Bootstrap4.0基础

一、Bootstrap的简介和作用

1.Bootstrap的简介

- Bootstrap是由twitter公司设计师基于html，css，js开发的简洁、开源、强大、优雅的UI框架。

- UI框架就是能够快速构建html架构的框架。

- 内置了大量的css类，供元素使用。

- Bootstrap由一个css库和一个js库组成，因为在整个ui框架中还会有许多动态需效果需要js来完成，因此，Bootstrap在动态部分是依赖jquery一个js类库完成的。`Bootstrap = jquery + css`

- 【注意】不需要背，查询文档知道每个类的作用即可

2.准备工作

【提示】讲解的是Bootstrap v4.0 版本，到5版本会有一些改变，大家注意查询手册，比如左右变成start和end

【查看】下载Bootstrap之后会得到很多文件，有很多不需要，为大家精简了一个版本，1个css和3个js。

【拷贝】将css文件放入一个新建的css文件夹下，3个js文件放到一个新建的js文件夹下

【操作】将boot.json内的文本创建全局代码片段boot

【安装】在vscode中安装 `HTML CSS Support`插件,可以提示一部分css

图形用户界面, 文本, 应用程序

描述已自动生成2.安装sass【作业】

- 安装插件

- 看视频<https://share.weiyun.com/FYgNL1M9> 《sass的更多安装方法与编译方法》和《依赖ruby的安装方法》

- 两个视频都先看一遍，“都看完”再选择一种进行安装(尽量使用npm安装方法可以少安装一个软件)

- 安装后编译成功，会出现css文件

（1）使用npm安装sass

- 查看是否安装npm，打开`cmd`,输入`npm -v`

- 打开`cmd`,输入`npm install -g sass`，安装成功后检查`sass -v`

- 如果上面的方法因为网的问题没有安装成功，可以使用淘宝镜像

- ` npm install -g cnpm --registry=https://registry.npm.taobao.org`

- 再使用cnpm安装sass`cnpm install sass`

（2）依赖ruby安装sass

- 依赖ruby

- 安装ruby，ruby软件在ready的tools中

- 查看自己电脑是32位还是64位，按照实际情况安装x86或者x64

- 查看ruby是否安装成功，打开`cmd`,输入`ruby -v`

- 安装sass

- mac：`sudo gem install sass`

- windows:`gem install sass`

- 安装完之后在`cmd`中检查 `scss -v`，如果sass -v提示没有这个指令，那么就输入`scss -version`出现版本号就可以了

- 文件编译和监听

- 在自己创建的一个文件夹demo中再创建两个文件夹scss 和 css，任何文件夹不能出现中文字符

- 使用终端或vscode终端在要编译的文件夹下打开demo文件夹，注意路径

- 在终端demo文件夹目录下，直接输入指令 `sass -w scss:css` css出现新的同名文件

- 以上编译方式参照视频

（3）编译出现的错误【重点】

编译中会出现很多错，比如编译之后没有任何效果，报错等

- sass项目的编译整个项目不允许有任何中文字符的文件夹

- 不运行可能是你的安全助手或者防火墙阻止了ruby，注意弹窗，不要阻止

- 路径写错，注意路径的关系

- 使用windows自带的cmd进行编译

- powershell编译报错，在终端切换成command prompt 也就是cmd编译，或者直接使用使用windows自带的cmd进行编译

二、布局容器

1.定宽容器 container

container的css,设置了一些基本样式以及响应式布局，会根据屏幕宽度变更宽度，并且居中，定死最大宽度。对比一个普通的div在不同的屏幕下都定义了最大宽度

|  |
| --- |
| .container {  width: 100%;  padding-right: 15px;  padding-left: 15px;  margin-right: auto;  margin-left: auto;  }  sm  @media (min-width: 576px) {  .container {  max-width: 540px;  }  }  md  @media (min-width: 768px) {  .container {  max-width: 720px;  }  }  lg  @media (min-width: 992px) {  .container {  max-width: 960px;  }  }  xl  @media (min-width: 1200px) {  .container {  max-width: 1140px;  }  } |

2.变宽容器 container-fluid

宽度随着body的变化而改变

|  |
| --- |
| .container-fluid {  width: 100%;  padding-right: 15px;  padding-left: 15px;  margin-right: auto;  margin-left: auto;  } |

三、工具类

1.颜色和排版

常用颜色：

图表, 背景图案

描述已自动生成- `primary` - 重要

- `success` - 成功

- `danger` - 危险

- `warning` - 警告

- `info` - 信息

- `light` - 亮色

- `dark` - 深色

- `secondary` - 浅灰

- `white` - 白色

- `transparent` - 透明色

常用排版字号

- `.h1 ~.h6` 让元素的文字呈现出标题效果

文本

低可信度描述已自动生成- `.display-1~.display-4`一种更大，更自以为是的标题样式

|  |
| --- |
| `primary` - 重要  `success` - 成功  `danger` - 危险  `warning` - 警告  `info` - 信息  `light` - 亮色  `dark` - 深色  `secondary` - 浅灰  `white` - 白色  `transparent` - 透明色 |

(1)文字颜色

- `.text-primary`

可以选择Bootstrap提供的色值

(2)背景颜色

- `.bg-primary`

可以选择Bootstrap提供的色值

2.文本样式

(1)文本对齐

- `.text-left` 左对齐

- `.text-center` 居中对齐

- `.text-right` 右对齐

(2)字体粗细和斜体

- `.font-weight-bold` 加粗

- `.font-weight-light` 细体

- `.font-italic` 斜体

3.调整大小

- 相对于父级的百分比，包括 25%、50%、75%、 100%

- w:宽度 `.w-50` 是父元素宽度的50%

- h:高度 `.h-25` 是父元素高度的50%

- 缺点：这能设置固定的几种百分比

4.边框

(1)边框属性

`.border` 基类属性 全边框 border: 1px solid #dee2e6

(2)边框方向

- `.border-left` 左边框

- `.border-right` 边框

- `.border-top` 上边框

- `.border-bottom` 下边框

(3)去掉边框

- `.border-right-0` 去掉某边框的宽度，left，right，top，bottom

- `.border-0` 去掉所有的边框

(4)边框颜色

- `.border-primary` 多为自定义颜色

(4)圆角

- `.rounded` 圆角属性

- `.rounded-top`圆角方向top，right，bottom，left

- `.rounded-0`去掉圆角

- `.rounded-circle` 最大圆角50%圆角

5.间距

- m ：margin值

- p ：padding值

- m-auto ：代表块级元素在父级里居中

- t、r、b、l：分别代表上右下左

- 如`.mt-1` 默认尺寸是1，0~5

- 如`.pb-1` 默认尺寸是1，0~5

- 响应式 `ml-lg-\*`、`pt-sm-\*`等 包括的响应式尺寸有xl，lg，md，sm

6.浮动

- `.float-left` ：元素左浮动

- `.float-right` ：元素右浮动

- `.clearfix` ：子元素float后，通过给父元素添加 `.clearfix` 达到清除浮动效果

7.定位

- `.position-relative` 相对定位,定位的方向需要单独写

- `.position-absolute` 绝对定位，注意父元素需要有定位属性，定位的方向需要单独写

- `.position-fixed` 固定定位，有底部固定和顶部固定两种设置，但其他需要单独设定

- `.fixed-top` 固定定位顶部

- `.fixed-bottom` 固定定位底部

- 缺点：top、left、bottom、right 还需要单独设置

8.列表样式

`.list-unstyled` 去掉列表原有样式

9.显示

- `.d-none`元素消失

- `.d-block`元素按照块级显示

- `.d-inline`元素按照内联显示

- `.d-inline-block`元素按照行内块显示

- `.d-flex`元素按照弹性布局显示

- 支持响应式`.d-\*-none` sm、md、lg、xl,需要注意的是看源码的区间

10.flex布局

(1)flex属性

- 所有需要使用flex布局的类都需要先在容器中加入以下两个类,否则不生效

- `.d-flex` 容器设置flex布局

- `.d-inline-flex` 内联元素设置flex布局

- 所有弹性都支持响应式`.d-sm-flex`,他规定了四个尺寸sm小尺寸,md中等尺寸,lg大尺寸,xl超大尺寸

(2)方向

- `.flex-row` 水平方向正向排版

- `.flex-row-reverse` 水平方向反向排版

- `.flex-column` 垂直方向反向排版

- `.flex-column-reverse` 垂直方向反向排版

- 同样他规定了四个尺寸sm小尺寸,md中等尺寸,lg大尺寸,xl超大尺寸

- `.flex-\*-row` 水平方向正向排版

- `.flex-\*-row-reverse` 水平方向反向排版

- `.flex-\*-column` 垂直方向反向排版

- `.flex-\*-column-reverse` 垂直方向反向

(3)内容对齐

- `.justify-content-\*` 代表水平轴的对齐方式

- 支持响应式，\*号位置代表sm,md,lg,xl

- `.justify-content-\*-start` (浏览器默认值)主轴起始位置对齐

- `.justify-content-\*-end` 主轴结束位置对齐

- `.justify-content-center` 居中

- `.justify-content-between` 两端对齐

- `.justify-content-around ` 项目左右间距相同,两元素间距是左右两侧的一倍

(4)对齐项目

- align-items-\* 代表主轴的交叉轴的对齐方式

- 支持响应式，\*号位置代表sm,md,lg,xl

- `.align-items-\*-start` 交叉轴起始位置对齐

- `.align-items-\*-end` 交叉轴结束位置对齐

- `.align-items-\*-center`交叉轴居中对齐

- `.align-items-\*-baseline`交叉轴及基线位置对齐

图形用户界面, 应用程序, 网站

描述已自动生成- `.align-items-\*-stretch`交叉轴占满宽高

【练习】