

BOOTSTRAP & LAYOUTS

DESCRIPTION & OBJECTIVES

- Learn Bootstrap basics
- Build **card grids** with Bootstrap
- Build **column layouts** with Flexbox

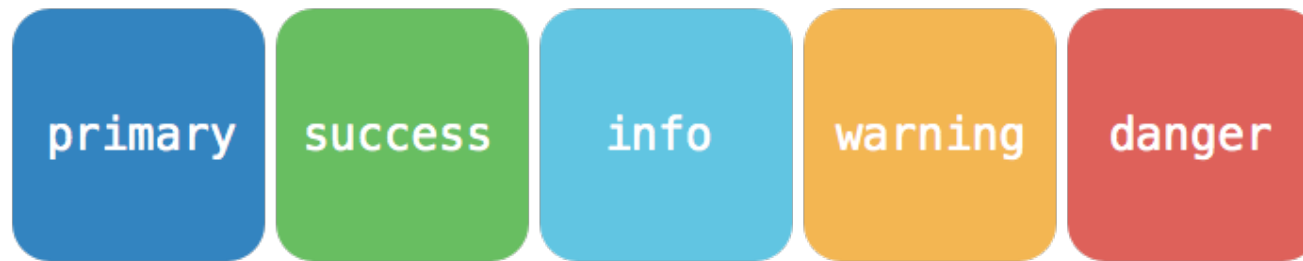
WHEN DO WE NEED BOOTSTRAP ?



- Simple components like buttons and forms
- Javascript components like dropdowns or modals
 - Responsive card grids
- Advanced components (cards, tabs, etc...)
 - => use your own design
- Column or grid layouts => use flexbox or CSS grid

WHAT ARE USEFUL BOOTSTRAP CLASSES ?

Semantic scheme

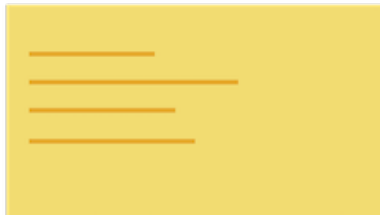


WHAT ARE USEFUL BOOTSTRAP CLASSES ?

Alignment



```
<div class='text-justify'></div>
```



```
<div class='text-left'></div>
```



```
<div class='text-center'></div>
```



```
<div class='text-right'></div>
```

WHAT ARE USEFUL BOOTSTRAP CLASSES ?

List inline

`ul.list-inline`

li

li

li

li



WHAT ARE USEFUL BOOTSTRAP CLASSES ?

Buttons



```
<a class='btn btn-primary' href='#'>Go</a>
```



```
<a class='btn btn-warning' href='#'>Go</a>
```



```
<a class='btn btn-danger' href='#'>Go</a>
```



```
<a class='btn btn-info' href='#'>Go</a>
```

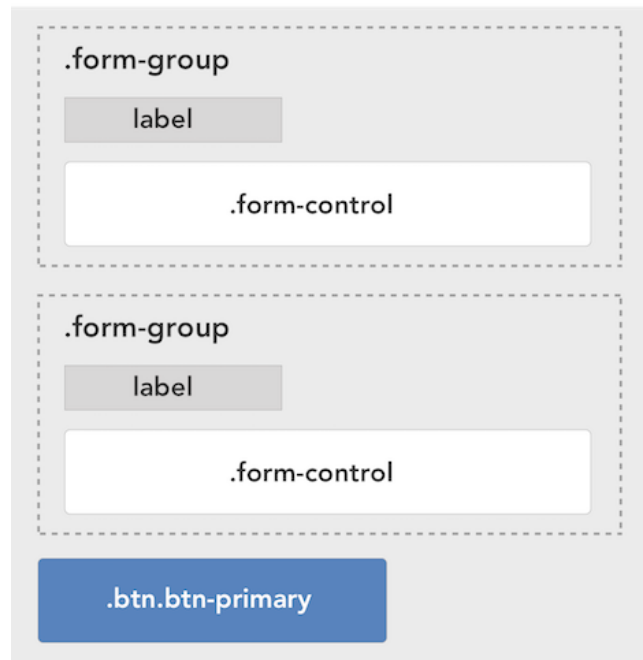


```
<a class='btn btn-success' href='#'>Go</a>
```

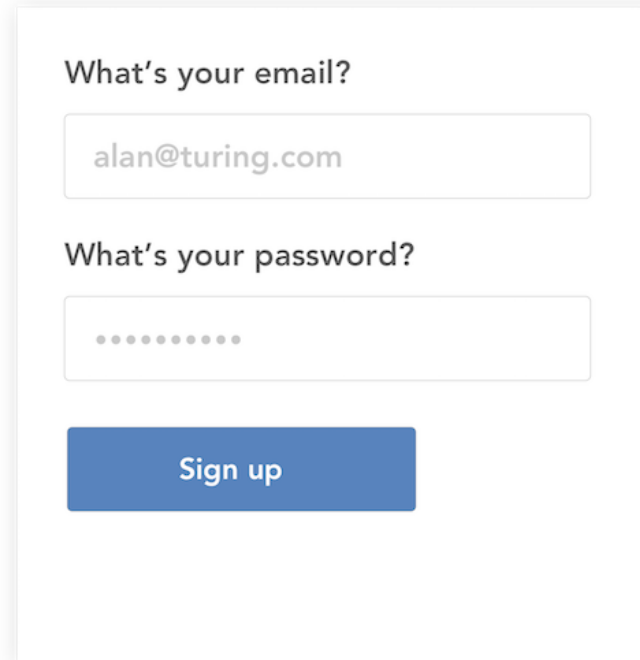
WHAT ARE USEFUL BOOTSTRAP CLASSES ?

Form with labels

form



The diagram illustrates the structure of Bootstrap form classes. It shows two identical form groups, each enclosed in a dashed box and labeled `.form-group`. Inside each group, there is a label box labeled `label` and a text input field labeled `.form-control`. Below these two groups is a blue button labeled `.btn.btn-primary`.

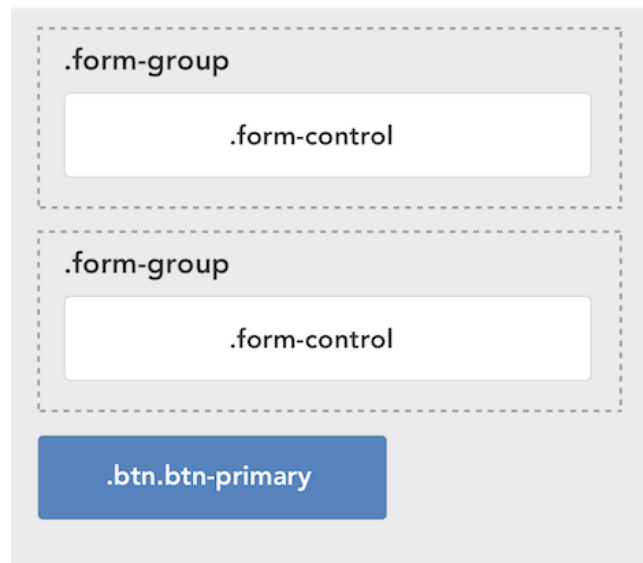


The visual representation shows a form with two input fields and a button. The first field is labeled "What's your email?" and contains the text "alan@turing.com". The second field is labeled "What's your password?" and contains a series of dots. Below the fields is a blue button labeled "Sign up".

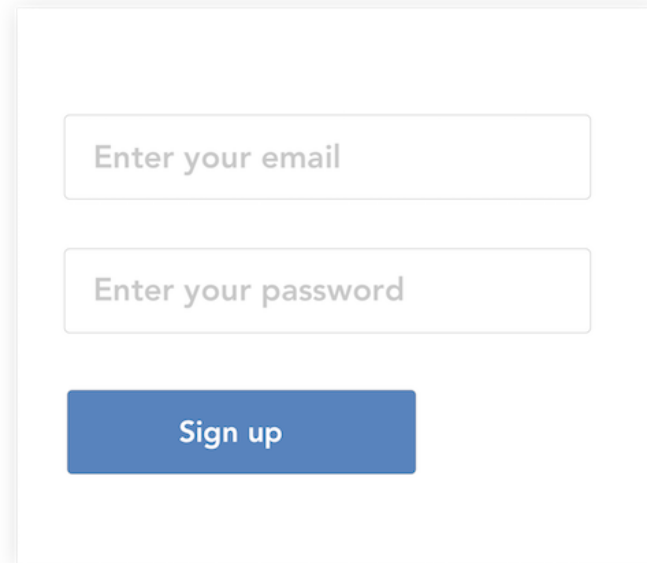
WHAT ARE USEFUL BOOTSTRAP CLASSES ?

Form without labels

form



A diagram illustrating the structure of a Bootstrap form without labels. It shows a light gray container with two dashed boxes, each labeled `.form-group`. Inside each `.form-group` is a white input field labeled `.form-control`. Below the second `.form-group` is a blue button labeled `.btn.btn-primary`.

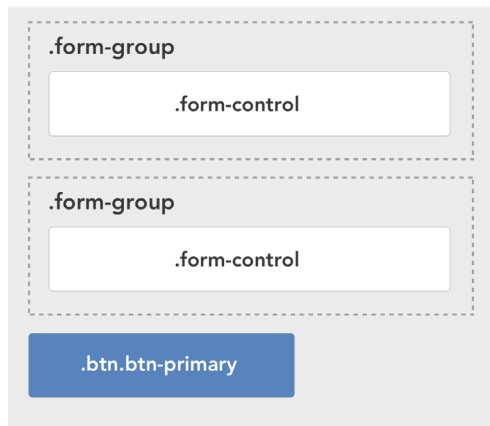


A visual representation of a form without labels. It features two white input fields with placeholder text "Enter your email" and "Enter your password". Below these fields is a blue button with the text "Sign up".

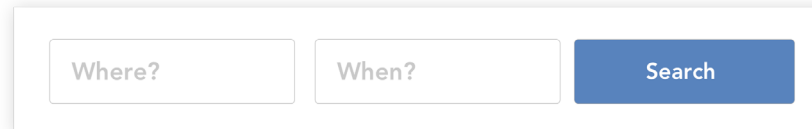
WHAT ARE USEFUL BOOTSTRAP CLASSES ?

Form inline

form.form-inline



A diagram illustrating the structure of a Bootstrap form using the `form.form-inline` class. It shows a light gray container with a dashed border. Inside, there are two `.form-group` elements, each containing a `.form-control` input field. Below these, there is a blue button with the text `.btn.btn-primary`.

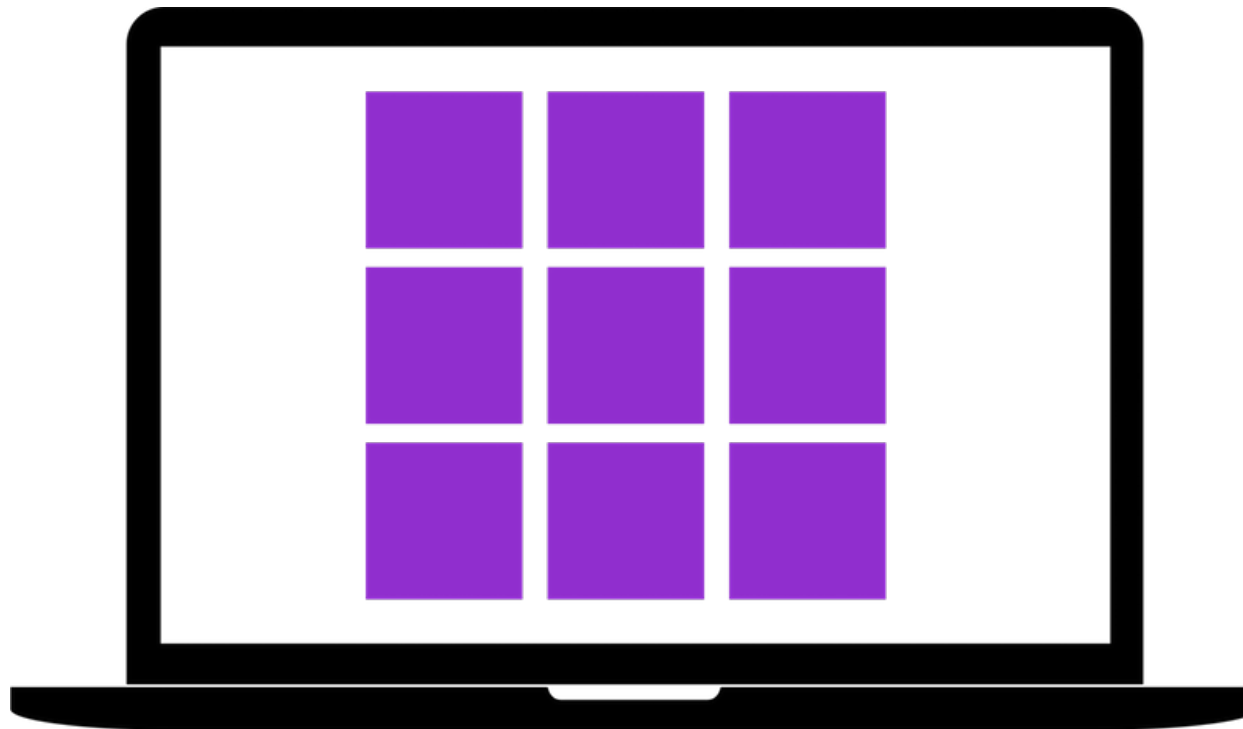


A visual representation of a Bootstrap form using the `form.form-inline` class. It shows a white container with a light gray border. Inside, there are two input fields labeled "Where?" and "When?", followed by a blue button labeled "Search".

BOOTSTRAP GRID

Responsive **card grids** :

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device size increases



BOOTSTRAP GRID

Responsive



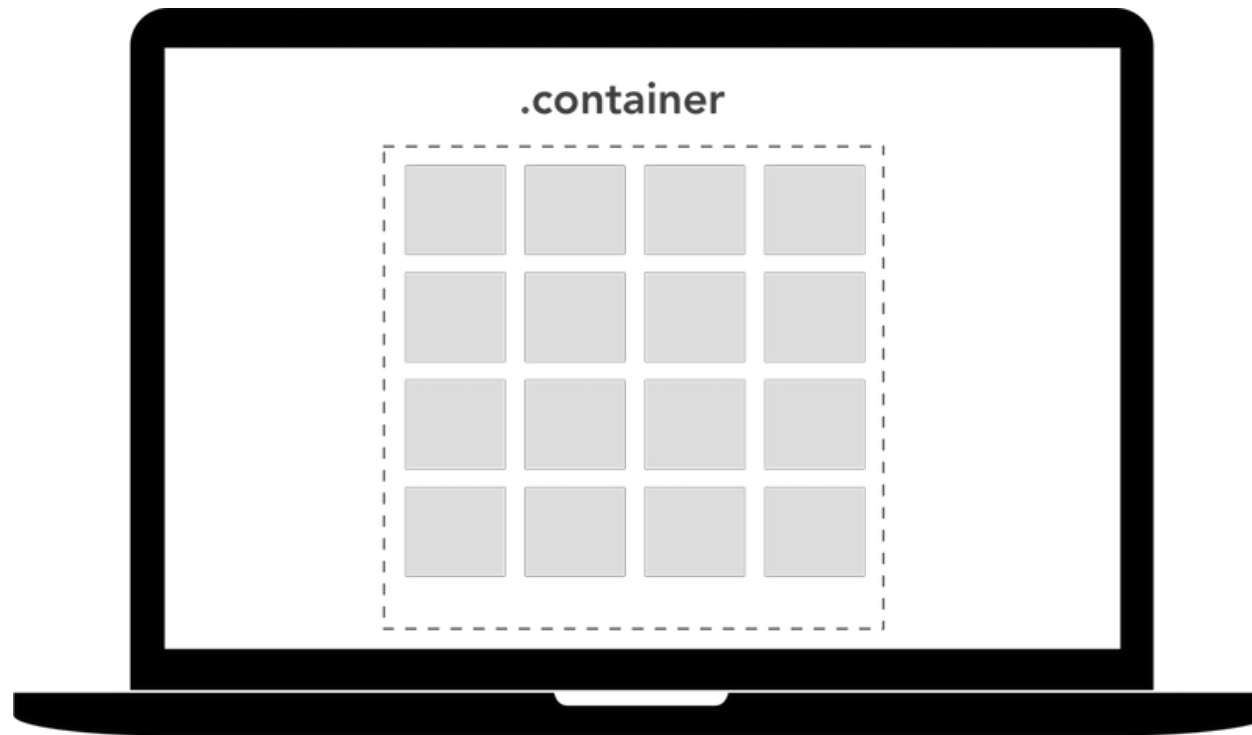
BOOTSTRAP GRID

Mobile-first



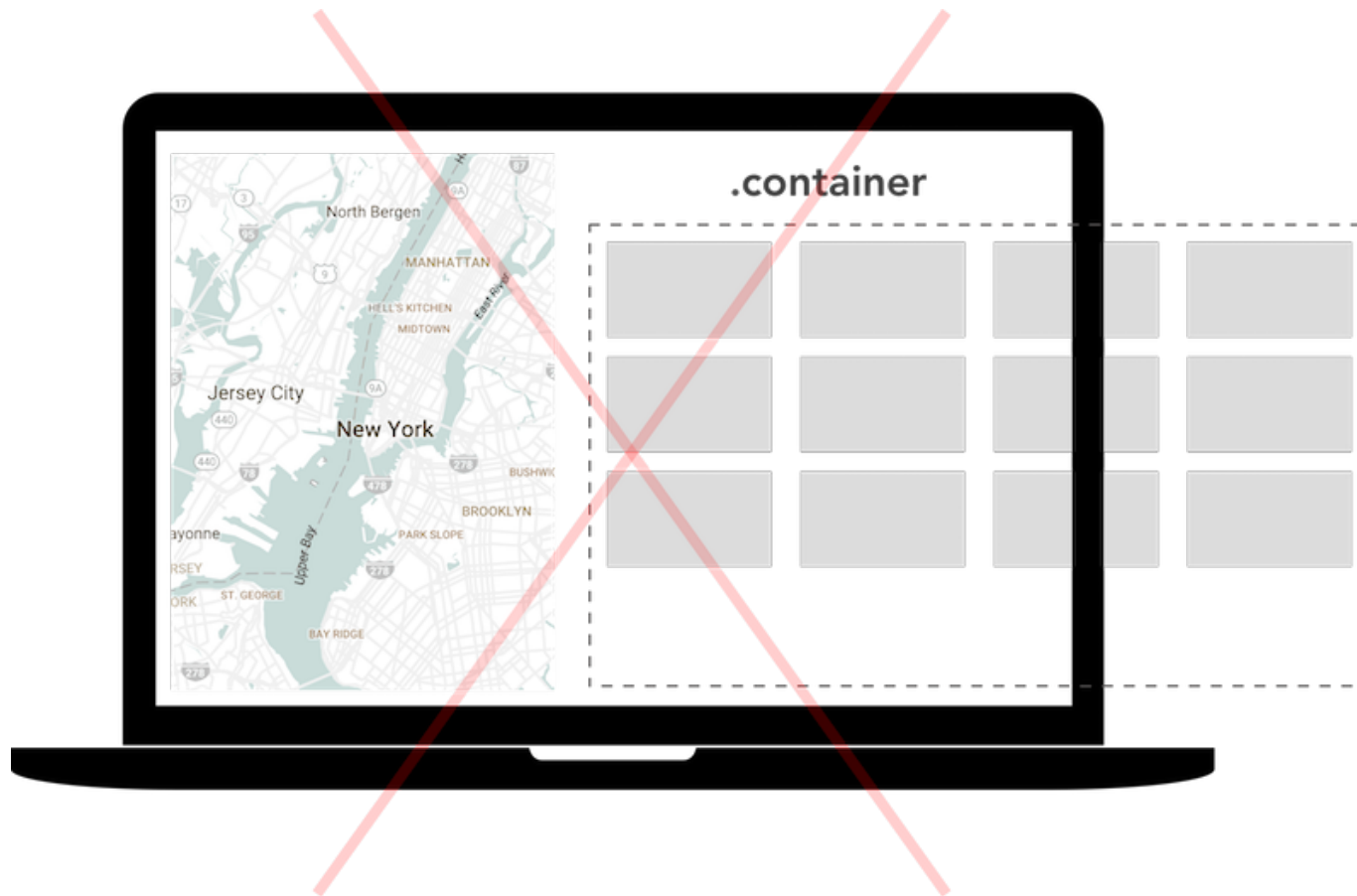
FULL-PAGE GRID

Start with a container as usual

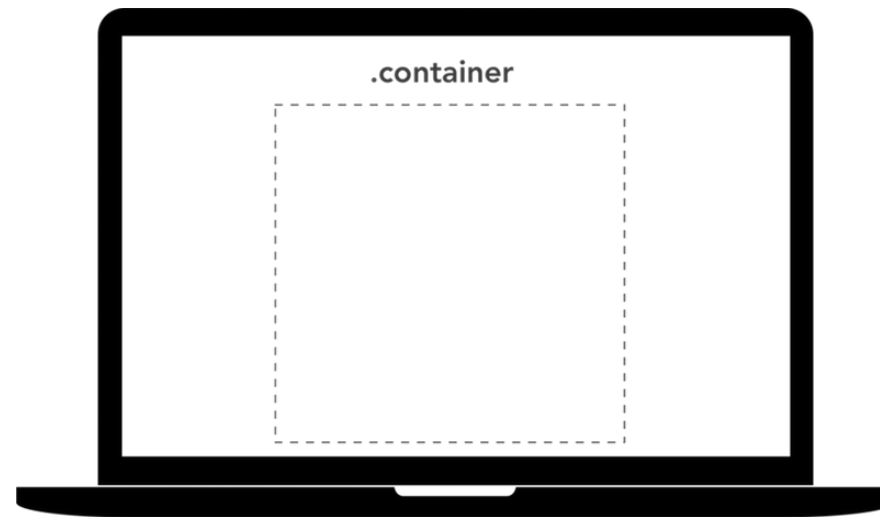


INNER-SECTION GRID

Container's width is set in pixels,
so don't use container

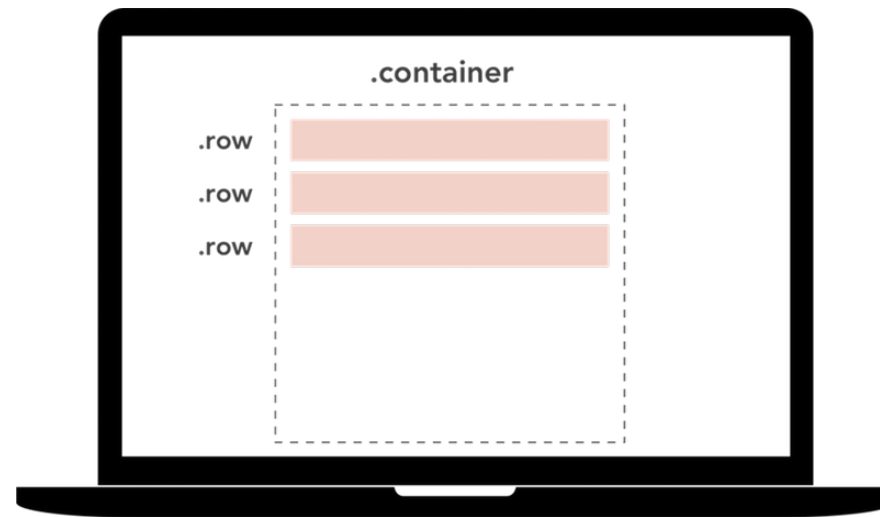


BUILD A FULL-PAGE GRID



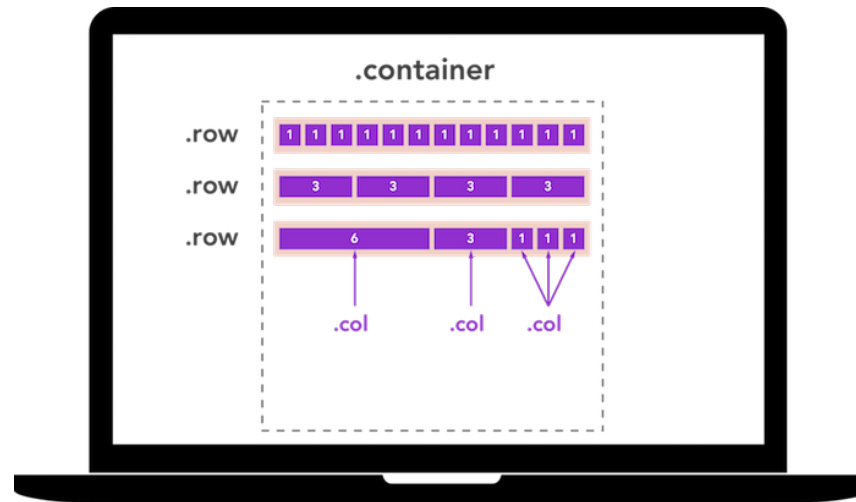
```
<div class="container">  
  <!-- Inject the grid here -->  
</div>
```


BUILD A FULL-PAGE GRID



```
<div class="container">  
  <div class="row"></div>  
  <div class="row"></div>  
  <div class="row"></div>  
</div>
```

BUILD A FULL-PAGE GRID



```
<div class="container">
  <div class="row">
    <!-- 12 cols of 1 column -->
  </div>
  <div class="row">
    <!-- 4 cols of 3 columns -->
  </div>
  <div class="row">
    <!-- 5 cols of various widths -->
  </div>
</div>
```

WHAT'S IN THE ROW?

```
<div class="col-xs-6">  
</div>
```

- **col**: because it fits a number of columns
 - **xs**: screen width
- **6**: number of columns taken by the block (max 12)

DEVICES

- **xs**: Extra small devices (Phones < 768px)
 - **sm**: Small devices (Tablets > 768px)
 - **md**: Medium devices (> 992px)
 - **lg**: Large devices (Desktops > 1200px)

QUIZZ

What **.col-xs-6** and **.col-md-3** mean
with simple words?

EXAMPLE (1 ROW, 2 COLUMNS)



```
<div class="container">
  <div class="row">

    <div class="col-xs-6">
    </div>

    <div class="col-xs-6">
    </div>

  </div>
</div>
```

MOBILE FIRST

If you just give the .col-xs classes it will apply
on **all larger screens**

NOW WHAT HAPPENS?

```
<div class="container">
  <div class="row">

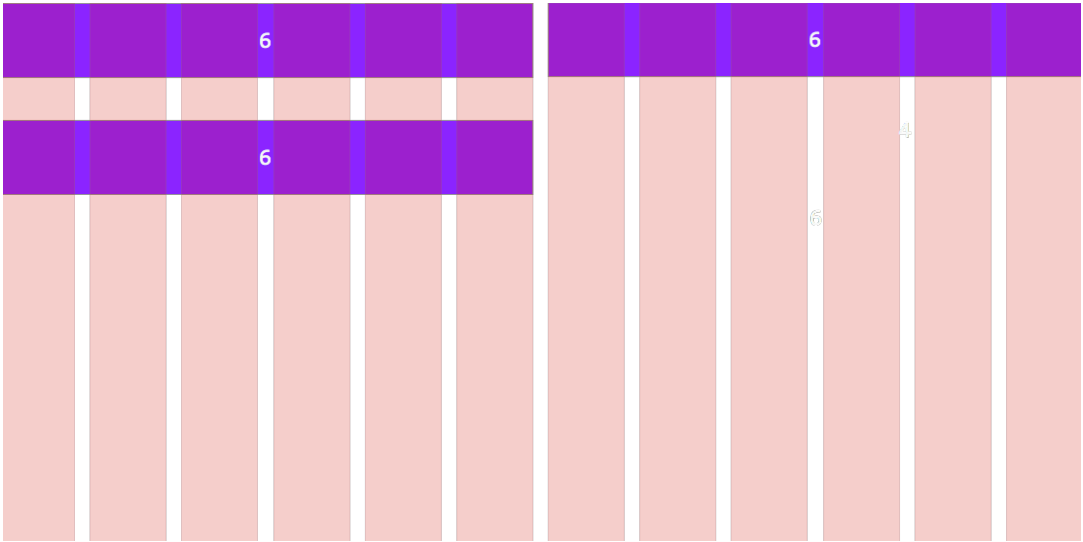
    <div class="col-xs-6">
    </div>

    <div class="col-xs-6">
    </div>

    <div class="col-xs-6">
    </div>

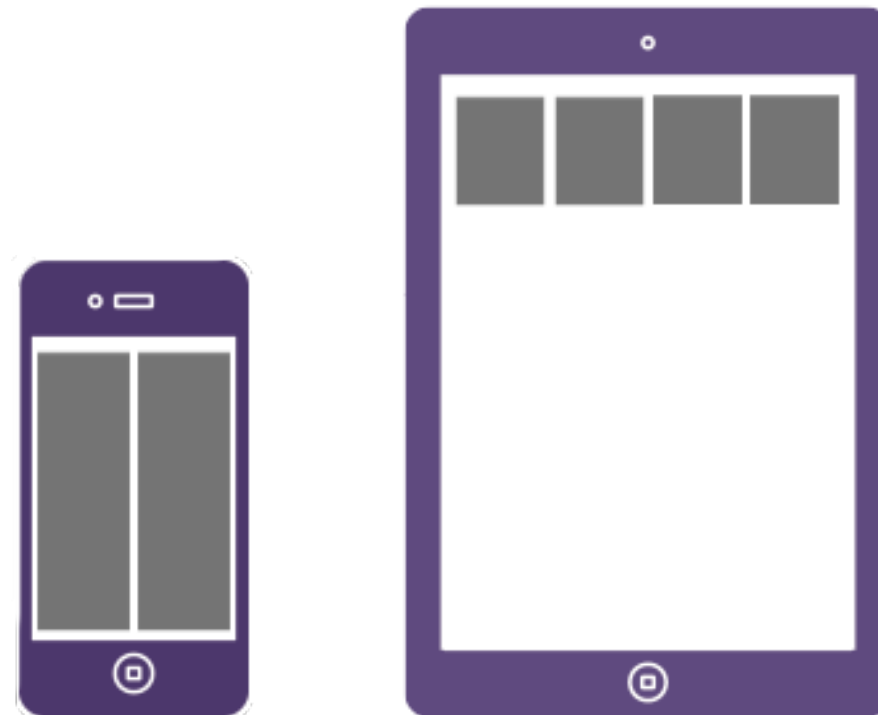
  </div>
</div>
```


JUST RETURN ON NEW LINE



RESPONSIVE EXAMPLE

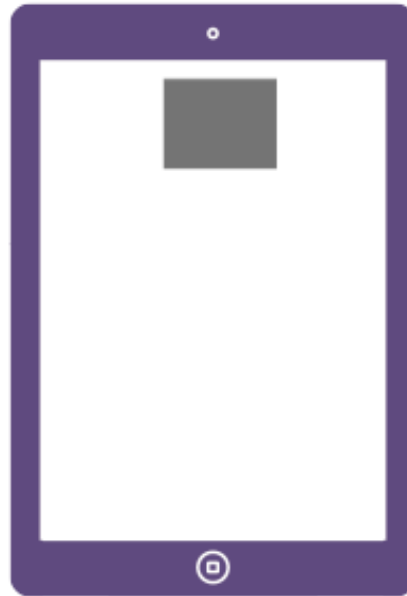
We want this **responsive** behavior



HOW WILL YOU DO ?

```
<div class="container">
  <div class="row">
    <div class="col-xs-6 col-sm-3"></div>
    <div class="col-xs-6 col-sm-3"></div>
    <div class="col-xs-6 col-sm-3"></div>
    <div class="col-xs-6 col-sm-3"></div>
  </div>
</div>
```

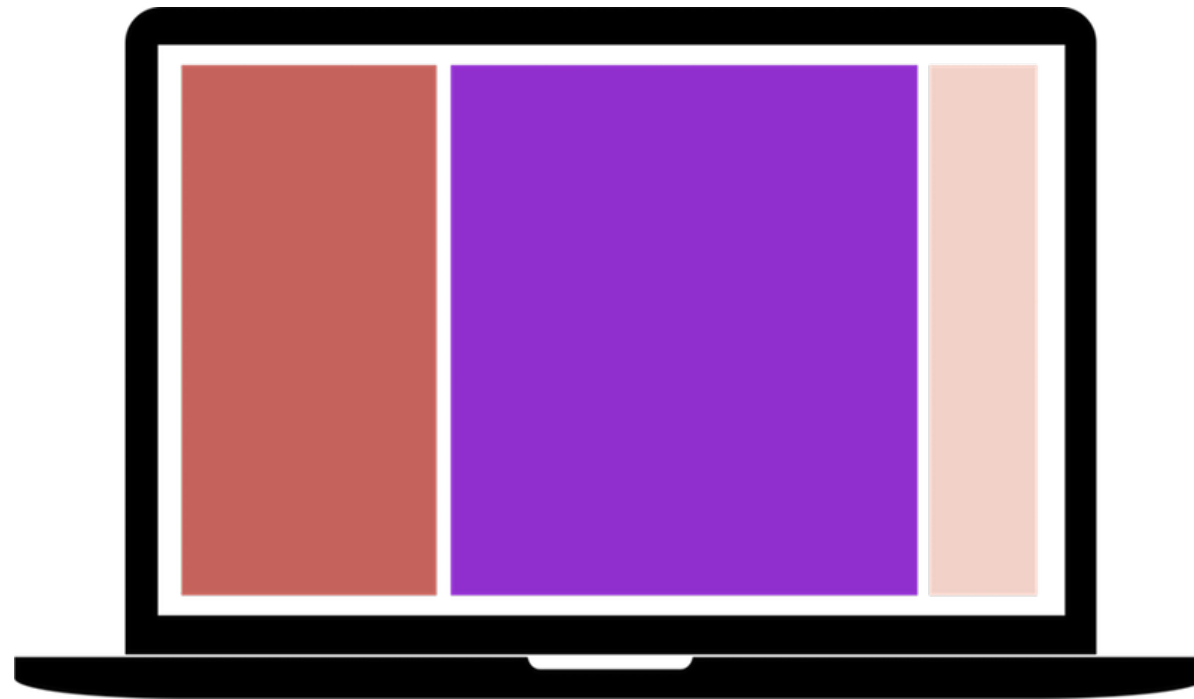
OFFSETTING BLOCKS



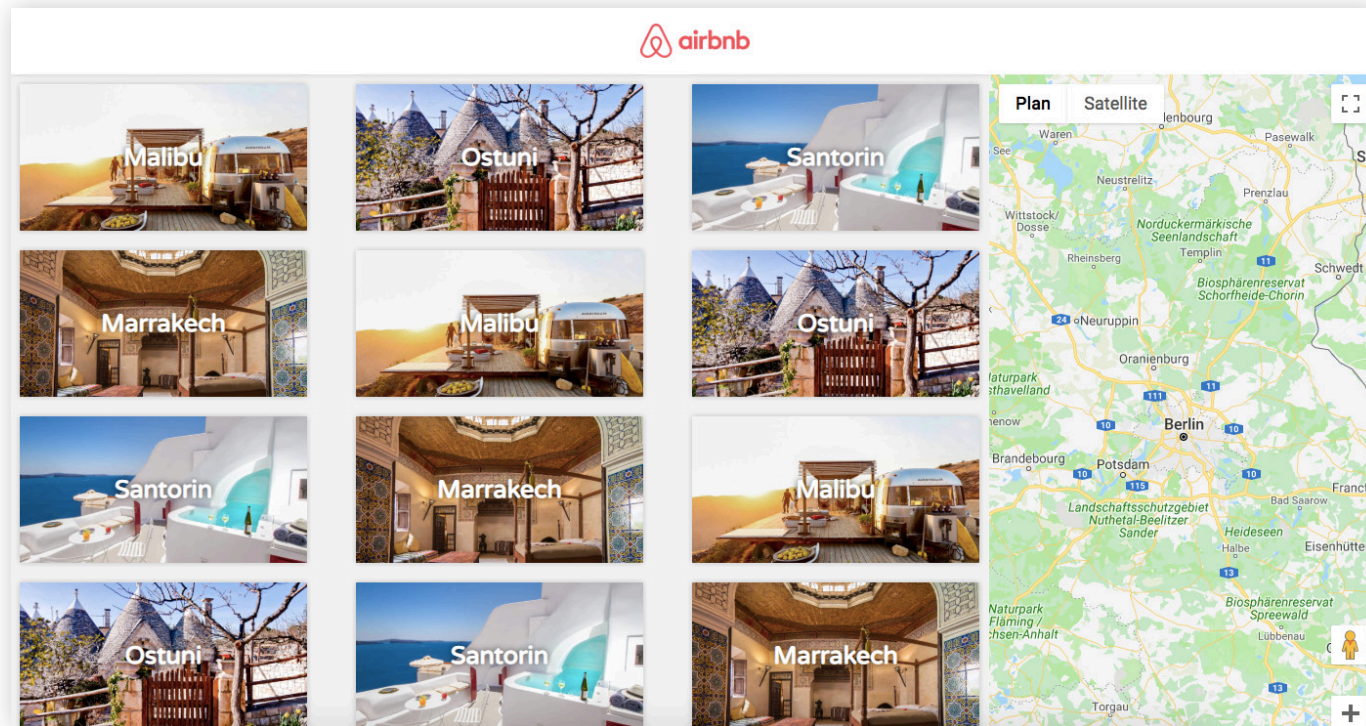
```
<div class="container">
  <div class="row">
    <div class="col-sm-offset-4 col-sm-4">
      <!-- This div is centered and takes 4 columns -->
    </div>
  </div>
</div>
```

COLUMN LAYOUTS

Split your page in columns with flexbox
(e.g. a sidebar, a map)



LET'S CODE THIS LAYOUT



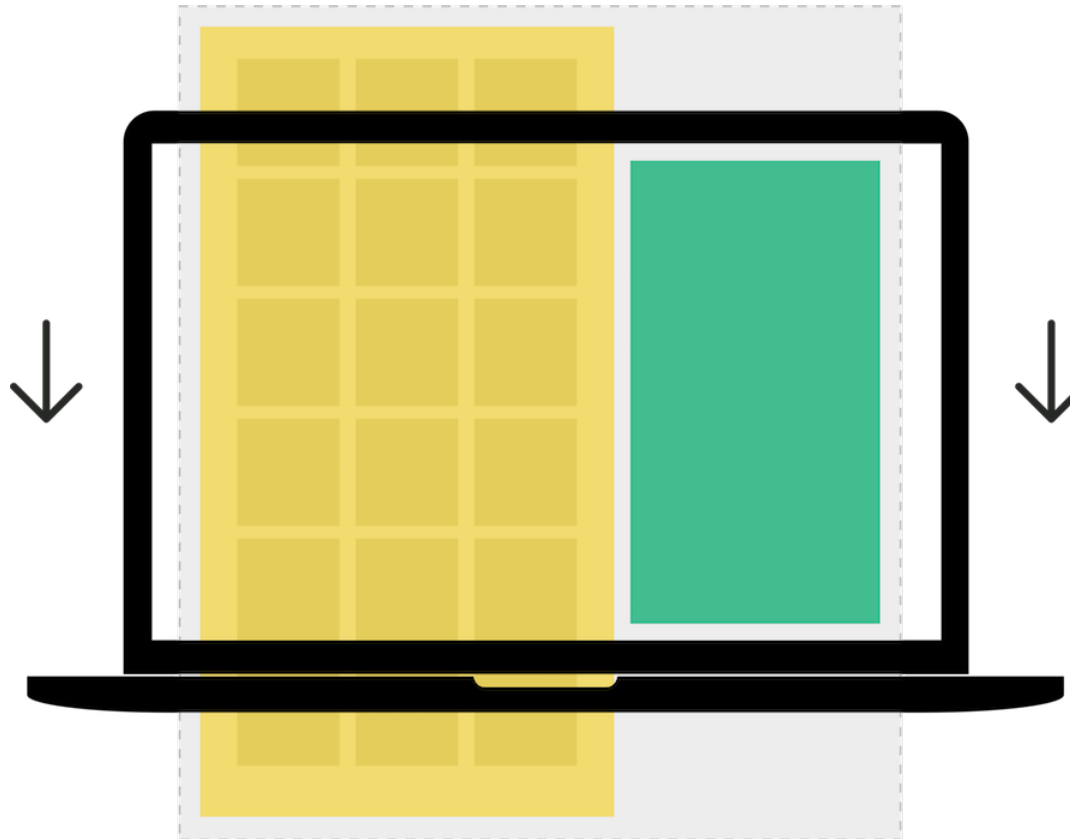
NEW CSS ORGANIZATION

```
.
├── css
│   ├── components
│   │   ├── avatar.css
│   │   └── card.css
│   ├── layouts
│   │   ├── profile.css
│   │   └── map.css
│   └── style.css
└── index.html
```

COLUMN LAYOUTS

```
.main-container {  
  display: flex;  
}  
#map {  
  flex: 0 0 450px;  
  height: 100vh;  
  position: sticky !important; /* !important to override Gmaps inline CSS */  
  top: 0 !important;  
}
```


COLUMN LAYOUTS



Let's code this layout