

Left

styles.css

styles.min.css

FastGulp

FastGulp

dist

assets

CSS

styles.css

styles.min.css

node\_modules

src

CSS

bootstrap.css

index.css

fonts

Right

styles.css

styles.min.css

```

6639 }
6640 }
6641 @media (min-width: 1200px) {
6642   .visible-lg-block {
6643     display: block !important;
6644   }
6645 }
6646 @media (min-width: 1200px) {
6647   .visible-lg-inline {
6648     display: inline !important;
6649   }
6650 }
6651 @media (min-width: 1200px) {
6652   .visible-lg-inline-block {
6653     display: inline-block !important;
6654   }
6655 }
6656 @media (max-width: 767px) {
6657   .hidden-xs {
6658     display: none !important;
6659   }
6660 }
6661 @media (min-width: 768px) and (max-width: 991px) {
6662   .hidden-sm {
6663     display: none !important;
6664   }
6665 }
6666 @media (min-width: 992px) and (max-width: 1199px) {

```

```

1  /*!
2   * Bootstrap v3.3.5 (http://getbootstrap.com)
3   * Copyright 2011-2015 Twitter, Inc.
4   * Licensed under MIT
5   (https://github.com/twbs/bootstrap/blob/master/LICENSE)
6   */
7  /*! normalize.css v3.0.3 | MIT License |
8   github.com/necolas/normalize.css */
9  .label,sub,sup{vertical-align:baseline}.btn,.btn-group,.btn-group-vertical,.caret,.checkbox-inline,.radio-inline,img{vertical-align:middle}hr,img{border:0}body,figure{margin:0}.navbar-fixed-bottom.navbar-collapse,.navbar-fixed-top.navbar-collapse,.pre-scrollable{max-height:340px}.btn-group>.btn-group,.btn-toolbar.btn,.btn-toolbar .btn-group,.btn-toolbar .input-group,.col-xs-1,.col-xs-10,.col-xs-11,.col-xs-12,.col-xs-2,.col-xs-3,.col-xs-4,.col-xs-5,.col-xs-6,.col-xs-7,.col-xs-8,.col-xs-9,.dropdown-menu{float:left}.remarked,html{webkit-text-size-adjust:100%}html{font-family:sans-serif;-ms-text-size-adjust:100%}article,aside,details,figcaption,figure,footer,header,hgroup,main,menu,nav,section,summary{display:block}audio,canvas,progress,video{display:inline-block;vertical-align:baseline}audio:not([controls]){display:none;height:0}[hidden],template{display:none}a{background-color:transparent}a:active,a:hover{outline:0}b,optgroup,strong{font-weight:700}dfn{font-style:italic}h1{margin:.67em 0}mark{color:#000;background:#ff0}sub,sup{position:relative;font-size:75%;line-height:0}sup{top:-.5em}sub{bottom:-.25em}svg:not(:root){overflow:hidden}hr{height:0;box-sizing:content-

```

# CSS: Disposición



Cuando hacemos una web?

universal

HOME ▾

FEATURES ▾

PORTFOLIO ▾

ALL PAGES ▾

CONTACT ▾

## 46 HTML PAGES FULL OF FEATURES

ARSHA

[Home](#)

[About](#)

[Services](#)

[Portfolio](#)

[Team](#)

[Drop Down ▾](#)

[Contact](#)

[Get Started](#)

## Better Solutions For Your Business

We are team of talented designers making  
websites with Bootstrap

[Get Started](#)

[Watch Video](#)



myob

BELIMO

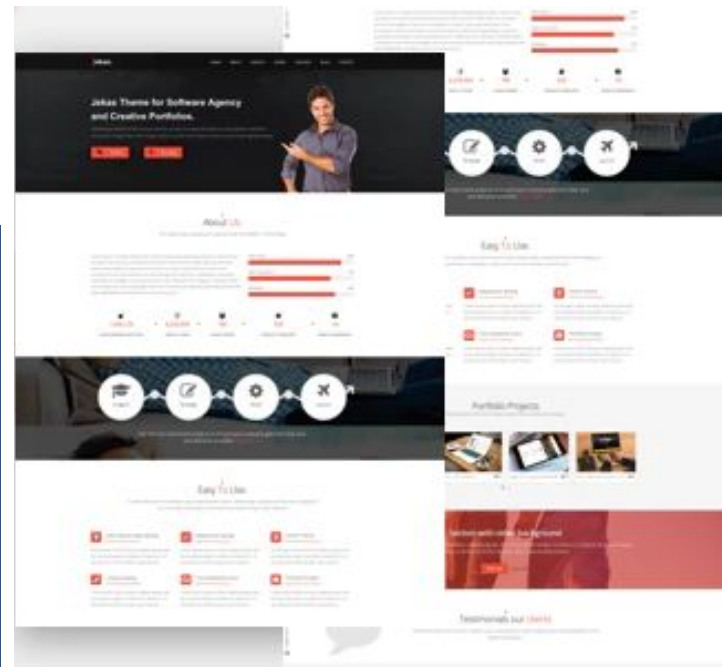
LifeGroups

Lilly

citrus

Trustly

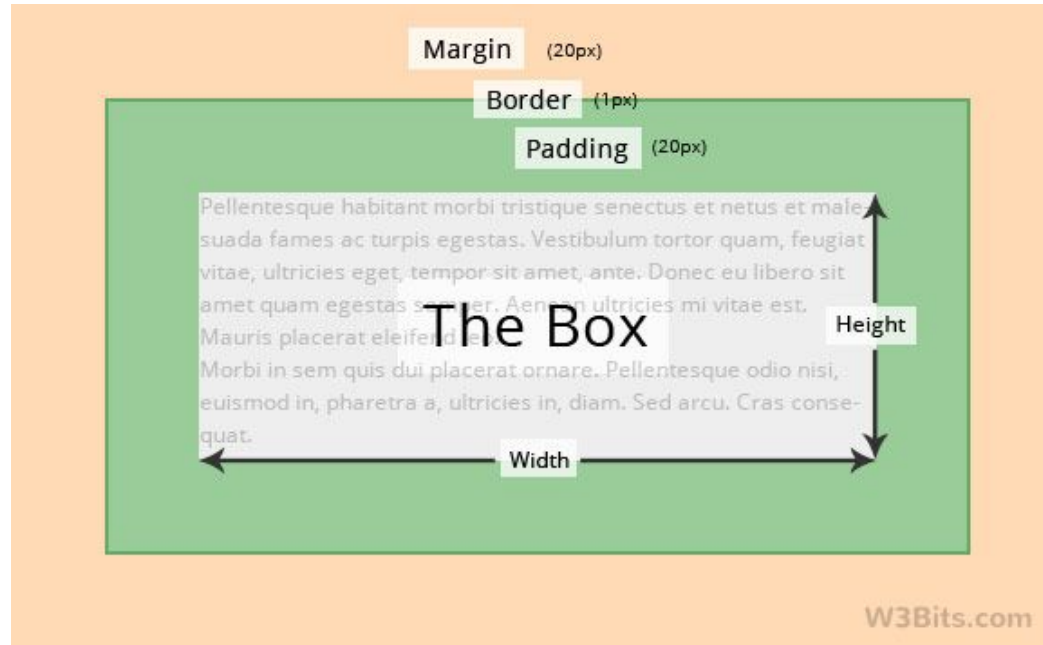
ABOUT US



Y para eso tenemos que entender unos conceptos

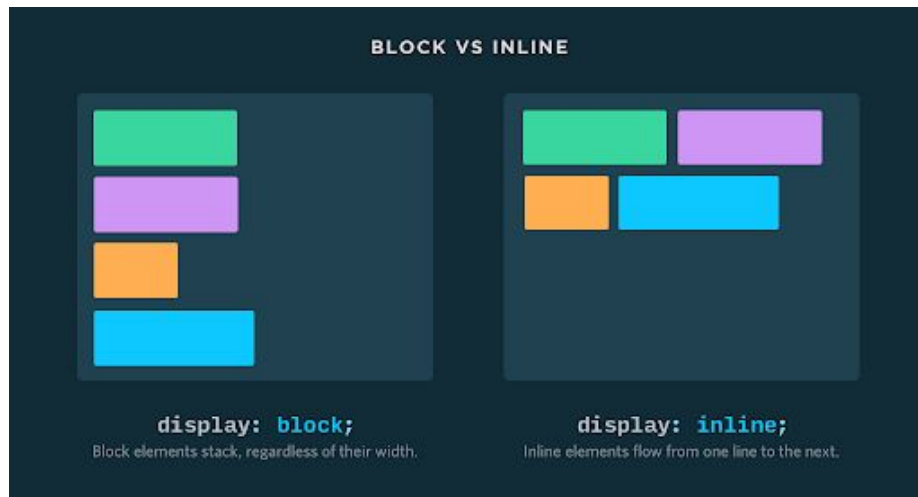
# Modelo de caja

...



# Display: Block, Inline

En HTML vamos a tener elementos que se van a presentar en forma de bloque (la mayoría, header, nav, div, p) y otros pocos elementos que se presentarán en la misma línea (span, label, input).



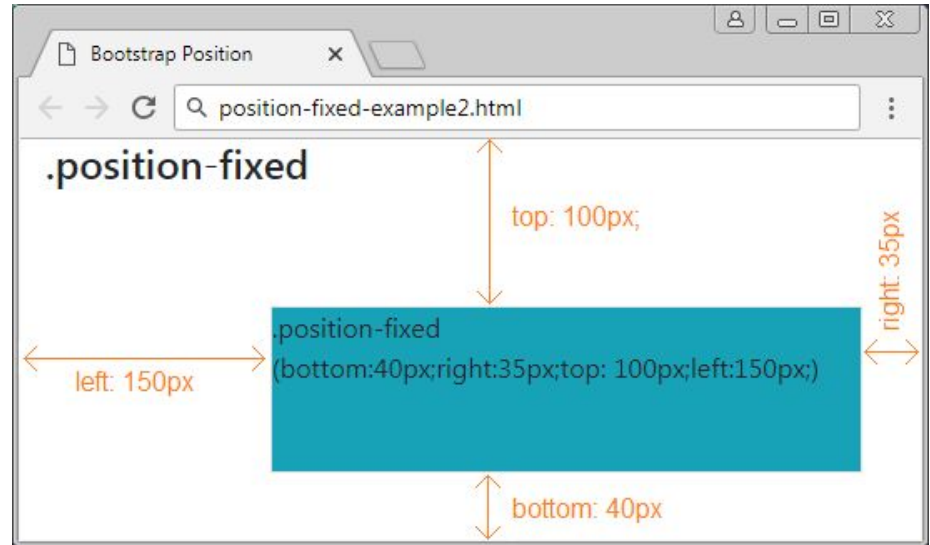
# position: relative, absolute, fixed

Nos va a permitir poner contenido “flotante” en unos de los lados.

position: relative



position: absolute





# position: relative, absolute, fixed

- The glass
- Ice cream scoops
- Whipped cream
- Cherries

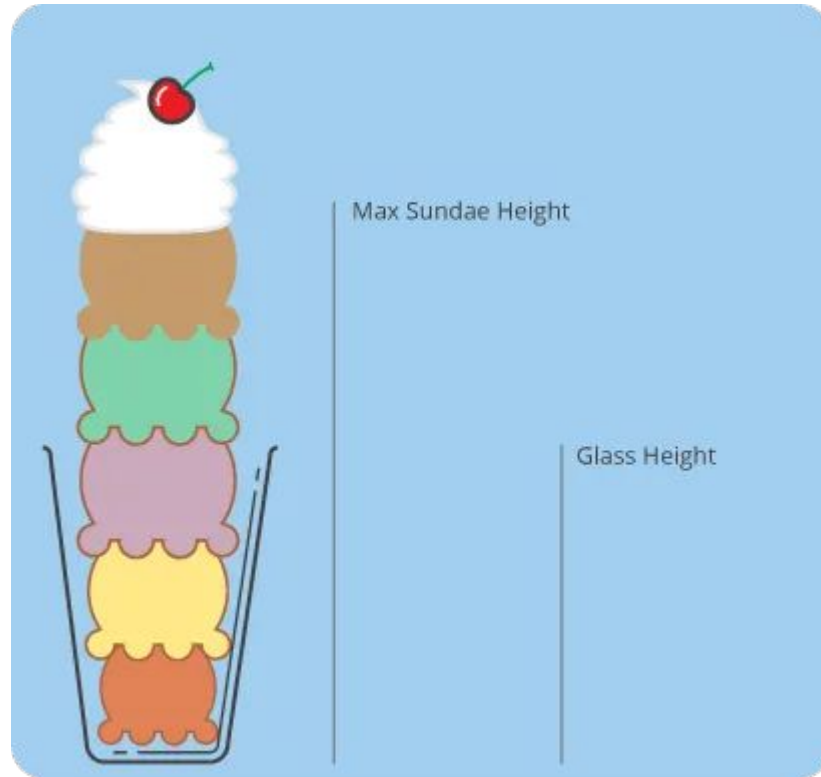
```
1 <body>
2   <div class='fullSundae'>
3     <div class='cherry'></div>
4     <div class='whippedCream'></div>
5     <div class='iceCreamScoop'></div>
6     <div class='iceCreamScoop'></div>
7     <div class='glass'>
8       <div class='iceCreamScoop'></div>
9       <div class='iceCreamScoop'></div>
10      <div class='iceCreamScoop'></div>
11    </div>
12  </div>
13 </body>
```



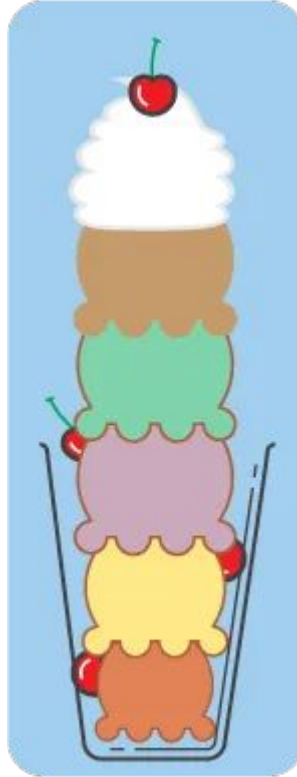
position: fixed - crema blanca



## position: relative - VASO



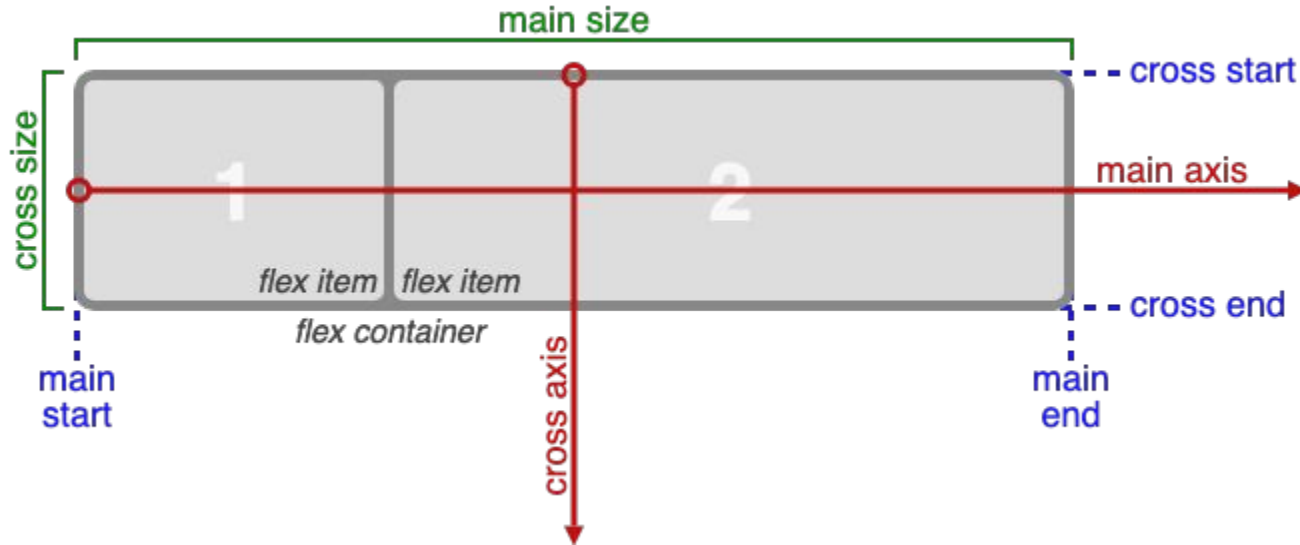
position: absolute cerezas



Y estas herramientas están diseñadas para disponer elementos en webs/aplicaciones?

# FLEXBOX

Para entender flexbox imaginemos que un contenedor (un div por ejemplo), tiene las siguientes propiedades:



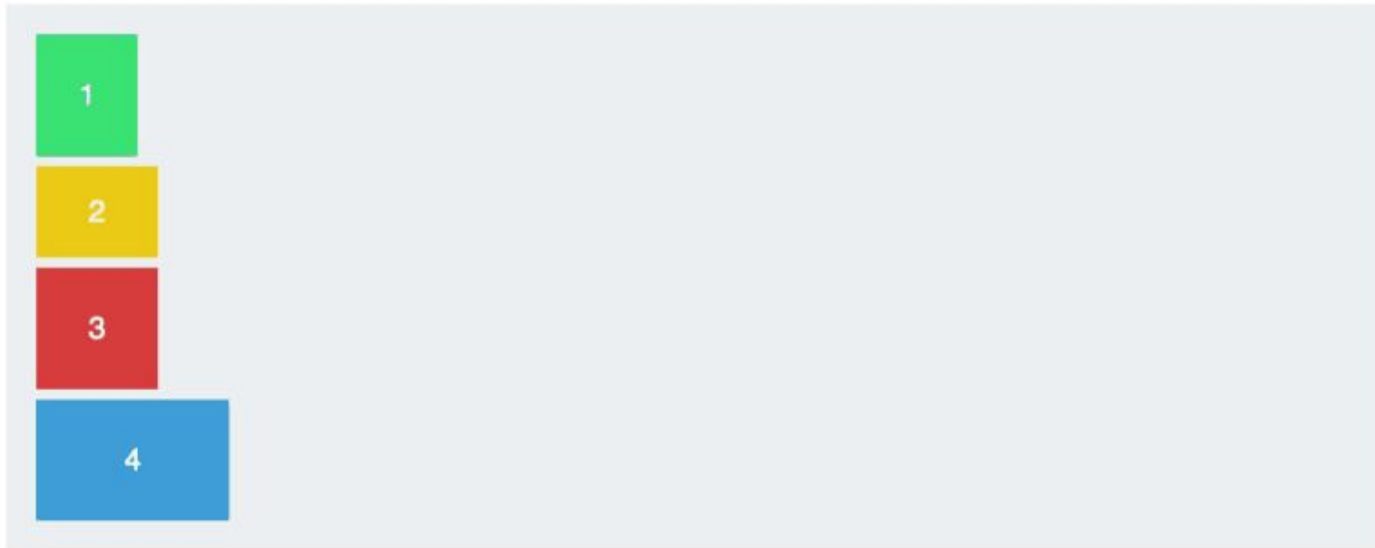
# display:flex;

A comparación de las herramientas anteriores donde escogemos un elemento y lo posicionamos, en el caso de flex vamos a utilizar **el PADRE** de esos elementos, para así cambiar el comportamiento de los hijos.

12

**display: block;**

13



# dirección

Podemos jugar con la dirección de las “propiedades” que vimos en flex;

**flex-direction: row;**





# Invirtiendo el orden

podemos jugar con la dirección de las “propiedades” que vimos en flex, invirtiendo su orden con `flex-direction` ,

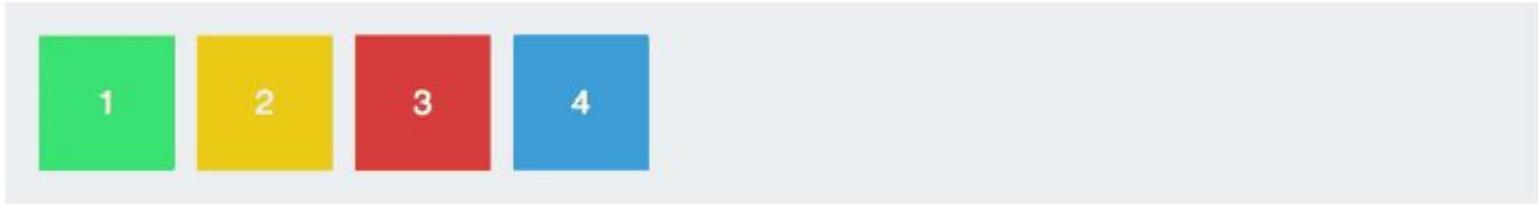
`row-reverse`, `column-reverse`



# Alineación en flex

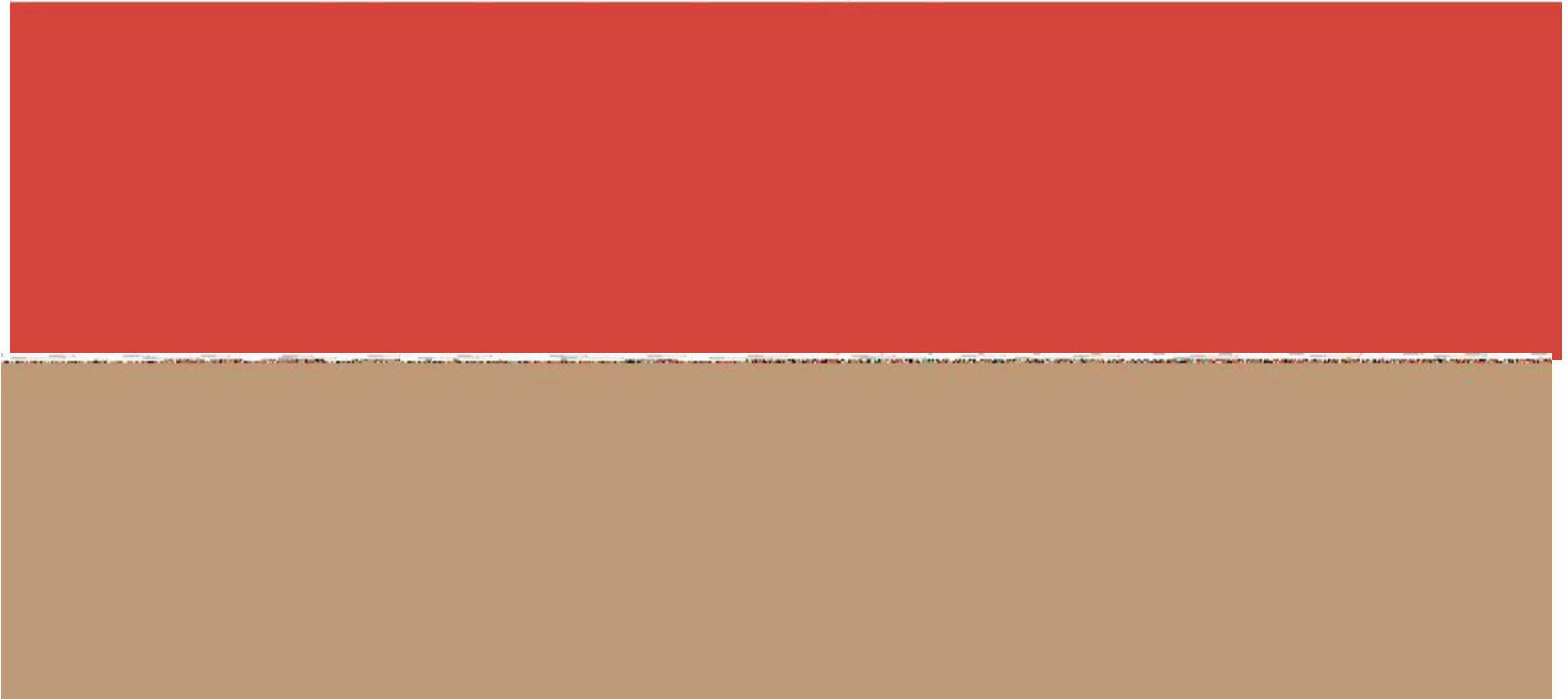
al utilizar justify content, podemos indicar como se pueden disponer estos elementos en flex. De forma horizontal

**justify-content: flex-start;**



# Tamaño de los elementos en flex.

Podemos controlar el tamaño de estos elementos con grow (crecer) y shrink (encojer).



# Contenido Extra:

Guia completa Flexbox

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Aprende flexbox ayudando a Froggy

<https://flexboxfroggy.com/#es>

