



# WHY BOOTSTRAP?

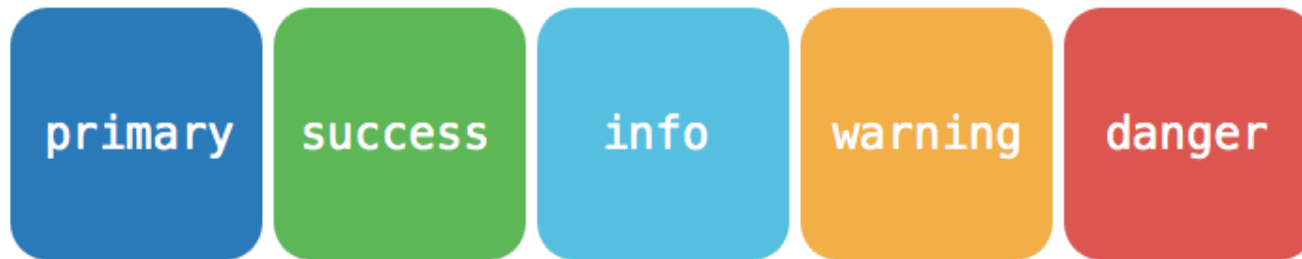
- You always need the same UI components
- You always need a responsive grid system

# **EVERYTHING STARTS WITH A NICE BOILERPLATE**

<https://github.com/lewagon/bootstrap-boilerplate>

# BASIC CSS

# BASIC CSS - SEMANTIC SCHEME



# BASIC CSS - ALIGNMENT



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



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



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



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

# **BASIC CSS - LIST INLINE**





# BASIC CSS - IMAGES




140x140

```
<img class='img-rounded' src='ex.png'>
```



140x140

```
<img class='img-circle' src='ex.png'>
```



140x140

```
<img class='img-thumbnail' src='ex.png'>
```

# BASIC CSS - BUTTONS

Go

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

Go

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

Go

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

Go

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

Go

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

# BASIC CSS - FORMS

```
<form action="/">  
  <input type='email' class='form-control' placeholder='alan@turing.org'>  
  <input type='password' class='form-control' placeholder='*****'>  
  <input type='submit' value='Apply' class='btn btn-primary'>  
</form>
```

# BASIC CSS - FORMS WITH LABEL

Your email

Your password

```
<form action='/>  
  <div class='form-group'>  
    <label for='email'>Your email</label>  
    <input id='email' type='email' class='form-control' placeholder='alan@turing.org'>  
  </div>  
  <div class='form-group'>  
    <label for='password'>Your password</label>  
    <input id='password' type='password' class='form-control' placeholder='*****'>  
  </div>  
  <input type='submit' value='Apply' class='btn btn-primary'>  
</form>
```

# BASIC CSS - FORMS INLINE

```
<form action="/" class='form-inline'>
  <input type='email' class='form-control' placeholder='alan@turing.org'>
  <input type='password' class='form-control' placeholder='*****'>
  <input type='submit' value='Apply' class='btn btn-primary'>
</form>
```

# DOCUMENTATION

<http://getbootstrap.com/css/>

# COMPONENTS

# COMPONENTS - LABELS

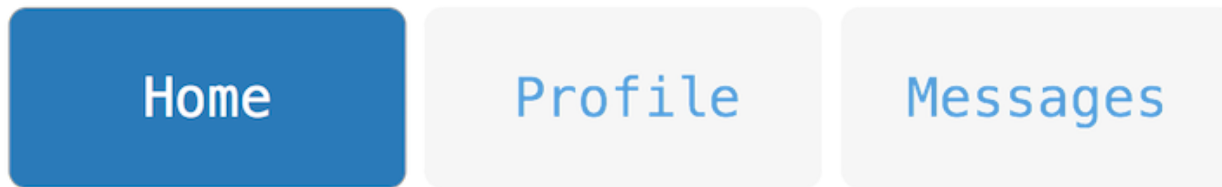
Challenge **hard** `<h2>Challenge <span class='label label-important'>hard</span></h2>`

Challenge **opt.** `<h2>Challenge <span class='label label-warning'>opt.</span></h2>`

Challenge **todo** `<h2>Challenge <span class='label label-success'>todo</span></h2>`



# COMPONENTS - NAV



```
<ul class='nav nav-pills'>  
  <li class='active'><a href='/home'>Home</a></li>  
  <li><a href='/profile'>Profile</a></li>  
  <li><a href='/messages'>Messages</a></li>  
</ul>
```

# COMPONENTS - ALERTS

You password is not secure enough

x

```
<div class='alert alert-danger alert-dismissible' role="alert">  
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  
    <span aria-hidden="true">&times;</span>  
  </button>  
  You password <strong>is not secure enough</strong>  
</div>
```

You are successfully logged in

x

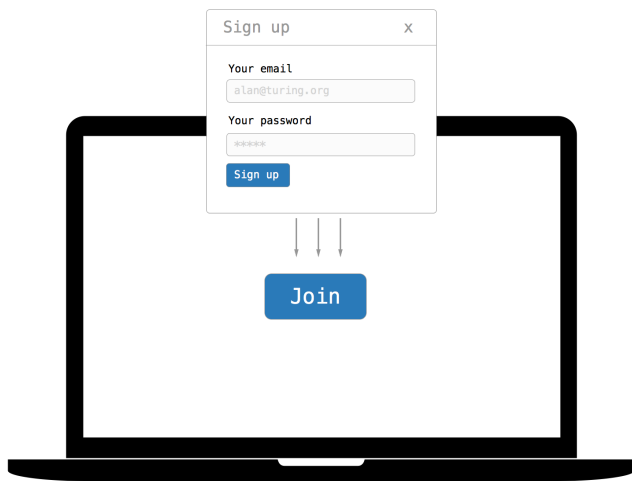
```
<div class='alert alert-success alert-dismissible' role="alert">  
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">  
    <span aria-hidden="true">&times;</span>  
  </button>  
  You are <strong>successfully</strong> logged in  
</div>
```

# COMPONENTS - NAVBAR

Bootstrap navbar is too complicated, here is a [light version](#)

# **DYNAMIC COMPONENTS**

# DYNAMIC COMPONENTS - MODALS



```
<!-- Button to trigger modal -->  
<button data-toggle="modal" data-target="#sign-up">Join</button>  
  
<!-- Modal -->  
<div class="modal fade" id="sign-up">  
  <!-- Modal content with Sign up form -->  
</div>
```

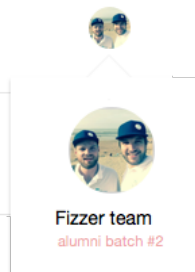
# DYNAMIC COMPONENTS - POPOVERS

Fizzer launch summer '14...

Posted on 7th July 2014 + ↗

Rent a roadstr car for Chri..

Posted on 6th Dec 2014 + ↗



```

                    <h2>Fizzer Team</h2>
                    <p>alumni batch #2</p>">
```

```
<script>
  $(function () {
    $(".popover-img").popover({
      html: true,
      trigger: "hover"
    });
  });
</script>
```

# DOCUMENTATION

<http://getbootstrap.com/javascript/>

# BOOTSTRAP GRID



# DEFINITION

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

# RESPONSIVE



# MOBILE-FIRST



# HOW IT WORKS?

# CONTAINER FIRST

```
<div class="container">  
  <!--Define a container-->  
</div>
```

# THEN INSERT ROWS

```
<div class="container">  
  <div class="row">  
    <!-- First row -->  
  </div>  
</div>
```

```
<div class="container">

  <div class="row">
    <!-- First row -->
  </div>

  <div class="row">
    <!-- Second row -->
  </div>

</div>
```

```
<div class="container">

  <div class="row">
    <!-- First row -->
  </div>

  <div class="row">
    <!-- Second row -->
  </div>

  <div class="row">
    <!-- Third row -->
  </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**)

# MEDIA

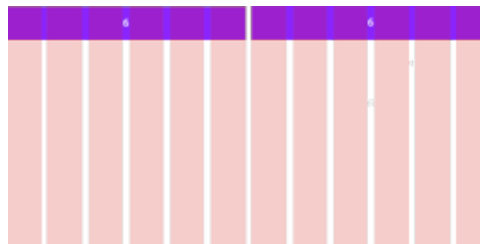
- **xs**: Extra small devices (Phones < 768px)
- **sm**: Small devices (Tablets > 768px)
- **md**: Medium devices (> 992px)
- **lg**: Large devices (Desktops > 1200px)

# EXAMPLE (2 COLUMNS)



# CODE

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

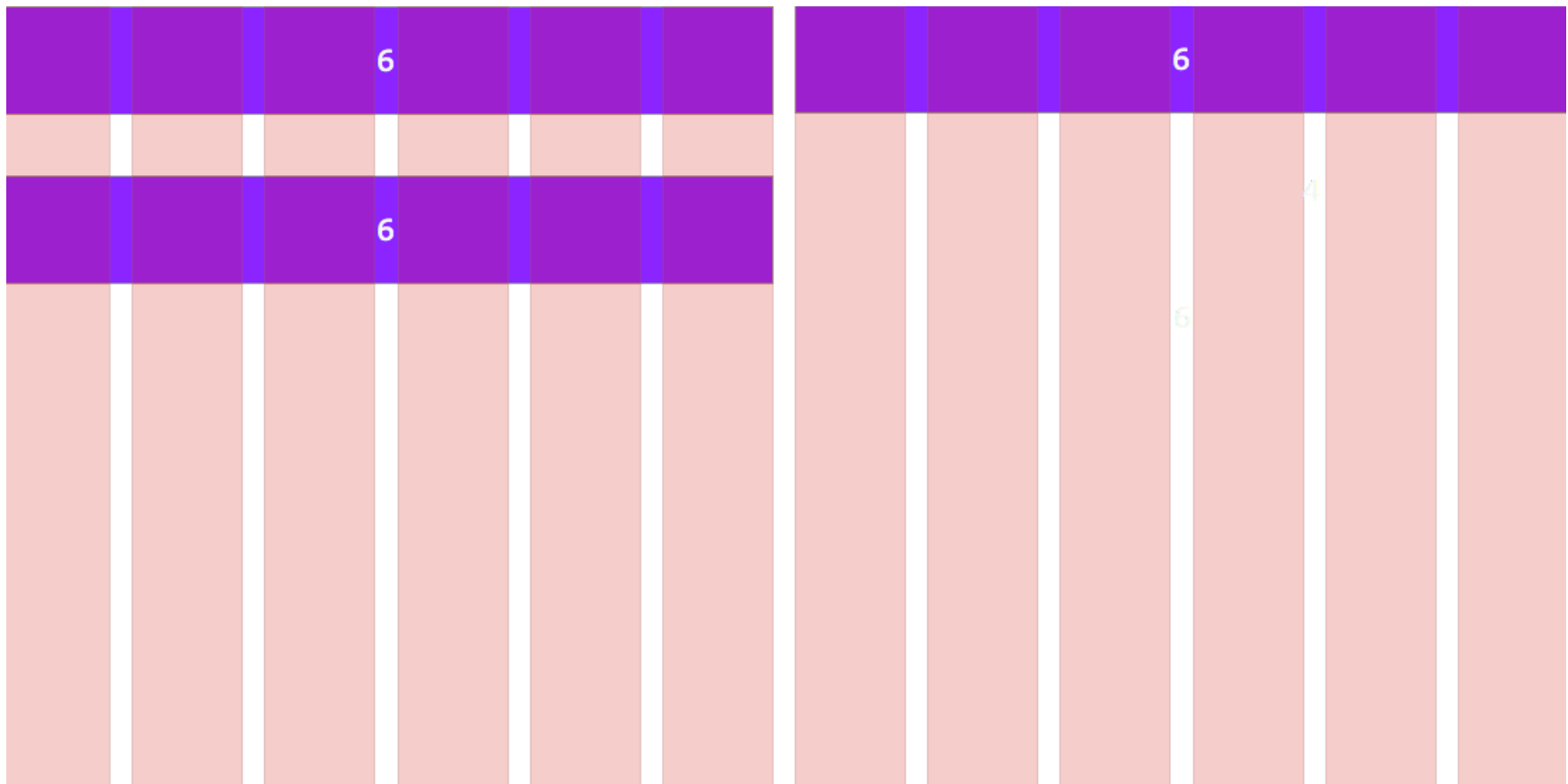


# MOBILE FIRST

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

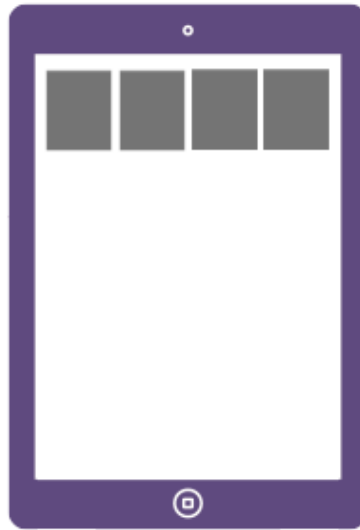


# JUST RETURN ON NEW LINE



# RESPONSIVE EXAMPLE

We want 4 blocks for tablets or larger screens





We want 2 rows for mobiles with 2 blocks on each row



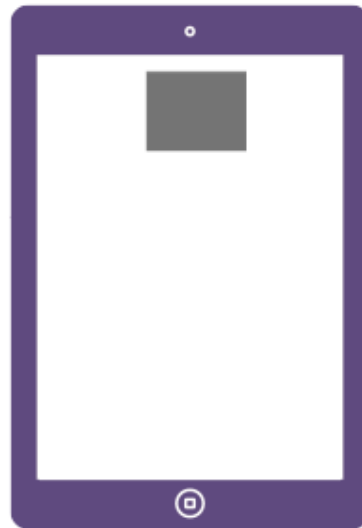
**HOW WILL YOU DO ?**

# CODE

```
<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>
```

Yeah baby !

# OFFSETTING BLOCKS



Can we offset a block ?

# YES WE CAN

```
<div class="container">
  <div class="row">
    <div class="col-sm-offset-4 col-sm-4">
      <!-- Le contenu de la div centrée -->
    </div>
  </div>
</div>
```

# GRID BEST PRACTICES (1)

- Never break the direct succession `.container > .row > .col`
- **Never touch the box model** of the grid elements (`.container`, `.row`, `.col`)
- **box-model == margin-border-padding-width**

# GRID BEST PRACTICES (2)

For styling, place div **around** the `.container` or **inside** the `.col`

```
<!--Example of good use of the grid-->
<div class="bg-primary padded">
  <div class="container">
    <div class="row">
      <div class="col-xs-4">
        <div class="social-box">
          <!-- your list of social links -->
        </div>
      </div>
      <div class="col-xs-8">
        <p>Le Wagon is the best company ever</p>
      </div>
    </div>
  </div>
</div>
```

# GRID BEST PRACTICES (3)

Then write your CSS on these divs.

```
/* You can touch at the box model of these divs */  
/* Because these are not direct grid elements */  
  
.padded{  
  padding: 50px 0;  
}  
  
.social-box{  
  border: 1px solid grey;  
  padding: 30px;  
  background: white;  
}
```



# GRID BEST PRACTICES (4)

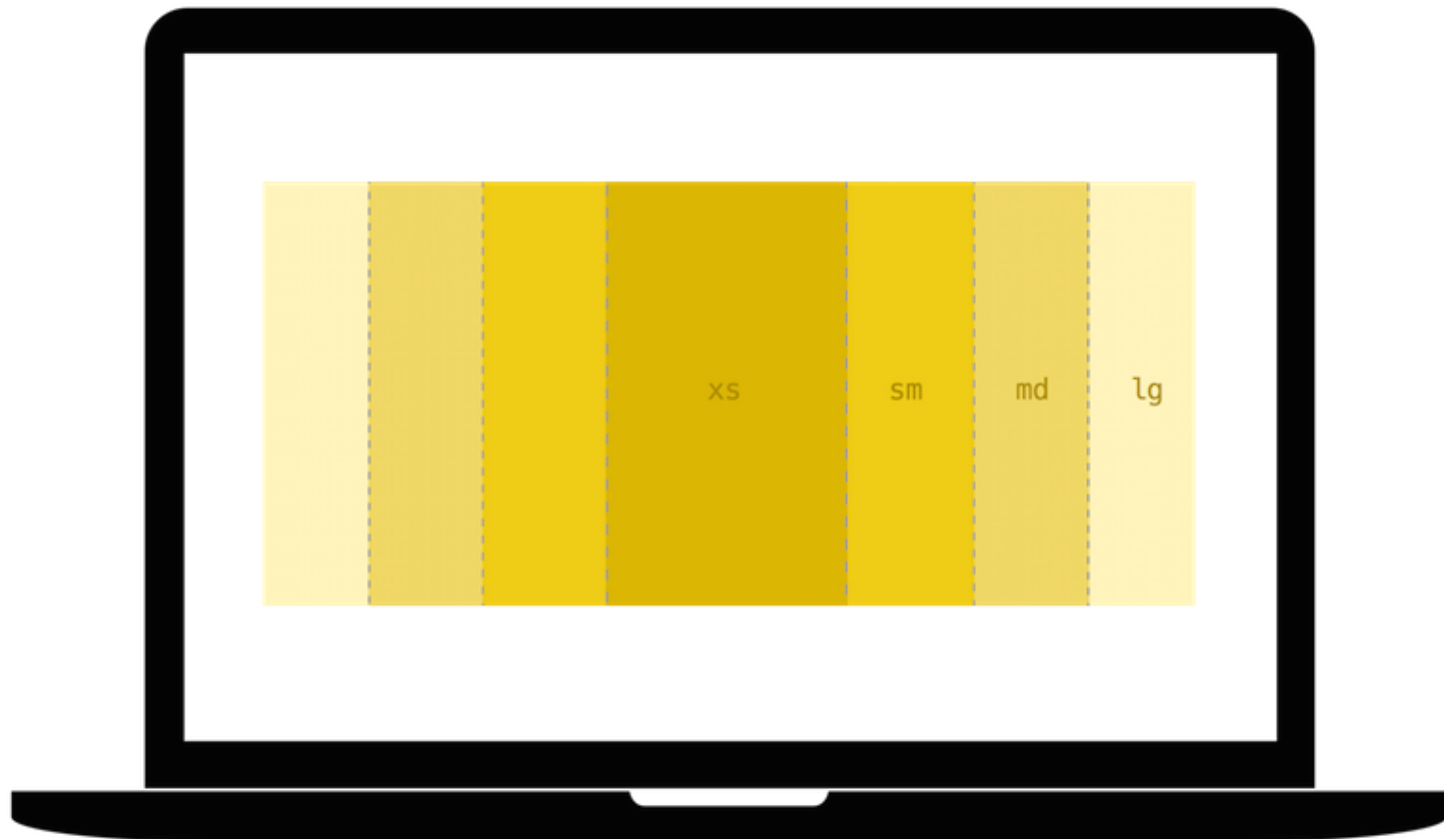
Images within the grid should hold the class `.img-responsive`

```

```

# CONTAINER

**Discrete** breakpoints



# CONTAINER FLUID

Continuous adaptation

