



CSS分层与面向对象

CSS分层 — 课程概要

- 为什么要分层?
- SMACSS
- BEM
- SUIT
- ACSS
- ITCSS

为什么要分层?

为什么要分层？

原因：

- CSS有语义化的命名约定和CSS层的分离，将有助于它的可扩展性，性能的提高和代码的组织管理。
- 大量的样式，覆盖、权重和很多!important，分好层可以让团队命名统一规范，方便维护。
- 有责任感地去命名你的选择器。



SMACSS

SMACSS(Scalable and Modular Architecture for CSS 可扩展的模块化架构的CSS)像OOCSS一样以减少重复样式为基础。然而SMACSS使用一套五个层次来划分CSS给项目带来更结构化的方法：

- Base -设定标签元素的预设值 `PS:html {} input[type=text] {}`
- Layout -整个网站的「大架构」的外观 `PS:#header { margin: 30px 0; }`
- Module -应用在不同页面公共模块 `PS:.button{}`
- State -定义元素不同的状态 `PS:.nav--main { .active {} }`
- Theme -画面上所有「主视觉」的定义 `PS: border-color、background-image`

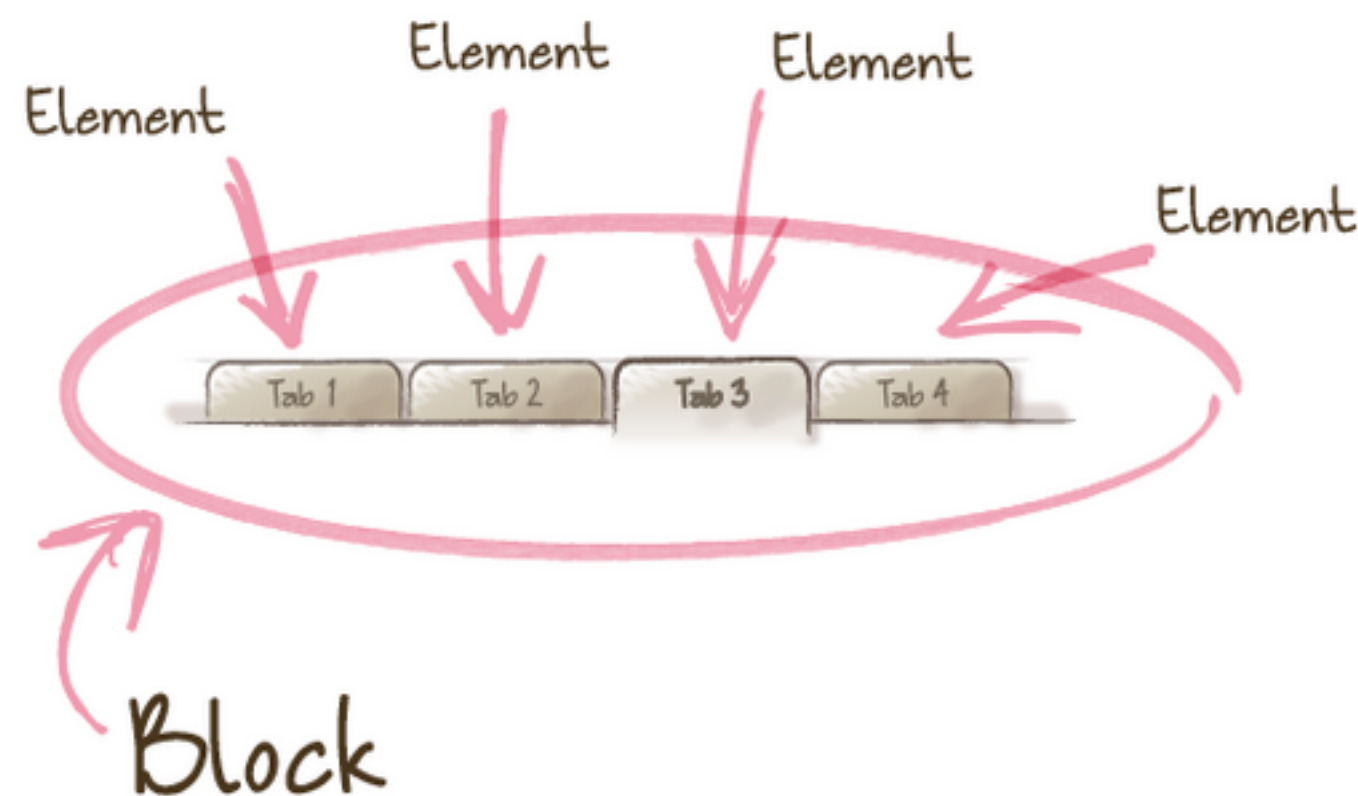
修饰符使用的是--，子模块使用__符号。

```
<div class="container">
  <div class="container-header">
    <div class="container-header__title">
      <h1 class="container-header__title--home">
```

BEM

BEM和SMACCS非常类似，主要用来如何给项目命名。一个简单命名更容易让别人一起工作。比如选项卡导航是一个块(Block)，这个块里的元素的是其中标签之一(Element)，而当前选项卡是一个修饰状态(Modifier)：

- block -代表了更高级别的抽象或组件
- block__element -代表.block的后代，用于形成一个完整的.block的整体。
- .block--modifier -代表.block的不同状态或不同版本。
- 修饰符使用的是_，子模块使用__符号。(不用一个-的原因是CSS单词连接)



```
<ul class="menu">
  <li class="menu__item">...</li>
  <li class="menu__item_state_current">...</li>
  <li class="menu__item">...</li>
</ul>
```


SUIT

Suit起源于BEM，但是它对组件名使用驼峰式和连字号把组件从他们的修饰和子孙后代中区分出来：

- 修饰符使用的是—，子模块使用__符号。(不用一个-的原因是CSS单词连接)

```
.u-utility {}  
.ComponentName {}  
.ComponentName--modifierName {}  
.ComponentName-descendantName {}  
.ComponentName.is-someState {}
```



```
.ProductDetails {}  
.ProductDetails-price {}  
.ProductDetails-title--sale {}
```

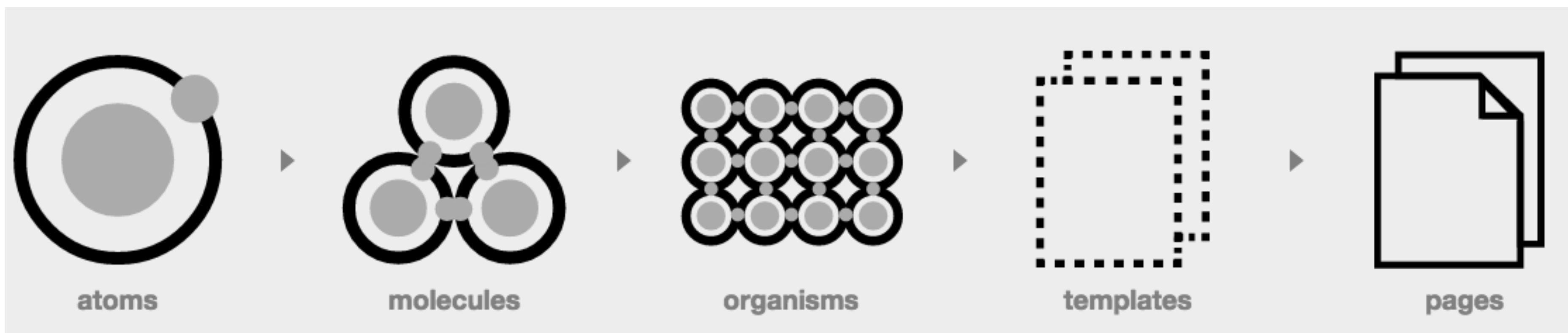
如果你不想使用如此严格或者复杂的命名规则，给每一个选择器加前缀同样可以达到这样的效果。

```
.s-product-details {}  
.t-product-details {}  
.js-product-details {}
```

结构属性将会被应用到s-product-details选择器中。主题属性将应用于t-product-details选择器。

ACSS

- 考虑如何设计一个系统的接口。原子(Atoms)是创建一个区块的最基本的特质，比如说表单按钮。分子(Molecules)是很多个原子(Atoms)的组合，比如说一个表单中包括了一个标签，输入框和按钮。生物(Organisms)是众多分子(Molecules)的组合物，比如一个网站的顶部区域，他包括了网站的标题、导航等。而模板(Templates)又是众多生物(Organisms)的结合体。比如一个网站页面的布局。而最后的页面就是特殊的模板。



```
.m-10 {  
  margin: 10px  
}  
  
.w-50 {  
  width: 50%;  
}
```


ITCSS

- Settings — 全局可用配置， 设置开关。 `$color-ui: #BADA55; $spacing-unit: 10px`
- Tools — 通用工具函数。 `@mixin font-color() {font-color: $color-ui;}`
- Generic — 通用基础样式。 `Normalize, resets, box-sizing: border-box;`
- Base — 未归类的HTML元素。 `ul {list-style: square outside;}`
- Objects — 设计部分开始使用专用类。 `.ui-list__item {padding: $spacing-unit;}`
- Components — 设计符合你们的组件。
`.products-list { @include font-brand(); border-top: 1px solid $color-ui; }`
- Trumps — 重写， 只影响一块的DOM。(通常带上我们的!important)

