bootstrap 栅格布局

目标：体会框架，学会查阅文档,无耻求demo

1.bootstrap框架

Bootstrap，来自 Twitter，是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、javaScript 的，它简洁灵活，使得 Web 开发更加快捷。Bootstrap提供了优雅的HTML和CSS规范，它即是由动态CSS语言Less写成。Bootstrap一经推出后颇受欢迎，一直是GitHub上的热门开源项目，国内一些移动开发者较为熟悉的框架，如WeX5前端开源框架等，也是基于Bootstrap源码进行性能优化而来。

对框架的初步感受：

<!DOCTYPE html>

<html lang="zh-CN">

  <head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- 上述3个meta标签\*必须\*放在最前面，任何其他内容都\*必须\*跟随其后！ -->

    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css" rel="stylesheet">

  </head>

  <body>

    <!--按钮-->

    <button class="btn btn-info">hello world</button>

    <h1>标题一</h1>

    <h2>标题二</h2>

    <h3>标题三</h3>

    <h4>标题四</h4>

    <h5>标题五</h5>

    <h6>标题六</h6>

    <span class="h1">标题一</span>

    <span class="h2">标题二</span>

    <span class="h3">标题三</span>

    <span class="h4">标题四</span>

    <span class="h5">标题五</span>

    <span class="h6">标题六</span>

    <!-- jQuery (Bootstrap 的所有 JavaScript 插件都依赖 jQuery，所以必须放在前边) -->

    <script src="https://cdn.jsdelivr.net/npm/jquery@1.12.4/dist/jquery.min.js"></script>

    <!-- 加载 Bootstrap 的所有 JavaScript 插件。你也可以根据需要只加载单个插件。 -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/js/bootstrap.min.js"></script>

  </body>

</html>

2.栅格布局

原理：网格系统的实现原理非常简单，仅仅是通过定义容器大小，平分12份(也有平分成24份或32份，但12份是最常见的)，再调整内外边距，最后结合媒体查询，就制作出了强大的响应式网格系统。Bootstrap框架中的网格系统就是将容器平分成12份。

在使用的时候大家可以根据实际情况重新编译LESS（或Sass）源码来修改12这个数值（也就是换成24或32，当然你也可以分成更多，但不建议这样使用）。

使用规则:

1 行(.row)必须包含在容器（.container）中，以便为其赋予合适的对齐方式和内距(padding)。

2 在行(.row)中可以添加列(.column)，最多分配12。

3 具体内容应当放置在列容器（column）之内，而且只有列（column）才可以作为行容器(.row)的直接子元素

4 类似 .row 和 .col-sm-4 这种预定义的类，可以用来快速创建栅格布局。

5 通过为“列（column）”设置 padding 属性，从而创建列与列之间的间隔

为了确保适当的绘制和触屏缩放，  需要在<head>中添加viewport元数据标签

<meta name="viewport" content="width=device-width, initial-scale=1">

width=device-width           宽度等于设备宽度

initial-scale=1.0            初始显示大小——原始大小，即不进行缩放

**container 的设计大致如下：**



Bootstrap 需要为页面内容和栅格系统包裹一个 .container 容器。我们提供了两个作此用处的类。注意，由于 padding 等属性的原因，这两种 容器类不能互相嵌套。

.container 类用于固定宽度并支持响应式布局的容器。

<div class="container">

  ...

</div>

.container-fluid 类用于 100% 宽度，占据全部视口（viewport）的容器。

<div class="container-fluid">

  ...

</div>

实现原理：

.container {

  padding-right: 15px;

  padding-left: 15px;

  margin-right: auto;

  margin-left: auto;

}

@media (min-width: 768px) {/\*当宽度大于768px时触发\*/

  .container {

    width: 750px;

  }

}

@media (min-width: 992px) {/\*当宽度大于992px时触发\*/

  .container {

    width: 970px;

  }

}

@media (min-width: 1200px) {/\*当宽度大于1200px时触发\*/

  .container {

    width: 1170px;

  }

}

.container-fluid { /\*这个是宽度默认\*/

  padding-right: 15px;

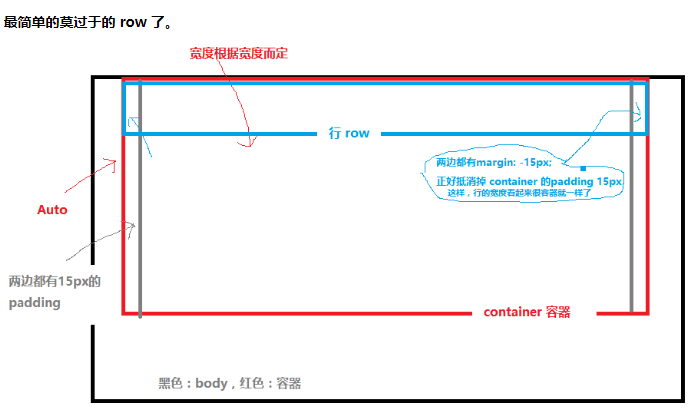
  padding-left: 15px;

  margin-right: auto;

  margin-left: auto;

}

**行 row**



**代码实现：**

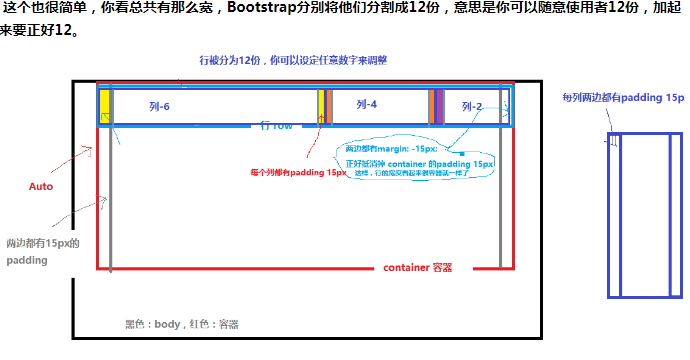
.row {

margin-right: -15px;

margin-left: -15px;

}

**列 Column**



不过要注意的一点是，Bootstrap实现了更好的响应式布局，列的种类是有很多种的。

.col-xs-\* 针对超小屏幕 手机（<768px）

.col-sm-\* 小屏幕 平板 (≥768px)

.col-md-\* 中等屏幕 桌面显示器 (≥992px)(栅格参数)

.col-lg-\* 针对特大的（≥1200px）

这些是什么意思？如何使用的呢？假设你写如下代码:

<div class="col-sm-10 col-md-8"></div><div class="col-sm-3 col-md-4"></div>

当这两个div在row里面之后。如果是这样 这两个div在小屏幕中会排2排 因为10+3>12 ,在中等屏幕中可以排同一排 8+4=12

说白了。就是在不同的宽度里面Bootstrap将帮助你选择不同的类，你可以用这些类定义你想在不同宽度的界面排版。

代码实现：

col-xs-\*的：

.col-xs-1, .col-sm-1, .col-md-1, .col-lg-1, .col-xs-2,/\*后面代码省略，选择了所有的列，所有不同的列类型全部都有这样的有这样\*/

{

  position: relative;

  min-height: 1px;

  padding-right: 15px;

  padding-left: 15px;

}

.col-xs-1, .col-xs-2, .col-xs-3, .col-xs-4, .col-xs-5, .col-xs-6, .col-xs-7, .col-xs-8, .col-xs-9, .col-xs-10, .col-xs-11, .col-xs-12 {

  float: left;

}

.col-xs-12 {

  width: 100%;

}

.col-xs-11 {

  width: 91.66666667%;

}

.col-xs-10 {

  width: 83.33333333%;

}

.col-xs-9 {

  width: 75%;

}

.col-xs-8 {

  width: 66.66666667%;

}

.col-xs-7 {

  width: 58.33333333%;

}

.col-xs-6 {

  width: 50%;

}

.col-xs-5 {

  width: 41.66666667%;

}

.col-xs-4 {

  width: 33.33333333%;

}

.col-xs-3 {

  width: 25%;

}

.col-xs-2 {

  width: 16.66666667%;

}

.col-xs-1 {

  width: 8.33333333%;

}

作业：

0.整理笔记，课堂案例

1.智能人工客服：

