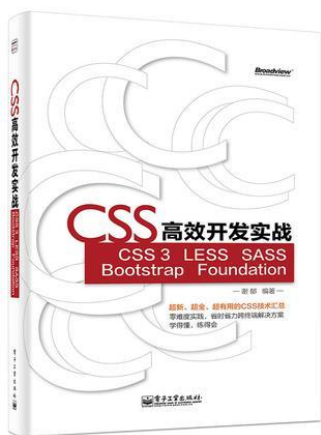


《CSS 高效开发实战—CSS 3、LESS、SASS、Bootstrap、Foundation》，电子工业出版社！



都说 CSS 的进阶顺序是

- 基础：能还原设计稿，兼容各种主流浏览器；
- 进阶：CSS 分层架构，CSS3，响应式，预处理....

本书全部搞定：

- 包括 CSS 基础中的盒子模型、浮动、定位等难点
- 包括 CSS 3 中的圆角、动画、个性边框、媒介查询等特色
- 包括 CSS 的固定布局、流式布局和响应式布局等设计模型
- 包括 Bootstrap 项目实战、Foundation 项目实战、DIV+CSS 实战、HTML 5+CSS 3 实战

本书的写作宗旨就是好学、好用、高效，提供更好的 css 样式解决方案。

1、一张图告诉你，只会 CSS 还不够!!!

会了 CSS 语法，你就真的会了 CSS 吗，来看这张图！是这本《CSS 高效开发实战—CSS 3、LESS、SASS、Bootstrap、Foundation》入门书的导览！熊孩子们，赶紧学习去吧！
曾就职于 CSDN 和淘宝的一线工程师力作。



2、作者简介

谢郁 毕业于电子科技大学，曾就职于卓衡科技、CSDN，目前在百度担任前端开发工程师，熟悉 HTML、CSS、JavaScript、Ruby 等多种开发语言，曾参与过多个互联网项目的产品研发，包括新闻出版总署官方网站、CSDN 论坛、CSDN 问答、CSDN Share 频道、Iteye 社区、CSDN 活动等产品。

3、目录

第一篇 揭开 CSS 3 的面纱

第 1 章 CSS 3 与现代 Web 标准

- 1. 1 未来 Web 标准的发展
 - 1. 1. 1 "去 Adobe"化
 - 1. 1. 2 基础功能集成
 - 1. 1. 3 客户端执行更多的逻辑和渲染任务
 - 1. 1. 4 适应移动设备的发展
- 1. 2 什么是 CSS 3
- 1. 3 CSS 3 的特性
 - 1. 3. 1 圆角
 - 1. 3. 2 阴影
 - 1. 3. 3 渐变
 - 1. 3. 4 变换
 - 1. 3. 5 动画
 - 1. 3. 6 媒介查询
- 1. 4 检测浏览器是否支持 CSS 3
 - 1. 4. 1 使用 Modernizr 检测支持 CSS 3
 - 1. 4. 2 支持 CSS 3 的浏览器
- 1. 5 用 CSS 3 实现的优秀网页
- 1. 6 小结

第 2 章 温故知新——HTML、CSS 基础

- 2. 1 W3C 标准、HTML 语言和 CSS
 - 2. 1. 1 W3C 组织与 W3C 标准
 - 2. 1. 2 什么是 HTML
 - 2. 1. 3 什么是 CSS
 - 2. 1. 4 HTML+CSS 之最佳拍档
- 2. 2 不可不知的 CSS 盒子模型
- 2. 3 跨浏览器的 CSS
- 2. 4 理解浏览器的属性前缀
 - 2. 4. 1 常用的属性前缀
 - 2. 4. 2 属性前缀的排序
- 2. 5 揭开浮动布局的秘密
 - 2. 5. 1 浮动导致的布局变动

- 2. 5. 2 清除浮动
- 2. 6 看穿 CSS 的定位技术
- 2. 6. 1 相对定位技术与实战
- 2. 6. 2 绝对定位技术与实战
- 2. 7 小结

第3章 使用 CSS 选择器让样式表更健壮

- 3. 1 基础选择器
 - 3. 1. 1 标签选择器
 - 3. 1. 2 类选择器
 - 3. 1. 3 id 选择器
 - 3. 1. 4 通配符选择器
 - 3. 1. 5 子元素选择器
 - 3. 1. 6 后代元素选择器
 - 3. 1. 7 相邻元素选择器
 - 3. 1. 8 属性选择器
 - 3. 1. 9 组选择器
 - 3. 1. 10 复合选择器
- 3. 2 伪类选择器
 - 3. 2. 1 结构化伪类
 - 3. 2. 2 目标伪类: **target**
 - 3. 2. 3 状态伪类
 - 3. 2. 4 否定伪类: **not(S)**
- 3. 3 实战演练——选择器
 - 3. 3. 1 伪类选择器的实战——新闻聚合类网页
 - 3. 3. 2 基础选择器的组合实战——新闻聚合类网页
- 3. 4 小结

第4章 设计更炫目的字体

- 4. 1 添加和使用自定义字体
 - 4. 1. 1 传统的字体定义
 - 4. 1. 2 个性化的字体定义
 - 4. 1. 3 个性化的字体图标
- 4. 2 使用反射让文字倒映
 - 4. 2. 1 反射的基本语法
 - 4. 2. 2 变幻多端的反射效果实例
- 4. 3 字体阴影——光晕、浮雕、投影效果
- 4. 4 字体描边
- 4. 5 字体分栏——让网页像报纸一样分栏排版
- 4. 6 实战演练——处理字体溢出和破字
- 4. 7 小结

第5章 背景和颜色

- 5. 1 设定背景图的大小
- 5. 2 利用图层叠加实现多背景
- 5. 3 使用图片背景的 **origin** 和 **clip** 属性
 - 5. 3. 1 **background-origin** 属性

- 5. 3. 2 background-clip 属性
- 5. 4 颜色模式
 - 5. 4. 1 RGBA 模式
 - 5. 4. 2 HSLA 模式
- 5. 5 透明颜色
- 5. 6 语法糖——currentColor 属性
- 5. 7 渐变——放弃图片的首选良方
 - 5. 7. 1 线性渐变
 - 5. 7. 2 放射渐变
- 5. 8 实战演练——渐变效果
 - 5. 8. 1 带有立体凸起效果的按钮
 - 5. 8. 2 构造尺寸更灵活的背景
 - 5. 8. 3 使用放射渐变制作光影效果
- 5. 9 小结

第 6 章 更个性的边框

- 6. 1 圆角边框
 - 6. 1. 1 圆角边框的基本用法
 - 6. 1. 2 使用百分比作为单位
 - 6. 1. 3 设置不同弧度的圆角
- 6. 2 边框阴影
 - 6. 2. 1 内外阴影
 - 6. 2. 2 偏移量
 - 6. 2. 3 阴影尺寸
 - 6. 2. 4 模糊距离
- 6. 3 图片边框——让图片环绕在元素周围
- 6. 4 通过 resize 属性来改变输入框的大小
- 6. 5 实战演练——CSS 3 边框效果
 - 6. 5. 1 边框圆角在 Bootstrap 和淘宝网中的应用
 - 6. 5. 2 边框阴影在苹果官网中的应用
- 6. 6 小结

第 7 章 变换和动画

- 7. 1 CSS 3 的变换类型
 - 7. 1. 1 rotate 旋转变换
 - 7. 1. 2 skew 扭曲变换
 - 7. 1. 3 scale 比例缩放
 - 7. 1. 4 translate 位移变换
 - 7. 1. 5 transform 小结
- 7. 2 使用 transition 制作交互动画
- 7. 3 使用 @keyframes 制作动画
 - 7. 3. 1 @keyframes 的基本语法
 - 7. 3. 2 用 @keyframes 制作循环动画
 - 7. 3. 3 @keyframes 小结
- 7. 4 实战演练——结合变换制作 3D 旋转卡片
- 7. 5 可参考的 CSS 动画资源

- 7. 5. 1 Hover. css——鼠标 hover 动画
- 7. 5. 2 iHover——hover 动画类库
- 7. 5. 3 CSS 3 和 JavaScript 的结合
- 7. 6 小结
- 第 8 章 媒介查询和响应式设计
- 8. 1 媒介类型=各种浏览终端
- 8. 2 认识响应式网页设计
- 8. 3 媒介查询的基本语法
- 8. 4 设备
- 8. 4. 1 常见设备的宽度和高度
- 8. 4. 2 检测设备翻转
- 8. 5 实战演练——应用媒介查询制作响应式导航栏
- 8. 6 小结
- 第 9 章 更简便的布局——弹性盒子
- 9. 1 认识弹性盒子
- 9. 2 弹性盒子的语法
- 9. 3 操作元素
- 9. 3. 1 控制子元素的方向
- 9. 3. 2 控制元素对齐
- 9. 3. 3 控制元素显示顺序
- 9. 4 实战演练——用弹性盒子设计阅读 APP
- 9. 5 小结
- 第 10 章 CSS 常用工具
- 10. 1 使用 Prefix free 处理 CSS 跨浏览器兼容
- 10. 2 应用 Normalize 统一不同浏览器下的样式
- 10. 3 应用 Grunt 进行 CSS 压缩
- 10. 4 小结
- 第二篇 使用 CSS 3 框架进行高效开发
- 第 11 章 流行的 CSS 布局设计
- 11. 1 固定布局
- 11. 1. 1 1960 的秘密
- 11. 1. 2 定义列宽
- 11. 1. 3 运用 CSS 实现固定列宽的栅格
- 11. 1. 4 实战演练——运用 960gs 实现固定布局的新闻页面
- 11. 2 流式布局
- 11. 2. 1 计算列百分比
- 11. 2. 2 使图片更加灵活
- 11. 2. 3 定义最大/最小宽度
- 11. 2. 4 实战演练——实现一个流式布局的新闻页面
- 11. 3 响应式布局
- 11. 3. 1 使用媒介查询
- 11. 3. 2 实战演练——实现一个响应式布局的新闻页面
- 11. 4 小结
- 第 12 章 Bootstrap 框架实战

- 12. 1 认识 Bootstrap
 - 12. 1. 1 初识 Bootstrap
 - 12. 1. 2 Bootstrap 为何如此流行
 - 12. 1. 3 Bootstrap 的版本发展
- 12. 2 Bootstrap 入门
 - 12. 2. 1 在自己的项目中引入 Bootstrap
 - 12. 2. 2 添加 Bootstrap 的 class 实现基本样式
 - 12. 2. 3 调用 Bootstrap 的通用组件
 - 12. 2. 4 添加 JavaScript 动态效果
- 12. 3 Bootstrap 的栅格系统
 - 12. 3. 1 固定布局的栅格系统
 - 12. 3. 2 流式布局的栅格系统
 - 12. 3. 3 响应式布局的栅格系统
- 12. 4 使用 Bootstrap 的基本样式
 - 12. 4. 1 字体排版
 - 12. 4. 2 表格
 - 12. 4. 3 表单
 - 12. 4. 4 按钮
 - 12. 4. 5 图片
 - 12. 4. 6 响应式工具
 - 12. 4. 7 工具类
- 12. 5 使用 Bootstrap 的组件
 - 12. 5. 1 下拉菜单
 - 12. 5. 2 按钮组
 - 12. 5. 3 input 控件组
 - 12. 5. 4 导航
 - 12. 5. 5 列表组
 - 12. 5. 6 分页
 - 12. 5. 7 标签与 Badge
 - 12. 5. 8 缩略图
 - 12. 5. 9 面板
 - 12. 5. 10 进度条
- 12. 6 Bootstrap 中的 JavaScript 特效
 - 12. 6. 1 模态对话框
 - 12. 6. 2 标签页切换
 - 12. 6. 3 Tooltip
 - 12. 6. 4 弹出框
 - 12. 6. 5 提示信息
 - 12. 6. 6 按钮
 - 12. 6. 7 折叠
 - 12. 6. 8 幻灯片
- 12. 7 定制 Bootstrap
 - 12. 7. 1 在官方网站进行 Bootstrap 的定制
 - 12. 7. 2 修改源代码定制 Bootstrap

- 12. 8 其他 Bootstrap 资源
- 12. 9 小结
- 第 13 章 Foundation 框架实战
 - 13. 1 认识 Foundation
 - 13. 2 Foundation 的安装和使用
 - 13. 2. 1 传统方式的下载安装
 - 13. 2. 2 使用 Compass 进行 Foundation 开发
 - 13. 2. 3 在 Rails 应用中引入 Foundation
 - 13. 3 使用 Foundation 栅格系统
 - 13. 3. 1 基本栅格系统
 - 13. 3. 2 块网格 (Block Grid)
 - 13. 4 Foundation 基本样式
 - 13. 4. 1 标题和段落
 - 13. 4. 2 列表
 - 13. 4. 3 按钮
 - 13. 4. 4 面板
 - 13. 4. 5 缩略图
 - 13. 4. 6 视频
 - 13. 4. 7 可见性
 - 13. 5 导航系统
 - 13. 5. 1 面包屑导航
 - 13. 5. 2 侧边栏导航
 - 13. 5. 3 头部导航
 - 13. 5. 4 子导航
 - 13. 6 Foundation 中的 JavaScript 特效
 - 13. 6. 1 幻灯片
 - 13. 6. 2 Clearing lightboxes
 - 13. 6. 3 弹出层显示
 - 13. 6. 4 长页面滚动效果
 - 13. 6. 5 其他特效
 - 13. 7 定制 Foundation
 - 13. 7. 1 在官方网站进行定制
 - 13. 7. 2 通过配置文件进行定制
 - 13. 8 小结
- 第 14 章 LESS 和 SASS
 - 14. 1 CSS 的缺陷
 - 14. 1. 1 法定义变量
 - 14. 1. 2 重复代码
 - 14. 1. 3 计算问题
 - 14. 1. 4 作用域和命名空间
 - 14. 1. 5 CSS 缺陷总结
 - 14. 2 LESS 其实更多
 - 14. 2. 1 LESS 介绍
 - 14. 2. 2 LESS 使用基础

- 14. 2. 3 使用变量和操作符
- 14. 2. 4 使用 Mixin 混入
- 14. 2. 5 内嵌规则
- 14. 2. 6 运算
- 14. 2. 7 LESS 总结
- 14. 3 使用 SASS
- 14. 3. 1 SASS 介绍
- 14. 3. 2 SASS 安装和使用
- 14. 3. 3 使用变量
- 14. 3. 4 计算
- 14. 3. 5 使用@import 导入
- 14. 3. 6 使用@extend 继承
- 14. 3. 7 使用@mixin 混入
- 14. 3. 8 使用@function 定义函数
- 14. 3. 9 控制语句
- 14. 3. 10 SASS 总结
- 14. 4 使用 SASS 的扩展库 Compass
- 14. 4. 1 CSS 3 模块
- 14. 4. 2 Reset 模块
- 14. 4. 3 Utilities 模块
- 14. 4. 4 Helpers 模块
- 14. 4. 5 Compass 总结
- 14. 5 小结

第 15 章 其他 CSS 框架简介

- 15. 1 轻量级框架代表——Pure CSS
- 15. 2 手机页面 UI 框架——Ratchet 框架
- 15. 3 优秀的国产 CSS 框架
- 15. 3. 1 阿里巴巴的 Alice 框架
- 15. 3. 2 网易的 NEC
- 15. 3. 3 百度的 GMU 框架
- 15. 3. 4 渴切
- 15. 3. 5 用于中文排版的 Typo. css
- 15. 4 小结

第三篇 CSS 实战项目

第 16 章 传统 DIV+CSS 设计的视频网站

- 16. 1 网站的页面效果图分析
- 16. 1. 1 页面头部和页脚分析
- 16. 1. 2 首页主体内容分析
- 16. 1. 3 内页主体内容分析
- 16. 2 网站的布局规划
- 16. 2. 1 页面布局规划
- 16. 2. 2 切割首页及导出图片
- 16. 2. 3 切割内页及导出图片
- 16. 3 网站 HTML 框架的编写

- 16. 3. 1 页面 HTML 框架搭建
- 16. 3. 2 页面头部和页脚的 HTML
- 16. 3. 3 页面公共部分的 HTML
- 16. 3. 4 首页主体内容的 HTML
- 16. 3. 5 内页主体内容的 HTML
- 16. 3. 6 首页 HTML 代码总览
- 16. 3. 7 内页 HTML 代码总览
- 16. 4 网站 CSS 样式的编写
- 16. 4. 1 页面公共部分的 CSS
- 16. 4. 2 页面框架的 CSS
- 16. 4. 3 页面头部和页脚的 CSS
- 16. 4. 4 首页主体内容的 CSS
- 16. 4. 5 内页主体内容的 CSS
- 16. 4. 6 网站 CSS 代码总览
- 16. 5 小结

第 17 章 使用 HTML 5+CSS 3 开发搜房网

- 17. 1 网站前期策划
- 17. 1. 1 理解 HTML 5 的语义性元素
- 17. 1. 2 搜房网网站结构
- 17. 1. 3 搜房网整站预览
- 17. 2 搜房网的首页设计
- 17. 2. 1 首页的布局
- 17. 2. 2 设计导航栏
- 17. 2. 3 设计宣传广告栏
- 17. 2. 4 CSS 布局
- 17. 3 搜房网的内容页设计
- 17. 3. 1 出售房源页面
- 17. 3. 2 购买房源页面
- 17. 3. 3 出租房源页面
- 17. 3. 4 房产过户页面
- 17. 3. 5 联系我们页面
- 17. 4 小结

第 18 章 使用 Bootstrap 实现论坛后台管理系统

- 18. 1 项目开始
- 18. 2 页面布局
- 18. 2. 1 引入 Bootstrap 3 框架
- 18. 2. 2 编写布局代码
- 18. 3 实现导航栏
- 18. 3. 1 构建导航的框架代码
- 18. 3. 2 填写标题和导航链接
- 18. 3. 3 添加搜索框和通知系统
- 18. 3. 4 添加管理员的登录信息
- 18. 3. 5 构建响应式导航
- 18. 4 实现左侧边栏

- 18. 5 实现主功能部分
 - 18. 5. 1 主功能的头部
 - 18. 5. 2 主功能的帖子列表
- 18. 6 小结
- 第 19 章 使用 Foundation 实现论坛首页
 - 19. 1 项目开始
 - 19. 2 页面布局
 - 19. 2. 1 引入 Foundation 需要的包
 - 19. 2. 2 移动优先的布局
 - 19. 3 实现头部导航栏
 - 19. 4 实现响应式版块列表
 - 19. 5 实现热门帖子推荐
 - 19. 6 小结
- 附录 A 网页制作的调试工具及使用
- 附录 B 提升 CSS 的性能和效率

4、样章试读

8.1 媒介类型=各种浏览终端

这里提到的媒介，不同于新闻传播学对媒介的定义，而是指我们浏览内容所使用的各种电子设备。在 CSS 2 标准中就已经可以根据不同的媒介类型（Media Type）来设置不同的输出样式了。`@media` 规则使开发者有能力在相同的样式表中，针对不同的媒介来使用不同的样式规则。

下面这个例子中的样式告诉浏览器在显示器上显示 14 像素的 Verdana 字体，但是假如页面需要被打印，将使用 10 像素的 Times 字体。

```
<html>
  <head>
    <style>
      @media screen{
        p.test {font-family:verdana,sans-serif; font-size:14px}
      }
      @media print{
        p.test {font-family:times,serif; font-size:10px}
      }
      @media screen,print{
        p.test {font-weight:bold}
      }
    </style>
  </head>
  <body>....</body>
</html>
```

表 8.1 是 Media Type 的详细类型，不过使用比较多的是 screen 和 print 类型，用于区分打印和屏幕显示。

表 8.1 Media Type的类型汇总

设备	指代
all	匹配所有设备
braille	匹配触觉反馈设备
embossed	凸点字符印刷设备
handheld	手持设备（尤其是小屏幕，有限带宽，PSP、NDS这种规格一般可以叫作 handheld）
print	打印机设备
projection	投影仪设备
screen	彩色计算机显示器设备
speech	语音合成器设备
tty	栅格设备（终端或电传打字机）
tv	电视设备

注意：现在的 Android、iPhone 都不是 handheld 设备，它们都是 screen 设备。所以，不要试图用 handheld 来识别 iPhone、iPad 或 Android 等设备。

通过表 8.1 可以看出，@media 只能做一个大概的区分，而现在桌面和移动设备拥有不同的分辨率，即使是同样类型的设备，也可能需要做出不同的适配。所以仅仅依靠 Media Type 已经无法满足时代的要求了。为了顺应这种需求，CSS 3 引入了 Media Query（媒体查询）。

注意：虽然@media 在 CSS 2 时代就已经引入，但是只有 Opera 支持 handheld 属性。

8.2 认识响应式网页设计

页面有能力去自动响应用户的设备环境。响应式网页设计就是一个网站能够兼容多个终端——而不是为每个终端做一个特定的版本。

响应式设计是一项不折不扣的“技术驱动”型设计模式，虽然对于设计师来说，把握响应式设计中的交互模式、色彩运用是一件颇费思量的事情，但是响应式设计本身是来源于移动互联网技术的兴起和新的 CSS 3 技术，没有这些，一切好的想法都是镜花水月。

下面就来简单认识一下响应式设计并了解响应式设计需要遵循的一些模式。首先来看图 8.1 和图 8.2，这是两个典型的响应式设计案例，读者可以直观地感受一下。

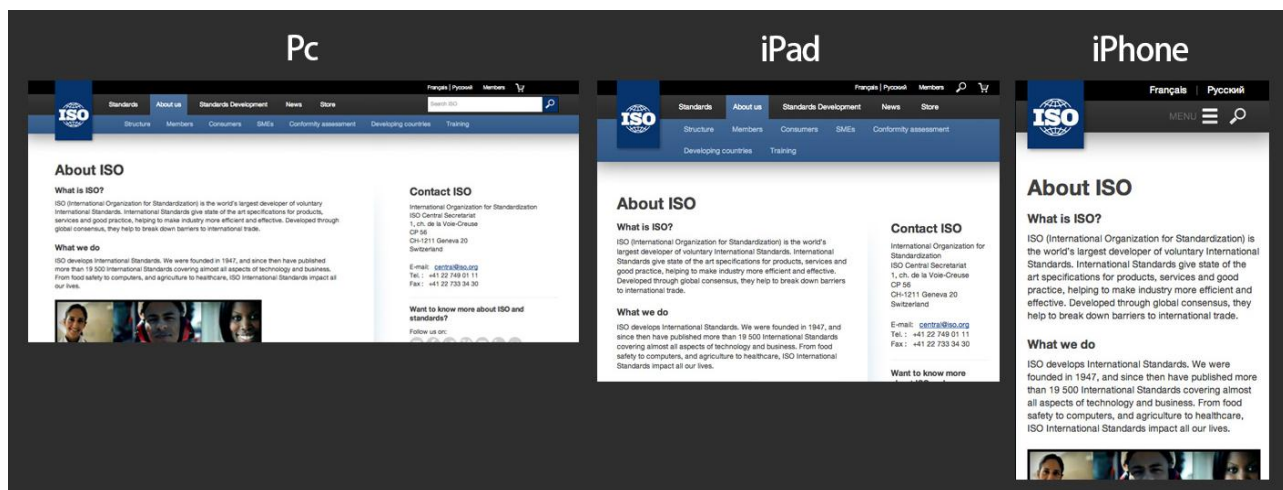


图 8.1 响应式设计案例 1

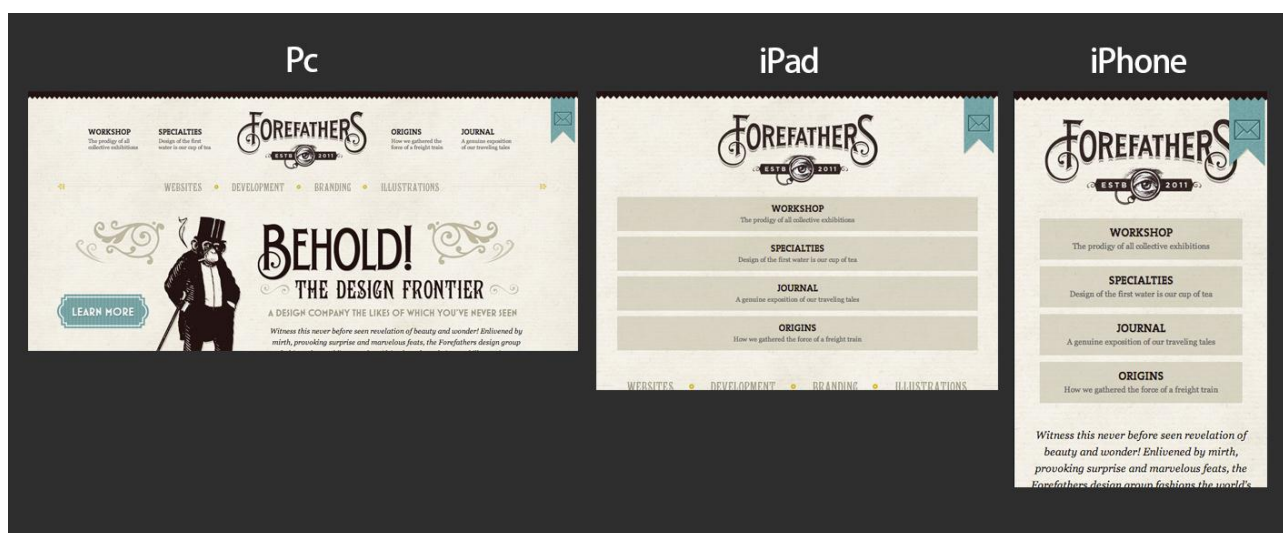


图 8.2 响应式设计案例 2

读者可能已经发现，响应式设计并不是同样内容的等比例缩小，也不像之前流行的 WAP 网站一样，和 PC 端差异巨大。响应式设计在设计风格和色彩搭配上保持了很大的一致性，又根据移动设备的特点对布局进行了调整。响应式设计一般需要遵循 4 个原则，如图 8.3 所示。

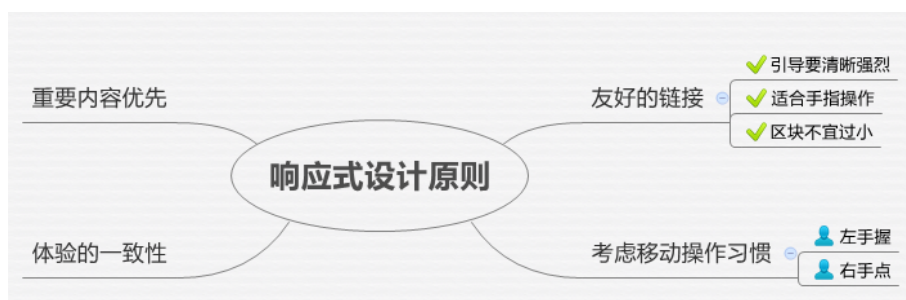


图 8.3 响应式设计原则

1. 小屏幕只显示高优先级内容

在响应式网页设计中，切换到小屏幕移动设备时，有时候需要对页面内容进行删减，按

照优先级显示内容，只显示高优先级内容是原则之一。在屏幕较小的移动设备上应该优先考虑主要内容并移掉那些小的栏目。在顶部显示高优先级内容，即把最重要的内容放置在顶部。导航是否一定要出现在页头或者重新布局在页尾，都要依网站具体规划去考虑。

2. 提供清晰和友好的手指操作链接

尤其在手机设备上，可点击操作的区块不宜过小，引导要清晰强烈，不忽略任何一款设备。

3. 体验的一致性

要让用户在不同的设备上仍保持对同一页面相同的视觉和感觉。这也遵循交互设计体验一致性的原则。读者可以参考 Oliver Russell 网站，一个设计非常灵活的网站，在不同的屏幕分辨率下可以保持一致的视觉和感受。

4. 考虑操作移动设备的习惯

大部分用户习惯于右手点击操作，左手负责握住设备。右侧的导航列表既方便右手的点击，又可以避免被握着设备的左手不小心触碰到。

8.3 媒介查询的基本语法

可以将 Media Query 看成是添加了 CSS 属性判断的 Media Type，其基本语法如下：

```
@media screen and (max-width: 600px) {
  .class {
    background: #ccc;
  }
}
```

这段代码定义了小于 600 像素的窗口所应用的样式。前半部分 @media screen 和 Media Type 的语法是一样的，一般来说选择 screen 或 only screen，因为所有现代的智能手机、平板、PC 在类型上都是 screen。后半部分使用 and 作为条件添加符号，可以使用多个 and 添加多个条件：

```
@media screen and (min-width: 600px) and (max-width: 900px) {
  .class {
    background: #333;
  }
}
```

这段代码表示宽度在 600~900 像素之间的窗口应用该样式。width 作为条件是最常用、最基本的，根据我们的需求，还可以限定更多的条件来更精确的对设备进行适配。比如通过 orientation 来判断设备翻转、通过 device-aspect-ratio 来判断屏幕的纵横比等。表 8.2 列出了可以使用的一些判断条件。

表 8.2 适配设备的判断条件

媒体特性	说明/值	可用媒体类型	接受min/max
width	窗体宽度	视觉屏幕/触摸设备	是
height	窗体高度	视觉屏幕/触摸设备	是
device-width	屏幕宽度	视觉屏幕/触摸设备	是
device-height	屏幕高度	视觉屏幕/触摸设备	是
orientation	设备手持方向（portrait横向/landscape竖向）	位图介质类型	否

aspect-ratio	浏览器、纸张长宽比	位图介质类型	是
device-aspect-ratio	设备屏幕长宽比	位图介质类型	是
color	颜色模式（例如旧的显示器为256色）整数	视觉媒体	是
color-index	颜色模式列表整数	视觉媒体	是
monochrome	整数	视觉媒体	是
resolution	解析度	位图介质类型	是
scan	progressive逐行扫描/interlace隔行扫描	电视类	否
grid	整数，返回0或1	栅格设备	否