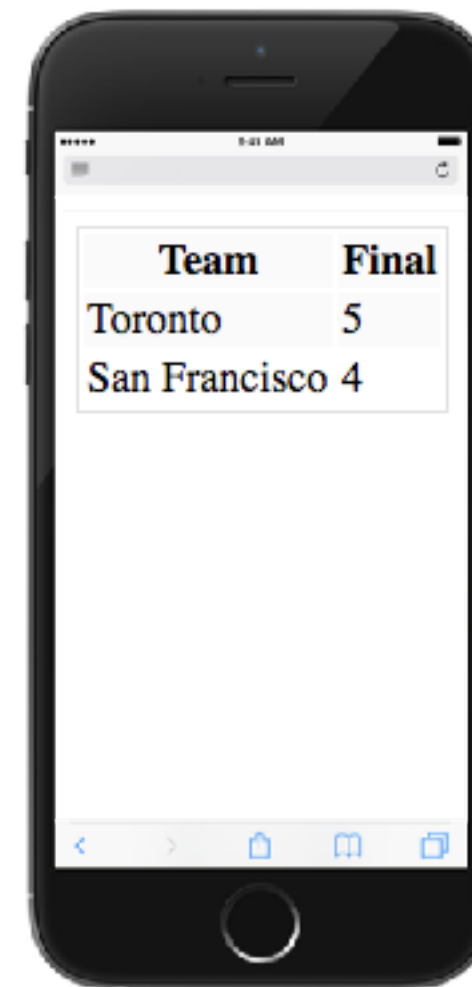


■ Columnas ocultas

Esta técnica básicamente permite ocultar ciertas columnas de la tabla para que la tabla entre completamente en la pantalla.

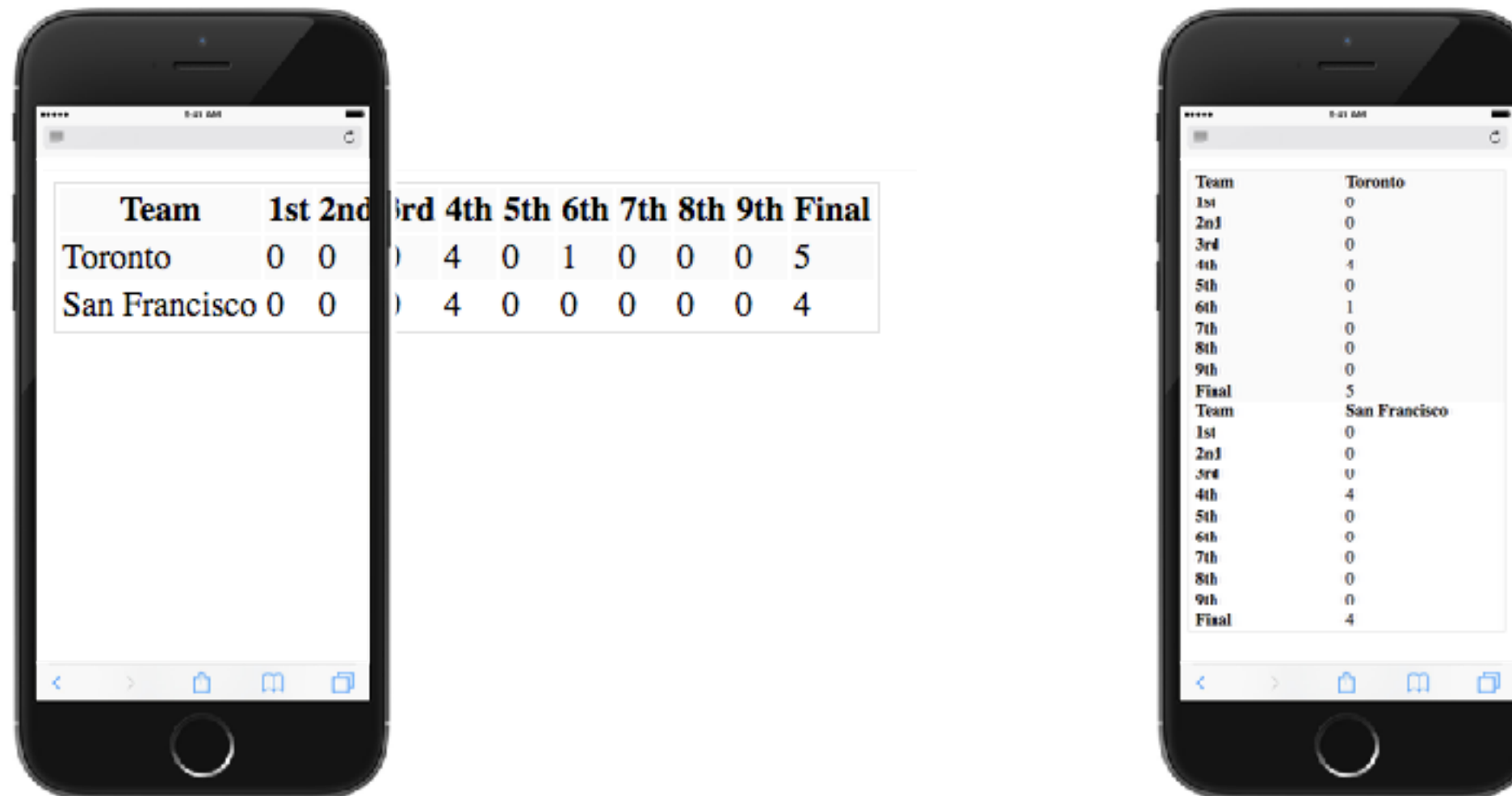


1st	2nd	3rd	4th	5th	6th	7th	8th	9th	Final
0	0	0	4	0	1	0	0	0	5
0	0	0	4	0	0	0	0	0	4



■ No más tablas

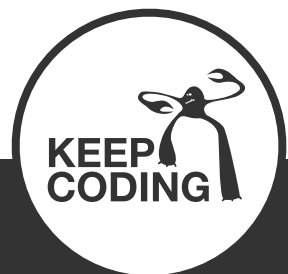
Esta técnica se basa en cambiar filas por columnas en la tabla, de manera que se pueda mostrar verticalmente.



Team	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	Final
Toronto	0	0	0	4	0	1	0	0	0	5
San Francisco	0	0	0	4	0	0	0	0	0	4

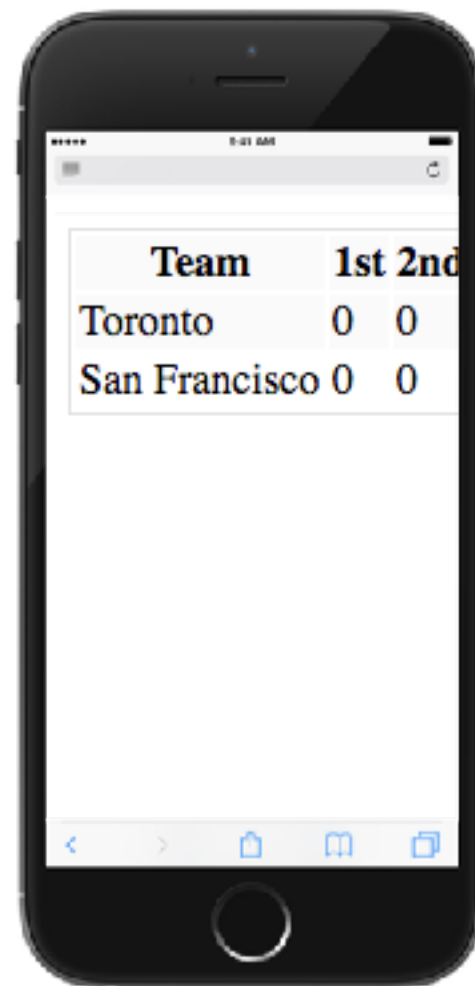
Team	Toronto
1st	0
2nd	0
3rd	0
4th	4
5th	0
6th	1
7th	0
8th	0
9th	0
Final	5
Team	San Francisco
1st	0
2nd	0
3rd	0
4th	4
5th	0
6th	0
7th	0
8th	0
9th	0
Final	4

<http://codepen.io/JohnMav/pen/BoGJNy>



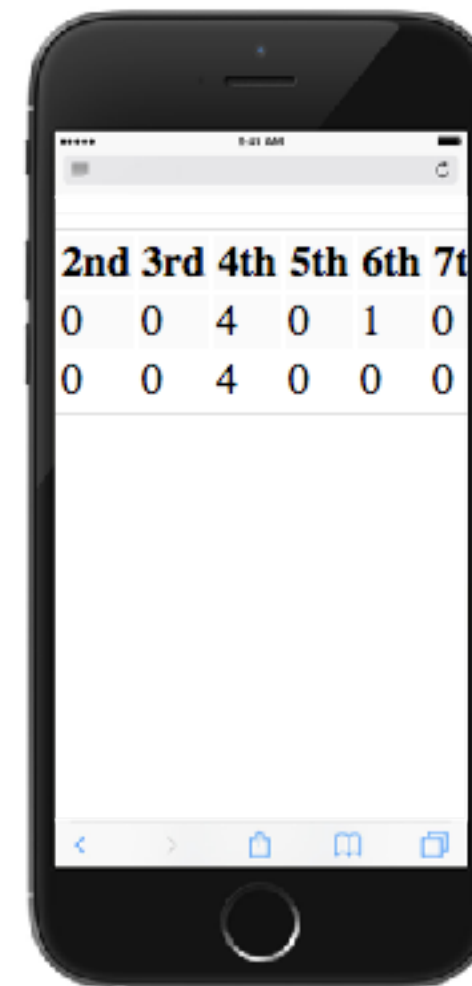
■ Scrollable

Otra opción, es hacer que la tabla permita realizar scroll horizontal.



A smartphone mockup displaying a table with vertical scrolling. The table has three columns: 'Team', '1st', and '2nd'. The data rows are 'Toronto' and 'San Francisco', both with '0' in the '1st' and '2nd' columns.

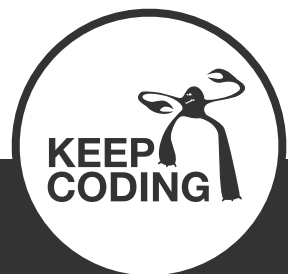
Team	1st	2nd
Toronto	0	0
San Francisco	0	0



A smartphone mockup displaying a table with horizontal scrolling. The table has seven columns: '2nd', '3rd', '4th', '5th', '6th', and '7th'. The data rows are 'Toronto' and 'San Francisco', with values '0', '0', '4', '0', '1', '0' for Toronto and '0', '0', '4', '0', '0', '0' for San Francisco.

2nd	3rd	4th	5th	6th	7th
0	0	4	0	1	0
0	0	4	0	0	0

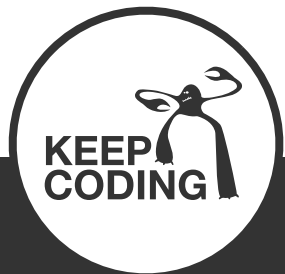
<http://codepen.io/JohnMav/pen/Mazrwm>





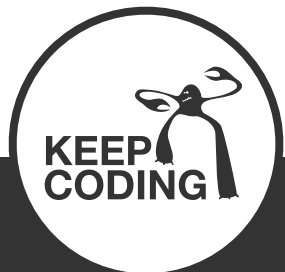
■ Imágenes

Uno de los mayores retos del RWD es la gestión de imágenes.

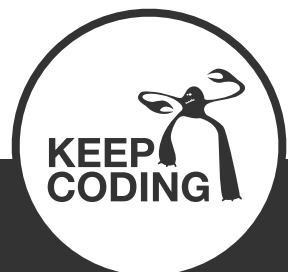


`max-width: 100%`

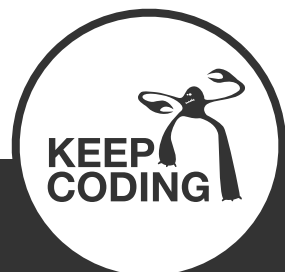
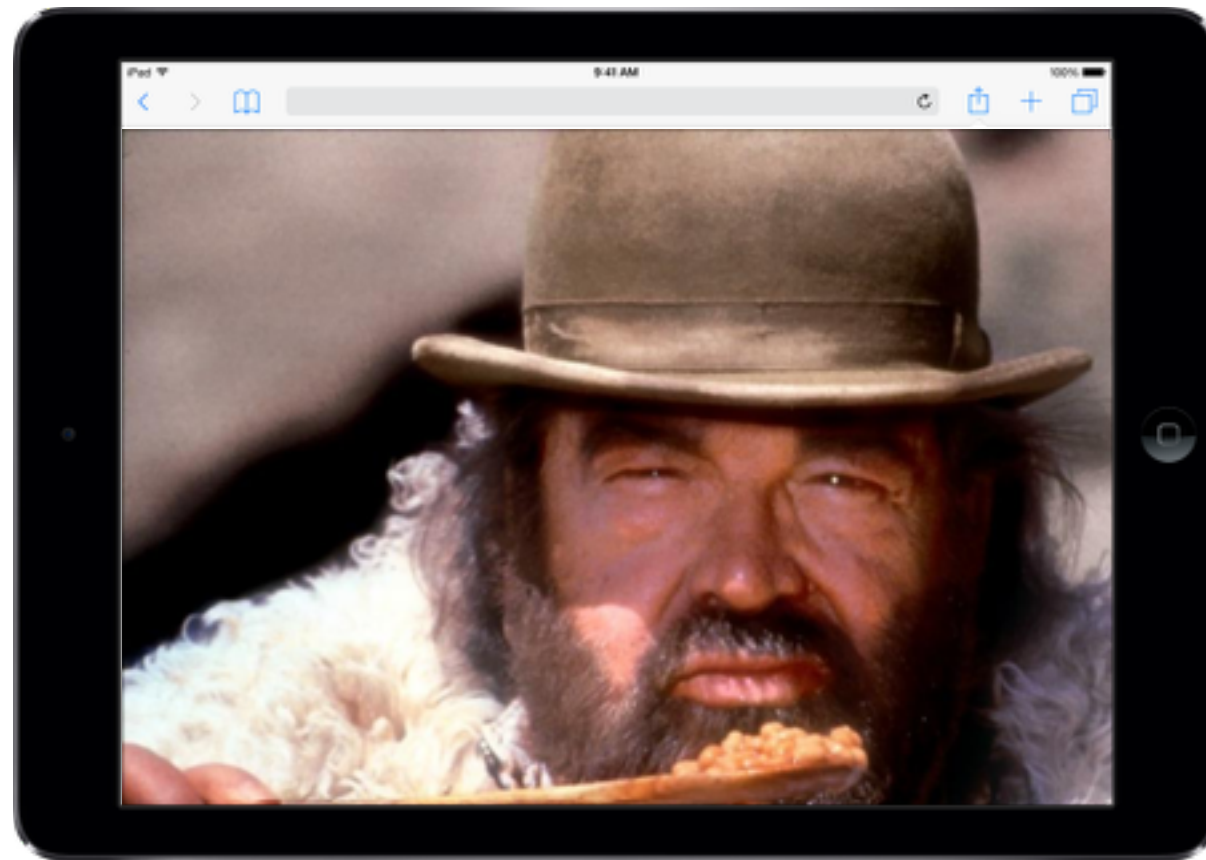
La propiedad `max-width` se usa para establecer el ancho máximo de un elemento. Evita que los valores usados por la propiedad `width` sean mayores que el valor especificado por `max-width`.



■ Sin max-width: 100%



■ Sin max-width: 100%



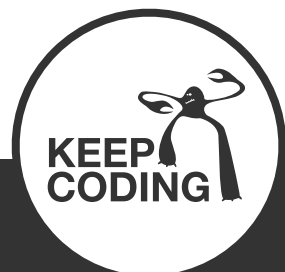
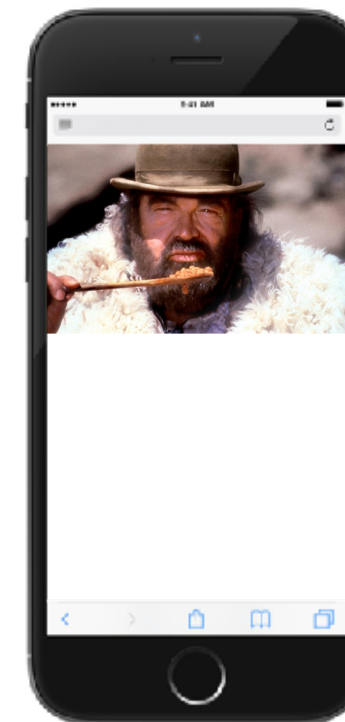
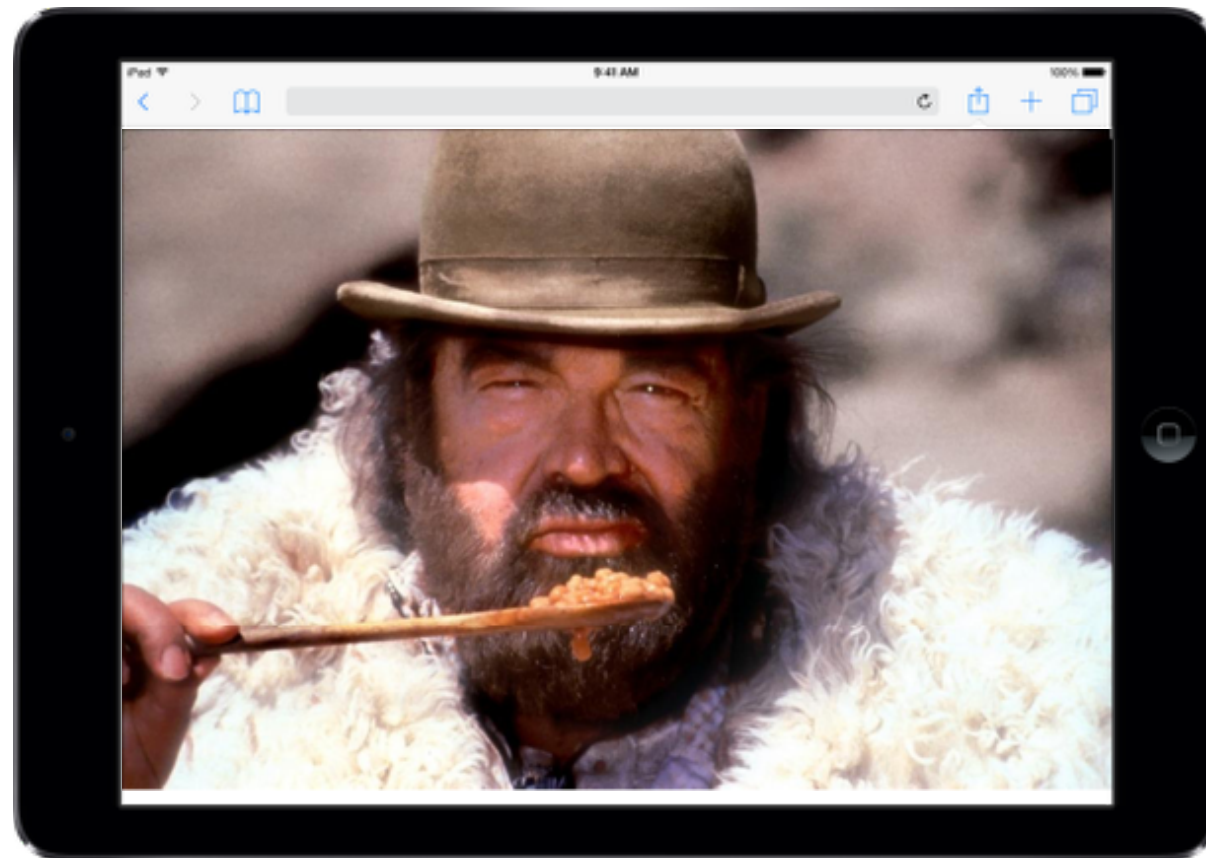
■ max-width: 100%

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

Hace imágenes, vídeos y “objetos” responsive por defecto.



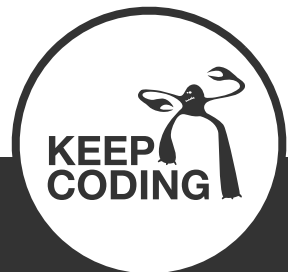
■ Con max-width: 100%



■ Optimización de imágenes

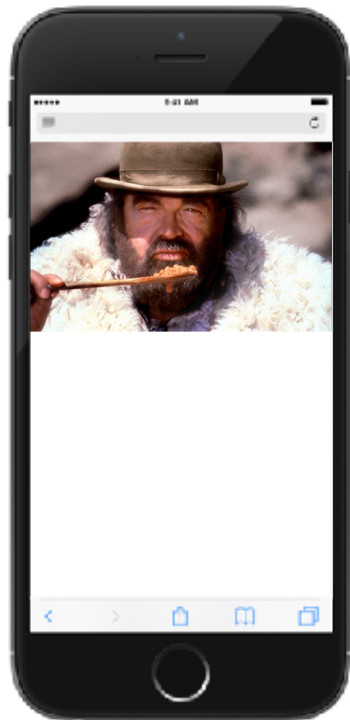
Es importante optimizar las imágenes que utilizamos para:

- Que la web cargue rápido
- Consumir menos ancho de banda (menos costes de hosting)
- No fundir la tarifa de datos de los usuarios de nuestra web



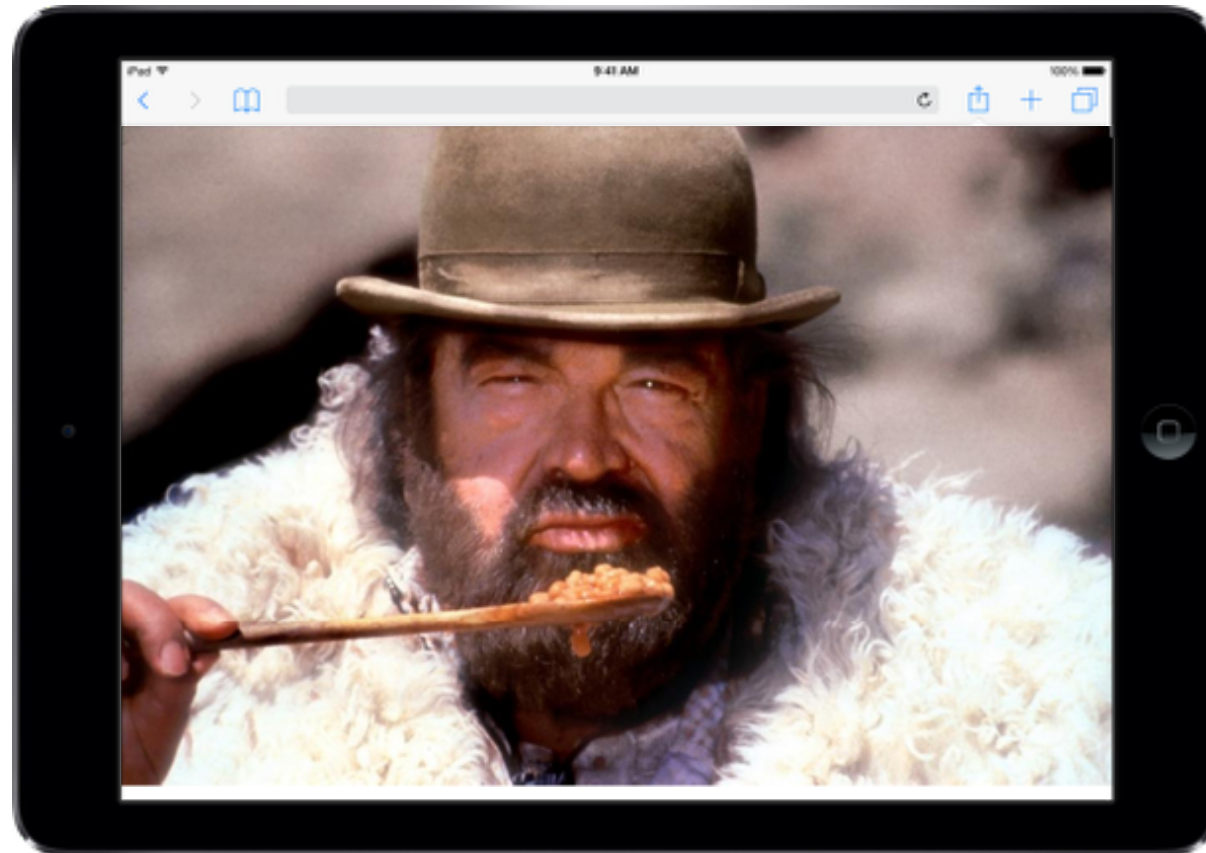
■ ¿Una imagen para dominarlos a todos?

375px
de ancho



Desperdicio:
 $1440 - 375\text{px} = 1065\text{px}$

1024px de ancho

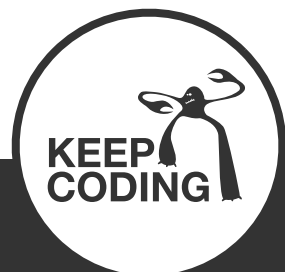


Desperdicio: $1440\text{px} - 1024\text{px} = 416\text{px}$

1440px de ancho



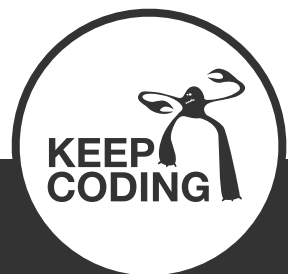
Desperdicio:
 $1440\text{px} - 1440\text{px} = 0\text{px}$



Usar la misma imagen para todos los dispositivos **no es una buena estrategia.**



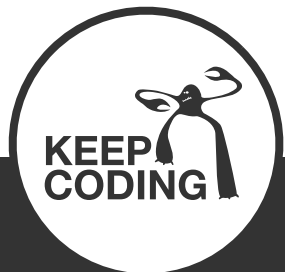
El atributo ***srcset*** al rescate



■ El atributo srcset al rescate

```
 <descriptor>">
```

- El atributo srcset permite indicarle al navegador los tamaños de las imágenes (no el tamaño que debe tomar la imagen).
- El navegador sólo descargará la versión correcta para el tamaño de la pantalla del dispositivo entre las opciones disponibles.
- Siempre debemos mantener el atributo **src** para garantizar la retro-compatibilidad con navegadores antiguos que no soporte **srcset**.



■ Descriptores de densidad de píxeles: x

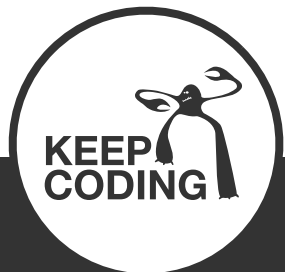
```

```

Carga logo-1x.png si la densidad de píxeles del dispositivo es 1x.

Carga logo-2x.png si la densidad de píxeles del dispositivo es 2x.

Carga logo-3x.png si la densidad de píxeles del dispositivo es 3x.



■ Descriptores de ancho de imagen: w

```

```

- El descriptor de ancho (w) indica al navegador el ancho que tiene la imagen.
- Con esta información, el navegador dividirá ese ancho entre el ancho del dispositivo para obtener la densidad de píxeles.
- Hecho esto, el navegador cargará la imagen que más se aproxime a la densidad de píxeles del dispositivo.



■ 320px de ancho y densidad de pixel 1

1. Leemos los anchos de imagen

```

```

2. Divide el ancho entre el ancho de pantalla

```

```

3. Buscamos el ratio que más se acerque a 1

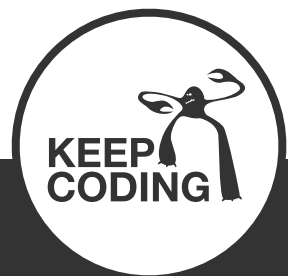
```

```

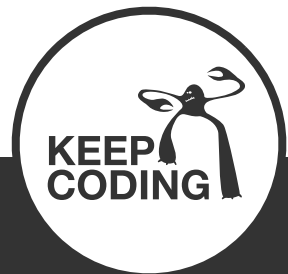
El navegador descargará
la imagen logo-200px.png



Pero a veces ***srcset*** no es suficiente



El descriptor de ancho calcula la densidad de pixels de una imagen en función del ancho del dispositivo.





Pero, ¿qué pasa si una imagen va a ocupar el 50% de la pantalla en desktop y el 100% en mobile?



Mejor usar **<picture>**



■ <picture> al rescate

```
<picture>
```

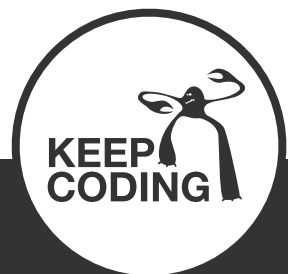
```
  <source srcset="baseball-xs.jpg 375w, baseball-sm.jpg 768w" media="(min-width: 768px)">
```

```
  <source srcset="baseball-md.jpg 1024w, baseball-@2x.jpg 1536w" media="(min-width: 992px)">
```

```
  
```

```
</picture>
```

- Con dentro de la etiqueta picture, podemos indicar varias fuentes y definir cual es la media query que se debe cumplir para utilizar dicha fuente.





GRACIAS
www.keepcoding.io

