**什么是面向对象的 CSS**

框架？工具？哲学？

Object-oriented CSS is a coding paradigm that styles “objects” or “modules”—nestable  
chunks of HTML that define a section of a webpage—with robust, reusable styles.

**很像语言的进化**

令 CSS 更强大

**有什么不同？**

ul{...}

ul li{...}

ul li a(②但是，结构在这里){①直至现在，我们只关心这里}

**CSS 大约 2005**

意大利面条

**CSS 大约 2008**

稍微好一点

**而不是使我们的代码变好**

我们筑了大栅栏

**性能呢？**

公认的毒药中心

**网站变慢**

文件大小和 HTTP 请求未作优化

**CSS 大约 2009**

**面向对象的CSS**

性能的最佳实践、可扩展性、和最重要的-容易使用

1. 创建对象而不是页面或模块
2. 在内容对象上设置好的公用默认值
3. 抽象重用元素
4. 分离结构和皮肤(为两个 class)
5. 分离容器和内容(开放的编辑区)
6. 使用继承
7. 对看起来 OO 的应用多个 class

**9条最佳实践**

1. 组件库：可重用和冗余
2. 一致性：避免位置依赖(location-dependent)的样式
3. 抽象化
4. 优化图像和 sprites
5. 灵活
6. 学会爱栅格
7. 避免非标准的浏览器字体
8. 避免高度对齐(alignment)
9. 谨慎卖弄你的技术(choose your bling carefully)

**9个陷阱**

1. 位置依赖的样式
2. 避免指定一个 class 的标签
3. 避免用 ID 定义主内容区域内的样式
4. 避免不规则背景上阴影和圆角
5. 不要拼合所有图片(触发只有少数几个页面)
6. 避免高度对齐
7. 文字就是文字，不要做成图片
8. 冗余
9. 避免过早优化

**创建组件库**

可重用的“乐高积木”

**重用元素使得它们**

性能“免费”

**组件就像乐高积木**

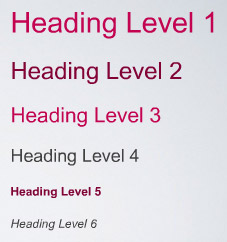
组合并匹配来创建不同的和有趣的页面

**从组件库创建 HTML**

新的页面一般不需要额外的 CSS

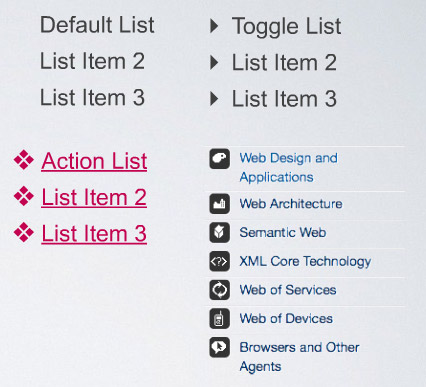
**标题**

根据你需要的语义来完成你想要的表现



**列表**

必须对页面中的所有模块可用



**SIDE-WIDE LOGES**

* 标题
* 列表(比如 action list, external link list, product list, 或 feature list)
* 模块 headers 和 footers
* 栅格
* 按钮
* 圆角 boxes
* Tabs, Carousel, toggle blocks

**避免重复**

在不能增加价值的元件上浪费性能资源

**近似相同的模块**

h3 和 h5 太相似了

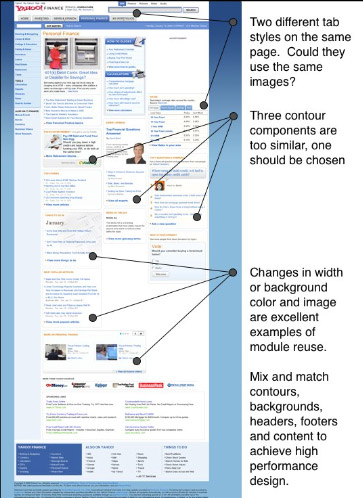


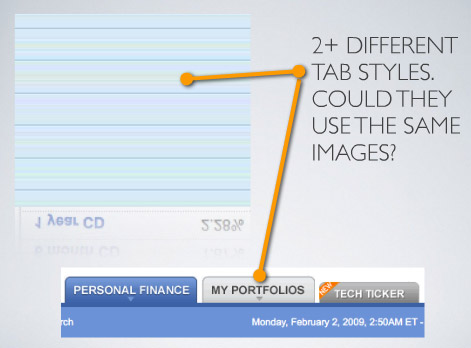
**经验法则：**

如果一个页面中的两个模块看起来太相似，它们在一个站点中太相似，选择一个！

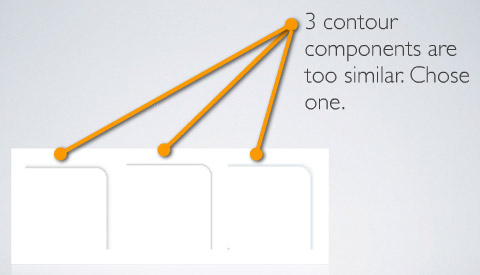
**例子**

Yahoo! 个人财经

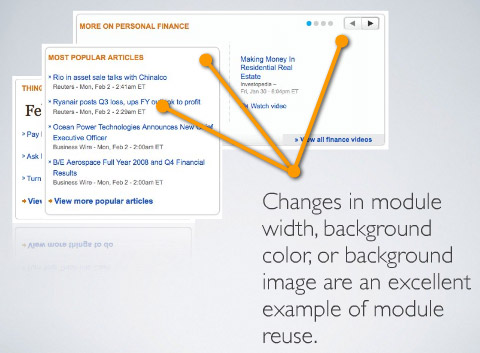




2+不同的 tab 风格。能用相同的图像吗？



3个元件的轮廓太相似了。选择一个。



模块宽度、背景色或背景图片的改变是个很好的模块复用的例子。

**避免位置依赖(location-dependent)的样式**

沙盒比意大利面条好，不过引起了性能问题

**避免什么？**

**运行区域**

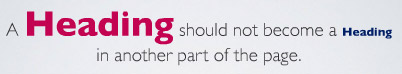
**不时…**

**返回直径**

**破坏**

**在 CSS 中我们一直这么做**

**破坏**



不好的：

#weatherModule h3{color:red;}

#tabs h3{color:blue;}

* h3 的全局颜色未定义，将导致
  + 在新模块中没有定义样式
  + 开发者被迫为相同的样式写更多 CSS

推荐：

h3{color:black;}

#weatherModule h3{color:red;}

#tabs h3{color:blue;}

* 定义了全局颜色(更好！)
* Weather 和 tabs 覆盖了缺省的 h3
  + h3 的3种样式在同一模块中不能并存
  + 缺省样式不能用在 weather 和 tabs 除非有更高的优先级
* Weather 和 tabs 的 h3 永远不能在其他模块中使用

**一致性**

写更多规则去重写之前的疯狂规则。

比如标题在任意模块的表现是可预见的。

**用这个来代替**

h1, .h1{...}

h2, .h2{...}

h3, .h3{...}

h4, .h4{...}

h5, .h5{...}

h6, .h6{...}

* 定义全局值
* 遵循语义(同时允许灵活的视觉)

**需要超过6个标题？**

真的吗？没有重复？没有相似的？

**仍然需要更多标题？**

.category{...}

.section{...}

.product{...}

.prediction{...}

* 通过对象本身的 class 扩展标题对象
* 避免使用继承来改变嵌套对象的表现

**抽象**

复用代码段

**重复编码**

是抽象不同水准的语义失败所导致的

**分离：**

* 容器和内容
* 结构和皮肤
* 轮廓和背景
* 对象和混合物

**分离容器和内容**

定义每个对象的界限

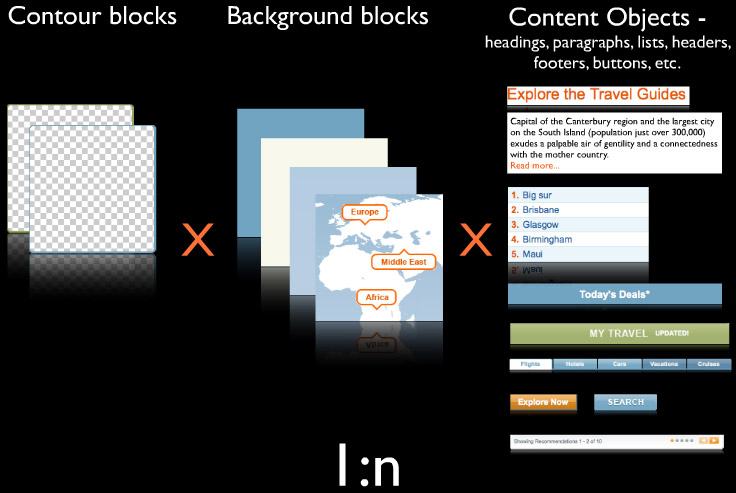
**开放的编辑区**

图像或 flash









**混合与匹配**

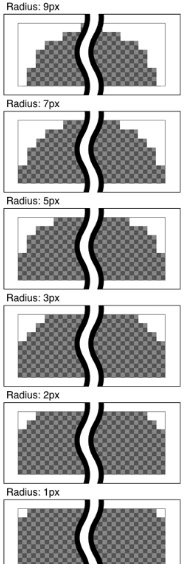
容器和内容对象达到高性能设计

**分离轮廓和背景**

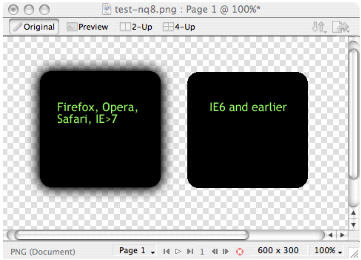
内部透明！

**MAKING IT LOOK FAB**

需要小心选择像素



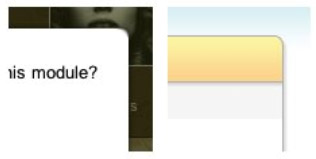
考虑 PNG8 来渐进增强





**陷阱**

可变的或渐变背景

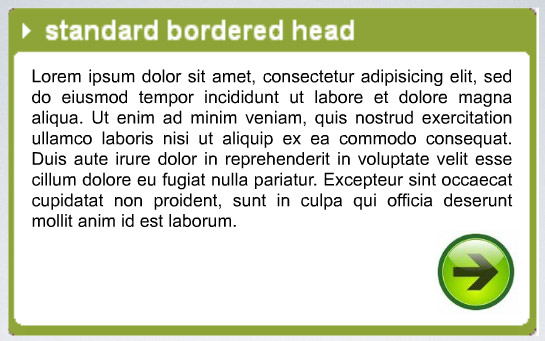


提防圆角后的可变或渐变背景

**分离结构和皮肤**

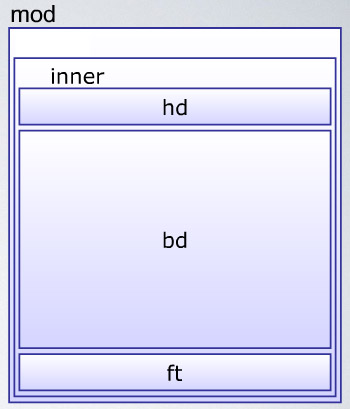
两个单独的 class

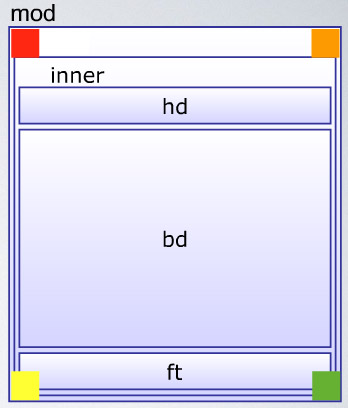
示例：模块

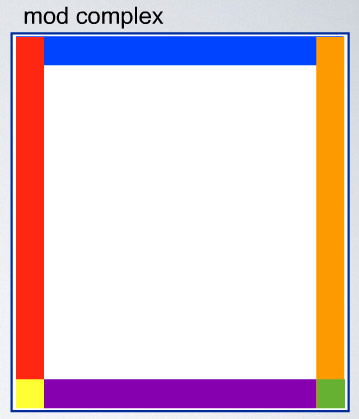


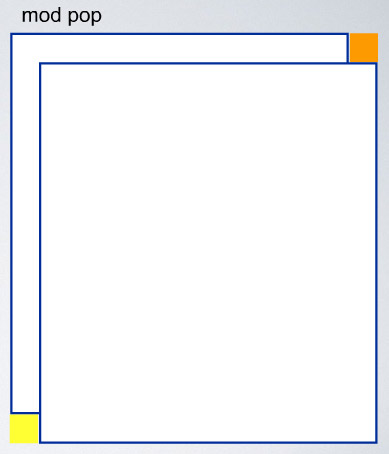
**结构**

Sloves borwser bugs, positions presentational elems, and generally does the heavy  
lifting of CSS





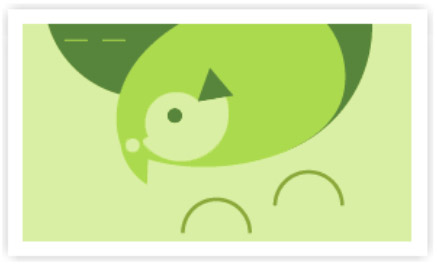




**皮肤**

弄漂亮些。

目标是非常明确的皮肤，复杂的被结构对象和跨网站共享所吸收(The goal is very predictable skins, complexity is  
absorbed by the structure object and shared across the site)。



/\* ----- simple (extends mod) ----- \*/

.simple .inner{

border:1px solid gray;

-moz-border-radius:7px;

-webkit-border-radius:7px;

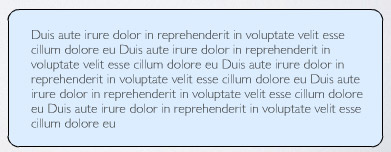
border-radius:7px;

}

.simple b{

\*background-image:url(skin/mod/corners.png);

}



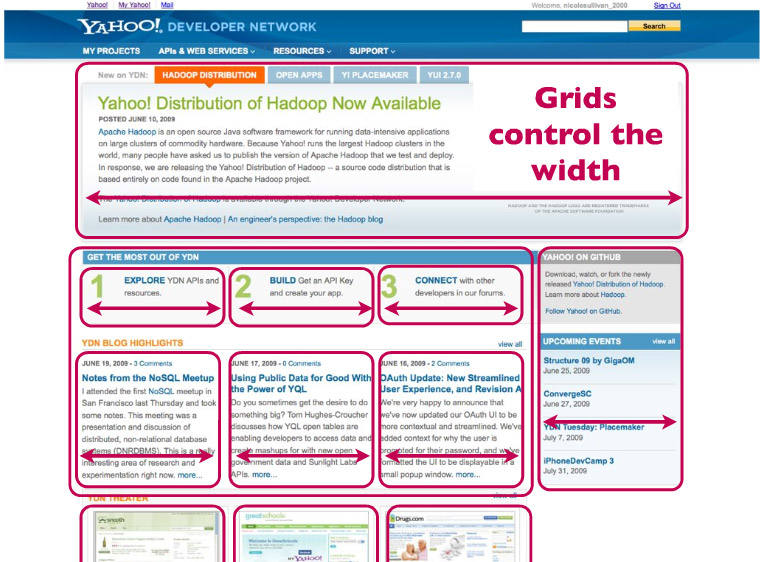
**什么属于皮肤？**

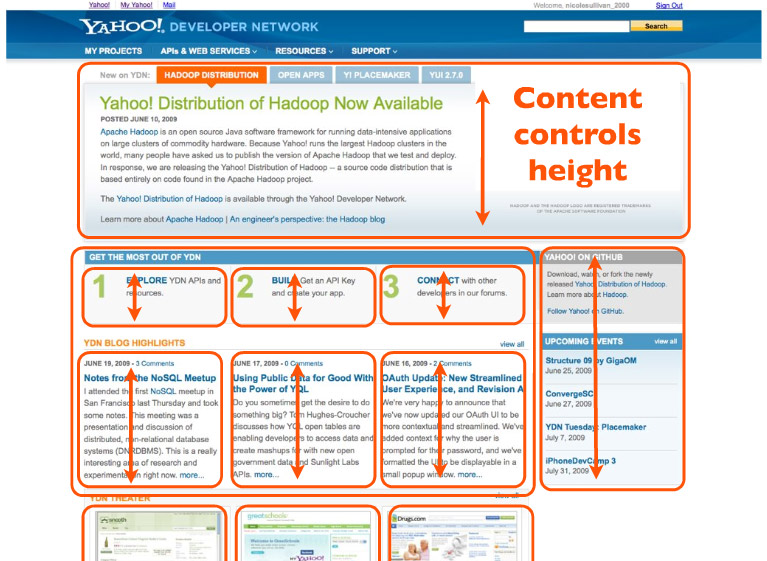
皮肤的每个值应该是确定的，容易预测并测量。

* 颜色
* 边框
* 图像

**灵活性**

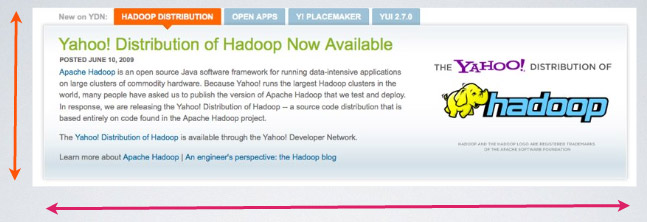
可延长的高度和宽度





**固定尺寸**

Limit the ways in which a module can be resued



**学着爱上栅格**

坚信其会很美

**传授 OO CSS**

给设计师和工程师

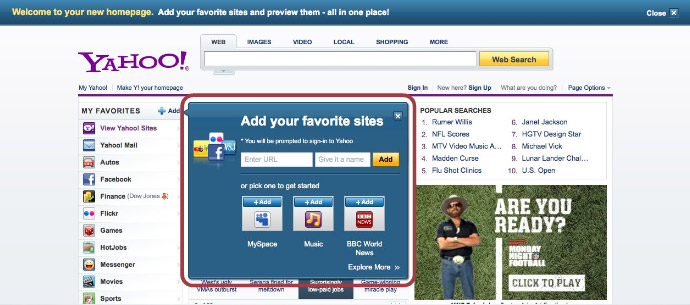
**自然改进**

从简单到复杂的任务



**一个真实案例**

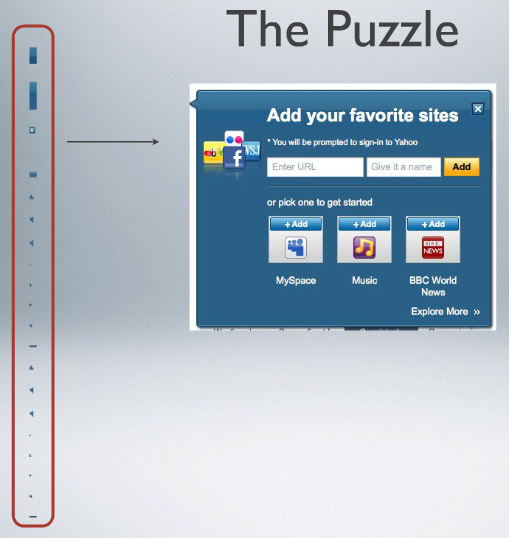
Gonzalo Cordero Yahoo! 前端开发工程师



**需要考虑的设计因素**

* 跨浏览器兼容
* 多语言支持
* 可访问性
* 按时完成全部布局和功能

**困惑**

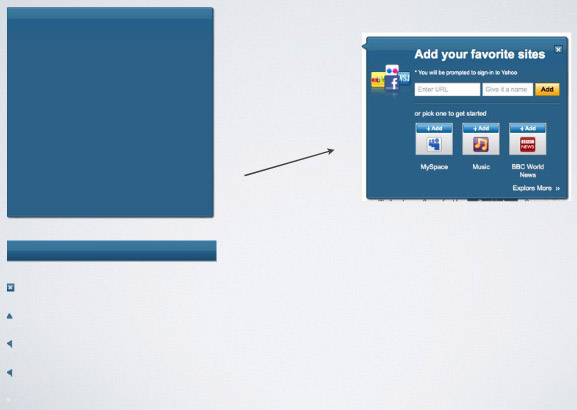


**WEB 妥协**

* 为什么“简单的东东”变复杂？
* 为什么要妥协？
* 为什么我们不能依固定的规则和结构？像讲台上那样？

**OOCSS SAVES THE DAY**

* 单个简单的标签结构
* 跨浏览器支持(甚至 IE 5.5！)
* 非常少的 CSS
* 可剥离的，容易应用在多个设计上



**译注**

这不仅仅是 OOCSS！

本文内容来源于：

* [Object Oriented CSS](http://www.slideshare.net/stubbornella/object-oriented-css) [高清视频下载](http://us.dl1.yimg.com/download.yahoo.com/dl/ydn/nicolesullivan.m4v)
* [What is Object Oriented CSS?](http://www.slideshare.net/stubbornella/what-is-object-oriented-css)
* [The Fast And The Fabulous](http://www.slideshare.net/stubbornella/the-fast-and-the-fabulous)
* [The Cascade, Grids, Headings, and Selectors from an OOCSS Perspective, Ajax Experience 2009](http://www.slideshare.net/stubbornella/the-cascade-grids-headings-and-selectors-from-an-oocss-perspective-ajax-experience-2009)
* [After YSlow “A”](http://www.slideshare.net/stubbornella/after-yslow-a)
* [Design Fast Websites](http://www.slideshare.net/stubbornella/designing-fast-websites-presentation)

由 [ytzong](http://www.99css.com) 翻译整理，算是把 Yahoo! 女性能工程师 Nicole Sullivan 给“榨干”了，嘿嘿。 另外，之前的《[驯服CSS选择器](http://www.99css.com/?p=55)》也是她的作品，富含 OOCSS 的思想。