

Bootstrap

1.0 响应式开发原理

1.1 响应式开发原理

就是使用媒体查询针对不同宽度的设备进行布局和样式的设置，从而适配不同设备的目的。

设备的划分情况：

- 小于768的为超小屏幕（手机）
- 768~992之间的为小屏设备（平板）
- 992~1200的中等屏幕（桌面显示器）
- 大于1200的宽屏设备（大桌面显示器）

1.2 响应式布局容器

响应式需要一个父级做为布局容器，来配合子级元素来实现变化效果。

原理就是在不同屏幕下，通过媒体查询来改变这个布局容器的大小，再改变里面子元素的排列方式和大小，从而实现不同屏幕下，看到不同的页面布局和样式变化。

父容器版心的尺寸划分

- 超小屏幕（手机，小于 768px）：设置宽度为 100%
- 小屏幕（平板，大于等于 768px）：设置宽度为 750px
- 中等屏幕（桌面显示器，大于等于 992px）：宽度设置为 970px
- 大屏幕（大桌面显示器，大于等于 1200px）：宽度设置为 1170px

但是我们也可以根据实际情况自己定义划分

2.0 bootstrap的介绍

2.1 Bootstrap简介

Bootstrap 来自 Twitter（推特），是目前最受欢迎的前端框架。Bootstrap 是基于HTML、CSS 和 JAVASCRIPT 的，它简洁灵活，使得 Web 开发更加快捷。

[中文网](#) [官网](#) [推荐网站](#)

框架：顾名思义就是一套架构，它有一套比较完整的网页功能解决方案，而且控制权在框架本身，有预制样式库、组件和插件。使用者要按照框架所规定的某种规范进行开发。

2.2 bootstrap优点

- 标准化的html+css编码规范
- 提供了一套简洁、直观、强悍的组件
- 有自己的生态圈，不断的更新迭代
- 让开发更简单，提高了开发的效率

2.3 版本简介

2.x.x: 停止维护,兼容性好,代码不够简洁, 功能不够完善。

3.x.x: 目前使用最多,稳定,但是放弃了IE6-IE7。对 IE8 支持但是界面效果不好,偏向用于开发响应式布局、移动设备优先的WEB 项目。

4.x.x: 最新版, 目前还不是很流行

2.4 bootstrap基本使用

在现阶段我们还没有接触JS相关课程, 所以我们只考虑使用它的样式库。

Bootstrap 使用四步曲:

1. 创建文件夹结构



2. 创建 html 骨架结构

```
<!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="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and
    media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file://
    -->
    <!--[if lt IE 9]>
      <script src="//cdn.bootcss.com/html5shiv/3.7.2/html5shiv.min.js">
    </script>
```

```

    <script src="//cdn.bootcss.com/respond.js/1.4.2/respond.min.js">
</script>
    <![endif]-->
</head>
<body>
    <h1>你好，世界! </h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files
as needed -->
    <script src="js/bootstrap.min.js"></script>
</body>
</html>

```

3. 引入相关样式文件

```

<!-- Bootstrap 核心样式-->
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">

```

4. 书写内容

直接拿Bootstrap 预先定义好的样式来使用

修改Bootstrap 原来的样式，注意权重问题

学好Bootstrap 的关键在于知道它定义了哪些样式，以及这些样式能实现什么样的效果

2.5 bootstrap布局容器

Bootstrap 需要为页面内容和栅格系统包裹一个 .container 或者.container-fluid 容器，它提供了两个作此用途的类。

.container

- 响应式布局的容器 固定宽度
- 大屏 (≥1200px) 宽度定为 1170px
- 中屏 (≥992px) 宽度定为 970px
- 小屏 (≥768px) 宽度定为 750px
- 超小屏 (100%)

.container-fluid

- 流式布局容器 百分百宽度
- 占据全部视口 (viewport) 的容器。

2.6 bootstrap栅格系统

Bootstrap提供了一套响应式、移动设备优先的流式栅格系统，随着屏幕或视口 (viewport) 尺寸的增加，系统会自动分为最多12列。

栅格系统用于通过一系列的行（row）与列（column）的组合来创建页面布局，你的内容就可以放入这些创建好的布局中。

- 按照不同屏幕划分为1~12 等份
- 行（row） 可以去除父容器作用15px的边距
- xs-extra small：超小； sm-small： 小； md-medium： 中等； lg-large： 大；
- 列（column） 大于 12，多余的“列（column）”所在的元素将被作为一个整体另起一行排列
- 每一列默认有左右15像素的 padding
- 可以同时为一列指定多个设备的类名，以便划分不同份数 例如 class="col-md-4 col-sm-6"

栅格嵌套

栅格系统内置的栅格系统将内容再次嵌套。简单理解就是一个列内再分成若干份小列。我们可以通过添加一个新的 .row 元素和一系列 .col-sm-* 元素到已经存在的 .col-sm-* 元素内。

```
<!-- 列嵌套 -->
<div class="col-sm-4">
  <div class="row">
    <div class="col-sm-6">小列</div>
    <div class="col-sm-6">小列</div>
  </div>
</div>
```

列偏移

使用 .col-md-offset-* 类可以将列向右侧偏移。这些类实际是通过使用 * 选择器为当前元素增加了左侧的边距（margin）。

```
<!-- 列偏移 -->
<div class="row">
  <div class="col-lg-4">1</div>
  <div class="col-lg-4 col-lg-offset-4">2</div>
</div>
```

列排序

通过使用 .col-md-push-* 和 .col-md-pull-* 类就可以很容易的改变列（column）的顺序。

```
<!-- 列排序 -->
<div class="row">
  <div class="col-lg-4 col-lg-push-8">左侧</div>
  <div class="col-lg-8 col-lg-pull-4">右侧</div>
</div>
```

响应式工具

为了加快对移动设备友好的页面开发工作，利用媒体查询功能，并使用这些工具类可以方便的针对不同设备展示或隐藏页面内容。

类名	超小屏	小屏	中屏	大屏
.hidden-xs	隐藏	可见	可见	可见
.hidden-sm	可见	隐藏	可见	可见
.hidden-md	可见	可见	隐藏	可见
.hidden-lg	可见	可见	可见	隐藏

3.0 阿里百秀案例制作

3.1 技术选型

方案：我们采取响应式页面开发方案

技术：bootstrap框架

设计图： 本设计图采用 1280px 设计尺寸

项目结构搭建

Bootstrap 使用四步曲：

- 1. 创建文件夹结构
- 2. 创建 html 骨架结构
- 3. 引入相关样式文件
- 4. 书写内容

container宽度修改

因为本效果图采取 1280的宽度， 而Bootstrap 里面 container宽度 最大为 1170px， 因此我们需要手动改下container宽度

```
/* 利用媒体查询修改 container宽度适合效果图宽度 */
@media (min-width: 1280px) {
    .container {
        width: 1280px;
    }
}
```