京程一灯

www.yidengxuetang.com

面向对象的CSS

面向对象的CSS— 课程概要

- OO CSS的概念解读
- OO CSS的作用和注意事项
- OO CSS代码实战

OO CSS的概念解读

OO CSS的概念解读

众多开发者忽视了CSS的表现（认为他太过简单，是一种机械的工作），而把更多关注在Javascript的性能上或者其他方面。

OO CSS将页面可重用元素抽象成一个类，用Class加以描述，而与其对应的HTML即可看成是此类的一个实例。

OO CSS的作用和注意事项

OO CSS的作用和注意事项— 作用

- 1.加强代码复用以便方便维护。
- 2.减小CSS体积。
- 3.提升渲染效率。
- 4.组件库思想、栅格布局可共用、减少选择器、方便扩展。

OO CSS的作用和注意事项— 注意事项

注意事项:

- 1.不要直接定义子节点，应把共性声明放到父类。
- 2.结构和皮肤相分离。
- 3.容器和内容相分离。
- 4.抽象出可重用的元素，建好组件库，在组件库内寻找可用的元素组装页面。
- 5.往你想要扩展的对象本身增加class而不是他的父节点。
- 6.对象应保持独立性。
- 7.避免使用ID选择器，权重太高，无法重用。
- 8.避免位置相关的样式。
- 9.保证选择器相同的权重。
- 10.类名 简短 清晰 语义化 OOCSS的名字并不影响HTML语义化。

OO CSS的作用和注意事项— 注意事项

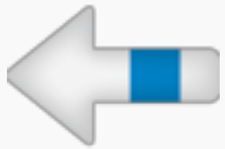
代码示例：

```
.mod .inner{……}          //.mod 下面的inner
.inner{……}               //不是很建议的的声明
```

OO CSS的作用和注意事项— 注意事项

代码示例：

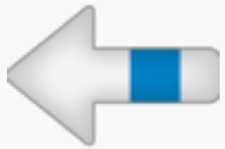
```
<div class="container simpleExt"></div>     //html 结构
.container {……}      //控制结构的class
.simpleExt{……}        //控制皮肤的class
```



OO CSS的作用和注意事项— 注意事项

代码示例：

```
<div class="container"><ul><li>排行</li></ul></div>     //html 结构
.container ul{……}      //ul依赖了容器
<div class="container"><ul class="rankList "><li>排行</li></ul></div>
.rankList ul{……}      //解除与容器的依赖，可以从一个容器转移到其他容器
```



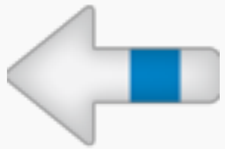
OO CSS的作用和注意事项— 注意事项

代码示例：

```
<div class="container"><div class="mod"></div></div>     //html 结构

.container {……} .container .mod{……} //控制结构的class

<div class="container mod"></div>
```

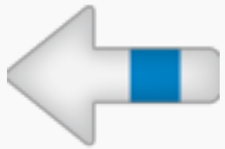


OO CSS的作用和注意事项— 注意事项

代码示例：

```
#header .container {……}, #footer .container{……}
.container{}
#header h1 {……}, #footer h1 {……}
```

h1,.h1{} h2,.h2{} <h1><class=“h6”></h1>



OO CSS的代码实战

面向对象的CSS

这本套课程中我们学习了OO CSS的具体概念。你应当掌握了以下知识：

- 1.OO CSS的概念解读。
- 2.OO CSS的作用和注意事项。
- 3.OO CSS的代码实战。

你可以使用这些技巧来制作改良一下你的页面，如果想继续提高，你可以继续在极客学院学习CSS在工程中的变化系列课程。