**SASS and Bootstrap Documentation**

**SASS( 'Syntactically awesome style sheets)**  **:**

Sass is an extension of CSS that helps us to use various elements like variables, nested rules, imports and more. It helps to keep things organised and creating style sheets faster.

It’s basically a scripting language providing programming kind of functionality in writing CSS.

Stylus, styled-components, PostCSS, Bootstrap, and Less are the most popular alternatives and competitors to Sass.

SCSS is the more popular one out of all others because of it’s similarities to CSS. And easy to understand.

Sass includes two syntax options:

* SCSS (Sassy CSS):
* **Indented (simply called 'Sass')**

Similar to programming Sass allows us to store variables, write function, inhertance etc.

**Variables:**

**For example:**

$primary-color: blue;

div{

background-color: $primary-color;

}

**It is same as:**

div{

background-color: blue;

}

**Nesting**

The hierarchy same as programming. When written properly is an excellent method for designing.

**For example :**

ul{

margin: 0;

padding: 0;

li{

padding: 1rem;

}

}

**Is same as:**

ul{

margin: 0;

padding: 0;

}

ul > li{

padding: 1rem;

}

**Partials**

A simple separated code written in \_file-name.scss which can be import whenever we need.

**Import**

Imports the partials.

**For Example :**

In file – ‘\_header.scss’

html{

padding: 0;

margin: 0;

}

In other file - ‘style.scss’

@import ‘./header’;

div{

color: red;

}

**Is same as :**

html{

padding: 0;

margin: 0;

}

div{

color: red;

}

**Mixins:**

Mixins are like functions in programming language they help us store css like in a function and include wherever we feel necessary and also allow arguments to be passed.

**For Example:**

@mixin display-flex($direction){

diplay: flex;

justify-content: $direction;

}

body{

@include display-flex(center);

}

**Is same as :**

body{

diplay: flex;

justify-content: $direction;

}

**Extend/Inheritance:**

@extend helps us inherit styles. Different to mixins in which we pass arguments and include code here we get the entire css of a selector into the inherited one.

**For Example:**

div1{

color: red;

}

div2{

@extend div1;

background-color: black;

}

**Is same as:**

div1{

color: red;

}

div2{

color: red;

background-color: black;

}

**Bootstrap :**

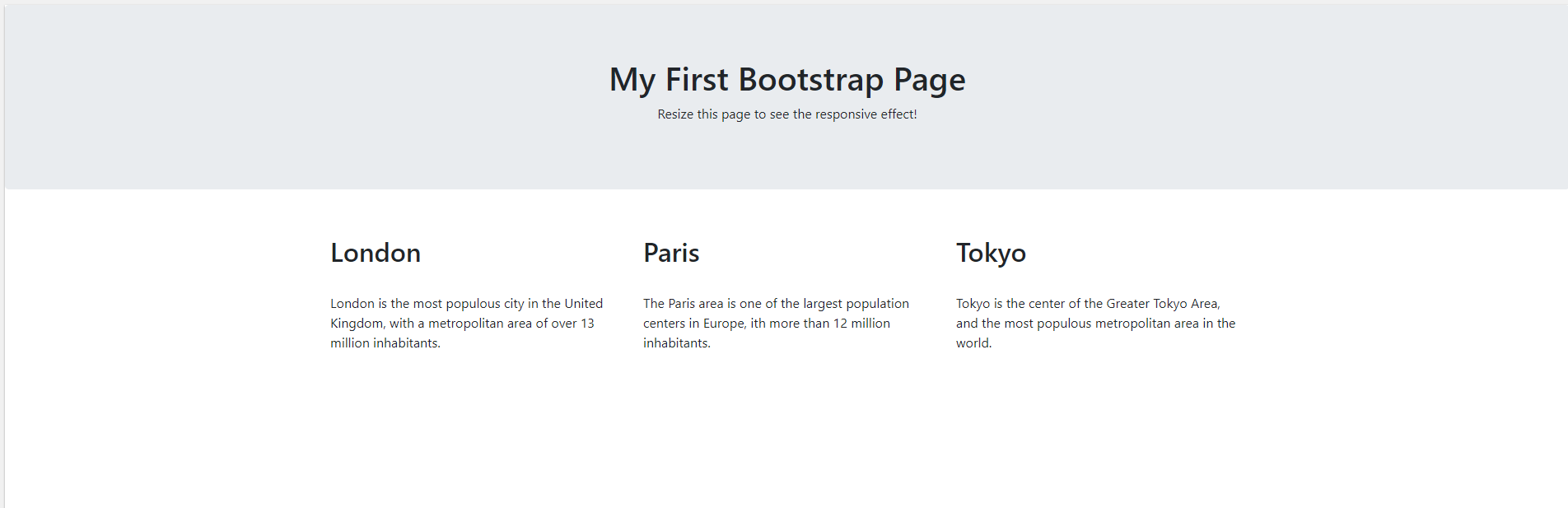
A free and open-source CSS Framework that helps us in creating responsive, mobile supportive pages and faster web development.

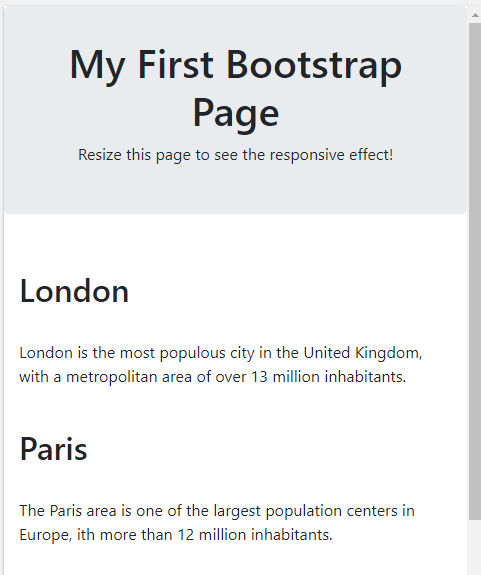
It also provides a lot of inbuilt JS function that provides a lot of interactiving allowing us to ease our workflow and speed of work.

**For Examples:**

<div class="jumbotron text-center">  
  <h1>My First Bootstrap Page</h1>  
  <p>Resize this page to see the responsive effect!</p>  
</div>  
  
<div class="container-fluid">  
  <div class="row">  
    <div class="col-sm-4">  
      <h2>London</h2>  
      <p>London is the capital city of England.</p>  
      <p>It is the most populous city in the United Kingdom,  
      with a metropolitan area of over 13 million inhabitants.</p>  
    </div>  
    <div class="col-sm-4">  
      <h2>Paris</h2>  
      <p>Paris is the capital of France.</p>  
      <p>The Paris area is one of the largest population centers in Europe,  
      with more than 12 million inhabitants.</p>  
    </div>  
    <div class="col-sm-4">  
      <h2>Tokyo</h2>  
      <p>Tokyo is the capital of Japan.</p>  
      <p>It is the center of the Greater Tokyo Area,  
      and the most populous metropolitan area in the world.</p>  
    </div>  
  </div>  
</div>

**Result :**

****

****

A simple code without writing the Vanilla CSS reduced our effort and work speed. Also providing us a responsive design.

By, L N S S Ravi Teja

A Full Stack Developer Intern