Section-End Project - Lesson 1

HTML, CSS + Bootstrap

## **Guiding notes**

Bootstrap offers a comprehensive documentation of all the features and components. We're not going to explain each feature in detail here, but instead, propose you some simple exercises where you can put some of those into practice.

We strongly suggest to have a look at the basics of HTML and CSS structure. Bootstrap is great, but it can't be of much help if you don't know the basics. These materials can be of help:

* HTML basics: <http://htmldog.com/guides/html/>
* CSS basics: <http://htmldog.com/guides/css/>

You can also check out these additional resources:

* HTML Introduction: <https://vimeo.com/34754506>
* Introduction to CSS: <https://vimeo.com/34754607>
* Inspecting the ID and class attributes: <https://vimeo.com/34754568>

## **Hands on exercises**

**Exercise 1. Hello world!**

There's no better way to start a relationship than saying hello. In this exercise we simply learn how to structure a basic html page, and to reference the proper resources.

<!DOCTYPE HTML><**html**>

<**head**>

<!-- This meta tags makes sure accents and other special caracters are displayed correctly -->

<**meta** http-equiv="Content-Type" content="text/html; charset=UTF-8">

<!--     We reference boostrap stylesheets. Notice that we reference the "online" version, the     one in ther content distribution network. If you download bootstrap, then you shoul reference    the one in your local system     -->

<**link** rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">

<!--      You can also override some of the styles that come by default, or create new ones. These extensions     should go in a separate files to make sure we don't lose them if we update the boostrap version in the future.     Note: Boostrap also offers the "source code" in \*less\* to facilitate extensions, but that's outsite the      scope of this course.     -->

<**link** rel="stylesheet" href="styles.css">

</**head**>

<**body**>

<!-- Visible part of the page goes here-->

<**h1**> Hello, wonderful and beautiful world! </**h1**>

</**body**></**html**>

**What should we do?**

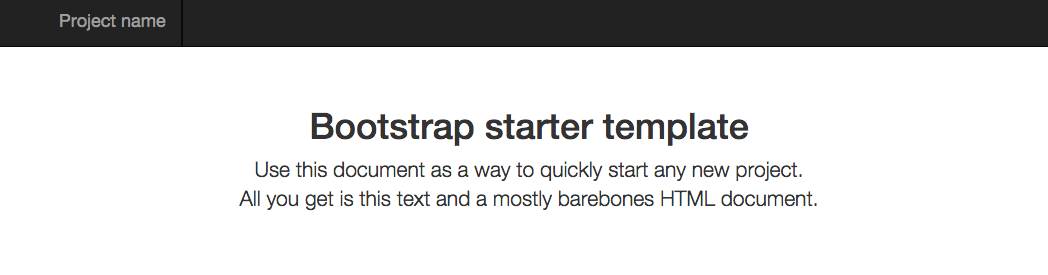
1. create an html page, call it hello.html. Here we will put the basic html structure above.
2. create styles.css. We will leave it empty for now, but in the next exercises we will put our specific styles here.
3. open the hello.html page in your browser

**Exercise 2. A basic page structure**

Using our hello world example, we can prepare the basic page structure. Our webpage will essentially have a navbar and a some content.

**To solve this exercise you need to look at:**

* navbar component: [http://getbootstrap.com/components/#navbar](http://getbootstrap.com/components/" \l "navbar)
* typography: [http://getbootstrap.com/css/#type](http://getbootstrap.com/css/" \l "type)

[](https://sites.google.com/site/webprogunitn/lab-sessions/lab-session-3/01-template.png?attredirects=0)

In the body tag, your result  should have a structure like the following:

<nav class="navbar navbar-default">

<**div** class="container-fluid">

<!-- this is for the header, the logo usually goes here -->

<**div** class="navbar-header">

<**a** class="navbar-brand" href="#">

Project name

</**a**>

</**div**>

</**div**>

</nav>

<!-- This is the main container of the entire page -->

<**div** class="container">

<**div** id="tagline">

<**h1**>Bootstrap starter template</**h1**>

<**p** class="lead">

Use this document as a way to quickly start any new project.

All you get is this text and a mostly barebones HTML document.

</**p**>

</**div**>

</**div**>

If you try it, you'll see that we don't get quite the result we wanted. Try the following:

1. Change the theme of the navbar from navbar-default to .navbar-inverse
2. Add a class to your styles.css file to center the element "tagline"

**Additional resources for beginners:**

* How do I reference a tag by "class" and "id"?

http://htmldog.com/guides/css/intermediate/classid/

* How do I center the text?

http://www.htmldog.com/guides/css/beginner/text/

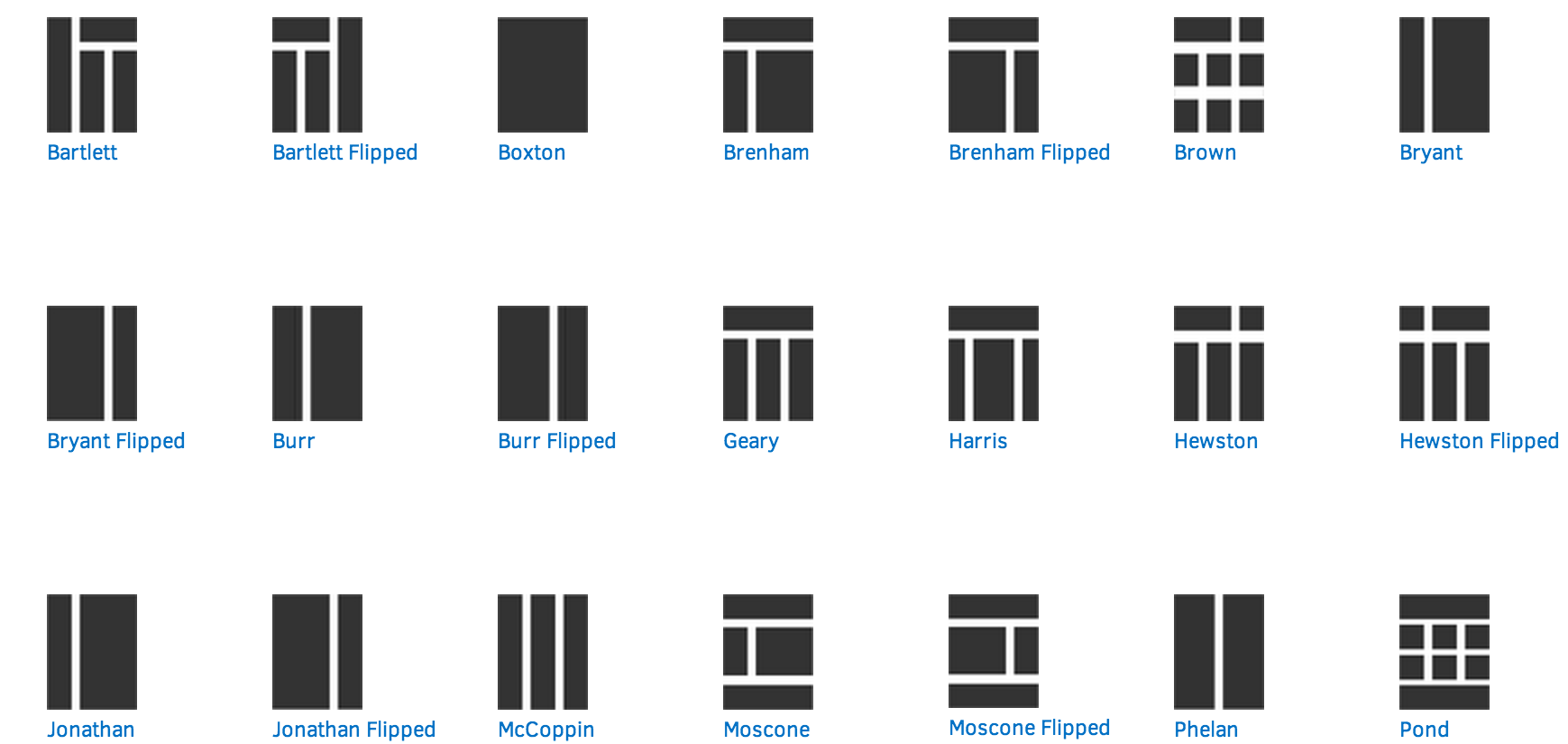
**Exercise 3. Playing with page layouts**

One of the most useful and life-saving features (students who have fought with low level properties will agree on this) is the the grid system. The grid system allow us to structure the document to reflect  given layout in a standard way.

In this exercise, we'll try to replicate the common page layouts (for desktop) that we see below by defining the proper html structure and using the proper classes. You can use the name of the class you're using as content.

**To solve this exercise you need to look at:**

* grid system: [http://getbootstrap.com/css/#grid](http://getbootstrap.com/css/" \l "grid)

[](https://sites.google.com/site/webprogunitn/lab-sessions/lab-session-3/02-grid.png?attredirects=0)

Let's do together one of the layouts: Bryant. This layout leaves spaces for a column on the left, probably for navigation or contextual information, and a larger area to the right, for the main content. We can distribute then the columns with the a row as follows:

<**div** class="container">

<**div** class="row">

<**div** class="col-md-3">.col-md-3</**div**>

<**div** class="col-md-9">.col-md-9</**div**>

</**div**>

</**div**>

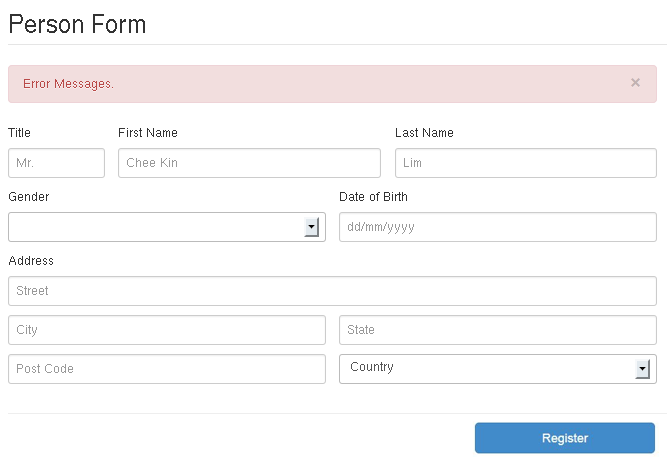
That should do the trick. Try now replicating the structure of the other layouts.

**Exercise 4. Defining form layouts**

A standard way of entering data is web pages is the form. Bootstrap provides a set of classes to organise forms in groups and define complex layouts, styles for input elements  and buttons, and some nice styles format messages. In this exercise we'll put them into practice in a representative "registration" form.

**To solve this exercise you need to look at:**

* Properties for forms: [http://getbootstrap.com/css/#forms](http://getbootstrap.com/css/" \l "forms)
* Buttons: [http://getbootstrap.com/css/#buttons](http://getbootstrap.com/css/" \l "buttons)
* Messages: [http://getbootstrap.com/css/#helper-classes](http://getbootstrap.com/css/" \l "helper-classes)

[](https://sites.google.com/site/webprogunitn/lab-sessions/lab-session-3/fluid-person-form.png?attredirects=0)

At this point you've probably looked at the documentation of forms and have an idea of how to define the form above. Still, we'll help you get started:

<**form**>

<!--              if you try using form-inline you won't get the desired output.              you'll see not only the form-groups, but all the elements inline.             here I go for the standard layout classes.        -->

<**div** class="row">

<**div** class="col-md-2">

<**div** class="form-group">

<!--                     'for' is used to link the label with the input element.                    if you click on the label, you'll see that the input gets                    the focus.               -->

<**label** for="ititle">Title</**label**>

<**input** type="text" class="form-control" id="ititle" placeholder="Mr.">

</**div**>

</**div**>

<**div** class="col-md-5">

<**div** class="form-group">

<**label** for="ifname">First name</**label**>

<**input** type="text" class="form-control" id="ifname" placeholder="Chee Kin">

</**div**>

</**div**>

<**div** class="col-md-5">

<**div** class="form-group">

<**label** for="ilname">Last name</**label**>

<**input** type="text" class="form-control" id="ilname" placeholder="Lin">

</**div**>

</**div**>

</**div**>

<!-- Well, here you should put the rest of the stuff -->

</**form**>

Complete the exercise with the rest of the input elements.

**Additional resources for beginners:**

* What are the most important components and properties of a form?: <http://www.htmldog.com/guides/html/beginner/forms/>
* What are the basic html elements to give structure to a form?: <http://www.htmldog.com/guides/html/advanced/forms/>

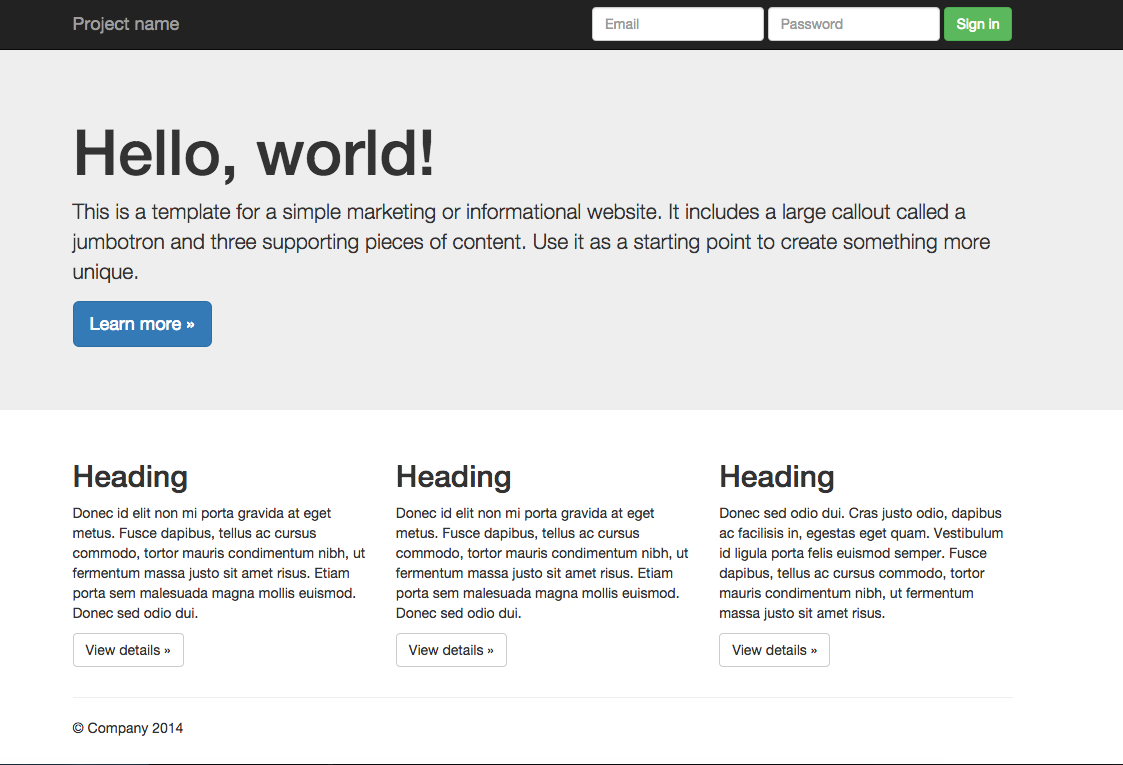
**Exercise 5. A standard landing page**

In this exercise we start putting things together in a complete "landing" page. We have seen how to structure navigation bars, page layouts,  typography and forms, and combining them requires essentially being able to go top-down from a macro structure to the specific structure of the individual components.

**To solve this exercise you need to look at:**

* All the above, plus
* the jumbotron, [http://getbootstrap.com/components/#jumbotron](http://getbootstrap.com/components/" \l "jumbotron)

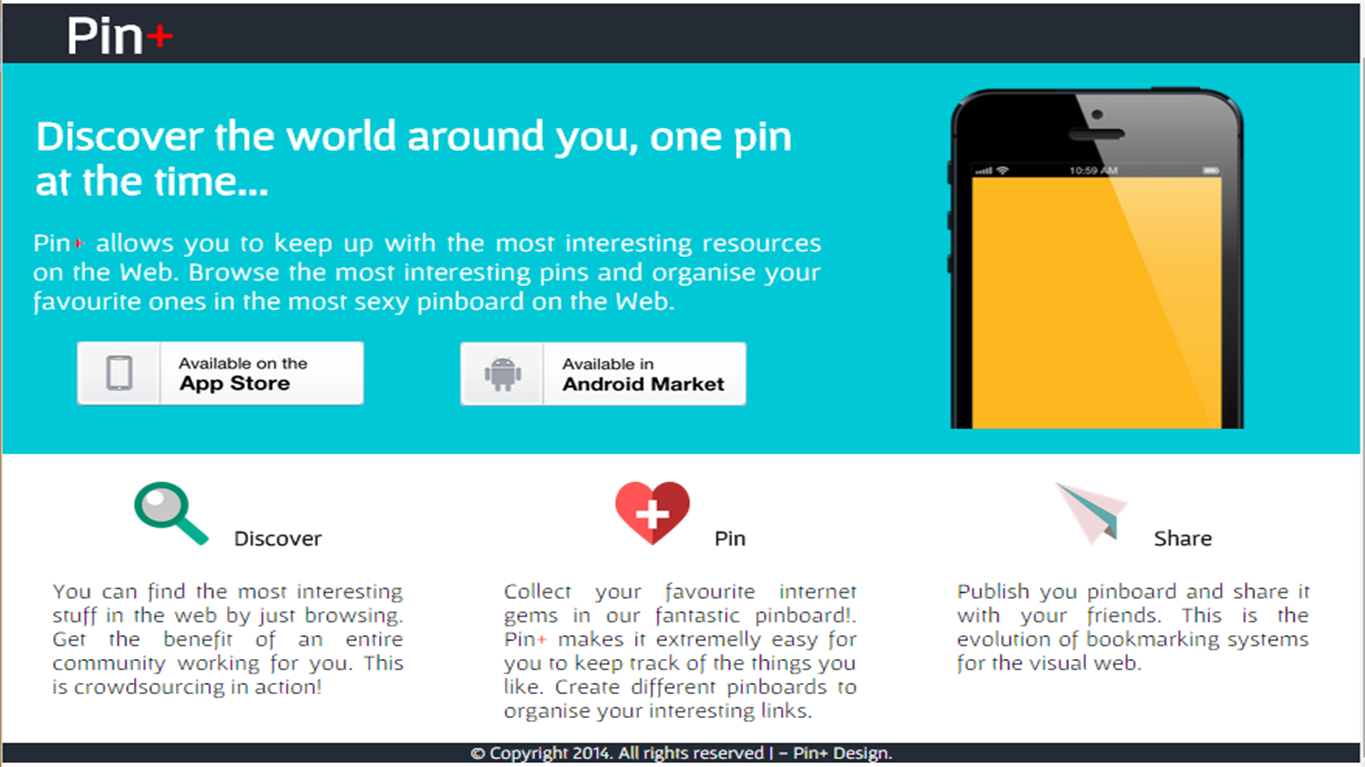
Your web page should replicate the structure and styles of the we page illustrated below.

[](https://sites.google.com/site/webprogunitn/lab-sessions/lab-session-3/04-landing.png?attredirects=0)

## **Homework**

In a real life scenario you'll need to add your own styles to a webpage. In this exercise we ask you to:

1. replicate the structure and layout using bootstrap
2. apply your own styles to approximate the graphical design of the mockup below

[](https://sites.google.com/site/webprogunitn/lab-sessions/lab-session-3/pinplus.png?attredirects=0)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*