day15

五.动画

六.CSS优化

|  |
| --- |
| 为了提升用户体验，节省服务器资源消耗  1.优化的原则  尽量减少请求个数--------使用缓存和精灵图(雪碧图)  页面的顶部，引入css文件  将css和js存放外部文件中  2.css代码优化  缩小样式文件的大小(用简写方式，代码重用)  不要出现空的src和href  代码压缩 |

七.css hack

|  |
| --- |
| 过时的知识点，低版本浏览器无法解析很多css样式  为了让css能够在低版本浏览器中正常运行。我们要对css代码有特殊的写法  background-img:-webkit-linear-gradient(top,#aaa,#bbb);  -webkit-  -moz-  -o-  -ms-  tmooc 第一个视频集合中，第三或者第四个--------20分钟支持面试 |

八.css reset和normalize.css

|  |
| --- |
| 不同浏览器的内核对同一篇代码，默认的解析方式不同  导致同一篇代码在不同浏览器可能会出现多种效果  我们在写代码之前，要把有冲突的代码进行重置（删除或者统一成一个值）  写这个重置，就叫做css reset  css reset没有标准格式和规则  css reset很霸道，把很多标签自带的语义效果都清除了 |

css reset的另外一种解决方案

|  |
| --- |
| normalize.css 很温和，保留元素的本身语义  把不同浏览器中，不兼容或者有bug的部分进行了调整  保留了大多数元素的基本语义  normalize.css有标准写法 |

Bootstrap

一.css的响应式布局(css3 2010提出)

|  |
| --- |
| Response web page 响应式/自适应网页  可以根据浏览器设备的不同，  而自动的改变布局，图片，文本效果，从而不影响用户体验 |

1.完成响应式布局的前提条件----------面向初学者

|  |
| --- |
| ①布局，尽量不要固定元素宽度，尽量使用流式布局(默认文档流+浮动)+弹性  ②尽量使用相对单位，让文本和图片随着容器改变大小  ③使用css3提供的媒体查询奇数  注意，复杂网页不适合使用响应式布局 |

2.如何测试响应式网页

|  |
| --- |
| ①使用真实设备测试  好处：真实可靠  缺点：成本太高了，测试任务量巨大  ②第三方模拟软件  好处：方便，快捷  缺点：测试效果有限  ③使用chrome自带的模拟器  缺点：极度不负责 |

3.编写响应式

①移动端适配

|  |
| --- |
| 只有在网页运行在移动端时才必须适配  <meta name="viewport" 设置视口  content="  width=device-width, 视口宽度为设备宽度  initial-scale=1.0, 默认不允许缩放  maximum-scale=1.0, 最大缩放比例1倍  user-scalable=0 不允许用户缩放  ">  简化写法  <meta name="viewport" content="width=device-width, initial-scale=1"> |

②媒体查询

|  |
| --- |
| CSS3 media query 做响应式布局的必备技术  Media 是指浏览网页的设备  query 根据浏览网页的设备不同，自动的匹配某些css样式，而忽略其他css样式 |
| Media 是指浏览网页的设备----条件  1.硬件设备 screen (pc、pad、phone)  TV  print  2.屏幕尺寸，业内把所有屏幕分5种尺寸  超大屏 xl view>=1200px  大屏 lg 992px<=view<1200px  中屏 md 768px<=view<992px  小屏 sm 576px<=view<768px  超小屏 xs view<576px max-width:575.99px  @media 设备 and 屏幕{  选择器{样式}  }  @media screen and (min-width:576px) and (max-width:767.99px) {  #d1{  font-size: 40px;  background: #afa;  color:#f0f;  }  } |

二.Bootstrap

|  |
| --- |
| boot www.bootcss.com  boot把很多的样式效果进行了封装，封装了类  我们只需要调用类名，就可以快速实现效果  问题：boot中关于距离，颜色只有几种，无法匹配所有项目需求  关于对齐，弹性，响应式等，项目中使用频繁  总结：不要求大家一定背熟类名  学习步骤：  1.起步，如何使用boot  2.全局css样式，一堆封装好的类  3.组件、js插件  4.定制 scss |

1.如何使用boot

①设置视口

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

②导入1个css和3个js文件，有顺序要求

|  |
| --- |
| <link rel="stylesheet" href="css/bootstrap.css"/>  <script src="js/jquery.min.js"></script>  <script src="js/popper.min.js"></script>  <script src="js/bootstrap.min.js"></script> |

③在body中编写div.container

|  |
| --- |
| <body>  <div class="container">  <!-- 编写代码的位置 -->  </div>  </body> |

2.全局css样式

①容器

|  |
| --- |
| .container 在不同屏幕下，都写死最大宽度，同时水平居中  定宽容器  . container-fluid 变宽容器 |

②按钮

|  |
| --- |
| btn 基本类  按钮颜色 btn-danger/success/warning/info/primary/secondary/dark/light  镂空按钮 btn-outline-danger  按钮大小 btn-lg/sm 通过内边距，字号，行高改变的按钮大小  btn-block 块级按钮 |

③图片相关的

|  |
| --- |
| rounded 圆角 0.25rem  rounded-circle 50%的圆角  img-thumbnail 带边框，带白色内边距的图片---缩略图  img-fluid 图片可以缩放，但是不能超过图片原始大小 |

④文本相关

|  |
| --- |
| 1.文本颜色  text-muted/danger/warning/info......  2.文本大小写  text-uppercase/text-lowercase/text-capitalize  3.文本加粗  font-weight-light/normal/bold  4.字号大小 .h1~~.h6  5.文本的水平对齐方式  text-left/center/right/justify |

⑤列表组的样式

|  |
| --- |
| 如果项目需求不需要列表组，就可以不写  list-unstyled 去除ul的列表标识，去除左内边距  ul.list-group 弹性，y轴主轴  li.list-group-item 设置列表组的列表项  list-group-item-danger li的颜色  active 选中  disabled 禁用 |

作业：

1.scss的安装，下周一之前能安装成功

2.使用媒体查询完成 3个屏幕



3. 使用媒体查询完成 3个屏幕

web第二阶段练习效果图示\day14\_boot\_01练习作业\课后作业\效果图

