

1. Bootstrap4 框架

- Bootstrap 是最受欢迎的 HTML \ CSS \ JS 框架，用于开发响应式布局、移动设备优先的web项目。(以下都是bootstrap4)
 - 特色：
 1. 响应式布局。
 2. 基于 flex 的栅格系统。
 3. 丰富的组件和工具方法。
 4. 常见交互使用。
 - 官网:<https://getbootstrap.com>
 - 中文:<https://getbootstrap.net/>
 - 下载完成原码后, dist 文件夹是 bootstrap 内容, scss 文件夹是关于源码的文件
 - bootstrap.css 、 bootstrap-grid.css 、 bootstrap-reboot.css 比较重要
 - bootstrap.css 包含 bootstrap-grid.css 和 bootstrap-reboot.css
 - bootstrap-grid.css 是栅格系统(flex)
 - bootstrap-reboot.css 是重置 css (重置默认样式)
- Containers
 - 屏幕断点: 默认(100%) | >=576px | >=768px | >=992px | >=1200px
 - container 居中，适配不同宽度的 max-width 尺寸。（版心）

container其他格式	描述
container-sm	和container断点相同
container-md	减少了 576 断点
container-lg	减少了 768 断点
container-xl	减少了 992 断点

- container-fluid ： 通栏(宽度 100%)

2. bootstrap-grid

- 要布局在 container 容器中

```
<div class="container">
  <div class="row">
    <div class="col-sm-6"></div>
    <div class="col-sm-6"></div>
  </div>
</div>
```

- row 是一行,行中可以有多列
- col 是一列,列后可以跟 sm , md , lg , xl ;决定了该列什么时候的宽度为什么
 - 只写 col ,列将不会有断点,任何分辨率下都会平分
 - 行的宽被分成 12 份,可以在列的断点后 跟数字决定分给这个列多大的宽度

- 不给列设置断点，列将会平分行的宽度
- 列默认带有左右 15px 边距
- `.w-100`：宽度 100%
- 根据内容自适应宽度
 - `col-auto` 或 `col-断点-auto`
 - `col-auto`：一直都是根据内容决定宽度
 - `col-断点-auto`：在断点之前(宽度 100% 之前)都是根据内容决定宽度
- 可以设置多个 `col-断点-长度` 来设置不同设备下,断点的宽

```
<div class="row g-0">
  <div class="col-md-6 col-lg-4 col-xl-3">aaa</div>
  <div class="col-md-6 col-lg-4 col-xl-3">aaa</div>
  <div class="col-md-6 col-lg-4 col-xl-3">aaa</div>
  <div class="col-md-6 col-lg-4 col-xl-3">aaa</div>
</div>
```

- `.row-cols-3`

```
<div class="row row-cols-3"> </div>
```

- 一行放三个列
- `row-cols-md-4`：在 768 之上 一行可以放三个列
- `row-cols-lg-4`：在992之上 一行可以放三个列
- 如果在一行内子 `div` 定义的栅格总数超过 12 列，Bootstrap会在保留列完整的前提下，将无法平行布局的多余列，重置到下一行，并占用一个完整的新行。
- 作用在行的 `class`

class	描述
<code>row-cols-3</code>	一行放三个列
<code>align-items-start</code>	行中元素贴上
<code>align-items-center</code>	行中元素居中
<code>align-items-end</code>	行中元素贴下
<code>justify-content-start</code>	行内元素贴左
<code>justify-content-center</code>	行内元素居中
<code>justify-content-end</code>	行内元素贴右
<code>justify-content-around</code>	元素间隔相等对齐
<code>justify-content-between</code>	两端对齐

- 作用在列中的属性

class	描述
col-auto	根据内容自适应宽度
align-self-start	在行内上部对齐
align-self-center	在行内居中对齐
align-self-end	在行内底部对齐
order-frist	将此列排序提升为第一
order-last	排在最后
order-1	数值越小,优先级越大 (取值为0-5) 如未定义排序优先级, 将不会改变位置
offset-md-3	控制间距, 决定列左边 空出几个栅

- margin: mr/ml/mt/mb/mx/my - auto|0~5 , padding同理
- 嵌套: 可以在 col 中嵌套 row (也是等分12份)

3. bootstrap-content(内容)

1. 将样式重置 -> 阅读原码 查阅重置样式

* 表单新属性hidden: `<input type="text" hidden>` 相当于 `display:none;`

2. 排版

- h1-h6 : 可以将其他标签字体大小和间距显示成 h1-h6 ,但不改变 display ;
 - `<p class="h1">这是测试</p>`
- display - 1~4 : 文字更醒目

4. bootstrap-Components(组件)

1. alert : `<div class="alert alert-primary">`
2. badge (徽章)

```
<div class="alert alert-success">
测试文本
<span class="badge badge-light">new</span>
</div>
```

- 徽章只是将容器变成合适的大小,需要通过 text-light \ bg-light 调控颜色

3. breadcrumb (导航)

```
<div class="bg-info"> <!-- 背景色 -->
  <ol class="breadcrumb px-5">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item"><a href="#">Library</a></li>
    <li class="breadcrumb-item active">Data</li>
  </ol>
</div>
```

4. button

```
<button type="button" class="btn btn-primary">Primary</button>
```

5. 按钮组(将多个按钮连接)

```
<div class="btn-group btn-group-lg">
  <button type="button" class="btn btn-secondary">Left</button>
  <button type="button" class="btn btn-secondary">Middle</button>
  <button type="button" class="btn btn-secondary">Right</button>
</div>
```

- btn-group-sm\lg: 添加在父元素,调控按钮大小 (只有 sm 和 lg 两种)

6. card (卡片): 图文描述 加 按钮

7. 滚动条: 滑动滚动条会实时滑动内容

- 安装 vscode 组件 bootstrap 4 ; 输入 b4 可以查看相关组件提示

5. Utilities(公共样式)

1. border :添加边框

- border : 默认1px浅灰色边框(#dee2e6)
- border - top\end\bottom\start - 0~5
- border - success/warning... : 设置颜色
- 圆角: rounded - circle/pill/sm/lg/top/bottom/left/right , 或只写 rounded

2. clearfix : 添加在父元素清除浮动

```
<div class="clearfix">
  <div class="float-left">float</div>
</div>
```

3. 对齐处理: text-center、text-end、text-start

4. 定

位: position-static、position-relative、position-absolute、position-fixed、position-sticky

5. 浮动: float-left、float-right、float-none

- float - sm\md\lg\xl - left\right\none

6. display : d-inline、d-block、d-inline-block、d-flex、

- d - sm\md\lg\xl - block

7. top/bottom/start/end - 0/50/100

- top:50% 定位使用, 只有三种值

8. overflow - auto\hidden

9. 响应式图片 `mw-100 + h-auto`

10. `text`bg``

- `text - primary\info\success...`
- `bg - primary\info\success...`

11. 宽和高

- `w - 25\50\75\100\auto` : 高同理

12. 隐藏文字: `text-hide`

- 更多查阅文档中文:<https://getbootstrap.net/>

6. 个人网页博客制作

- bootstrap 样板网页:<https://getbootstrap.com/docs/5.1/examples/>

7. Sass和Less

- Sass 和 Less 都属于 CSS 预处理器, CSS 预处理器定义了一种新的语言,其基本思想是,以一种专门的编程语言,为 CSS 增加了一些编程的特性,如:变量、语句、函数、继承等概念。将 CSS 作为目标生成文件,然后开发者就只要使用这种语言进行 CSS 的编码工作。
 - CSS 预处理器: Sass less stylus
 - less 官网: <http://lesscss.org/>
 - Less 中文: <http://lesscss.cn/>
 - VSCode 插件: Easy LESS
- Sass 官网: <https://sass-lang.com/>
- Sass 中文: <https://sass-lang.cn/>
- VSCode 插件: Easy Sass
 - 文件名.less -> 生成 CSS 文件
 - 文件名.scss -> 生成对应的 CSS 和 min.css 文件

8. Sass和Less的基本语法

- 注释
- 变量、插值、作用域
- 选择器嵌套、伪类嵌套、属性嵌套(Sass)
- 运算、单位、转义、颜色
- 函数
- 混入、命名空间(Less), 继承
- 合并、媒体查询
- 条件、循环
- 导入...

9.注释、变量、插值、作用域

- 注释: Less 和 Sass 单行注释都不会被编译,多行注释会被编译, sass 的 min.css 文件注释不编译
- 变量:

- Less: @变量名 : 值;

```
@number: 123px;
.box{
    width: @number;
}
```

- Sass: \$变量名: 值;

```
$number: 123px;
```

- 插值:类似字符串拼接

- Less: @{变量名}

```
@{key}: margin;
@i: 2;
.box@{i}{
    @{key}: auto;
}
.box2{
    margin: auto;
}
```

- Sass

```
#{$变量名}
$key:margin;
$i: 3;
.box#{i}{
    #{key}:10rem;
}
```

- 作用域:

- Less : 就近原则

```
@number: 200px;
.box{
    width: @number; // 100px;
    @number: 100px;
    margin: @number; // 100px;
}
```

- Sass : 先后顺序

```
$number: 200px;
.box{
    width: $number; // 200px;
    $number: 100px;
    margin: $number; // 100px;
}
```

10. Sass和Less选择器嵌套、伪类嵌套、属性嵌套(Sass)

- 选择器嵌套: Less和Scss同理

```
ul{
  list-style: none;
  li:first-child{
    text-align: left;
  }
  li{
    text-align: center;
    p{
      color: red;
    }
  }
}
```

◦ 生成

```
ul { list-style: none; }
ul li:first-child { text-align: left; }
ul li { text-align: center; }
ul li p { color: red; }
```

• 伪类嵌套：Less和Scss同理 `&:hover`

```
ul{
  &:hover{
    color: red;
    li{
      color: red;
    }
  }
}
```

◦ 生成

```
ul:hover { color: red; }
ul:hover li { color: red; }
```

• 属性嵌套：Sass 拥有

```
.box5{
  font : {
    size: 20px;
    weight:bold;
  };
  background : {
    color: red;
    repeat:repeat-x;
  };
}
```

◦ 生成

```
.box5 {  
  font-size: 20px;  
  font-weight: bold;  
  background-color: red;  
  background-repeat: repeat-x;  
}
```

11. Sass和Less运算、单位、转义、颜色

- 运算

- Less: `less` 中进行运算时,会以前面的单位为标准

```
@num : 100px;  
.box4{  
  width: @num * 3; // 300px  
  height: @num + 10em; // 110px  
  margin: 10em + @num; // 110em  
}
```

- **Sass中不同单位是不能运算的**

- 转义

- Less

```
padding: ~"20px / 1.5" // ~为拒绝转义字符,引号内的内容不会运算
```

- Scss

```
padding: (20px / 1.5) // '/'默认进行分割,使用小括号进行运算
```

- 颜色: Scss和Less同理

```
color: #010203 *2; // #020406; 颜色也会进行运算
```

- 以下情况Sass和Less可以运算 `2 + 1px`、`1px + 2`

12. Sass和Less函数

```
// Sass自定义函数  
@function sum($m,$n){  
  @return $m+$n;  
}  
sum(3px,2px)
```

13. Sass和Less混入、命名空间(Less)、继承

- 混入

- Less: `.类名{ .类名 }`


```
.show{ display: 'block'; }
.box{
  width: 90%;
  .show; // display:'block';
}
```

- 在标签名后加 `()` , `.show(){}` 之后,此标签样式不会被渲染

```
.hide(@color){
  color: @color;
}
.box{
  color : .hide(blue);
}
```

- Sass : `@mixin 类名{}`

```
@mixin show{
  display:block;
}
.box{
  @include show;
}
```

- Sass的混入不会渲染，可以传参

```
@mixin hide($color){
  display: none;
  color: $color;
}
.box{
  @include hide(blue);
}
```

- 命名空间(Less)： `#名称{ 类名{ } }`

```
.show{ width:20px; }
#nm(){
  .show(@h){
    width:10px;
    height: @h;
  }
}
.box{
  .show; // width: 20px;
  #nm.show(20px); // width:10px; height:20px;
}
```

- 继承

- Less: `&:extend(类名)`

```
.line{ display:inline; }
.box2{ &:extend(.line); }
.box3{ &:extend(.line); }
```

- 结果:

```
.line, .box2, .box3 {  
  display: inline;  
}
```

- Scss: @extend 类名

```
.line{ display: block; }  
.box6{ @extend .line; }  
.box7{ @extend .line; }
```

- 结果:

```
.line, .box6, .box7 {  
  display: block;  
}
```

- sass占位符 %, 此处样式不会被渲染

```
%line{ display: block; }  
.box7{ @extend %line; }
```

- 结果:

```
.box7 { display: block; }
```

14. Sass和Less合并、媒体查询

- 合并

- Less

```
.box9{  
  background+: url();  
  background+: contain;  
  transform+_: scale();  
  transform+_:translate();  
}
```

- 结果:

```
.box9 {  
  background: url(), contain;  
  transform: scale() translate();  
}
```

- 使用 + 会让属性 隔开, 使用 +_ 会让属性 空格 隔开

- Scss

```

$background:(
  a:url(),
  b:red
);
$transform:(
  a:scale(2),
  b:rotate(30deg)
);

.box9{
  background: map-values($background);
  transform: zip(map-values($transform)...);
}

```

■ 结果:

```

.box9 {
  background: url(), red;
  transform: scale(2) rotate(30deg);
}

```

• 媒体查询: Less和Scss一样

```

.box10{
  width:100px;
  @media all and (min-width:768px) {
    width: 200px;
  }
}
.box11{
  width:100px;
  @media all and (min-width:768px) {
    width: 200px;
  }
}

```

- 以上会生成单独的 @media ,和在Less中使用以下方式没有区别

```

@media all and (min-width:768px) {
  .box10{ }
  .box11{ }
}

```

15. Sass和Less条件、循环、导入

- 条件(尽量使用js,此条件不能实时渲染)
 - Less

```

@num:40;
.get(@cn) when ( @cn > 4 ){
    width: 100px + @cn;
}
.get(@cn) when (@cn <= 4){
    width: 10px + @cn;
}

.box12{
    .get(@num); -> 140px;
}

```

- 类似于 Less混入, 后面 when 相当于条件判断,判断成功就渲染 (Less混入也可以进行传参)
- 可以存在多个相同名的条件判断,组成 if...else

◦ Scss

```

$count:4;
.box12{
    @if($count > 4){
        width: 100px + $count;
    }@else if($count <=3 ){
        width: $count + 20px;
    }@else{
        width: $count + 10px;
    }
}

```

• 循环

◦ Less -> 通过 条件 + 递归

```

@count:0;
.get2(@cn)when(@cn < 3){
    .box-@{cn}{
        width: 10px + @cn;
    }
    .get2((@cn+1));
}
.get2(@count);

```

- 结果:

```

.box-0 { width: 10px; }
.box-1 { width: 11px; }
.box-2 { width: 12px; }

```

◦ Scss : 提供了 for 、 while 循环

```

@for $i from 0 through 2{ // 循环 0,1,2
    .box-#{ $i }{
        width: 10px + $i;
    }
}

```

- 结果:

```
.box-0 { width: 10px; }  
.box-1 { width: 11px; }  
.box-2 { width: 12px; }
```

- 导入：类似于导入模块，可以将其他 `.less`、`.scss` 文件内容,导入本文件中
 - Less和scss相同

```
@import '文件路径.less'
```