

1.CSS架构与文件组织

- 在一个大型项目中,由于页面过多,导致CSS代码难以维护和开发。所以CSS架构可以帮助我们解决文件管理与文件划分等问题。
- 首先要对CSS进行模块化处理,一个模块负责一类操作行为。利用Sass或Lass来实现。
- CSS架构：将不同功能的 .css 文件放入不同的文件夹

文件夹	含义
base	一些初始的通用CSS,如重置默认样式,动画,工具,打印等。 工具：浮动简单封装的 class ,清除浮动的 class 、简单定位的 class 等 打印：需要进行打印机打印的设备样式
components	用于构建页面的所有组件, 如按钮,表单,表格,弹窗等。
layout	用于页面布局的不同部分,如页眉、页脚、弹性布局、网格布局等。
pages	放置页面之间不同的样式, 如首页特殊样式,列表页特殊样式等 一个网站页面特别多,但是重复的样式很多, 这时只需要建立一个 .css 编写其中的特殊样式
themes	应用不同的主体样式时, 如管理员、买家、卖家等
abstracts	放置一些如 变量、函数,响应式等辅助开发的部分
vendors	放置一些第三方独立的CSS文件, bootstrap 、 iconfont 等

- 在文件夹内,建立对应的 css 文件 如 components ->(_button.scss , _form.scss , _table.scss ...), 最后通过 main.scss 将所有样式引入
- 还可以使用vue等框架开发

2.css新特性之 自定义属性

- CSS自定义属性(也称之为 ‘CSS变量’), 在目前所有现代浏览器中都得到了支持。
 - Less: @变量名：属性值
 - Sass: \$变量名：属性值
 - 原生CSS定义变量:

```
:root{
  --color:red; /* --属性名 : 值; */
}
.box{
  /* 标签中也可以定义变量 */
  --color : blue;
  color: var(--color); /* blue , 看作用域*/
}
```

- `:root` 相当于文档根元素 `html`
- 老式浏览器不支持 自定义属性的,可以使用 `PostCss` 中 `postcss-cssnext` 进行降级

◦ 计算:

```
:root{
  --number: 100;
}
.box{
  width: calc( var(--number) * 1px )
}
```

◦ 默认值

```
.box{
  width: var(--wid, 100px);
}
```

- 当前面的变量找不到时, 会使用后面的值 (后面的值不能为变量,但是可以为 `var(--变量)`) `color: var(--c,var(--cc));`

- **只能拥有一个备用值**

◦ 作用域: 按照会影响到标签的优先级

- `style` 行间 > `id` > `class` > `tag` (标签) > `*` > 继承
 - `Less` 是就近原则 `Sass` 是先后顺序

1.

```
:root{
  --color: blue;
}
.box{
  color: var(--color);    /* red */
  --color: red;
}
```

2.

```
:root{
  --color: blue;
}
div{
  --color: gray;
}

.box{
  color: var(--color);
}
```

- `class` 为 `box` 的 `div` 颜色为 `gray`,其他的为 `blue`

◦ 在 `:root` 中声明的一些样式, 是默认最高级别的(需要是可继承的样式)

```
:root{
  font-size:12px; /* 在不覆盖的情况下, 所有标签默认字体大小都是12px */
}
```

3.CSS新特性之 shapes

- CSS shapes 布局可以实现不规则的文字环绕效果,需要 配合浮动使用
- 布局

```
<div class="container">
  <div class="float"></div>
  文本文本文本
</div>
```

- Shape 共有两种属性：
 - shape-outside 让文本围在图形外，和 shape-margin 属性一起使用
 - shape-inside 把文本包装在形状的内部，和 shape-padding 一起使用
- 先给 .float 设置 border-radius:50%;
 - 给 .float 设置浮动, 然后通过样式, 控制 外部文本的环绕方式
- shape-outside: 决定 外部文字环绕方式

shape-outside属性值	描述
margin-box	以 margin 区域以外环绕(默认)
border-box	以 border 以外区域环绕
padding-box	以 padding 以外的区域环绕
content-box	以 内容 以外的区域进行环绕
polygon(x1 y1, x2 y2, x3 y3)	多边形， 首尾两点自动相连
circle(i [at x y])	半径长为 i 的圆形， 可以指定圆心位置 x y； 不指定位置默认容器中央为圆心
ellipse(x y [at x2 y2])	椭圆， x 轴半径为 x， y 轴半径为 y 的椭圆； 可以指定圆心位置 x, y； 不指定位置默认容器中央为圆心
inset(10px 20px round 5px)	距离外部盒子的上，下各 10px 距离外部盒子的左，右各 20px 还有半径为 5px 的圆角

- CSS shapes 通过参考盒来定义并且创建,这个盒子用来绘制元素上的形状。默认情况下,使用 margin box 盒模型作为参考,也可以使用 padding-box、border-box、content-box；
 - 简述: 默认使用 margin-box 为画板, margin-box 左上角为 0,0；
 - 图形盒子: 分别是 margin-box，border-box，padding-box 和 content-box。要来指定文字环绕的时候是依照哪个盒子的边缘来计算的。
 - 基本图形函数: 如 circle(), ellipse() 等
 - 图像类: URL 链接图片(要具有透明度的 png), 渐变图像, cross-fade(), element() 等

- 图像类需要服务器环境

- 格式: `shape-outside: circle(50px at 20px) padding-box;`

- `shape-image-threshold: .3;`

- 决定什么透明度之内的值,可以环绕文字

- `.5` `50%` 之内透明度的值可以环绕文字

- `clip-path`: 将当前容器, 剪切, 被裁去的部分将隐藏, 不会改变容器大小

取值	描述
<code>circle(i [at x y])</code>	半径长为 <code>i</code> 的圆形, 可以指定圆心位置 <code>x</code> , <code>y</code> ; 不指定位置默认容器中央为圆心
<code>polygon(x1 y1, x2 y2, x3 y3)</code>	多边形, 首尾两点自动相连
<code>ellipse(x y [at x2 y2])</code>	与 <code>shape-outline: ellipse()</code> 相同
<code>inset(10px 20px round 5px)</code>	距离外部盒子的上, 下各 <code>10px</code> 距离外部盒子的左, 右各 <code>20px</code> 还有半径为 <code>5px</code> 的圆角

- 通过以下设置 `shape-outside` 和 `clip-path` , 可以达到 文字环绕容器的效果

```
shape-outside: polygon(0 0,0 100px, 100px 100px);
clip-path: polygon(0 0,0 100px, 100px 100px);
```

- `shape-margin`: 调节容器 和 外部文字之间的间隙

4.CSS新特性之scrollbar (伪元素)

- CSS `scrollbar` 用于实现自定义滚动条样式。

伪元素	描述
<code>::-webkit-scrollbar</code>	定义滚动条宽度
<code>::-webkit-scrollbar-thumb</code>	定义滑块样式
<code>::-webkit-scrollbar-track</code>	定义滑块轨道部分样式

- 伪元素需要给父容器添加, `body` 高 `2000px` , 给 `html` 添加伪元素

- `::-webkit-scrollbar` : 定义滚动条宽度, 横向时使用 `height`

```
body::-webkit-scrollbar{
  width: 10px;
  /* height: 1px; 横向时使用height*/
}
```

- `::-webkit-scrollbar-thumb` : 定义滑块样式

```
body::-webkit-scrollbar-thumb{
  background: red;
  border-radius:100vw;
}
```

- `::-webkit-scrollbar-track` : 定义滑块轨道部分样式

```
body::-webkit-scrollbar-track{
  background: gray;
  box-shadow: inset -1px -1px 10px black;
}
```

- 还可以配合伪类使用

```
body::-webkit-scrollbar-track:hover{}
body::-webkit-scrollbar-track:active{}
```

- 滚动条还有其他的 不常用伪元素,可以查阅资料

5.CSS新特性 scrollsnap

- CSS *ScrollSnap*(CSS滚动捕捉), 允许你在用户完成滚动后 锁定特定的元素或位置。
- `scroll-snap-type` : 添加给父容器, 决定捕捉模式和捕捉轴, 父容器里面的内容需要溢出,产生滚动条才会生效

```
scroll-snap-type: x/y轴 mandatory/proximity;
/* 第一个参数: 捕捉的轴, 第二个参数: mandatory(精准捕捉)/proximity(非精准捕捉)*/
scroll-snap-type: x mandatory;
```

- `scroll-snap-align` : 添加给子元素, 决定子元素被捕捉后, 与父元素的对齐方式

```
scroll-snap-align: center;
/* 可选值三个: start center end 分别于父元素 起始处对齐, 居中对齐, 结束对齐 */
```

- `scroll-padding` : 添加给父元素, 决定 捕捉后距离父元素的 `padding`, 使其具有一定边界
 - 还有 `scroll-padding-left/top/right/bottom` 等
- `scroll-margin` : 添加给子元素, 决定 捕捉后距离父元素的 `margin`, 使其具有一定边界和 `scroll-padding` 差不多

6.css与js结合之 钟表

7.css与js结合之 折叠菜单