

网易微专业之《前端开发工程师》

学习笔记

开始时间：2016.2.26

《页面架构》

响应式布局

响应式：网站自动适配不同的设备

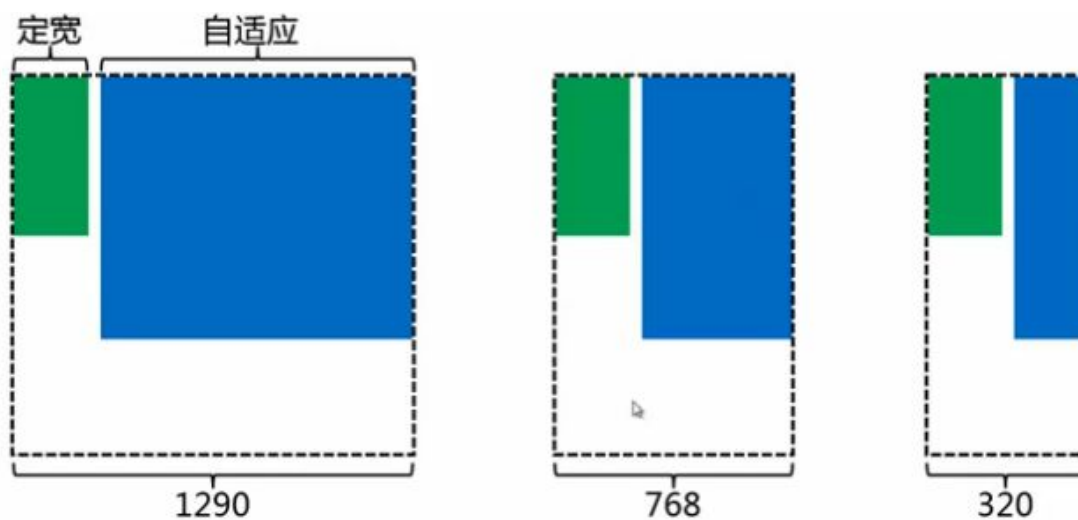
- 优点：不用为不同的设备定制不同的网站，开发成本、人力及维护成本低；
- 缺点：资源冗余（对于手机等设备时）

Viewport 视窗

```
<meta name="viewport" content="
    width=device-width
    ,initial-scale=1.0
    ,user-scalable=no
"/>
```



自适应布局

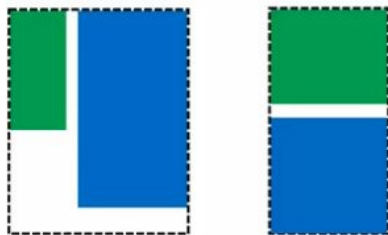


@media 媒体查询

本笔记由西风潇潇编写，欢迎浏览博客访问更多内容：<http://www.xifengxx.com>

```
@media screen and (max-width:320px){
    /* 视窗宽度 <= 320px */
}
@media screen and (min-width:769px){
    /* 视窗宽度 >= 769px */
}
@media screen and (min-width:769px) and (max-width:1000px){
    /* 769px <= 视窗宽度 <= 1000px */
}
```

```
<div class="parent">
  <div class="left">
    <p>left</p>
  </div>
  <div class="right">
    <p>right</p>
    <p>right</p>
  </div>
</div>
```



```
.left{
    float: left;
    width: 100px;
    margin-right: 20px;
}
.right{
    overflow: hidden;
}
@media screen and (max-width:320px){
    .left{
        float: none;
        width: auto;
        margin-right: 0;
        margin-bottom: 20px;
    }
}
```

参考资料：

移动端 H5 页面高清多屏适配方案：

<http://www.cocoachina.com/webapp/20150715/12585.html>

页面优化

为什么要优化？

- 提升网页响应速度
- 对搜索引擎、屏幕阅读器友好
- 提高可读性，可维护性

如何优化：

- 减少请求
- 减少文件大小
- 提升页面性能
- 可读性、可维护性

减少请求

- 图片合并
- 减少 CSS 文件合并

图片合并

通过 `sprite`，把小图标合并成一张图片。

减少 CSS 文件合并

- 多个 CSS 文件合并为一个
- 少量 CSS 样式内联
- 避免使用 `import` 的方式引入 `css` 文件

```
@import "base.css";  
@import "grid.css";  
@import "module.css";  
@import "unit.css";
```

每个 `import` 都会产生一个请求，而且请求是同步的，第一个请求完成后才会进行第二个请求。

减少文件大小

1. 减少图片大小

选择合适的图片格式：PNG / JPG

压缩图片

- ImageOptim
- ImageAlpha
- JPEGmini

2. CSS 值缩写

margin/padding/border/border-radius/font/background

```
p{  
  margin-top: 5px;  
  margin-right: 10px;  
  margin-bottom: 5px;  
  margin-left: 10px;  
}
```



```
p{  
  margin: 5px 10px;  
}
```

3. 省略值为 0 的单位

```
.m-box{  
  margin: 0px 10px;  
  background-position: 50% 0%;  
  line-height: 0.5  
}
```

```
.m-box{  
  margin: 0 10px;  
  background-position: 50% 0;  
  line-height: .5  
}
```

4. 颜色值最短表示

- red
- rgb(255,255,0)
- rgba(0,0,0,0.5)
- #333

```
.m-box{  
  color: #ccc;  
}
```

5. CSS 选择器合并

本笔记由西风潇潇编写，欢迎浏览博客访问更多内容：<http://www.xifengxx.com>

CSS 雪碧图：

```
.m-logo, .m-help, .m-list li, .u-tab li, .u-tab li a{  
    background: url(../images/sprite.png) no-repeat 9999px 9999px;  
}
```

清除浮动：

```
.g-hd:after, .g-bd:after, .g-ft:after{  
    display: block;  
    visibility: hidden;  
    clear: both;  
    height: 0;  
    content: ".";  
}
```

6. 文件压缩

- YUI Compressor
- cssmin

提升页面性能

1.加载顺序

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  ...
  <link rel="stylesheet" href="index.css">
</head>
<body>
  <div>...</div>
  <script src="index.js"></script>
</body>
</html>
```

2.减少标签数量

```
<div class="list">
  <ul>
    <li>一</li>
    <li>二</li>
    ...
  </ul>
</div>
```



```
<ul class="list">
  <li>一</li>
  <li>二</li>
  ...
</ul>
```

3.选择器长度

```
<div class="m-nav">
  <ul>
    <li><a href="">导航一</a></li>
    <li><a href="">导航二</a></li>
  </ul>
  <div>...</div>
</div>
```

.m-nav ul li a:hover{color:red;} → .m-nav a:hover{color:red;}

4.耗性能属性

- expression

```
.class{width: expression(this.width>100?'100px':'auto');}
```

- filter

```
.class{filter: alpha(opacity=50);}
```

- border-radius

- box-shadow

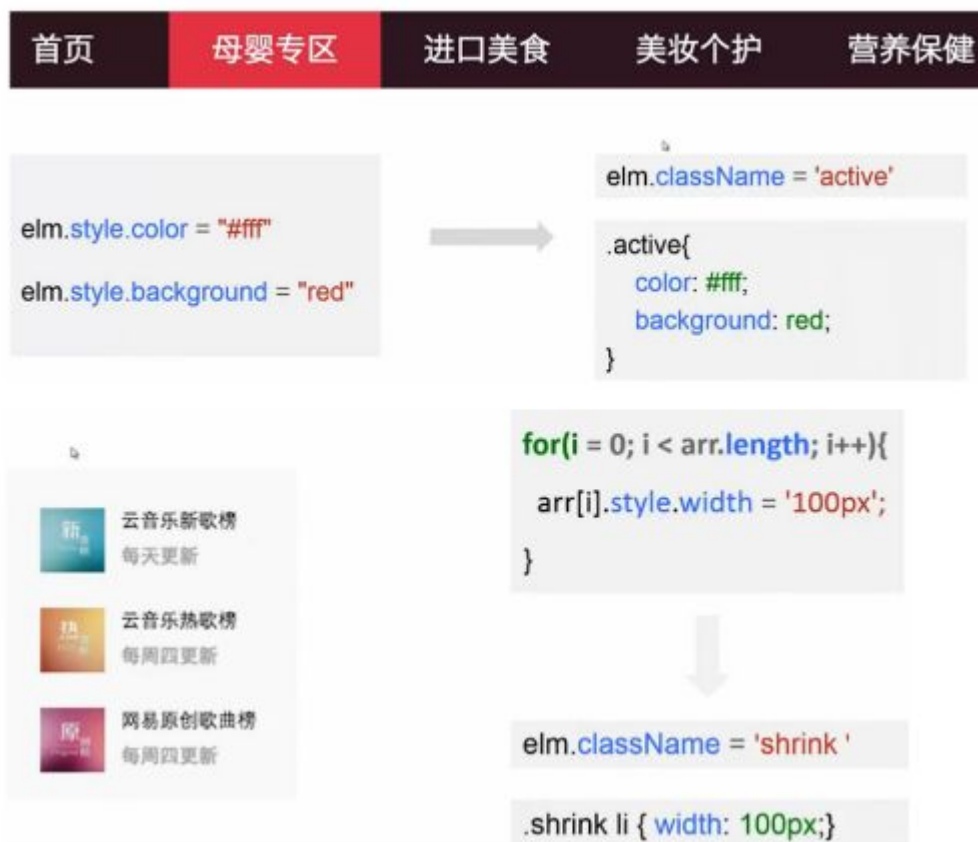
- gradients

5. 图片设置宽高

```

```

6. 所有表现用 CSS 实现



用 JS, 每次都需要重绘列表, 而使用 CSS, 可以把多次渲染合并成一次渲染。

可读性、可维护性

- 规范
- 语义化
- 尽量避免 Hack（IE7 用*号，IE6 用_下划线）
- 模块化
- 注释

规范与模块化

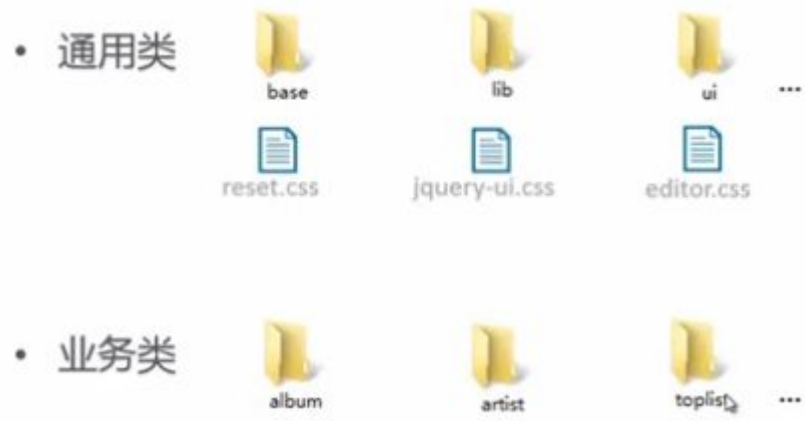
规范



文件规范

1、文件分类

根据文件的用处分类归在不同的文件夹里；



2、文件引入

- 行内样式（不推荐）

- 外部引入
- 内联样式
- 避免使用@import 引入样式

3、文件本身

- 文件名：以小写字母、数字和下划线组成，以字母开头，下划线分隔；
- 统一编码；UTF-8

注释规范

- 块状注释
 - 统一缩进
 - 在被注释对象之上

```
/* 块状注释文字
 * 块状注释文字
 * 块状注释文字
 */
.m-list{width:500px;}
...
```
- 单行注释
 - 文字两端需空格
 - 在被注释对象之上

```
/* 单行注释文字 */
.m-list li em{color:#666;}
```
- 行内注释
 - 文字两端需空格
 - 在分号之后

```
.m-list li a{color:#333;}/* 对color的行内注释 */}
```

命名规范

1、分类命名： 全局还是模块内

```
<!-- demo -->
...
<div class="g-header">...</div>
<div class="g-section">
  <div class="article">
    <div class="header">...</div>
    ...
  </div>
</div>
<div class="g-footer">...</div>
...
```

/* 布局样式 */
.header{color:black;}

/* 文章样式 */
.article{color:blue;}
.article .header{font-size:12px;}

样式污染

```
<!-- demo -->
...
<div class="g-header">...</div>
<div class="g-section">
  <div class="m-article">
    <div class="header">...</div>
    ...
  </div>
</div>
<div class="g-footer">...</div>
...
```

/* 布局样式 */
.g-header{color:black;}

/* 文章样式 */
.m-article{color:blue;}
.m-article .header{font-size:12px;}

2、命名格式：小写

- 大小写

建议：小写

```
/* 大写选择器 */  
.LIST{font-size:12px;}  
.LIST .TITLE{color:#333;}
```

```
/* 小写选择器 */  
.list{font-size:12px;}  
.list .title{color:#333;}
```

- 长度

权衡长度和可读性

```
.subnavigator{font-size:12px;}
```

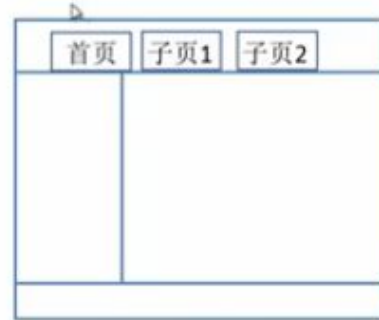
```
/* 缩写名称 */  
.subnav{font-size:12px;}
```

```
/* 再缩写 */  
.sbnav{font-size:12px;}
```

3、语义化命名

• 以内容语义命名

```
/* 以结构命名 */  
.top{font-size:12px;}  
.top .link{color:#333;}  
  
/* 以内容语义命名 */  
.nav{font-size:12px;}  
.nav .link{color:#333;}
```



四、书写规范

1、单行与多行

```
/* 单行 */  
.logo{width:200px;height:50px;font-size:12px;color:#666;}  
.logo a{float:left;padding:0 10px;}  
  
/* 多行 */  
.logo{  
    width:200px;  
    height:50px;  
    font-size:12px;  
    color:#666;  
}  
.logo a{  
    float:left;  
    padding:0 10px;  
}
```

二选一 遵守之

2、空格与分号

• 空格

- 缩进（必须有） 2个 or 4个
- 规则内空格

```
/* css规则内空格 */  
.logo{width: 200px; height: 50px; color: #666;}
```

不要省略

• 分号

- 保留最后一个属性值的分号

3、属性书写顺序

- 根据属性的重要性按顺序书写

→	显示属性	自身属性	文本属性和其他修饰
	display	width	font
	visibility	height	text-align
	position	margin	text-decoration
	float	padding	vertical-align
	clear	border	white-space
	list-style	overflow	color
	top	min-width	background

4、关于 Hack

- 统一各浏览器的 Hack 方式

IE6	_property:value	/* IE7显示#888, IE6显示#fff, 其他浏览器显示#000 */
IE6/7	*property:value	.m-list{color:#000;*color:#888;_color:#fff;}

- 不要滥用 Hack

5、统一属性值格式

- color: white or #fff or rgb(255,255,255)
- url(): icon.png or "icon.png" or 'icon.png'

五、HTML 规范

• 文档声明	如 “<!DOCTYPE html>” ，且首行顶格开始
• 闭合	闭合标签： <code><div></div></code> 自闭合标签： <code><input></code> or <code><input /></code>
• 属性	<code><h1 class="logo"></h1></code> or <code><h1 class='logo'></h1></code> <code><input readonly></code> or <code><input readonly="readonly"></code>
• 层级	用缩进体现层级，提高可读性 标签正确嵌套，但嵌套不宜太深
• 注释	<code><!-- LOGO --></code> <code><h1 class="logo">LOGO</h1></code> <code><!-- /LOGO --></code>
• 大小写	标签、属性均小写

六、图片规范

1.文件名称：

- 语义
- 长度

2.保留源文件

- 保留图片处理前的源文件，所有修改都在源文件上操作并导出

3.图片合并

- 尽可能使用 `sprite` 技术
- `sprite` 图片可按模块、业务、页面来划分

模块化

什么是模块化？

- 一系列相关联的结构组成的整体
- 带有一定语义，而非表现。

如下图中的分页器、图片轮播图，都可以成为一个模块。



怎么做？

- 为模块分类命名（如.m-,.md-）
- 以一个主选择器开头（模块根节点）
- 使用以主选择器开头的后代选择器（模块子节点）

```
<!-- 模块1 -->
<模块1>
  <子元素1></子元素1>
  <子元素1></子元素1>
</模块1>

<!-- 模块2 -->
<模块2>
  <子元素1></子元素1>
  <子元素1></子元素1>
</模块2>
```

```
/* 模块1 */
模块1{}
模块1 子元素1{}
模块1 子元素2{}

/* 模块2 */
模块2{}
模块2 子元素1{}
模块2 子元素2{}
```

案例：

```
<!-- NAV MODULE -->
<div class="m-nav">
  <ul>
    <li class="z-crt"><a href="#">index</a></li>
    <li><a href="#">link1</a></li>...
  </ul>
</div>
<!-- /NAV MODULE -->
```

```
/* 导航模块 */
.m-nav{ /* 模块容器 */
.m-nav li,.m-nav a{}
.m-nav li{}
.m-nav a{}
.m-nav .z-crt a{ /* 交互状态变化 */
```

```
<!-- NAV-1 MODULE -->
<div class="m-nav m-nav-1">
  <ul>
    <li class="z-crt"><a href="#">index</a></li>
    <li><a href="#">link1</a></li>...
  </ul>
  <a class="btn">Login</a>
</div>
<!-- /NAV-1 MODULE -->
```

```
/* 导航模块扩展 */
.m-nav-1{}
.m-nav-1 a{border-radius:5px;}
.m-nav-1 .btn{}
```

为什么要模块化？



```
<div class="m-slides">
  <ol>
    <li class="slide z-crt"></li>
    <li class="slide"></li>
    <li class="slide"></li>
    <li class="slide"></li>
  </ol>
  <span class="ctrl"><i class="z-crt"></i><i></i><i></i><i></i><i></i></span>
</div>

/* 未模块化样式 */
.slides{}
.slide{}
.slide img{}
li.z-crt{}
.ctrl{}
.ctrl i{}
i.z-crt{}

/* 未模块化样式 */
.m-slides{}
.m-slides .slide{}
.m-slides .slide img{}
.m-slides li.z-crt{}
.m-slides .ctrl{}
.m-slides .ctrl i{}
.m-slides .ctrl i.z-crt{}

```

模块化的好处

- 利于多人协同开发
- 便于扩展和重用
- 可读性、可维护性好
- 不会引起样式污染

参考资源：

<http://nec.netease.com/standard>

网易内部的 CSS 规范以及模块化方案——NEC：

规范页：包含了 CSS 规范、HTML 规范和工程师规范；

代码库：包含了常用的布局方式、常见模块和元件的实现以及一些 bug、技巧等；

帮助你对页面的规范和模块有进一步认识。