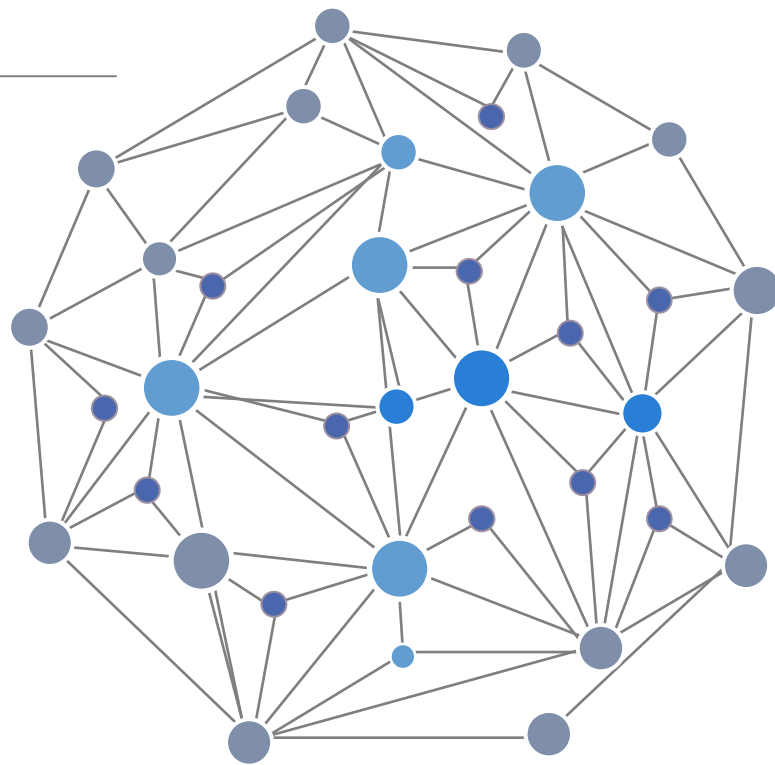

Web 程序设计

第五讲 层叠样式表 CSS(2) CSS 排版布局

福州大学 计算机与大数据学院
软件工程系 陈昱

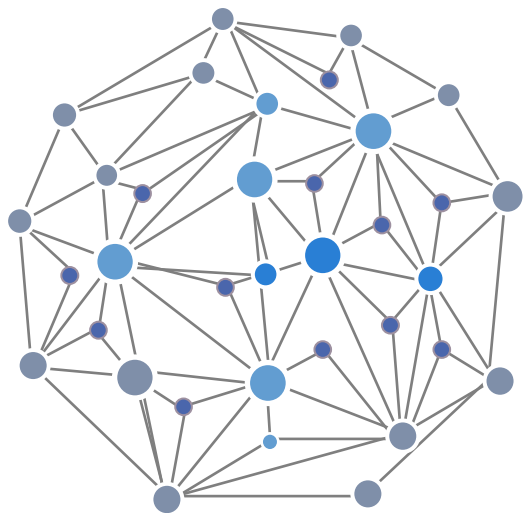


内容提要

- 盒状模型：边界，边框和补白
- 布局，定位，显示属性
- CSS 布局实例
- 阅读材料

基于 CSS 的排版布局

- 如何用 CSS 替代表格进行布局？
- 透彻理解三大概念：
 - 盒状模型 Box-Model
 - 定位 Position
 - 浮动 Float



盒状模型

Box-Model: margin, border and padding

盒状模型 Box Model

- CSS 假设所有元素都是矩形（盒子），一个个元素按照顺序排列在网页上
 - 内联元素在块元素内部也是按照矩形排列
- 元素的外围都被一系列矩形的框（间距）包围着

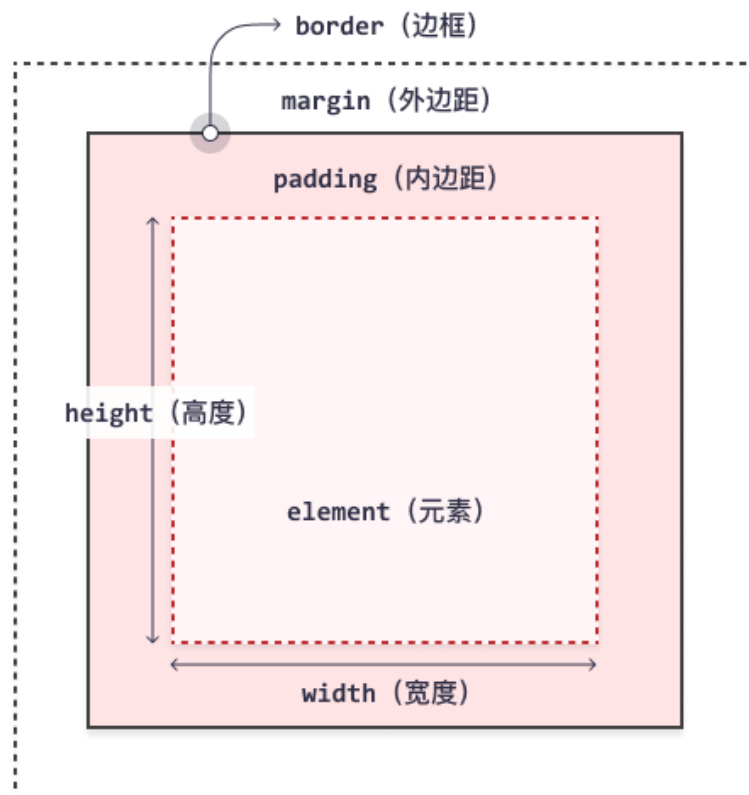
h1

p

.....

盒状模型

- margin (外边距) 是元素边框外部与其他元素的空白距离;
- 而 padding (内边距) 则是元素内部边框与内容的空白距离;
- 它们是设置元素间距最常用的两个属性。



盒状模型

- 中间是元素区域，从里到外依次包围着内边距、边框和外边距
- 它们决定了一个元素与其他元素之间的间距
- 元素的背景（background）是填充在由**内容**和**内边距、边框**组成的区域

margin & padding

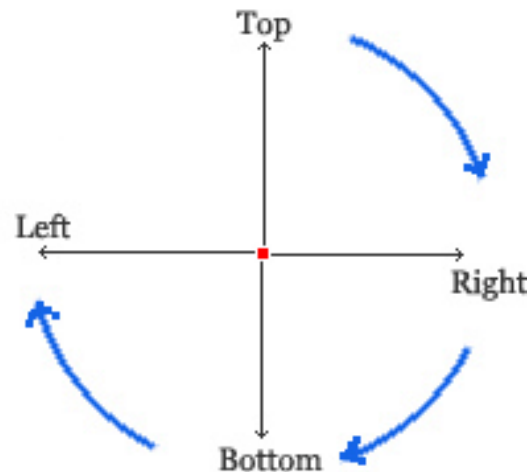
- margin-top
 - margin-right
 - margin-bottom
 - margin-left
 - 简写属性
 - margin
- padding-top
 - padding-right
 - padding-bottom
 - padding-left
 - 简写属性
 - padding

Sample

```
h2 {  
  font-size: 1.5em;  
  background-color: #ccc;  
  margin-top: 5px;  
  padding-bottom: 0.5em;  
}
```

简写属性 (1)

- margin: 上边距 右边距 下边距 左边距;
 - 顺时针顺序
 - margin: 20px 10px 20px 10px;
- margin: 上边距 左右边距 下边距;
 - margin: 20px 10px 10px;



简写属性 (2)

- margin: 上下边距 左右边距;
 - margin: 20px 10px;
- margin: 四个边距;
 - margin: 10px;

简写属性总结

- 一个值=所有的边距;
 - 两个值=上下边距, 左右边距;
 - 三个值=上边距, 左右边距, 下边距;
 - 四个值=遵循顺时针顺序
-
- 注意, 简写属性的值之间使用空格隔开

简写属性例子

```
margin-left: 10px;  
margin-right: 10px;  
margin-top: 20px;  
margin-bottom: 20px;
```

简写为:

```
margin: 20px 10px 20px 10px;
```

或:

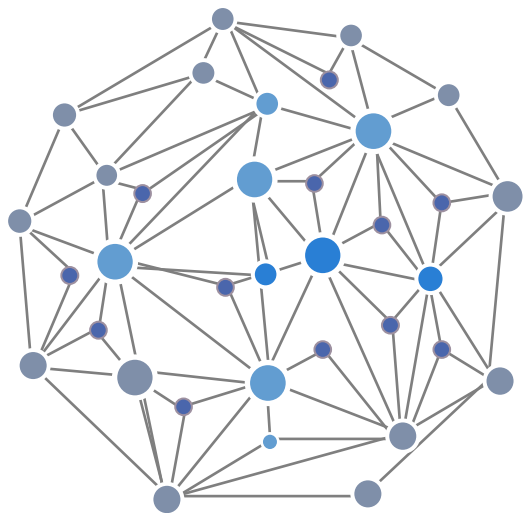
```
margin: 20px 10px;
```

边框 border

- 边框可以应用到在主体元素 body 中的绝大部分元素上
- 边框样式 border-style 2
- 边框宽度 border-width 1
- 边框颜色 border-color 3
- 简写属性 border

注意，继承属性的例外

- 以下属性不会被子元素继承
 - 背景 background
 - 边界 margin
 - 补白 padding
 - 边框 border
 - 元素宽度 width
 - 元素高度 height



尺寸、定位，布局 和显示属性

Dimension, Positioning, Layout and Visibility

Dimension 尺寸

- height 设置元素的高度
- width 设置元素的宽度
- line-height 设置两行间的距离（行高）
- max-height/min-height
 设置元素的最大/最小高度
- max-width/min-width
 设置元素的最大/最小宽度

流动布局与固定宽度布局

- 固定宽度布局 (Fixed-Width Layout)
 - 页面宽度为固定值, 比如 960px
 - 经典布局方式
- 流动布局 (Fluid Layout)
 - 页面的宽度不固定, 可以在屏幕上自由延伸
 - 可以充分利用屏幕空间, 适应多种屏幕分辨率
 - 在 Web 2.0 时代开始流行
 - 通过 max-width 和 min-width 设定一个宽度的流动区间

定位属性

- position
- top
- bottom
- left
- right
- z-index

position 定位

- 定位属性 position 用于定义一个元素用什么方式定位。利用 position 定位属性，可以精确地设定元素的位置和参照物
- 通常定位的参照物有三种，相对于：
 - 父元素
 - 其他元素
 - 浏览器窗口

position 定位

- 定位属性 position 的属性值有 4 种
 - static (静态, 默认值)
 - absolute (绝对定位)
 - relative (相对定位)
 - fixed (相对与浏览器窗口定位)

上下左右 四个属性

- top, bottom, left, right
- 配合 position, 用于指定对象的定位距离
- 值可以是 *auto* 或是 **长度值** 或是 **百分数**

```
position: absolute;  
top: 100px; left: 100px;
```

position: static;

- 默认定位方式，对象遵循 (X)HTML 定位规则

<p>

.....

R2D2

Torquil

Fred

.....

</p>

.....
_orem ipsum dolor sit amet, consectetur adipisicing
et dolore R2D2 **Torquil** Fred magna al
exercitation ullamco laboris nisi ut aliquip ex ea com
prehenderit in voluptate velit esse cillum dolore eu
.....

[Sample](#)

position: static;

- p、h1、div 等元素称为块元素，显示为一块内容
- strong 和 span 等元素称为内联元素，显示在行内
- 在文档流中，块级元素从上到下一个接一个排列；内联元素在一行中水平布置

position: relative;

```
em {  
  position: relative;  
  top: 20px;  
  left: 20px;  
}
```

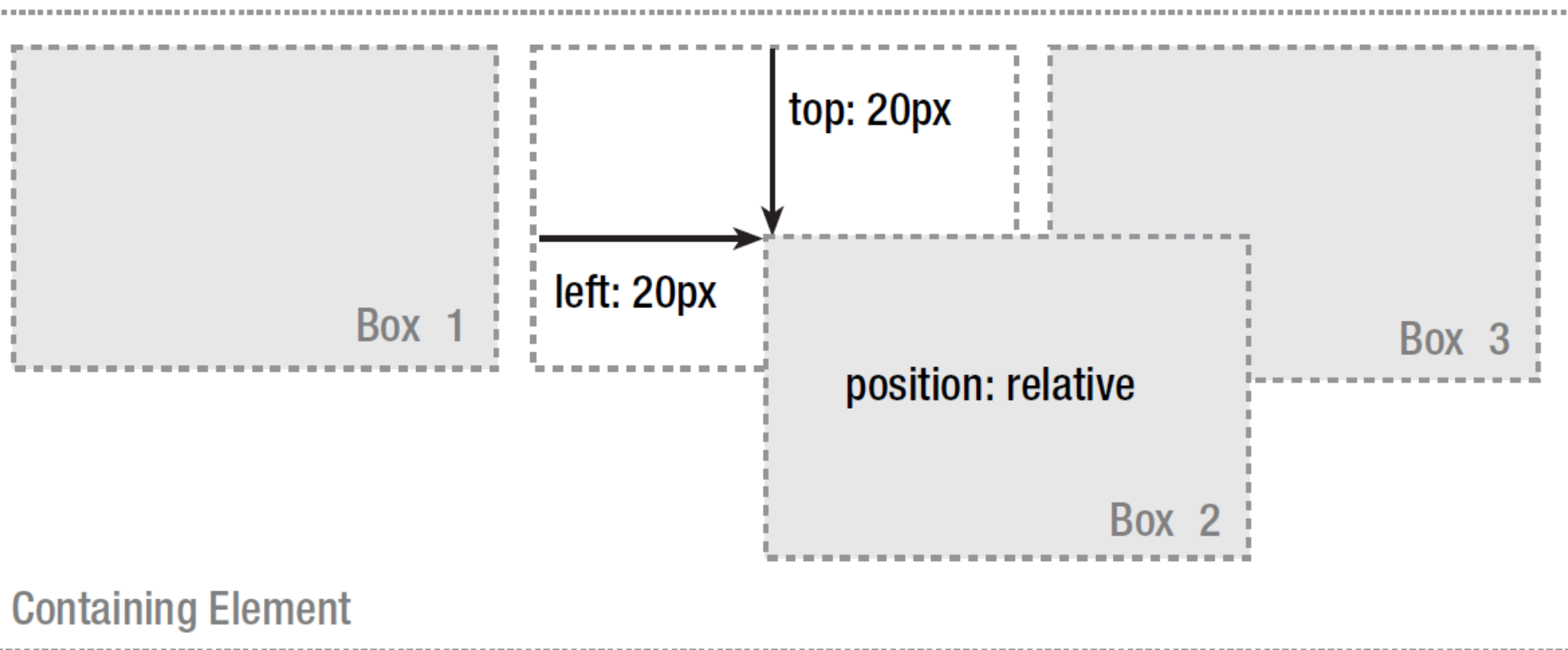
```
<p>  
<span>R2D2</span>  
<em>Torquil</em>  
<span>Fred</span>  
</p>
```

et, consectetur adipisicing elit, sed do eiusmod
ore **R2D2** **Fred** magna ali
iis nostrud exercitation **Torquil** laboris nisi ut a
is aute irure dolor in reprehenderit in voluptate
la pariatur. Excepteur sint occaecat cupidatat n
officia deserunt mollit anim id est laborum.

[Sample](#)

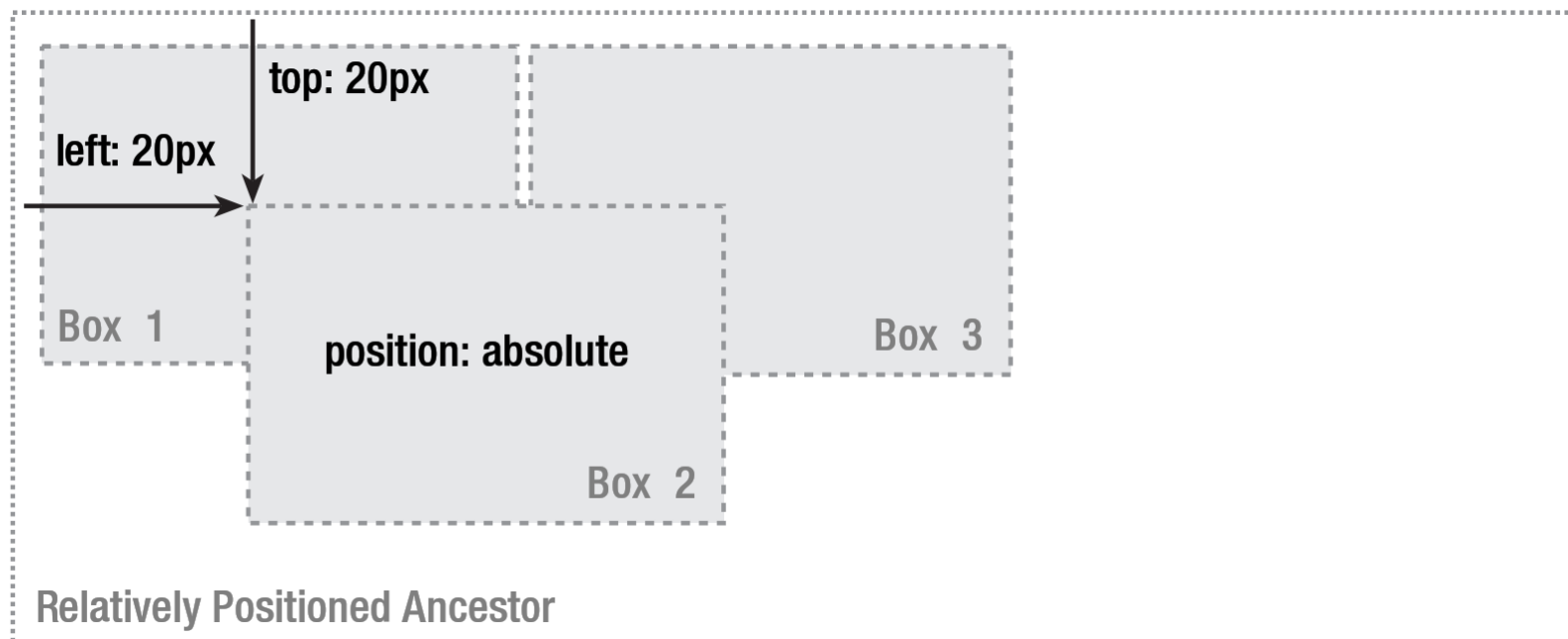
position: relative;

- 元素相对于它在文档流中的位置进行移动
- 元素原来的空间仍会保留



position: absolute;

- 绝对定位是相对于最近的已定位（即有设置 position 属性）的祖先元素移动位置
- 如果没有已定位的祖先元素，那么它的位置相对于最初的包含块（html 或是 body）



position: absolute;

```
p {  
  position: relative;  
}  
em {  
  position: absolute;  
  top: 2em;  
  left: 2em;  
}  
<p>  
<span>R2D2</span>  
<em>Torquil</em>  
<span>Fred</span>  
</p>
```

dolore eu fugiat nulla pariatur. Excepteur si
deserunt mollit anim id est laborum.

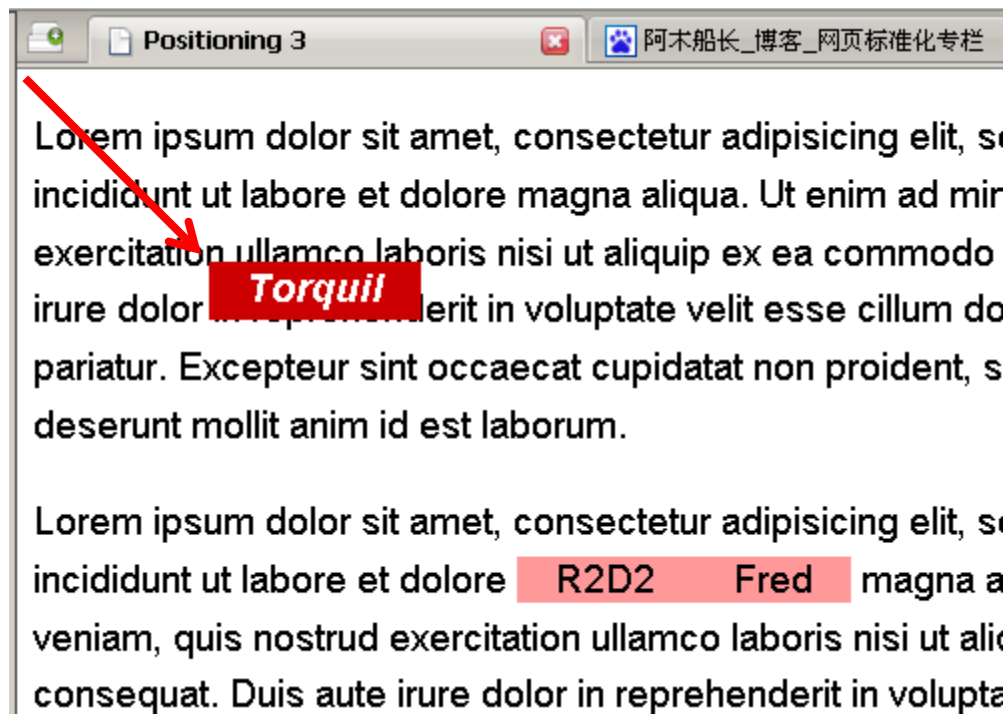
Lorem ipsum dolor sit amet, consectetur ac
et do R2D2 Fred magna aliqu
ullamco laboris nisi ut aliquip ex ea comm
voluptate velit esse cillum dolore eu fugiat r
proident, sunt in culpa qui officia deserunt r

[Sample](#)

position: absolute;

```
em {  
  position: absolute;  
  top: 5em;  
  left: 5em;  
}
```

```
<p>  
<span>R2D2</span>  
<em>Torquil</em>  
<span>Fred</span>  
</p>
```



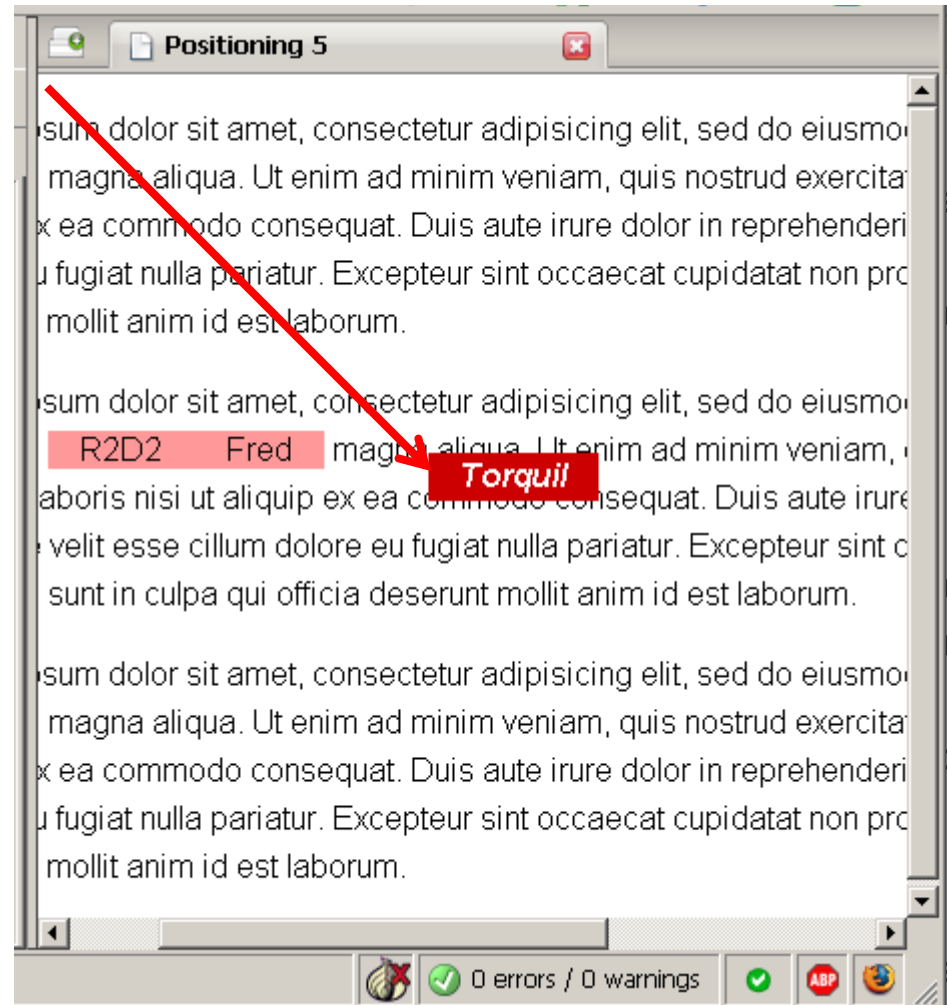
IE6 不支持, IE7 以上支持

position: fixed;

[Sample](#)

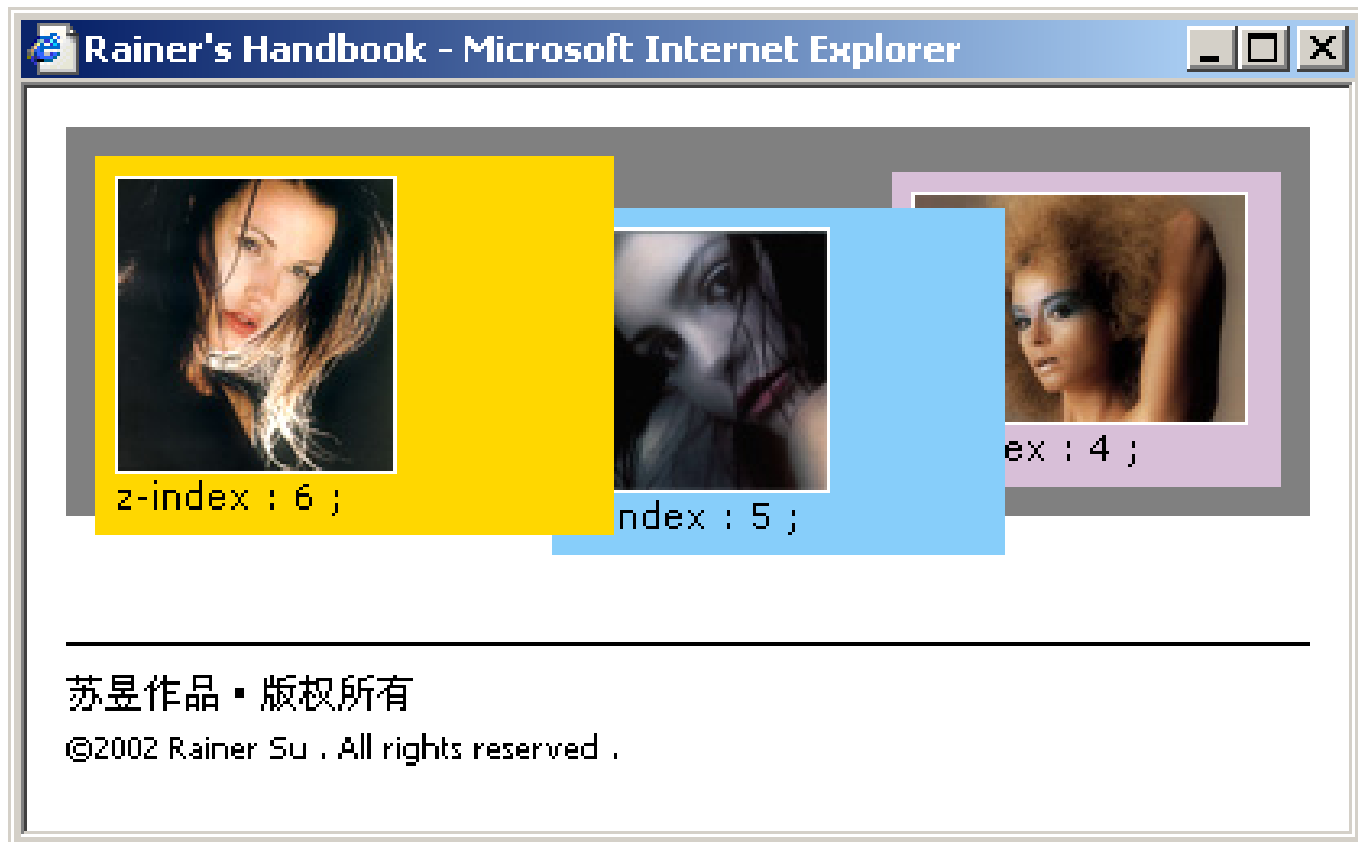
```
em {  
    position: fixed;  
    top: 200px;  
    left: 200px;  
}
```

```
<p>  
<span>R2D2</span>  
<em>Torquil</em>  
<span>Fred</span>  
</p>
```



z-index

- 定义对象在 z 轴方向的坐标，越小的数值在越底层，可以为负值。



布局属性

- float
- clear

float 浮动

- CSS 允许元素是浮动的
 - 浮动属性将一个元素盒子向左或向右移动，直到它的外边缘碰到包含元素或另一个浮动元素的边框为止。
- 可能的取值： `left`, `right` 或 `none`

```
p.aside {  
    float: right;  
    width: 10em;  
    border: 2px solid;  
    padding: 0.5em;  
}
```

float 浮动



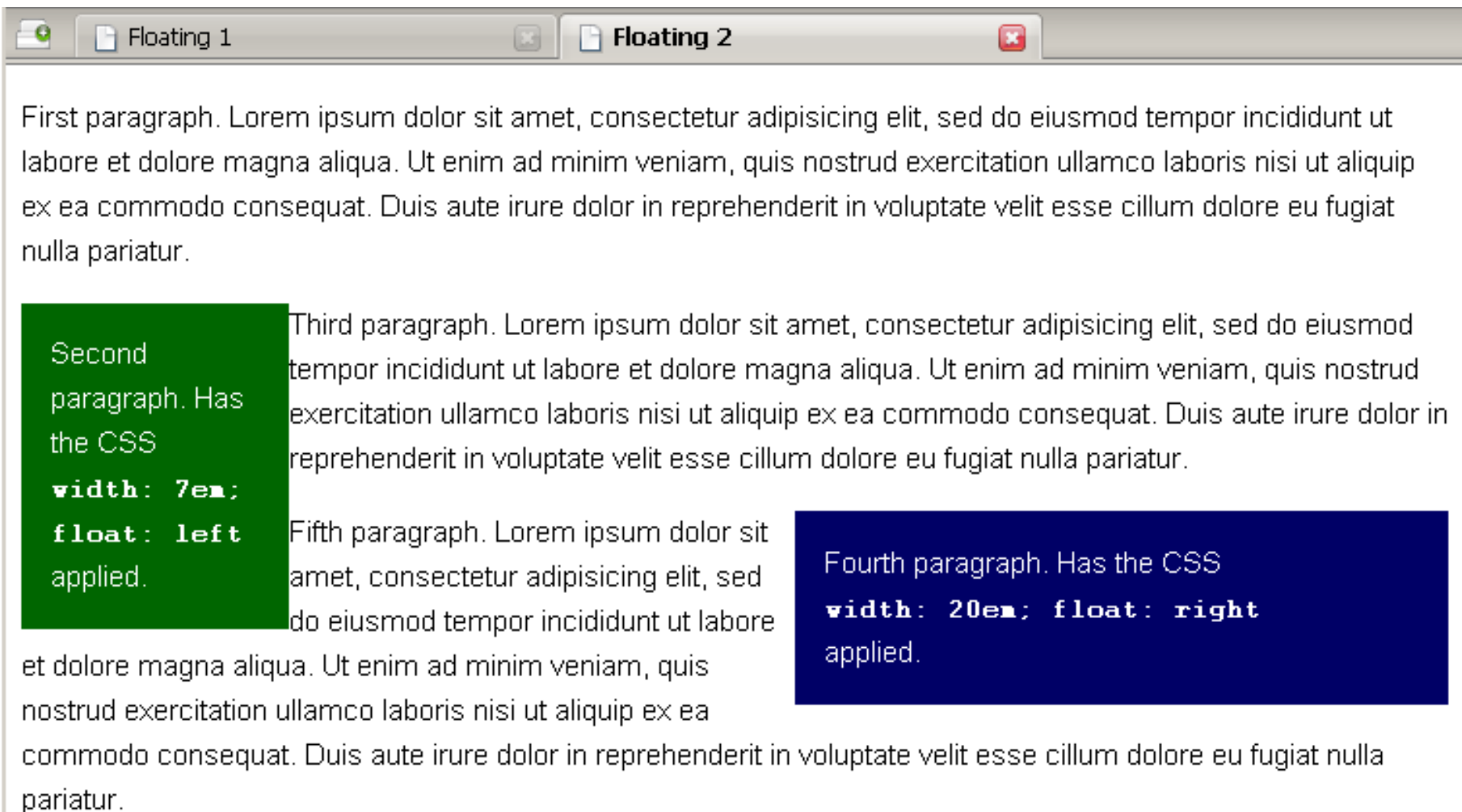
Second paragraph. Has the CSS `width: 7em;` `float: left` applied.

Third paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Fourth paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

voluptate velit esse cillum dolore eu fugiat nulla pariatur.

float 浮动



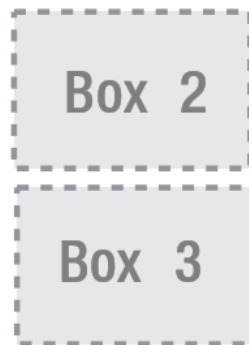
float 浮动

- 当把Box 1 向右浮动时，它脱离文档流并且向右移动，直到它的右边缘碰到包含元素的右边缘：

No boxes floated



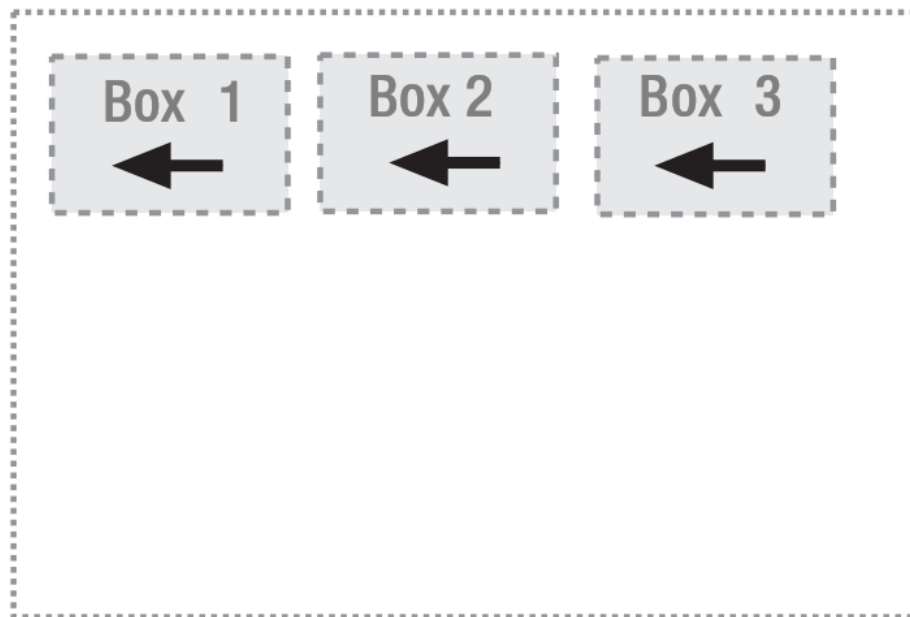
Box 1 floated right



float 浮动

- 如果三个框都向左移动，那么Box 1 向左浮动直到碰到包含盒，另外两个Box向左浮动直到碰到前一个浮动 Box。

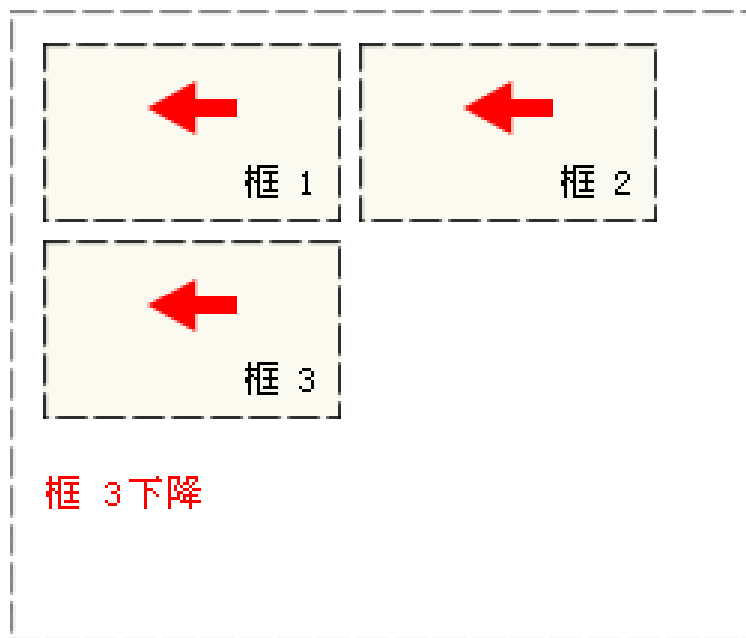
All three boxes floated left



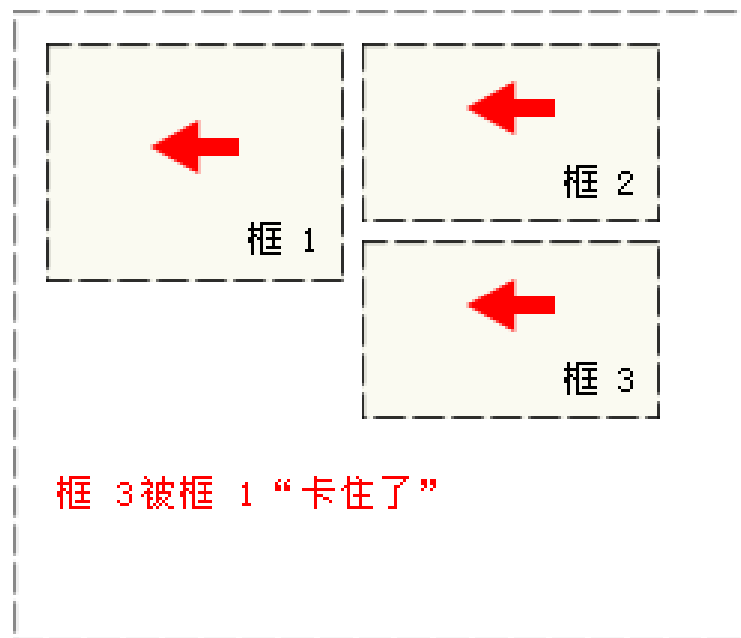
float 浮动

- 如果包含框太窄，无法容纳水平排列的三个浮动元素，那么其它浮动块向下移动，直到有足够的空间。如果浮动元素的高度不同，那么当它们向下移动时可能被其它浮动元素“卡住”：

框 1 向左浮动



所有三个框向左浮动



clear 清除

- 有时候不允许某个元素两边有浮动对象
 - 可以用 `clear` 属性实现
- `clear` 可能的值
 - `left`, `right`, `both`, `none`
- 通常用于页脚
 - `clear: both;`
 - `/* 页脚左右两边不允许浮动对象 */`

clear 清除



First paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Second paragraph. Has the CSS **width: 7em;**
float: left applied.

Third paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Fourth paragraph. Has the CSS **clear: left** applied.

Fifth paragraph. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip

显示属性

- display
- visibility

display 属性

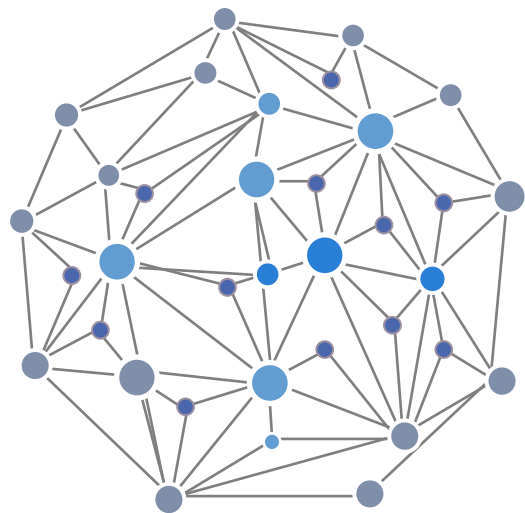
- 改变元素的显示类型
 - 内联 inline、块 block或是清除 none
 - 从而我们可以使块元素像内联元素一样使用
- display: none; 可以完全隐藏一个元素

visibility 属性

- inherit: 默认值。继承父对象的可见性
 - visible: 对象可视
 - hidden: 对象隐藏
-
- 用 visibility: hidden; 隐藏的元素，页面上仍然会保留元素原来所在空间的空白

回顾 CSS 排版相关属性

- 盒状模型
 - margin, border, padding
- 尺寸
 - width, height, ...
- 定位
 - position, top, bottom, left, right, z-index
- 布局
 - float, clear
- 显示属性
 - display, visibility



CSS 布局实例

设计一个文档的步骤

- 从考虑文档的结构开始
- 使用 `<div>` 按照内容进行分块
 - `class`和`id`的名字尽可能反映内容，而不是表现
- 尽可能选择反映结构的元素，如：
 - `<h1>`表示最高级别的标题
 - `` 和 `` 表示列表
 - 只使用 `table` 存储表格数据，而不是布局排版
- 最后考虑布局，将每一块放到合适的位置

网页十六种基本布局

十六种常见布局形式

第一排是四个基本型，后三排是简单混合十二种

这四个是基本型



CSS 布局实例

页眉 header

导航栏 navigation bar

内容栏
main content

边栏
sidebar

页脚 footer

Simple 2 column CSS layout, final layout

[Option 1](#) [Option 2](#) [Option 3](#) [Option 4](#) [Option 5](#)

Column 1

[456 Berea Street Home](#) | [Simple 2 column CSS layout](#) | [Developing with web standards index](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna. Mauris risus nunc, tristique varius, gravida in, lacinia vel, elit. Nam ornare, felis non faucibus molestie, nulla augue adipiscing mauris, a nonummy diam ligula ut risus. Praesent varius. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Nulla a lacus. Nulla facilisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce pulvinar lobortis purus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec semper ipsum et urna. Ut consequat neque vitae felis. Suspendisse dapibus, magna quis pulvinar laoreet, dolor neque lacinia arcu, et luctus mi erat vestibulum sem. Mauris faucibus iaculis lacus. Aliquam nec ante in quam sollicitudin congue. Quisque congue egestas elit. Quisque viverra. Donec feugiat elementum est. Etiam vel lorem.

Aenean tempor. Mauris tortor quam, elementum eu, convallis a, semper quis, purus. Cras at tortor in purus tincidunt tristique. In hac habitasse platea dictumst. Ut eu lectus eu metus molestie iaculis. In ornare. Donec at enim vel erat tempor congue. Nullam varius. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla feugiat hendrerit risus. Integer enim velit, gravida id, sollicitudin at, consequat sit amet, leo. Fusce imperdiet condimentum velit. Phasellus nonummy interdum est. Pellentesque quam.

Consectetur adipiscing elit

Nulla dictum. Praesent turpis libero, pretium in, pretium ac, malesuada sed, ligula. Sed a urna eu ipsum luctus faucibus. Curabitur fringilla. Suspendisse sit amet quam. Sed vel pede id libero luctus fermentum. Vestibulum volutpat tempor lectus. Vivamus convallis tempus ante. Nullam adipiscing dui blandit ipsum. Nullam convallis lacus ut quam. Mauris semper elit at ante. Vivamus tristique. Pellentesque pharetra ante at pede. In ultrices arcu vitae purus. In rutrum, erat at mollis consequat, leo massa consequat libero, ac vestibulum libero tellus sed risus. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas eu velit nec magna venenatis consequat. In neque. Vivamus pellentesque, lacus eu aliquet semper, lorem metus rhoncus metus, a ornare orci ante a diam. Nunc sem risi, aliquet quis, elementum nec, imperdiet in, wisi. Proin in lorem. Etiam molestie diam eget ante. Morbi quis tortor id lacus mollis venenatis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam vel orci sit amet tellus mollis eleifend. Donec urna diam, viverra eget, ultricies gravida, ultrices eu, erat. Proin vel arcu. Sed diam. Cras hendrerit arcu sed augue. Sed justo felis, luctus a, accumsan in, tincidunt facilisis, libero. Phasellus eu eros.

Footer

Column 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel magna.

- [Link 1](#)
- [Link 2](#)
- [Link 3](#)
- [Link 4](#)
- [Link 5](#)
- [Link 6](#)
- [Link 7](#)
- [Link 8](#)
- [Link 9](#)
- [Link 10](#)
- [Link 11](#)
- [Link 12](#)
- [Link 13](#)
- [Link 14](#)
- [Link 15](#)

Step 1a. Basic structure

- 建立 HTML 文件基本结构

```
<div id="wrap">  
  <div id="header"> </div>  
  <div id="nav"> </div>  
  <div id="main"> </div>  
  <div id="sidebar"> </div>  
  <div id="footer"> </div>  
</div>
```

Step 1a. Basic structure

- **Why wrap?**
- 我们经常会在 body 的里面，其他 div 的外面包一层 wrapper，可以方便我们对版面的宽度进行设置
- 同时又能为 body 指定一个背景图片

Step 1b. Basic structure

- 填充内容

```
<div id="wrap">
  <div id="header"> <h1>Document Heading</h1> </div>
  <div id="nav">
    <ul>
      <li><a href="#">Option 1</a> </li>
      <li><a href="#">Option 2</a> </li>
      ...
    </ul>
  </div>
  <div id="main">
    <h2>Column 1</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...
    </p>
  </div>
```


Step 2. 调整 html 元素

- 让内容与浏览器窗口的边框间没有距离

```
html, body {
```

```
  margin:0;
```


```
  padding:0;
```

```
  background:#a7a09a;
```

```
  color:#000;
```

```
}
```

指定颜色
方便调试



有些浏览器错误地将body作为根元素，因此将html和body一并写上

Step 3. 主容器

- 让 #wrap “居中显示” ， 并指定一个宽度

```
body {  
    min-width:750px;  
}  
#wrap {  
    background:#99c;  
    margin:0 auto;  
    width:750px;  
}
```

Step 3. 主容器

- 为每个部分赋予不同的背景色
- 方便调试，让我们排版时能够清楚看到它们的区域

```
#header {  
    background:#ddd;  
}  
#nav {  
    background:#c99;  
}  
#main {  
    background:#9c9;  
}  
#sidebar {  
    background:#c9c;  
}  
#footer {  
    background:#cc9;  
}
```

Step 4. 内容栏与边栏

- 为了让它们并排显示，采用 float 方式排版
- 一个浮动到 left，一个浮动到 right

```
#main {  
  background:#9c9;  
  float:left;  
  width:500px;  
}
```

```
#sidebar {  
  background:#c9c;  
  float:right;  
  width:250px;  
}
```


Step 5. 让页脚下沉

- 用 `clear:both;` 让页脚的左右没有浮动元素

```
#footer {  
  background:#cc9;  
  clear:both;  
}
```

Step 6. 修正背景色

- 我们发现，sidebar 和 main 不等长
- 通过将 sidebar 的背景色设为和 wrap 一样来弥补

```
#sidebar {  
    background:#99c;  
    float:right;  
    width:250px;  
}
```

Step 7. 让导航栏横向显示

- 修改 ul 和 li 的默认显示方式

```
#nav ul{  
  margin:0;  
  padding:0;  
  list-style:none;  
}  
#nav li{  
  display:inline;  
  margin:0;  
  padding:0;  
}
```

li 默认是块元素，会产生换行；
通过 display:inline 将其变成内联元素，就不会换行了

Step 8. 细节调整

```
#header h1 {  
  padding:5px;  
  margin:0;  
}  
#nav {  
  background:#c99;  
  padding:5px;  
}  
#main h2, #main h3, #main p {  
  padding:0 10px;  
}
```

```
#sidebar ul {  
  margin-bottom:0;  
}  
#sidebar h3, #sidebar p {  
  padding:0 10px;  
}  
#footer p {  
  padding:5px;  
  margin:0;  
}
```

2008 阿里巴巴前端招聘考题

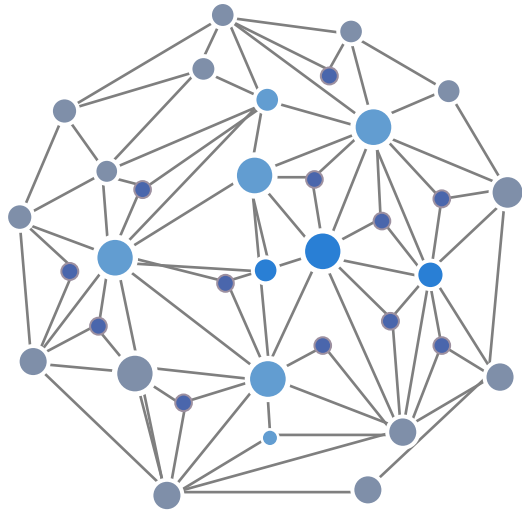
divA

divC

divB

```
<div id="wrap">  
  <div id="divA">  
  </div>  
  <div id="divB">  
  </div>  
</div>
```

```
<div id="divC">  
</div>
```



CSS Framework

CSS Framework

- 设计师的框架
 - Frameworks for Designers
- 可以帮助设计师快速高效率的书写CSS文档
- 你可以将他理解成一套模板，里面包含了大多数站点所需要的那些 CSS 类

比较著名的 CSS 框架

- 可以学习下面框架的源代码
- Twitter Bootstrap
 - <http://twitter.github.com/bootstrap/>
 - <http://www.bootcss.com/>
- 960 Grid System
- Blueprint
- ...



Blueprint

第五讲第二部分课后练习

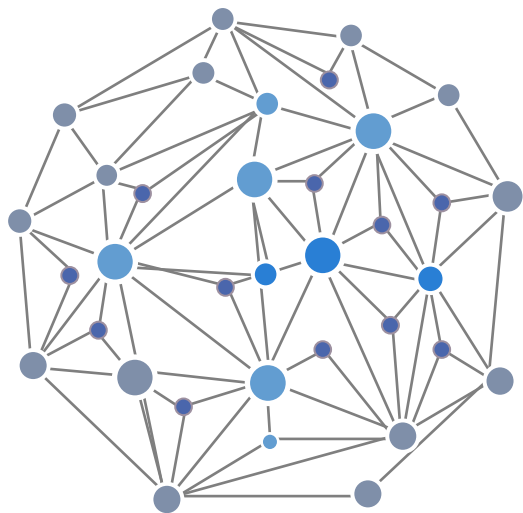
- 学习

- 《HTML5 和 CSS3 基础教程（第9版）》
第 15 章，可延伸阅读 16-20 章
- HTMLDog 指南 之 CSS 中级、高级指南
- [CSS 教程 \(w3school.com.cn\)](http://w3school.com.cn)

- 编程作业 hw3 另行通知

在线资源

- 资源
 - CSS 禅意花园
 - <http://www.csszengarden.com>
 - 优设网（国内网页设计师学习平台）
 - <http://www.uisdc.com/>
 - Eric Meyer：CSS 界泰山北斗
 - <http://www.meyerweb.com/eric/css/>



CSS 参考手册

CSS Reference

CSS Reference

- W3school CSS 中文参考手册
 - <http://www.w3school.com.cn/cssref/>
- 在线练习与参考手册
 - <http://www.css-ref.com/>

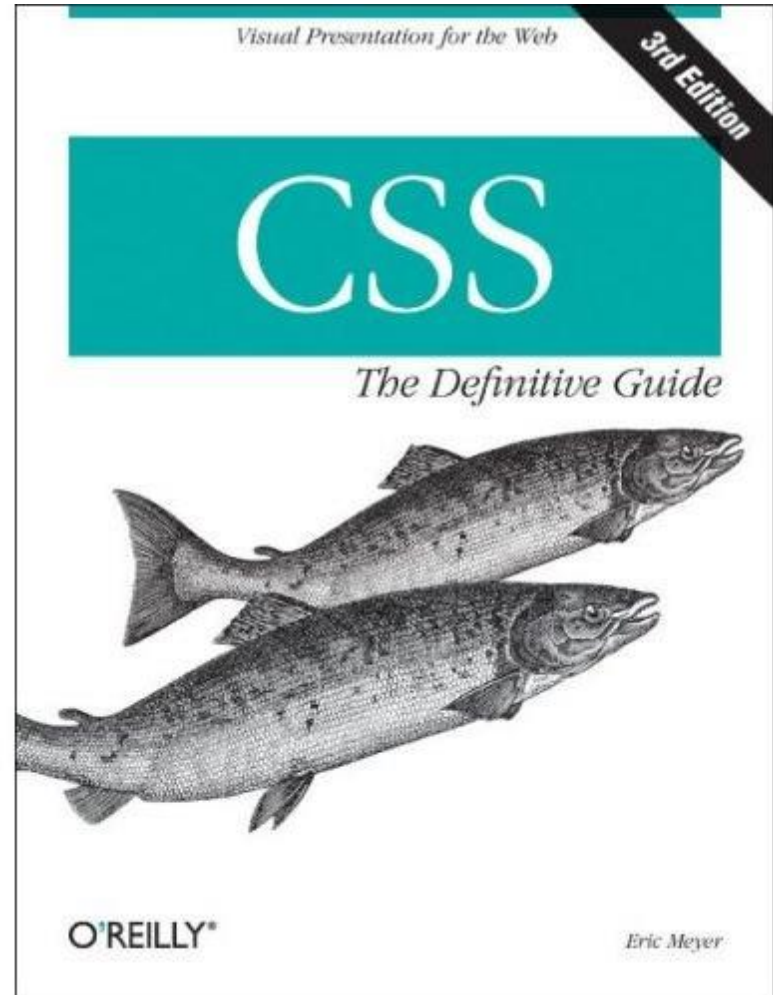
CSS Reference

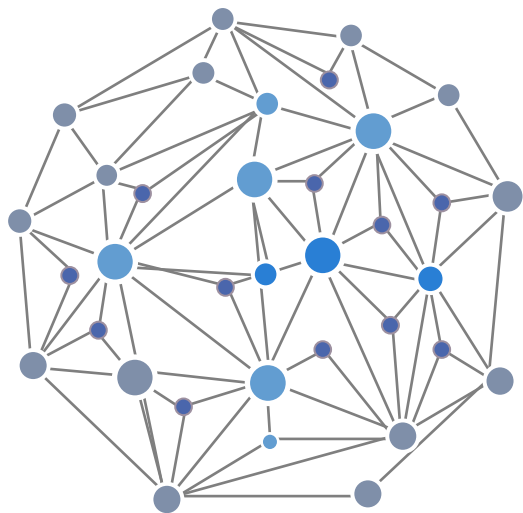
- 苏昱(苏沈小雨) 《样式表中文手册》 2.0



CSS Reference

- O'Reilly
- CSS: The Definitive Guide, 3rd Edition
- By Eric Meyer





CSS 推荐阅读材料

光光课本是不够的

CSS 推荐书籍与阅读顺序

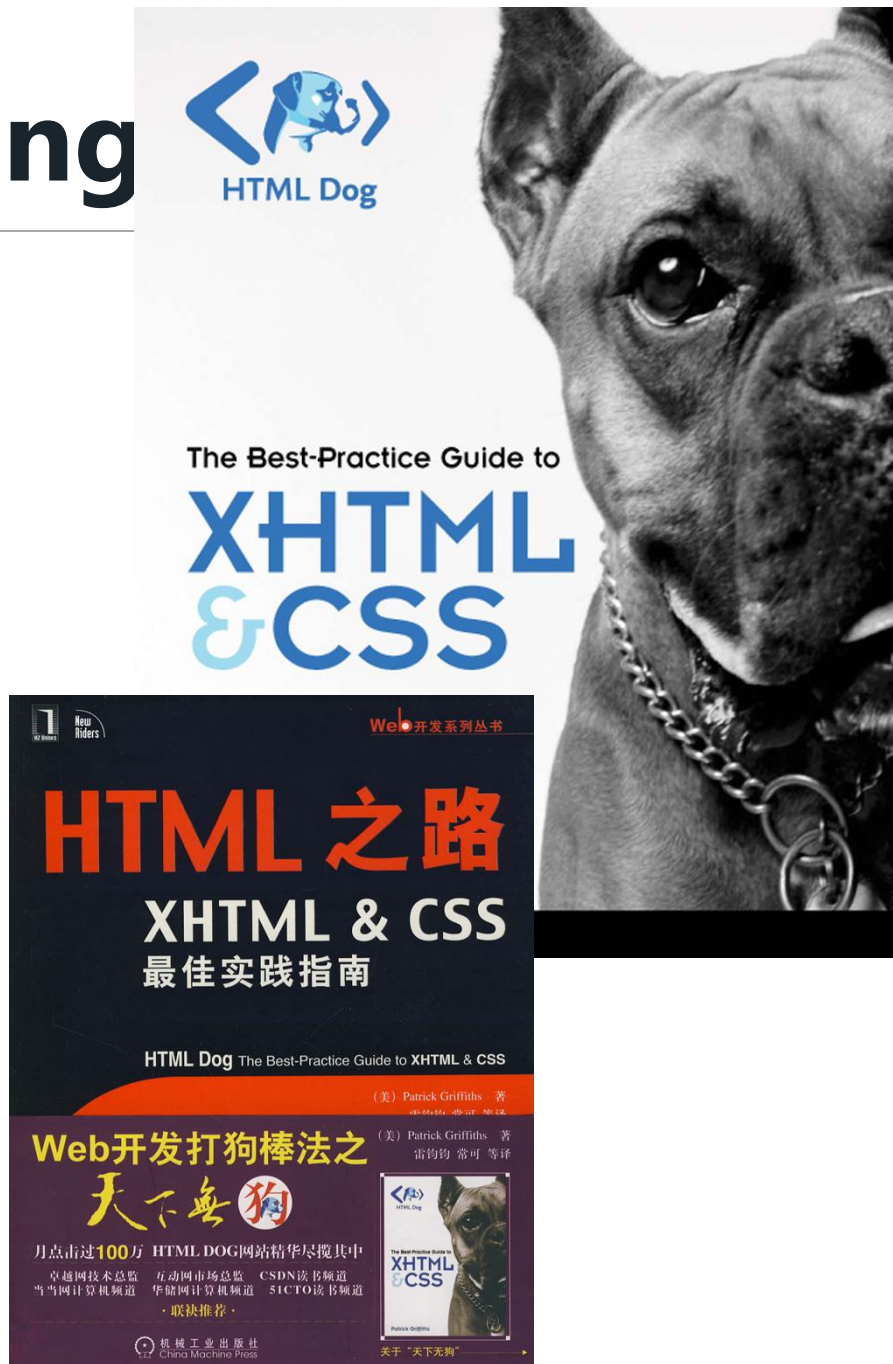
1. 课本 第3章
2. HTMLDog HTML 和 CSS 网页标准指南 之
 - CSS 初级、中级、高级指南 (属入门简明教程)
3. Stylin with CSS: A.Designers.Guide
 - 这本也可以做中级教程, 比《精通CSS》简单些, 容易理解、示例恰当、阐述简明
4. CSS Mastery
 - 精通 CSS (高级教程)
5. Bulletproof Web Design
 - 无懈可击的Web设计 (技巧类)
6. Eric Meyer on CSS 和 More Eric Meyer on CSS
 - Eric Meyer 谈 CSS 卷1 和 卷2 (技巧类)

面向页面设计师的书

- Stylin with CSS: A.Designers.Guide
 - 这本也可以做中级教程，比《精通CSS》简单些，容易理解、示例恰当、阐述简明
- The Zen of CSS Design
 - CSS 设计禅机
- The Principles of Beautiful Web Design
 - 完美网页设计艺术

Extended Reading

- HTML.Dog.The Best-Practice.Guide.to.XHTML.and.CSS
- HTML 之路：
XHTML&CSS 最佳实践指南





隐藏



后退



前进



主页



字体



选项(O)

目录(C)

索引(N)

搜索(S)

书签(U)

HTML和CSS网页标准指南

HTML初级指南

快速启动

标签、属性和元素

文档标题

段落

标题

列表

连接

图象

表格

表单

综合在一起

CSS初级指南

应用CSS

选择符、属性和值

颜色

文本

边界和补白

边框

综合

HTML中级指南

span和div

元标签

表格

定义列表

Javascript

不良标签

CSS中级指南

类class和标识id选择符

组合和嵌套

伪类

快捷属性

背景图片

HTML高级指南

CSS高级指南

HTML和CSS网页标准指南

HTMLDog指南

介绍

用符合Web标准的HTML和CSS构建你的网站

欢迎来到**传承标准的HTML和CSS网页标准指南**，此标准指南来自著名的国外XHTML和CSS的实用指南网站HTMLDog，经Neo翻译而成。最初由Ly_V整理制作CHM电子书，后由CY重新整理制作。在本指南内，你不仅学到最常用的制做网页的技术，最重要的是，你还可以深入了解实用的、最新的**Web标准（Web Standards）**。

总共六个指南，基本上覆盖了HTML和CSS的范围。初级指南是按部就班的，你应该按顺序逐个学习。而中级和高级指南相对比较独立，包含各种技巧和技术，你可以按照你的需求选择性地学习。

目录

HTML初级指南

这是按部就班的最基础教程，如果你尚未入门，你应该从这里开始。

CSS初级指南

也是按部就班的，如果你学完了HTML出家指南，进一步学习吧。

HTML中级指南

对初学者和精通者都有用的一些HTML技术。

CSS中级指南

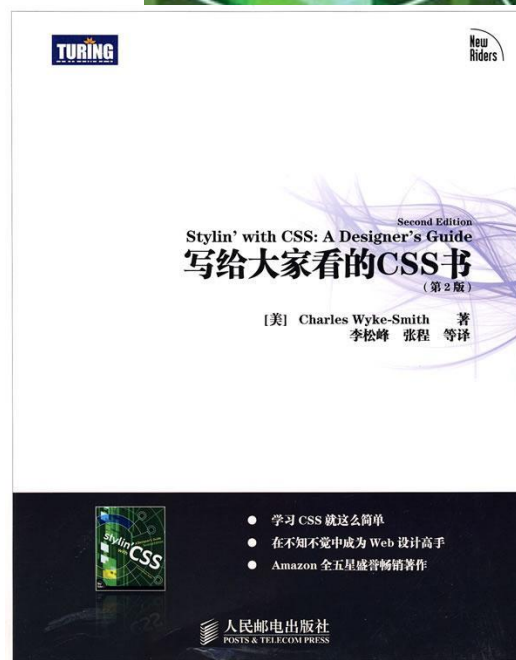
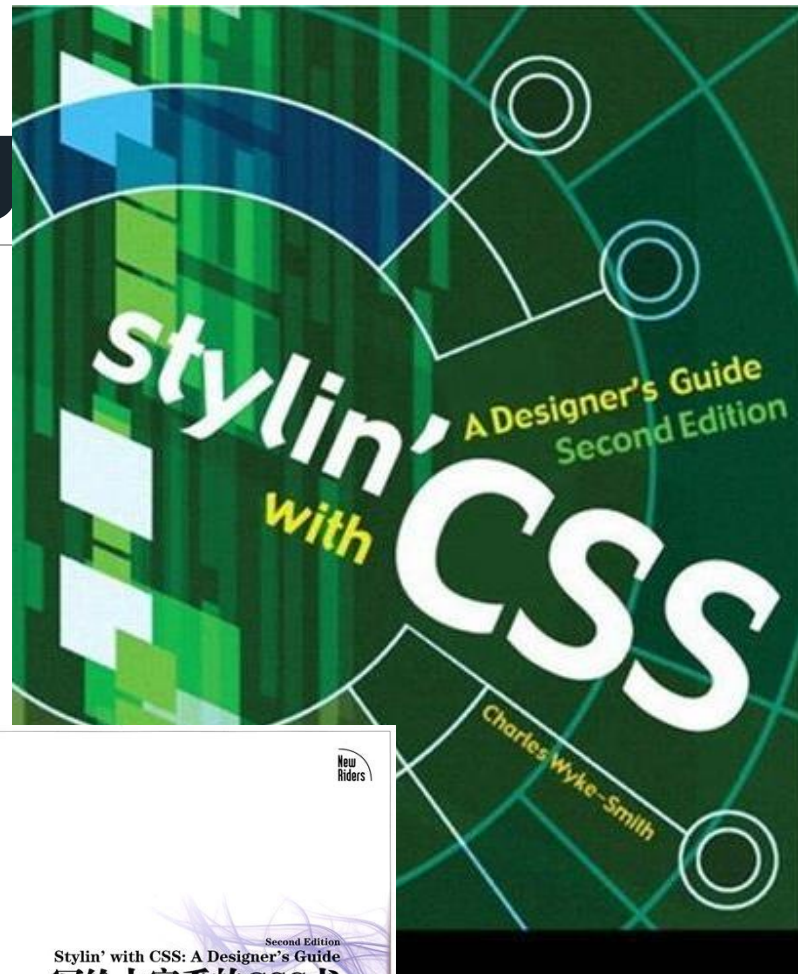
基于基本的CSS的一些“多余”和“古怪”的东西。

HTML高级指南

发挥HTML的兼容标准，增强亲和力和易用性。

Extended Reading

- **Stylin' with CSS: A Designer's Guide (2nd Edition)**
- 写给大家看的CSS书 (第2版)
- 介绍了很多排版和美工的概念，非常不错的 CSS 中级教程

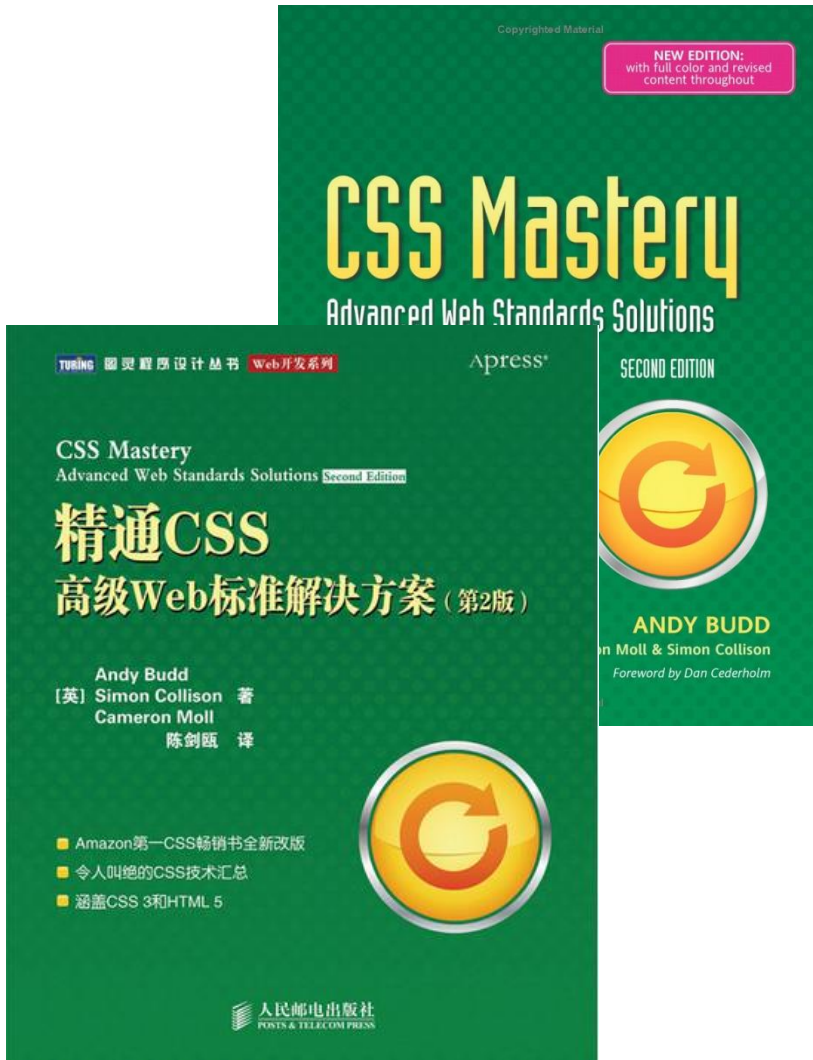


Extended Reading



- CSS: The Missing Manual
- CSS 实战手册
- 17th Annual Jolt Award Book
- 适合从初级到高级的读者

Extended Reading



- CSS Mastery: Advanced Web Standards Solutions, Second Edition
- 精通CSS：高级Web标准解决方案（第2版）
- 包括许多 hack 技巧介绍

Extended Reading

- The Zen of CSS Design: Visual Enlightenment for the Web
- CSS设计禅机——参透Web 视觉艺术玄机
 - 字体的使用
 - 色彩理论
 - 版式设计技巧
 -



Extended Reading

- **The Principles of Beautiful Web Design**

by Jason Beaird, 2007
Mar

- www.sitepoint.com

- 《完美网页设计艺术》



THANKS

本章结束

陈昱

福州大学 计算机与大数据学院 软件工程系

