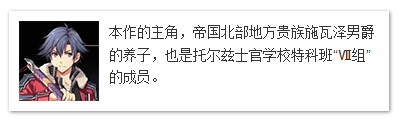
**OOCSS**

OOCSS（Object Oriented CSS），是面向对象的CSS，是由Nicole Sullivan提出的css理论，旨在编写高可复用、低耦合和高扩展的CSS代码。其主要的两个原则是：

    Separate structure and skin（分离结构和样式）  
    Separate container and content（分离容器和内容）

用一个例子来说明。请看下面这样的图文排列：  


<div class="media media-shadow">

<div class="media-image-container">

<img class="media-image" src="rean.jpg" alt="">

</div>

<div class="media-body">

<p class="media-text">本作的主角，帝国北部地方贵族施瓦泽男爵的养子，也是托尔兹士官学校特科班“Ⅶ组”的成员。</p>

</div>

</div>

.media{

padding: 10px;

}

.media:after{

display: table;

clear: both;

content: " ";

}

.media-image-container{

float: left;

margin-right: 10px;

}

.media-image{

display: block;

}

.media-body{

overflow: hidden;

}

.media-shadow{

box-shadow: 1px 1px 3px rgba(0, 0, 0, .5);

}

上面这段代码用media表示了这种图文排列的页面元素。如果把构成它的html、css及javascript（如果有）看做一个整体，那就相当于这是一个元件，或者说对象（object）。它可以在站点的任何地方被重用。

这样是如何体现OOCSS的两个原则的呢？  
**Separate structure and skin**

分离结构和样式是在于将一些视觉样式效果（例如background、color）作为单独的“样式”来应用。在上面的例子中的阴影效果，没有被直接写在media的样式规则内，而是被单独写在了一个名为media-shadow的class中。因此，它成为了可选择、可拆分的样式。如果不需要对应样式，什么也不要加，如果需要，加上对应的class，就是这样的思路。  
**Separate container and content**

分离容器和内容要求使页面元素不依赖于其所处位置。在上面的例子中，css的选择符都很短，无继承选择符（例如.header .media { }），所以，这个图文排列的元件，可以在任何地方使用，且会有一致的外观。

如果需要在特定的地方让这个元件看起来不一样一些，继续为这个元件增加class，将“不一样的部分”作为可配置的选项。元件的外观仍不依赖其所处位置。

**操作指南**

可以看出，OOCSS风格的css可以描述为两点：

    增加class，追求class的复用性  
    不使用继承选择符

OOCSS追求元件的复用，其class命名比较抽象，一般不体现具体内容。

**BEMCSS**

BEM是由Yandex团队提出的一种CSS Class 命名方法，旨在帮助开发者实现模块化、可复用、高可维护性和结构化的CSS代码。BEM的意思就是块（block）、元素（element）、修饰符（modifier）



官方是这么描述它的：

BEM是一种让你可以快速开发网站并对此进行多年维护的技术。

以往开发组件，我们都用“重名概率小”或者干脆起个“当时认为是独一无二的名字”来保证样式不冲突，这是不可靠的。

理想的状态下，我们开发一套组件的过程中，我们应该可以随意的为其中元素进行命名，而不必担心它是否与组件以外的样式发生冲突。

BEM解决这一问题的思路在于，由于项目开发中，每个组件都是唯一无二的，其名字也是独一无二的，组件内部元素的名字都加上组件名，并用元素的名字作为选择器，自然组件内的样式就不会与组件外的样式冲突了。

这是通过组件名的唯一性来保证选择器的唯一性，从而保证样式不会污染到组件外。

这也可以看作是一种“硬性约束”，因为一般来说，我们的组件会放置在同一目录下，那么操作系统中，同一目录下文件名必须唯一，这一点也就确保了组件之间不会冲突。

BEM的命名规矩很容易记：

block-name\_\_element-name--modifier-name，

也就是模块名 + 元素名 + 修饰器名。

**Block**

使用简洁的前缀名字来命名一个独立且有意义的抽象块或组件。

e.g.

.block

.header

.site-search

**Element**

使用\_\_连接符来连接Block 和 Element。

e.g.

.block\_\_element

.header\_\_title

.site-search\_\_field

**Modifier**

使用--连接符来连接Block 或 Element 和 Modifier。

e.g.

.block--modifier

.block\_\_element--modifier

.header--hide

.header\_\_title--color-red

.site-search\_\_field--disabled

BEM禁止使用子代选择器，BEM中修饰器的样式不依赖于任何结构关系，也就是说，元素的状态改变只会影响自身，不对其他元素进行影响

所以，BEM（或BEM的变体）是一个非常有用，强大，简单的命名约定，以至于让你的前端代码更容易阅读和理解，更容易协作，更容易控制，更加健壮和明确而且更加严密。

二者的区别：

两者都禁止采用子代选择器，使用目的是为了避免改变结构及样式覆盖等问题

BEM主要是针对class命名唯一性

OOCSS主要是提高class的复用性