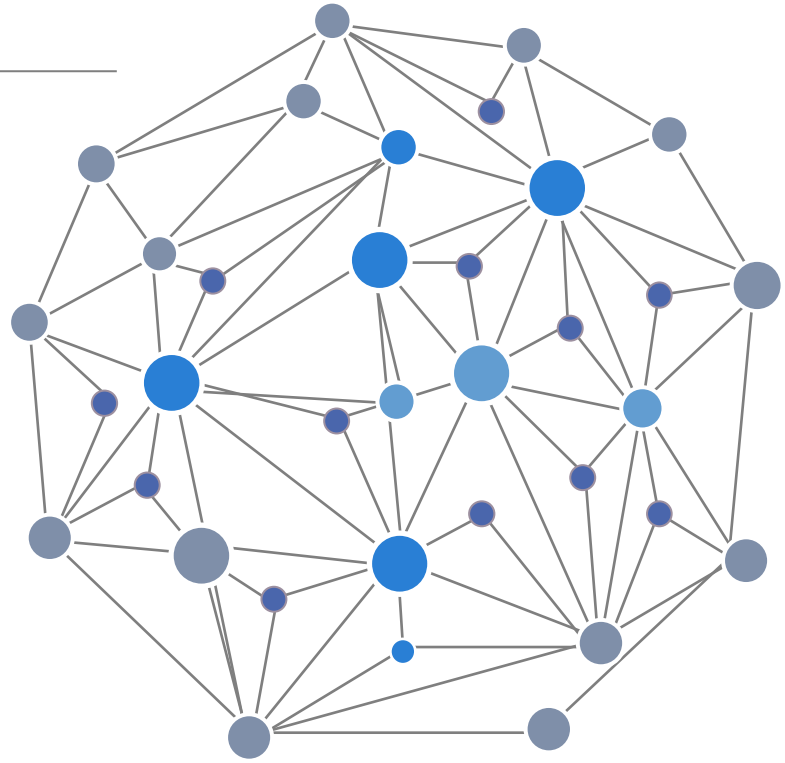
Web 程序设计

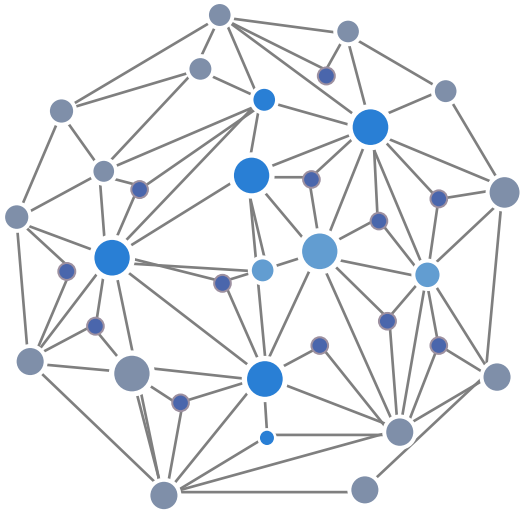


第五讲 层叠样式表 CSS (1)
CSS 基础

福州大学 计算机与大数据学院
软件工程系 陈昱
2022/10/25

内容提要

- 表现与内容分离
- 什么是 CSS?
- CSS 语法基础
- CSS 常用属性
- CSS 常用选择器



表现与内容分离

Separation of Content and Presentation

什么是表现(Presentation)与内容(Content)？

- 内容是指信息包含的实质性事物，表示信息正文及其它组成部分，如标题、表格、图片等
- 表现是指信息的展示形式，如标题使用16号黑体，正文使用12号宋体、图片采用文绕图间距10px的排列方式等

什么是表现(Presentation)与内容(Content)？

- 使用传统的HTML来表现
  - 设计方式复杂，控制区域较多
  - 代码繁杂，易读性差

- 表现与内容分离的技术
  - 设计简单，易于控制
  - 代码简单，易于修改

内容和表现相混合

```
<h1 align="center">
  <font size="7"
        face="宋体"
        color="#800080">
    <strong>一个紫色的标题</strong>
  </font>
</h1>

<h1 align="center">
  <font size="7"
        face="宋体"
        color="#800080">
    <strong>另一个紫色的标题</strong>
  </font>
</h1>
```

内容和表现相分离的版本(采用 CSS)

```
<h1>一个紫色的标题</h1>
<h1>另一个紫色的标题</h1>
```

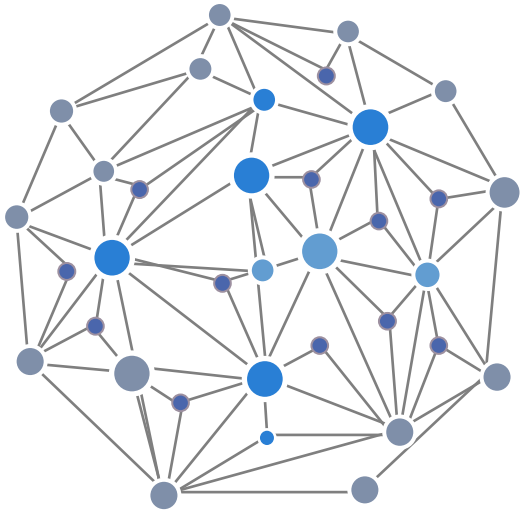
HTML

```
h1 {
    font-family: "宋体";
    font-size: 3em;
    font-weight: bold;
    color: #800080;
    text-align: center;
}
```

CSS

为什么要实现表现与内容的分离?

- 分离使得网站代码可维护性提高
- 代码重用得到可能，大幅提高开发效率
- 重用使得代码量可以成倍减少，减轻服务器负担，进而降低带宽成本
- 加快浏览器解析速度，使用户阅读更快捷迅速



什么是 CSS？

What's CSS ?

什么是 CSS？

- CSS = Cascading Styles Sheets

- CSS 描述应该如何显示元素，它是描述 HTML/XML 文档样式的语言，引入样式表的宗旨就是将结构(内容)和样式分离。

- HTML 是文档的内容，则样式表是文档的样式，或者说外观，一般保存在后缀为 *css* 的文件中

CSS 的版本 (Level)

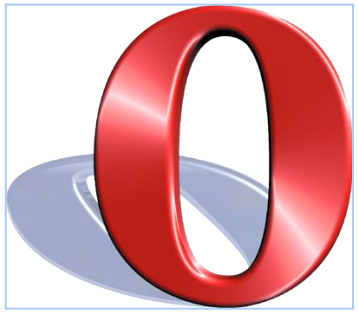
- 目前有三个版本
- CSS1
  - HTML 文档采用的样式表版本
- CSS2
  - 兼容 CSS1
  - 更好地支持基于 XML 的语言
  - 元素的绝对和相对定位
  - CSS2 revision 1
    - 修正了 CSS2 中的一些错误
    - 添加了一些新特性
- CSS3
  - 不断发展中，但已获得新一代浏览器的支持
  - CSS3 向后兼容

兼容性问题

- 历史上浏览器对 CSS 支持程度不一
  - 相同浏览器的不同版本、不同操作系统下显示不同
  - 不同浏览器对 CSS 不同版本的支持也不同
- 现代浏览器已都能较好支持标准


Internet Explorer


Firefox


Opera


Safari


Chrome

关于 CSS 的几个偏见

- 写CSS是美工做的事情，和程序员没关系？
- 从分工合作角度来说是这样
- 不过一般的美工缺乏程序设计的经验，通常需要页面制作师配合完成
- CSS 是一种领域编程语言（DSL）

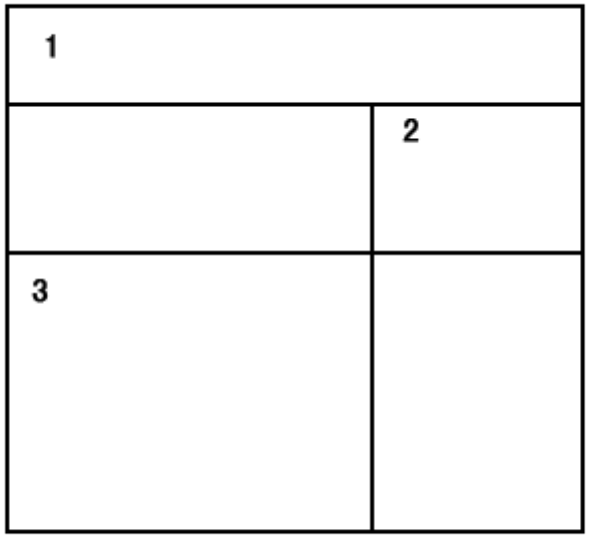
不用表格可以制作出漂亮的页面布局吗?

- No Problem ~
- CSS可以实现几乎所有用 table 实现的布局

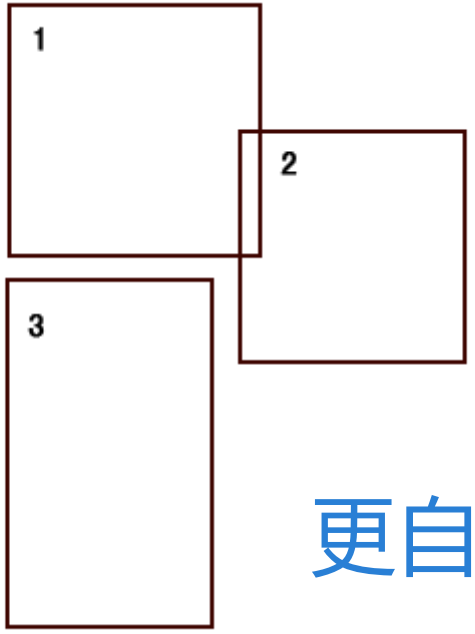
- 展现 CSS 的魅力
  - http://www.csszengarden.com/
    - http://www.mezzoblue.com/zengarden/alldesigns/

排版布局方式比较

- HTML 的掌握是表现与内容分离的基础



传统表格式排版



更自由

DIV+CSS 排版

CSS Zen Garden

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample **html file** and **css file**

Select a Design:

Under the Sea!
by Eric Stoltz

Make 'em Proud
by Michael McAghon and Scotty Reifsnyder

Orchid Beauty
by Kevin Addison

Oceanscape
by Justin Gray

CSS Co., Ltd.
by Benjamin Klemm

Sakura
by Tatsuya Uchida

Kyoto Forest
by John Politowski

A Walk in the Garden
by Simon Van Hauwermeiren

Archives

next designs »

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What Is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so

C.S.S. ZEN GARDEN

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

DOWNLOAD THE SAMPLE HTML FILE AND CSS FILE

The Beauty of Old Design

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it

Fig. A

SELECT A DESIGN

- **Under the Sea!** *by Eric Stoltz*
- **Make 'em Proud** *by Michael McAghon and Scotty Reifsnyder*
- **Orchid Beauty** *by Kevin Addison*
- **Oceanscape** *by Justin Gray*
- **CSS Co., Ltd.** *by Benjamin Klemm*
- **Sakura** *by Tatsuya Uchida*
- **Kyoto Forest** *by John Politowski*
- **A Walk in the Garden** *by Simon Van Hauwermeiren*

Fig. B

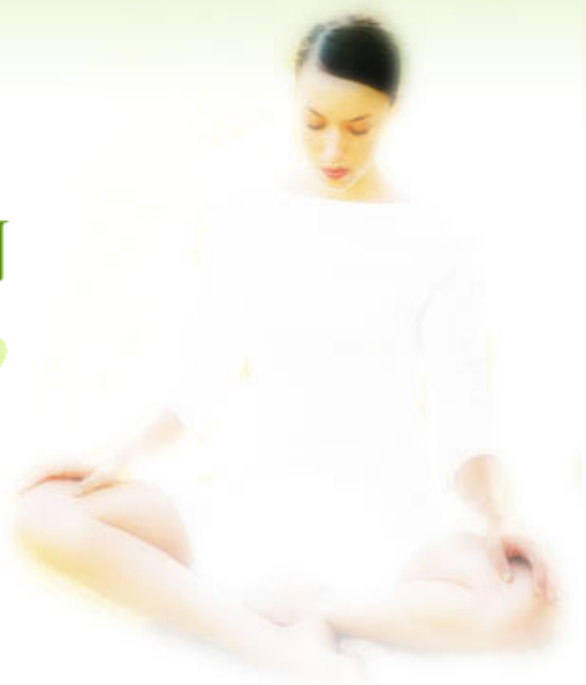
ARCHIVES

- next designs »
- View All Designs

RESOURCES

THE BEAUTY OF CSS DESIGN

CSS ZEN GARDEN



THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WASP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

A demonstration of what can be accomplished visually through CSS-based design.

Select any style sheet from the list to load it into this page

DOWNLOAD THE SAMPLE **HTML FILE** AND **CSS FILE**

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and

SELECT A DESIGN

RETRO THEATER
by **Eric Rogé**

LILY POND
by **Rose Thorogood**

CSS ZEN GARDEN

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample HTML FILE and CSS FILE

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The

Select a design

RETRO THEATER
by Eric Rogé

LILY POND
by Rose Thorogood

ICICLE OUTBACK
by Timo Virtanen

ZEN ARMY
by Carl Desmond

THE ORIGINAL
by Joachim Shotter

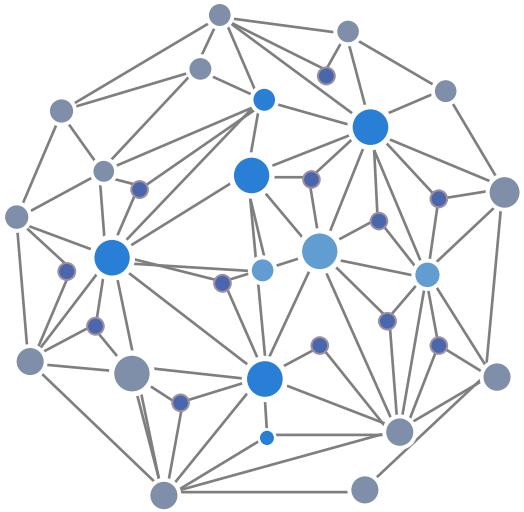
FLORAL TOUCH
by Jadas Jimmy

ELEGANCE IN SIMPLICITY
by Mani Sheriar

DAZZLING BEAUTY
by Deny Sri Supriyono

Archives

next designs »



CSS 语法基础

CSS 语法基础

- HTML 中应用 CSS 的方法
- 样式表的类型
- CSS 规则
- 选择器、属性和值

应用 CSS 的方法 — 外链式

- 把所有的 CSS 规则都写在一个或几个**单独**的文件中
- 可以使用 <link> 标记来把一个 CSS 链接到 HTML 文档中:

```
<link rel="stylesheet" type="text/css" href="styles.css" />
```

- 外部样式表可为网站的多个页面服务

应用 CSS 的方法 — 内嵌式

- 内嵌式 CSS 是写在<style> 标记中，放在文档的 <head> 里
- 里面的规则只对本文档有效

```
<head>
    <style type="text/css">
        h1 { font-size: 3em; }
    </style>
</head>
```

应用 CSS 的方法 — 内联式

- 内联样式通过元素的 **style 属性**直接套进 HTML 元素中

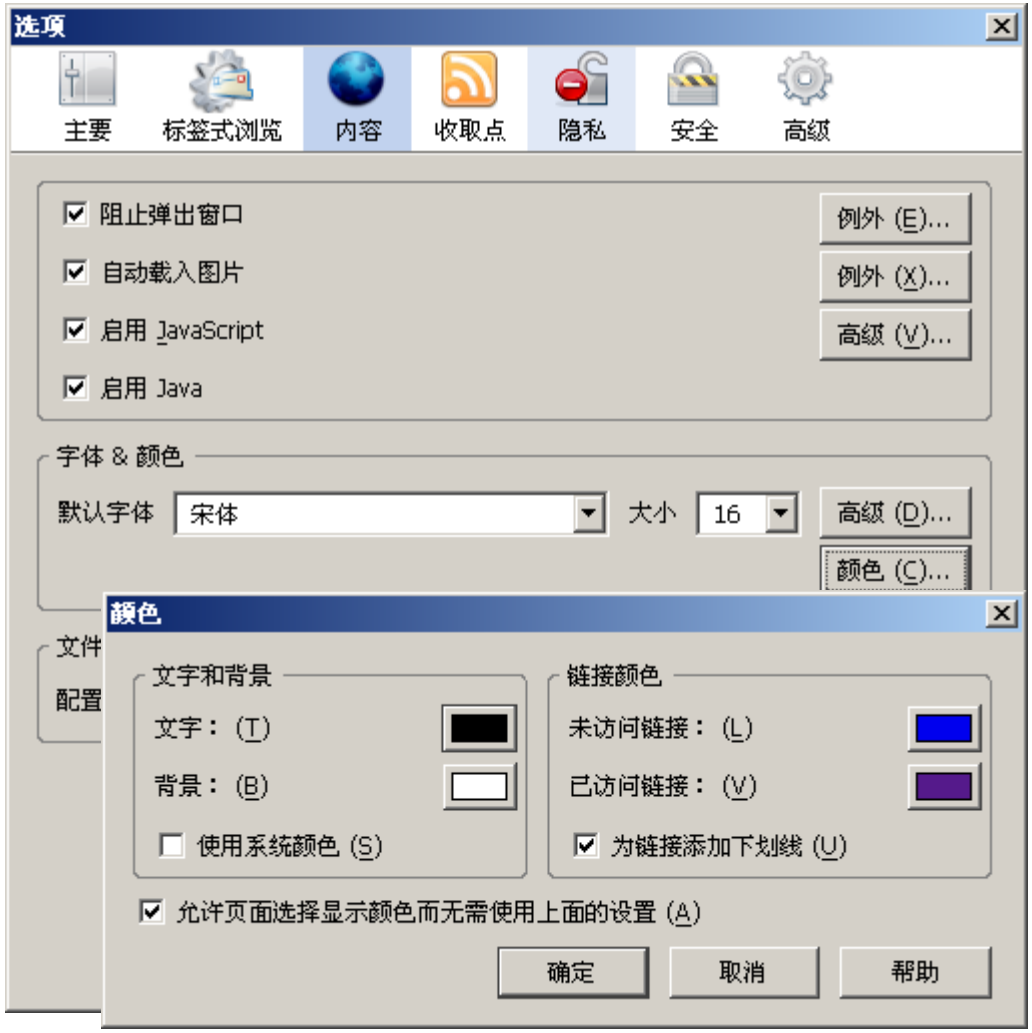
```
<p style="color: red;">text</p>
```

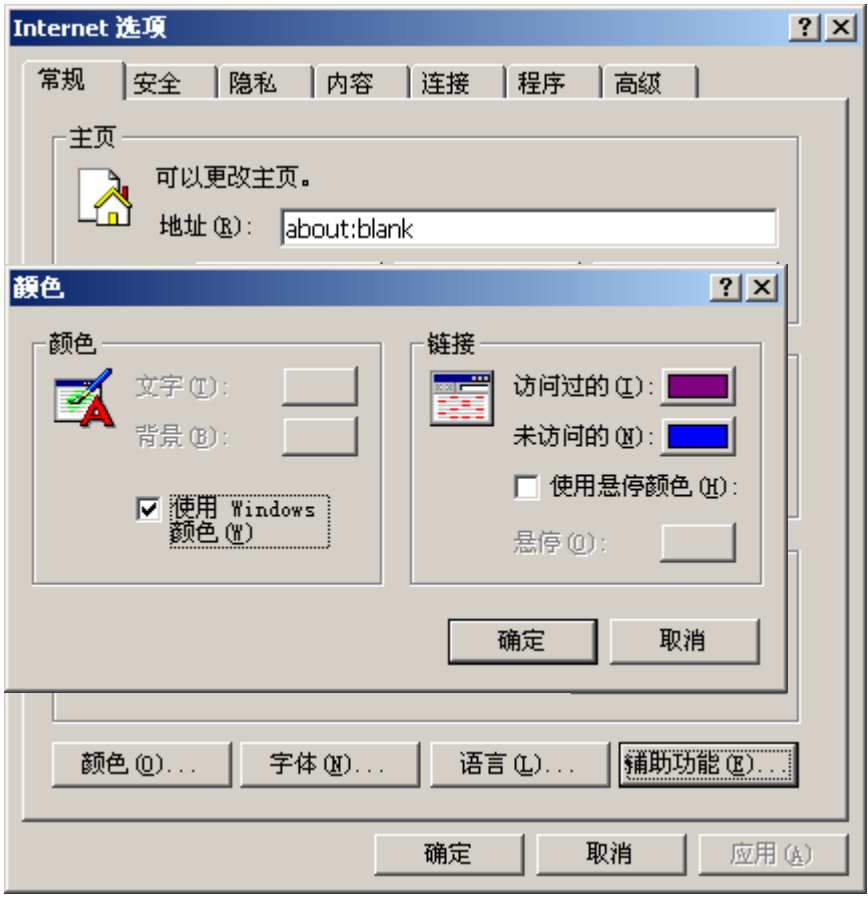
样式表的层次

样式表的层次关系到样式表规则的优先级

1. 浏览器样式表
   - 浏览器的默认样式
   - 比如：黑色的文本，蓝色的链接
2. 作者样式表
   - 由网页作者创建
   - 我们主要关注的对象
3. 用户样式表
   - 由浏览器用户创建
   - 可以覆盖其他两类样式

浏览器样式表


选项
主要
标签式浏览
内容
收取点
隐私
安全
高级
阻止弹出窗口
例外 (E)...
自动载入图片
例外 (X)...
启用 JavaScript
高级 (V)...
启用 Java
字体 & 颜色
默认字体
宋体
大小
16
高级 (D)...
颜色 (C)...
颜色
文字和背景
文字： (T)
背景： (B)
使用系统颜色 (S)
链接颜色
未访问链接： (L)
已访问链接： (V)
为链接添加下划线 (U)
允许页面选择显示颜色而无需使用上面的设置 (A)
确定
取消
帮助


Internet 选项
常规
安全
隐私
内容
连接
程序
高级
主页
可以更改主页。
地址 (R):
about:blank
颜色
颜色
文字 (T):
背景 (B):
使用 Windows 颜色 (W)
链接
访问过的 (I):
未访问的 (N):
使用悬停颜色 (H):
悬停 (O):
确定
取消
颜色 (O)...
字体 (N)...
语言 (L)...
辅助功能 (E)...
确定
取消
应用 (A)

作者样式表：分三类

- 内联样式表

```
<p style="color: red;">text</p>
```

- 内嵌样式表

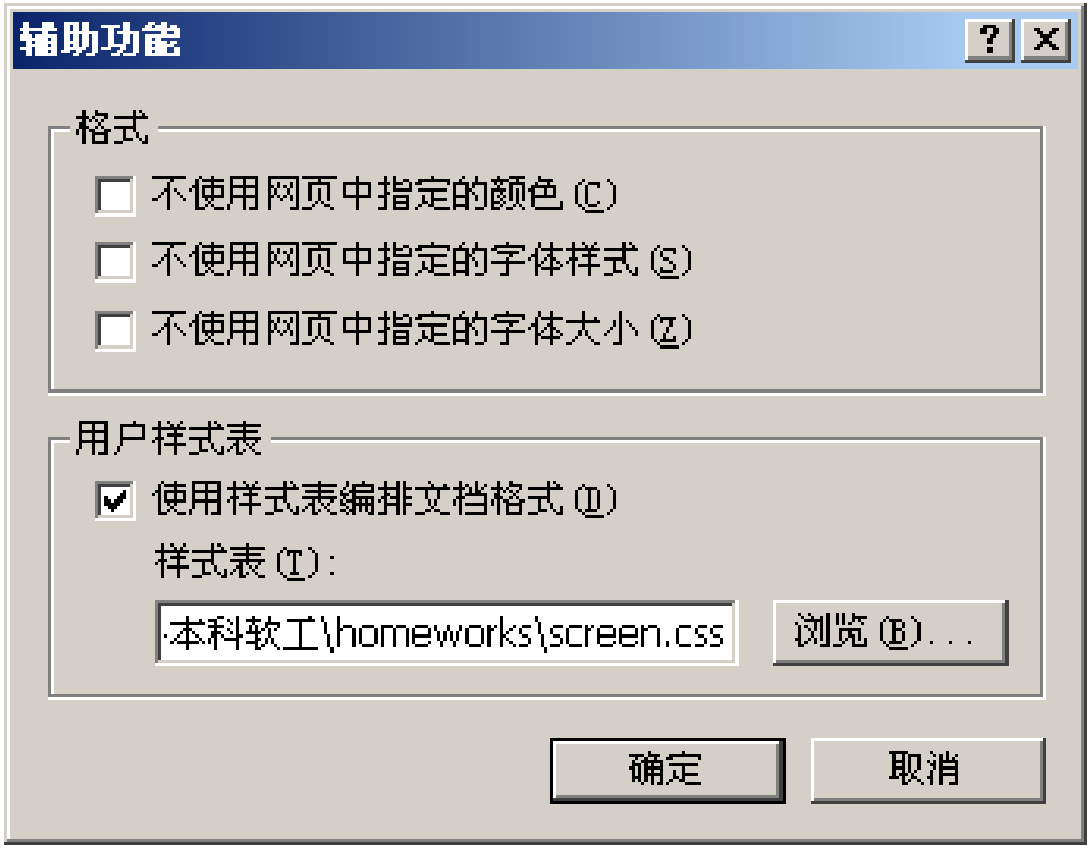
```
<style type="text/css">... </style>
```

- 外链样式表

```
<link rel="stylesheet" type="text/css" href="styles.css" />
```

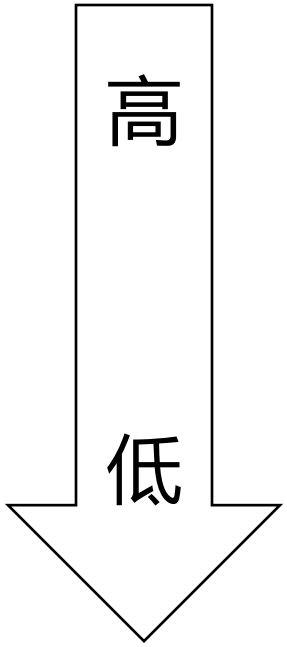
用户样式表

- 用户样式表由上网者编写，并在浏览器中调用 (不常用)



层叠优先级

- 用户样式表
- 内联样式表
- 内嵌样式表
- 外链样式表
- 浏览器样式表



- 当一个元素的样式在不同地方被重复定义的时候，依照以上顺序决定使用哪条规则

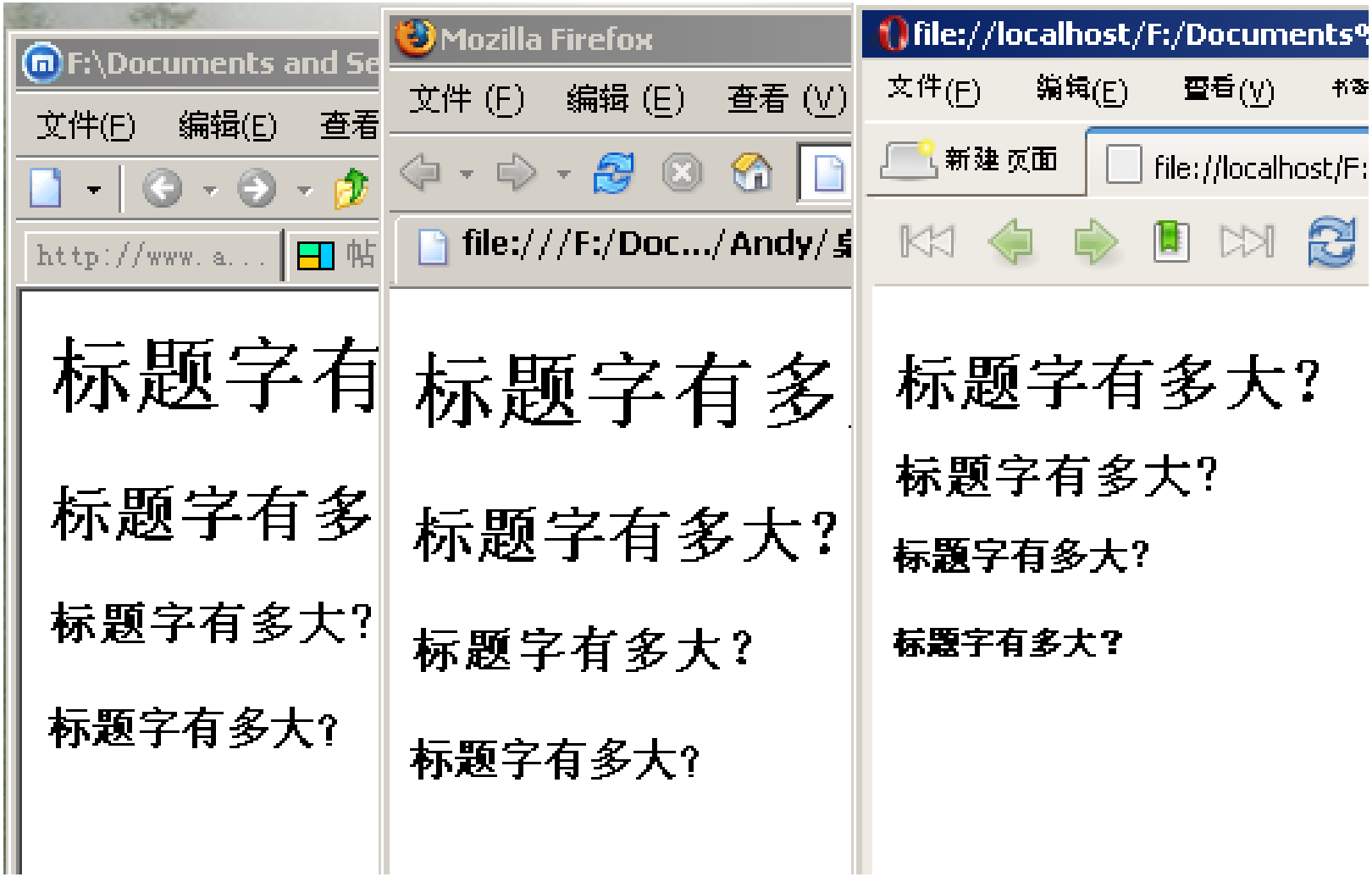
注意：
HTML 中并没有指定标记的样式！

- 在 HTML 中，所有标记都不具备默认样式

- 我们在浏览器中所看到的默认样式不过是不同浏览器自带的默认样式，而不是 HTML 自身所具备的样式

- 标记与样式没有直接的关系！

不同浏览器默认样式的不同



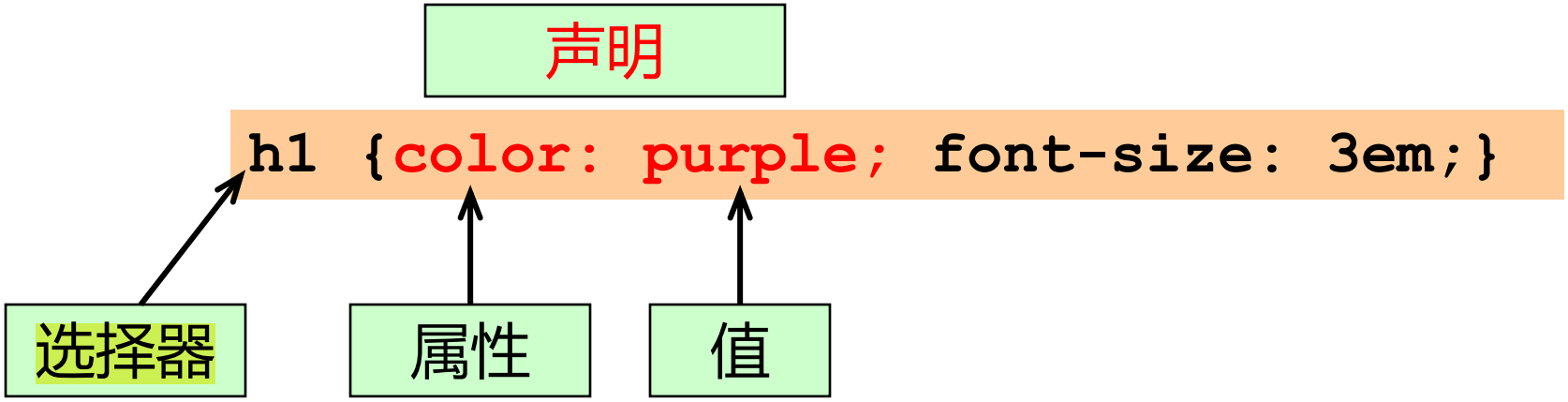
CSS 代码示例

```
/* begin : seaside-theme */
h1 {
    color: #696969;
    font-size: 14pt;
    padding : 0 0 0 0;
}
p {
    color: #4682B4;
    font-size: 12pt;
    margin: 0 4px 0 0;
}
```

规则

CSS 规则（CSS Rule）

- 一条 CSS 的规则可以为特定的元素指定具体的样式
  - eg: h1 { color: purple; }
  - 一条规则是由一个选择器和一个或多个声明组成。



选择器 （Selector）

- 选择器用来选择要赋予样式的对象
- 选择器区分大小写
- 选择器可以分组使用

```
h1, h2, p {color: purple}
```

- HTML 元素名称是最基础的一种选择器
- 此外，* 为全局选择器可以匹配所有元素

```
* {color: purple;}
```

属性 (properties) 与值 (value)

- 每个选择器都通过大括号 { } 中的属性来指定样式，诸如 color, font-size 或者 background-color

- 值在半角英文引号(:)后面，多个值之间用半角英文逗号(,)隔离

- 属性与值合称一个声明，声明之间用半角英文分号(;)隔离

属性 (properties) 与值 (value)

- 如果属性值中包含空格等特殊字符，需要用引号包裹起来，单引号和双引号皆可
  - font-family: "New Century Schoolbook", serif

- 如果属性值中包含引号，分号，感叹号，逗号等特殊字符，则必须将其进行转义
  - 例如：\' \" \: \!

注释

- /* ... */

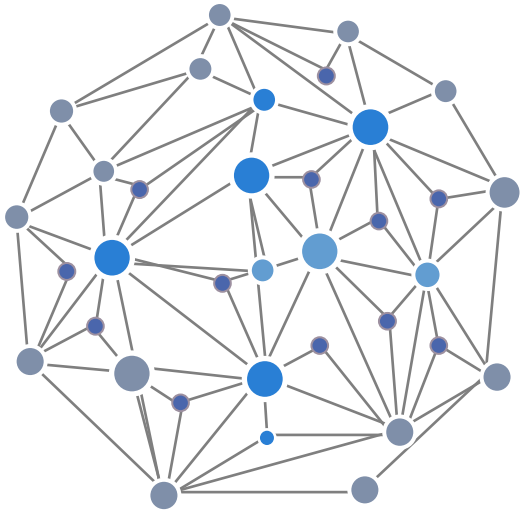
```
/* This is a comment */
p {
    text-align: center;
/* This is another comment */
    color: black;
    font-family: Arial;
}
```

W3C CSS 语法校验

- http://jigsaw.w3.org/css-validator/

CSS 的基础属性介绍

- 颜色及背景
- 字体
- 文本
- 列表



颜色及背景

颜色及背景

- 前景 color

- 背景 background
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position

前景色和背景色

- 某个元素的前景色和背景色可以通过属性 color 和 background-color 来指定

```
h1 {
    color: yellow;
    background-color: #cc9;
}
```

颜色的属性值

- 命名的颜色
  - CSS 2.1 定义了 17 种颜色名字
  - CSS 3 将定义 140 种颜色名字
- RGB 颜色
  - 用 RGB 来表示一个颜色
  - 可以用百分数或是 0-255 的数字表示
- 十六进制颜色
  - 使用十六进制代码（hex）值来表示
  - #RRGGBB 或 #RGB

颜色的属性值

- h1 { color: red; }

- h1 { color: rgb(132,20,180); }
  h1 { color: rgb(12%,200,50%); }

- h1 { color: #FF0000; }
- h1 { color: #F00; }

补充：CSS3 颜色模型

- CSS3 增强了对颜色模型的支持
  - RGB 模型添加了 Alpha 通道成为 rgba()
  - 增加了 HSL 模型：hsl() 和 hsla()
- 例如：

```
p { color: rgba(0,0,255,0.5) }
* { color: hsl(120, 100%, 75%) } /* 浅绿 */
p { color: hsla(240, 100%, 50%, 0.5) } /* 半透明的蓝色 */
```

背景图片

- background-image：图片的 URL 地址
- background-repeat：图片如何平铺
- background-attachment：图片随内容滚动
- background-position：图片的摆放位置

- 可以合并到一个 ***background* 简写属性** 中

```
background: color image repeat attachment position;
```

背景简写属性

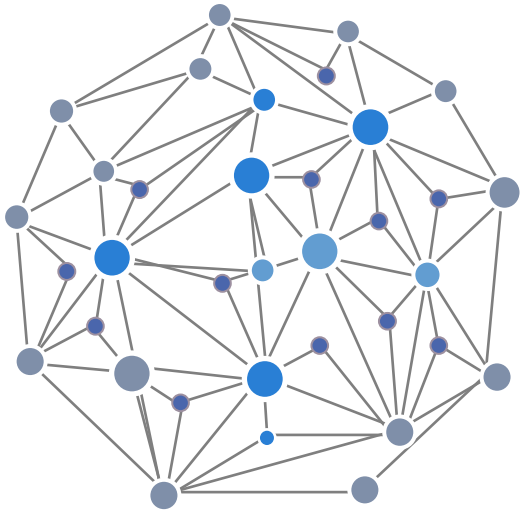
- 可以省略其中某些项，但是顺序不能错
- 属性值之间用空格隔开

```
body {
    background: white url(images/bg.gif)
            no-repeat fixed top right;
}
```

背景图片的应用

- 注意，页面上的图片根据用途分两种：

- 一种是与内容相关的图片，放在 HTML 中
  - 使用 img 标记
- 一种是纯粹装饰用的图片，放在 CSS 中
  - 使用 background-image



字体样式

字体属性

- 字体族　　font-family
- 字体尺寸　　font-size
- 字体加粗　　font-weight
- 字体风格　　font-style
- 字体变种　　font-variant

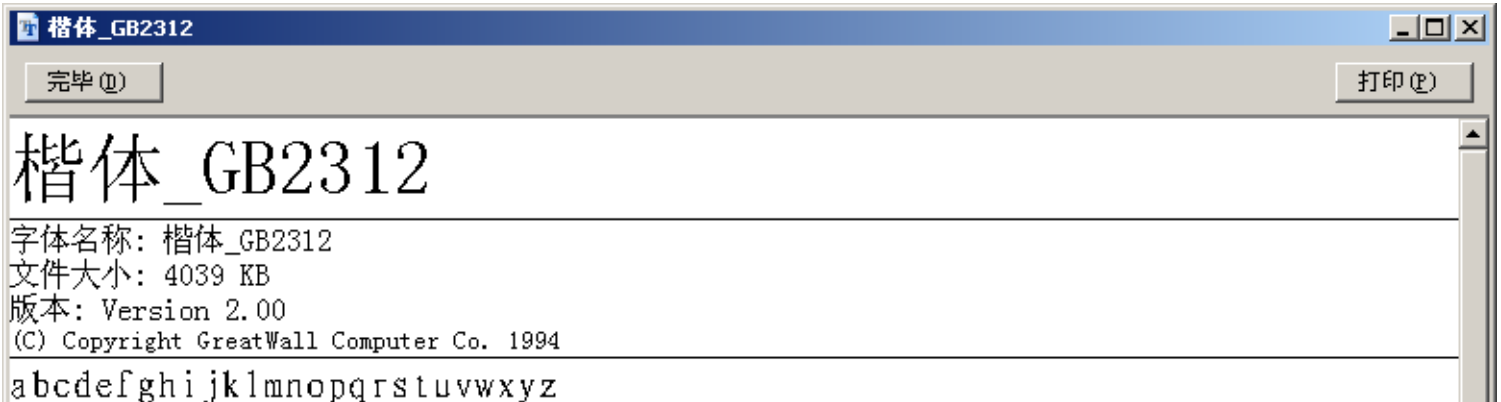
字体族 font-family

- 该属性用于指定字体名称
  - 字体本身的名字
    - “Times News Roman”, “Arial”或“Courier New”
  - 或是通用字体族的名字
    - serif, sans-serif, monospace, cursive, fantasy
    - 有衬线，无衬线，等宽字体，草书体，幻想体

- font-family: Tahoma, SimSun, Sans-Serif;

字体名称的写法

- 大多数的浏览器对字体名称以大小写无关处理，也可使用中文(不过要小心文件编码)
  - Times 与 times，Serif 与 serif 是一样的
- 字体的准确名称可以从 Windows 字体查看器查得
  - 比如 WinXP 自带楷体名称为"楷体_GB2312"



通用字体族 Generic Font-Family

Generic Name	Examples
serif	Times New Roman, Garamond
sans-serif	MS Arial, Helvetica
cursive	Caflisch Script, Zapf-Chancery
fantasy	Critter, Cottonwood
monospace	Courier, Prestige

各族常见字体

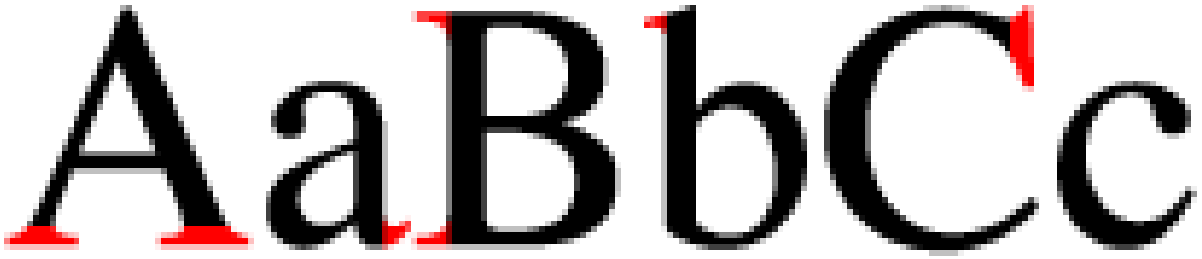
- ***Serif***: Times, Times New Roman, 宋体
- ***Sans-Serif***: Helvetica, Verdana, Tohoma, Arial, Lucida Sans
- ***Monospace***: Courier, Courier New, Consolas (Vista), Monaco (苹果机), 新宋体

- 网页正文一般推荐用无衬线字体

衬线划分

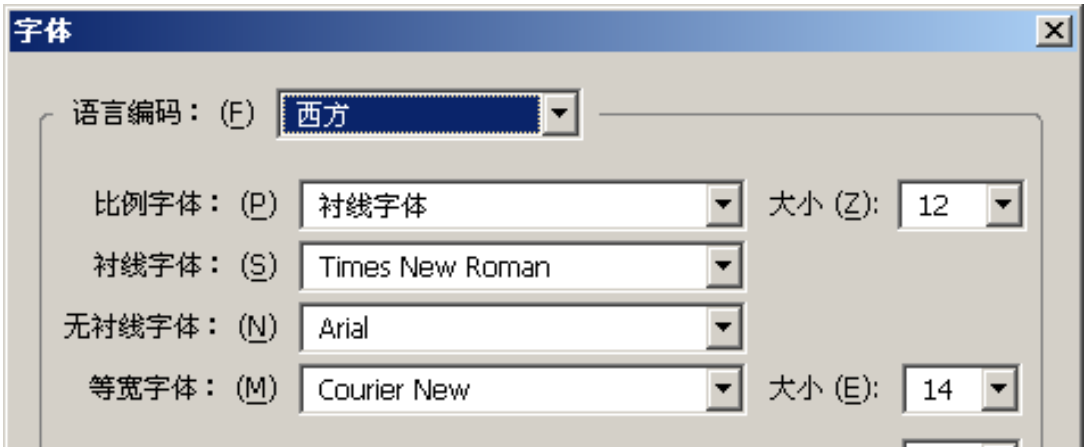
- 无衬线
- Sans-Serif

- 有衬线
- Serif



字体族 font-family

- 按照 W3C 的 CSS 规则，在 font（或者 font-family）的最后都要求指定一个 Serif 这样的字体族 (Generic-family) 的名字
  - font-family: Tahoma, SimSun, ***Sans-Serif***;
- 避免客户端在没有找到指定字体时都使用本机上的默认字体



CSS3 @font-face 规则

- 通过 CSS3，设计师可以使用他们喜欢的任意字体
- 字体文件为 eot 或 ote 或 ttf

```
@font-face {
        font-family: "French Script MT";
        font-style:  normal;
        font-weight: normal;
        src: url(frenchs4.eot);
    }
p { font-family: "French Script MT"; }
```

字体尺寸 font-size

- 可以使用绝对单位或是相对单位指定
- 或是一些关键字和百分比

- p { font-size: 12pt; }
- p { font-size: 1.2em; }

- p { font-size: small; }
- p { font-size: 80%; }

绝对长度单位

- 五个可用的绝对长度单位
  - in (Inches) 英寸
  - cm (Centimetres) 厘米
  - mm (Millimetres) 毫米
  - pt (Points, 1pt = 1/72 in) 点
  - pc (Picas, 1pc = 12pt) 12点活字
- Web 上绝对长度单位比较少用
  - 显示的大小受到显示设备大小和分辨率的影响

相对长度单位

- 三种相对长度单位
  - em　　(字体中M的高度)
  - ex　　(字体中x的高度)
  - px　　(像素,pixel)
- em 和 ex 的大小随字体大小而改变

- px = pt * 显示器DPI / 72
- 通常 DPI = 96，因此 12pt = 16px

关键词

绝对大小关键词：

- xx-small | x-small | small | medium | large | x-large | xx-large

相对大小关键词：

- larger | smaller

百分比

- 使用百分比设定的大小，将在容器元素（即父元素）的文字大小基础上改变

```
body { font-size: 12px; }
p { font-size: 80%; }
```

- 那么 p 的字体大小为：12px * 80%

继承 CSS 属性

- HTML 中，子元素会继承父元素的 CSS 属性
- 子元素定义的属性与父元素重复时，子元素属性覆盖父元素属性

```
body { color: blue; font-size: 12px; }
p      { font-size: 80%; }

<body> <p> text </p> </body>
```

字体加粗

- **字体加粗 font-weight**
  - normal（普通）
  - bold（粗体）
  - bolder
  - lighter
  - 100，200，～，900

其他字体属性

- **字体变种 font-variant**
  - font-variant: normal　　(缺省值，正常显示)
  - font-variant : small-caps（以小型大写字母方式显示)
- **字体风格 font-style**
  - font-style: normal（正常）
  - font-style: italic（意大利体）
  - font-style: oblique（倾斜体）

字体的简写属性

- font 简写属性完整形式：
  - **font : font-style || font-variant || font-weight || font-size/line-height || font-family**
- **注意：顺序不能错，但是可以省略其中某几项**

```
p { font: italic small-caps 600 12px/18px Courier; }
```

文本属性

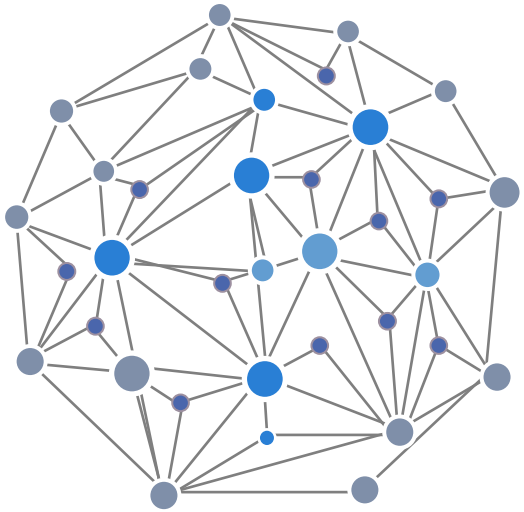
- 字母间隔 letter-spacing
- 单词间隔 word-spacing
- 设置行高 line-height
- 文本对齐 text-align
- 文本缩进 text-indent
- 文本装饰 text-decoration
- 文本转换 text-transform

文本属性

- **文本装饰 text-decoration**
  - text-decoration: overline
  - text-decoration: line-through 删除线
  - text-decoration: underline
  - text-decoration: none 取消下划线（常用于超链接）

文本属性

- **文本转换 text-transform**
  - text-transform: capitalize
    - 把每个单词的首字母转换成大写
  - text-transform: uppercase
    - 把所有的字母都转换成大写
  - text-transform: lowercase
    - 把所有的字母都转换成小写
  - text-transform: none



列表

列表

- list-style-type 可以改变列表项标记的类型
  - ul { list-style-type: decimal; }
  - ol { list-style-type: lower-roman; }
  - 注意：无序列表和有序列表不再局限于默认的样式，无序列表前面可以有编号，反之亦然
- list-style-image 设置列表项标记的图像
  - list-style-image : url(images/rdl_arrow.gif);
- list-style-position 改变列表项标记的位置

详细说明：http://www.w3school.com.cn/css/pr_list-style.asp

显示自定义风格的列表

Aircraft Types

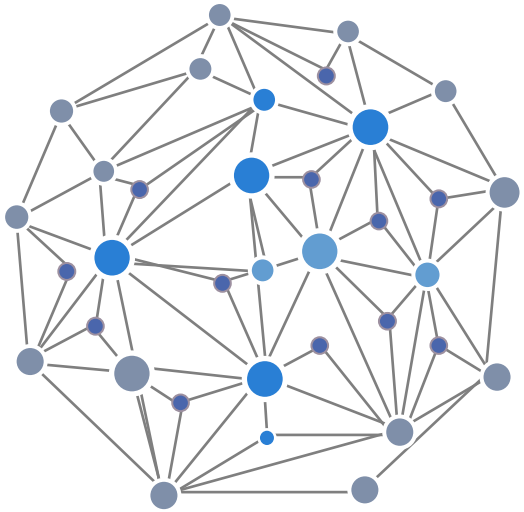
I. General Aviation (piston-driven engines)
    A. Single-Engine Aircraft
        1. Tail wheel
        2. Tricycle
    B. Dual-Engine Aircraft
        1. Wing-mounted engines
        2. Push-pull fuselage-mounted engines
II. Commercial Aviation (jet engines)
    A. Dual-Engine
        1. Wing-mounted engines
        2. Fuselage-mounted engines
    B. Tri-Engine
        1. Third engine in vertical stabilizer
        2. Third engine in fuselage

list-style-image

```
ul {
  list-style: square inside url('/i/eg_arrow.gif')
}
```

```
<ul>
  <li>咖啡</li>
  <li>茶</li>
  <li>可口可乐</li>
</ul>
```

- 咖啡
- 茶
- 可口可乐



CSS 的选择器类型

CSS Selectors

Selector: 为样式找到目标元素

- 元素选择器
- 全局选择器
- 类(class)选择器
- 标识(id)选择器
- 伪类选择器
- 伪元素选择器
- 后代选择器
- 子代选择器
- 相邻兄弟选择器

元素选择器

- 元素选择器：几乎所有的 HTML 标记名都可以作为选择器，也称为“类型选择器”
- 全局选择器：*，匹配所有元素
- 同样的样式定义可以用于多个选择器，用逗号隔开（选择器分组）

```
*        { margin: 0; padding: 0; }
a        { text-decoration:none; }
th, td { font-size:14px; width:120px; }
```

class 和 id 属性

- 绝大多数 HTML 元素都可以设定两个属性 class , id

- 同一个 class 属性的值可以应用于文档中的多个元素，表明它们是同一类事物，具有某些相同的样式

- 一个 id 属性值只能应用一个元素

class 和 id 属性

- XHTML 中没有用于定义内容区域或者导航栏的专用元素
- 虽然可以使用 XML 的方法自定义标记，不过太复杂了，现阶段不太现实
- 折衷的办法是通过添加 class 和 id 为现有元素赋予额外的意义

Sample

```
<ul id="mainNav">
  <li><a href="#">Home</a></li>
  <li><a href="#">About Us</a></li>
  <li><a href="#">Contact</a></li>
</ul>
```

类（class）选择器

- 为多个相同类名的元素指定相同的样式
- 在CSS中，类选择器放在一个**半角英文句点（.）**之后

Sample

```
.ppl {color: purple;}
```

```
<h1>A normal heading</h1>
<h1 class="ppl"> and a purple
one</h1>
<h1>or a heading with<span
class="ppl">some purple</span> in
the middle
```

特定的类选择器

- 可以为某种指定的XHTML元素应用选择器
- 元素名放在类名前面

Sample

```
.ppl {color: purple;}
span.ppl {font-style: italic}
```

```
<h1>A normal heading</h1>
<h1 class="ppl"> and a purple
one</h1>
<h1>or a heading with<span
class="ppl">some purple</span> in
the middle
```

联合类选择器

- 在CSS和标记中都可以联合使用类选择器

```
.red {color: red;}
.alert {font-weight: bold}
.alert.red {background: black}
```

```
<h1 class="red">A red heading</h1>
<h1 class="alert">An alert</h1>
<h1 class="red alert">Red Alert!!!!</h1>
```

Sample

IE6 支持联合类选择器有问题

标识（id）选择器

- id 选择器放在**半角英文井号**（#）之后
- id 和 class 不同之处在于，id 用在唯一的元素上，而 class 则可用在不止一个的元素上

```
#special {color: purple; background: black}
```

```
<h1>A normal heading</h1>
<h1 id="special"> and a special one</h1>
```

Sample

伪类（Pseudo-class）选择器

- 伪类选择器可以将样式应用到元素的动态属性上

selector:pseudo-class { property: value; }

:link	用在未访问的链接上
:visited	用在已经访问过的链接上
:active	用于选定（被点击）的链接上
:hover	用于鼠标光标置于其上的链接
:focus	用于获得焦点的输入空间上(如 input)

提示：在 CSS 定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。a:active 必须被置于 a:hover 之后，才是有效的。

Pseudo-class example

```
a {font-family: Arial; font-size: 32px;}
a:link {color: red}
a:visited {color: green}
a:hover {color: black; font-size: 48px;}
a:active {color: yellow}
```

```
<a href="http://www.fzu.edu.cn/">FuZhou
University</a>
<h1>Everything else gets pushed around</h1>
<a href="http://www.google.com/">Google Web
site</a>
```

Sample

伪元素(Pseudo-element)选择器

- 使用形式和伪类选择器类似

:first-letter	应用样式到元素的第一个字
:first-line	应用样式到元素的首行
:before	应用样式到元素的前面
:after	应用样式到元素的后面

- 配合 content 属性，在不更改 HTML 的情况下改变内容的两边事物

Pseudo-element example

```
body{background-color: rgb(80%,80%,80%);}
p {font-family: Arial; font-size: 24px;}
p:first-line {color: purple}
p:first-letter {font-size: 200%}
p:before {content: "before content ";
color: red}
p:after {content: " after content"; color:
blue}
```

Sample

- IE6 不支持 :before 和 :after

后代选择器

- 用来寻找特定元素的后代
- **E1 E2 {** *sRules* **}**

- 选择所有被 E1 包含的 E2
- **不管 E2 是 E1 的第几层的子元素**

- li a {text-decoration: none;}
- …

子代选择器

- 后代选择器选择一个元素的所有后代
- 而子代选择器只选择元素的直接后代，即子元素
- **E1 > E2** { *sRules* }

- #nav > li { font-size:14px; }

- IE6 不支持

相邻兄弟选择器

- 相邻兄弟选择器使用了加号（+）
- **E1 + E2 { *sRules* }**
- h1 + p {margin-top:50px;}
  - 选择紧接在 h1 元素后出现的段落，h1 和 p 元素拥有共同的父元素

结合使用各类选择器

- html > body table + ul {margin-top:20px;}

- 解释为：选择紧接在 table 元素后出现的所有兄弟 ul 元素，该 table 元素包含在一个 body 元素中，body 元素本身是 html 元素的子元素。

更多选择器写法

- 参见选择器手册
  - http://www.w3school.com.cn/cssref/css_selectors.asp

第五讲课后

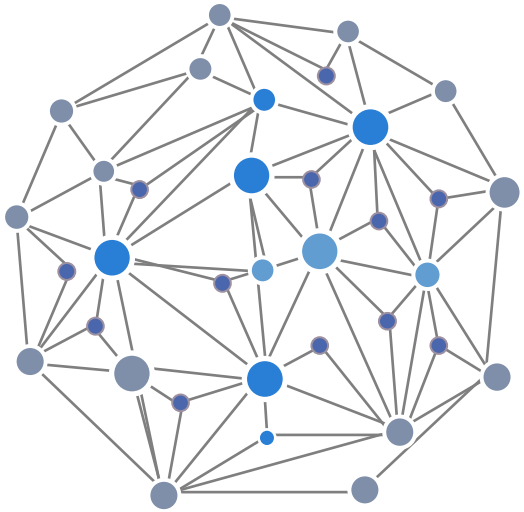
- 学习
  - 《HTML5 和 CSS3 基础教程（第9版）》
    第 11，12，13，14 章

  - HTMLDog 指南 之 HTML 中级指南
  - HTMLDog 指南 之 CSS 初级、中级指南

在线教程

- W3School CSS 教程
  - http://www.w3school.com.cn/css/
  - http://www.w3school.com.cn/css3/

- 许多实例
  - http://www.w3school.com.cn/example/csse_examples.asp
  - http://www.htmldog.com/examples/



附录

题外话：浏览器的分类

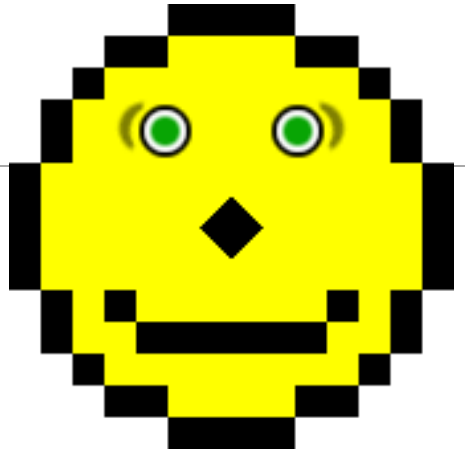
- 按照渲染引擎(rendering engine)进行分类
- 目前的3大渲染引擎
  - **Trident (也叫MSHTML, IE的核心 IE4~IE8)**
  - **Gecko (Mozilla 系, v1.8->v1.9)**
  - **KHTML and WebKit**

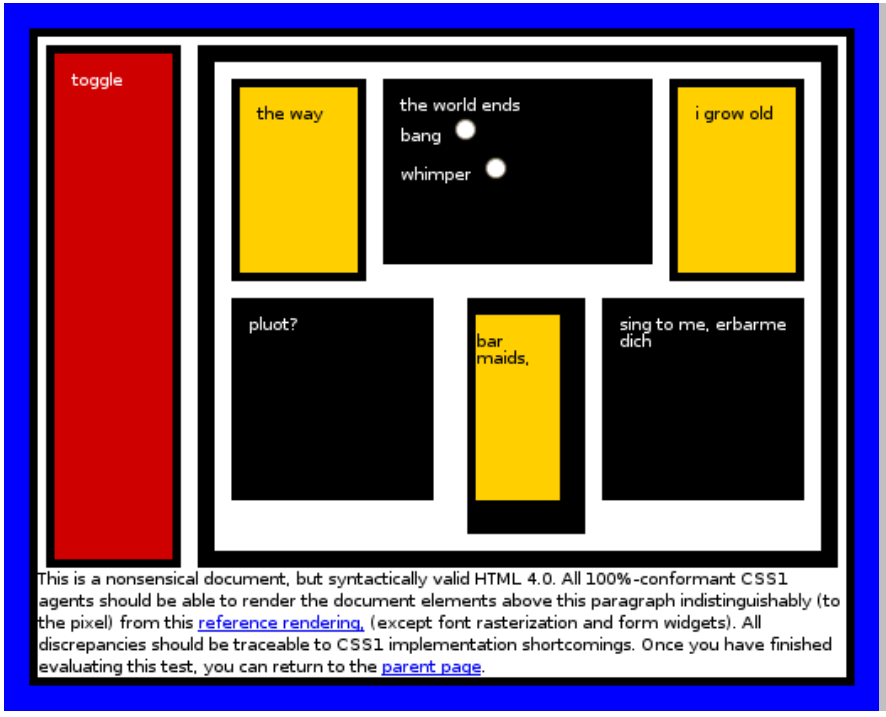
    (KDE Konqueror, GNOME Epiphany, NOKIA S60, Apple Safari, Google Chrome/Android)
- **网页代码->渲染引擎->屏幕上看到的图像**

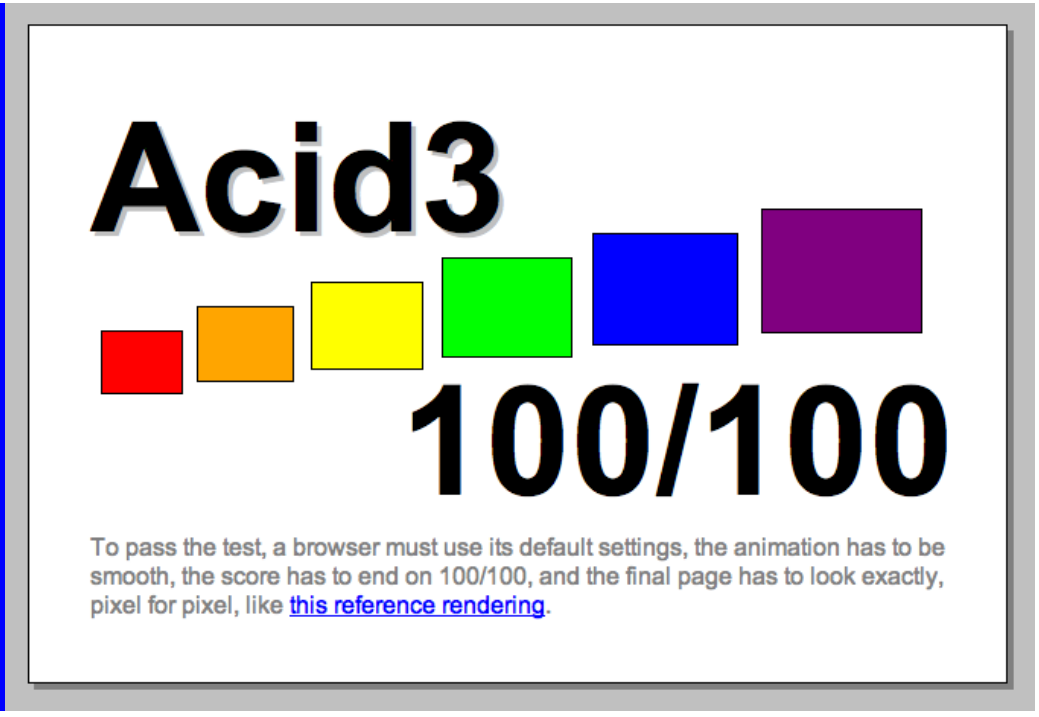
题外话： ACID 测试

- ACID 是网页标准计划小组（WaSP）设计的一套浏览器与标准的兼容性的测试软件
- Acid 1　CSS 1.0
  - http://acid1.acidtests.org/
- Acid 2　HTML/CSS 2.0/PNG
  - http://acid2.acidtests.org
- Acid 3　ECMAScript / DOM level 3
  - http://acid3.acidtests.org/

ACID 1/2/3 测试图




toggle
the way
the world ends
bang
whimper
i grow old
pluot?
bar
maids,
sing to me, erbarme dich
This is a nonsensical document, but syntactically valid HTML 4.0. All 100%-conformant CSS1 agents should be able to render the document elements above this paragraph indistinguishably (to the pixel) from this reference rendering, (except font rasterization and form widgets). All discrepancies should be traceable to CSS1 implementation shortcomings. Once you have finished evaluating this test, you can return to the parent page.


Acid3
100/100
To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like this reference rendering.

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