

# 网页编程基础 CSS3基础

WebBasic CSS3BASIC

Unit01

# 内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	CSS 概述
	10:30 ~ 11:20	CSS 语法
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	尺寸与边框
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



# CSS概述

---

## CSS 概述

```
graph LR; A[CSS 概述] --> B[CSS 概述]; A --> C[使用 CSS 样式表]; B --> D[CSS 的作用]; B --> E[什么是 CSS]; B --> F[CSS 与 HTML 之间的关系]; B --> G[HTML 属性与 CSS 样式的使用原则]; C --> H[使用 CSS 样式表的方式]; C --> I[内联方式使用 CSS]; C --> J[内部样式表]; C --> K[外部样式表];
```

### CSS 概述

CSS 的作用

什么是 CSS

CSS 与 HTML 之间的关系

HTML 属性与 CSS 样式的使用原则

### 使用 CSS 样式表

使用 CSS 样式表的方式

内联方式使用 CSS

内部样式表

外部样式表

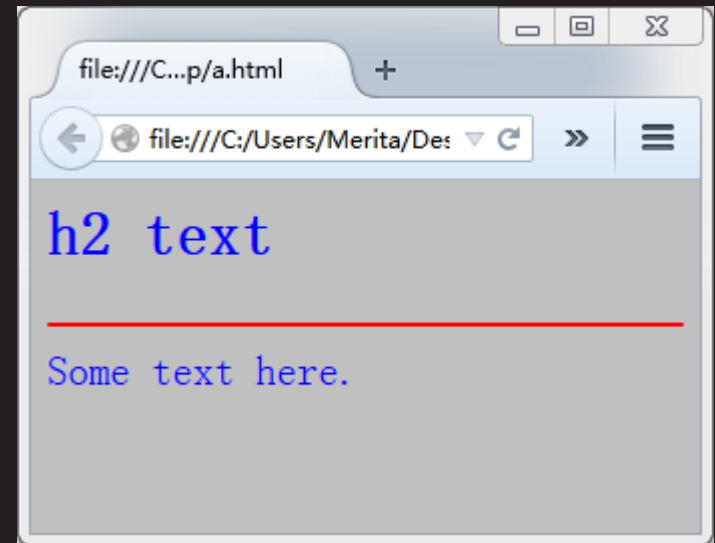
# CSS 概述



# CSS 的作用

- 查看如下代码

```
<body bgcolor="silver" text="blue">
  <h2>h2 text</h2>
  <hr color="red" />
  Some text here.
</body>
```



如果使用 HTML 标记中的属性来定义样式，各元素对于样式的属性定义各不相同！



# CSS 的作用（续1）

```
<html>
  <head>
    <style type="text/css">
      body
      {
        background-color:silver;
        color:blue;
      }
      h2
      {
        background-color:orange;
        color:green;
      }
      hr { color:red; }
    </style>
  </head>
```

使用 CSS 定义样式



# CSS 的作用（续2）

```
<body>
  <h2>h2 text</h2>
  <hr />
  Some text here.
</body>
</html>
```



使用 CSS：

各元素使用统一的样式声明！

且提高了样式的可重用性和可维护性！



# 什么是 CSS

- CSS ( Cascading Style Sheets ) : 层叠样式表, 又叫级联样式表, 简称样式表
- 用于 HTML 文档中元素的样式定义
  - 实现了将内容与表现分离
  - 提高代码的可重用性和可维护性





# CSS 与 HTML 之间的关系

- HTML 用于构建网页的结构
- CSS 用于构建 HTML 元素的样式
- HTML 是页面的内容组成，CSS 是页面的表现

```
<style type="text/css">
p{
  color:red;
  font-size:14px;
  text-align:center;
}
</style>
```

表现

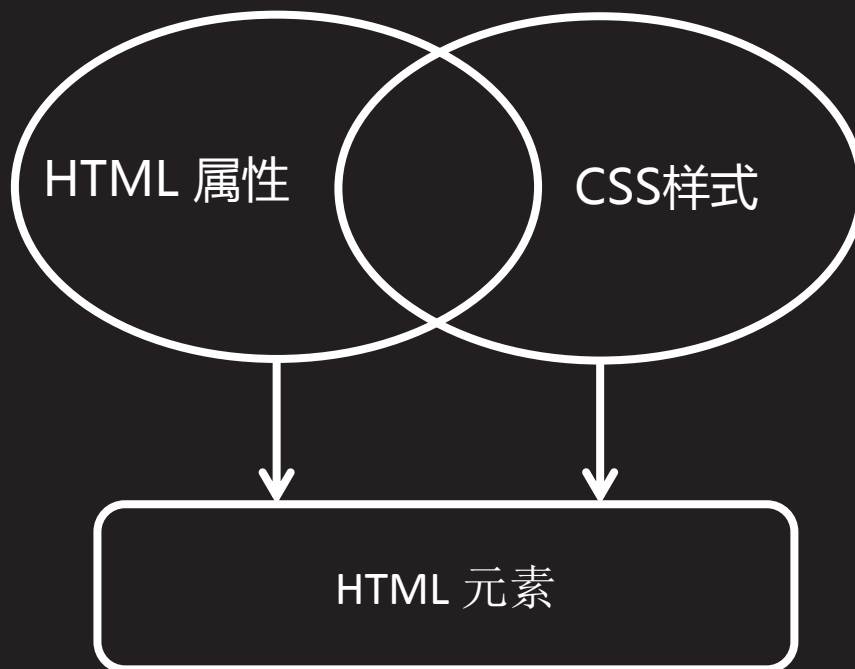
```
<p>段落1</p>
<p>段落2</p>
<p>段落3</p>
```

内容



# HTML 属性与 CSS 样式的使用原则

- W3C 建议尽量使用 CSS 样式取代 HTML 属性
  - 实现内容和表现的分离
  - 如果为 HTML 所特有的属性，则使用 HTML 属性



# 使用 CSS 样式表



# 使用 CSS 样式表的方式

- 内联方式
  - 样式定义在单个的 HTML 元素中
- 内部样式表
  - 样式定义在 HTML 页的头元素中
- 外部样式表
  - 将样式定义在一个外部的 CSS 文件中 ( .css 文件 )
  - 由 HTML 页面引用样式表文件



# 内联方式使用 CSS

- 样式定义在 HTML 元素的标准属性 style 里
- CSS 语法
  - 只需要将分号隔开的一个或者多个属性/值对作为元素的 style 属性的值
  - 属性和属性值之间用：连接
  - 多对属性之间用；隔开

```
<h1 style = "background-color : silver ; color : blue ;" >
    文本
</h1>
```

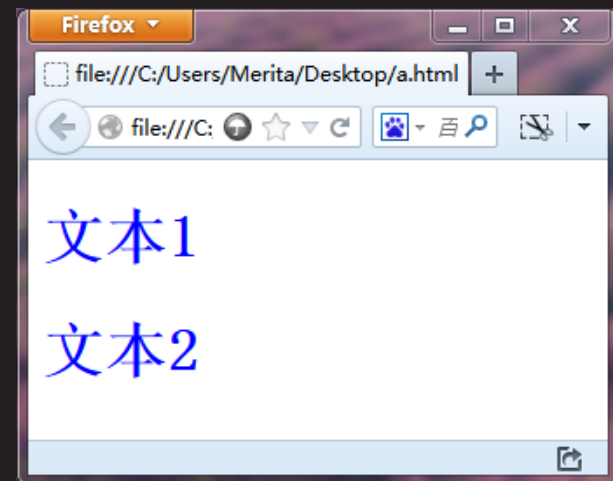


# 内部样式表

- 样式表规则位于文档头元素中的 `<style>` 元素内
  - 在文档的 `<head>` 元素内添加 `<style>` 元素
  - 在 `<style>` 元素中添加样式规则

```
<html>
  <head>
    <style type="text/css">
      h1 { color : blue ; }
    </style>
  </head>
  <body>
    <h1>文本1</h1>
    <h1>文本2</h1>
  </body>
</html>
```

可以在当前页面范围内重用



# 内部样式表（续1）

- 在 `<style>` 元素中添加样式规则
  - 可以定义多个样式规则
  - 每个样式规则有两个部分：选择器和样式声明
- 选择器：决定哪些元素使用这些规则
- 样式声明：一对大括号，包含一个或者多个属性/值对

`<head>`

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

选择器

`h1`

`{`

`color : blue ;`

样式声明

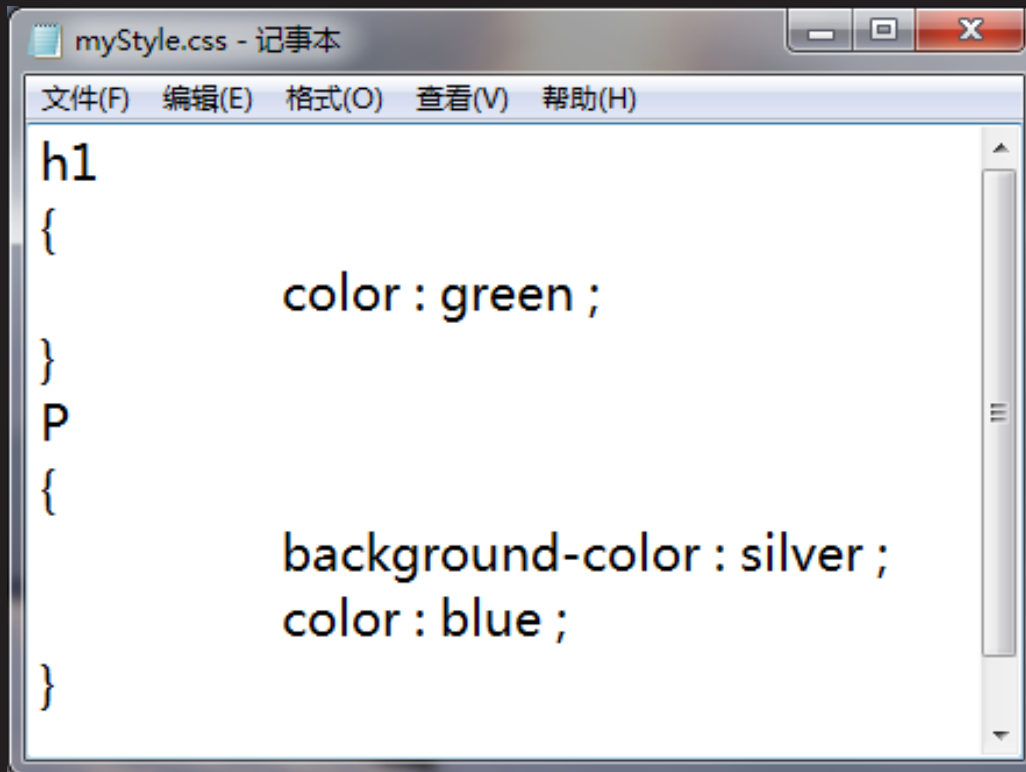
`}`

`</style>`

`</head>`

# 外部样式表

- 第一步：创建一个单独的样式表文件用来保存样式规则
  - 一个纯文本文件，文件后缀为 .css
  - 该文件中只能包含样式规则
  - 样式规则由选择器和样式声明组成



```

myStyle.css - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

h1
{
    color : green ;
}
p
{
    background-color : silver ;
    color : blue ;
}
    
```

CSS 文件：  
myStyle.css



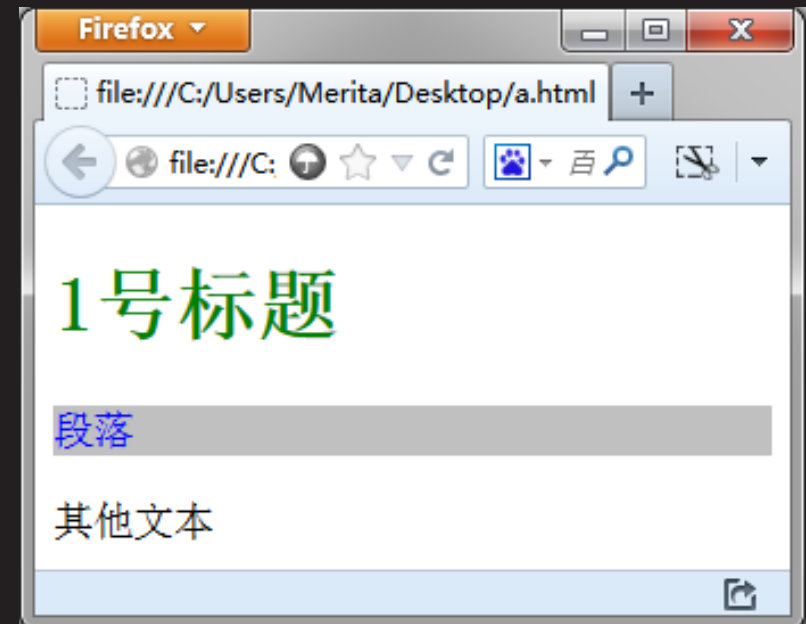


# 外部样式表（续1）

- 第二步：在需要使用该样式表文件的页面上，使用 `<link>` 元素链接需要的外部样式表文件
  - 在文档的 `<head>` 元素内添加 `<link>` 元素

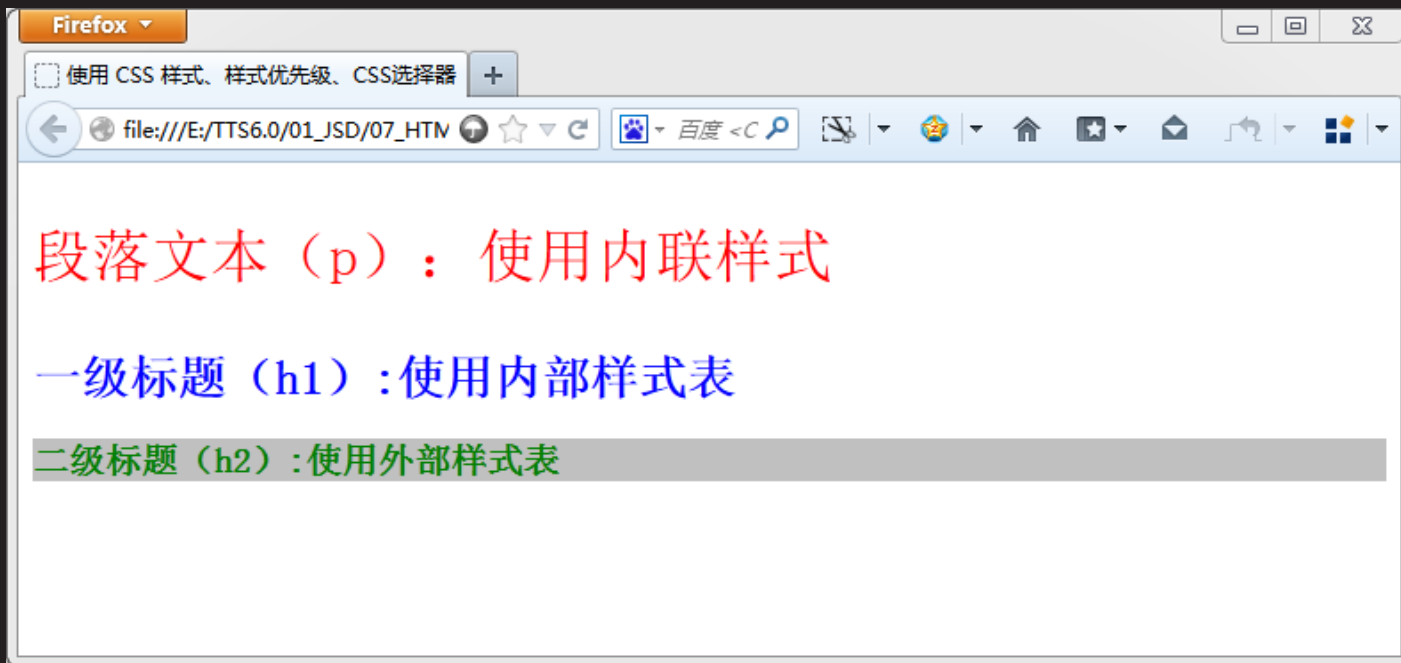
```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="myStyle.css" />
  </head>
  <body>
    <h1>1号标题</h1>
    <p>段落</p>
    <span>其他文本</span>
  </body>
</html>
```

可以实现内容与表现分离  
可以被站点中的所有页面重用



# 使用 CSS 样式

- 【参见 COOKBOOK】
  - 1、使用内联样式为 <p> 元素设置样式
  - 2、使用内部样式表为 <h1> 元素设置样式
  - 3、使用外部样式表为 <h2> 元素设置样式



# CSS语法

## CSS 语法

### CSS 语法规范

CSS 语法规范总结

CSS 样式表特征

样式优先级

!important 规则

### CSS 选择器

通用选择器

元素选择器

类选择器

id 选择器

群组选择器

后代选择器

子代选择器

伪类选择器

选择器优先级

# CSS 语法规范

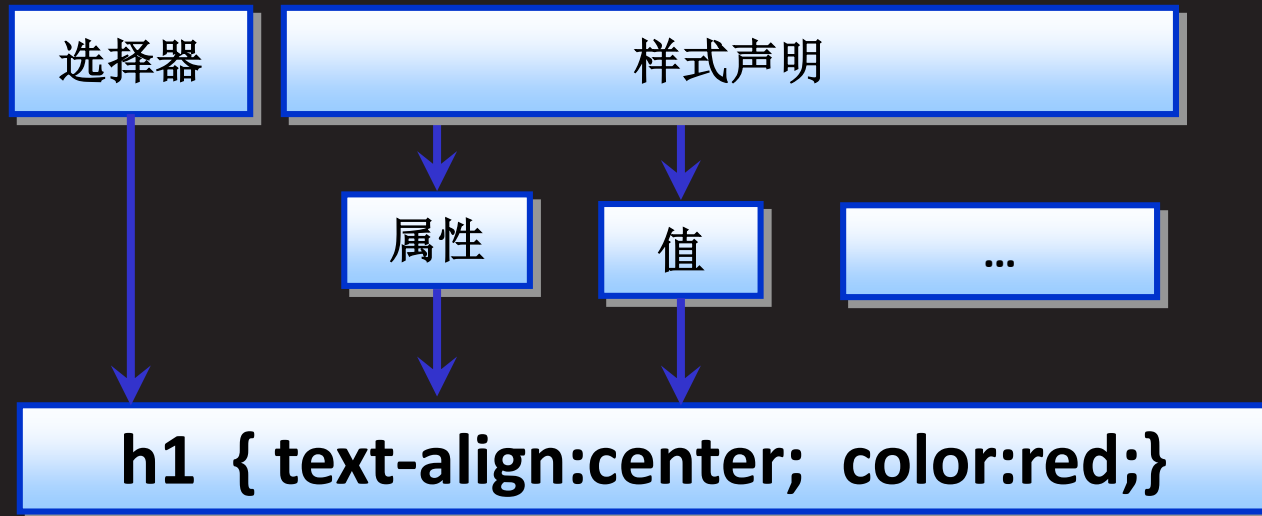


# CSS 语法规则总结

- 内联样式：由样式声明组成

<h1 style="background-color:silver;color:blue;">文本</h1>

- 样式表（内部样式表或者外部样式表）
  - 由多个样式规则组成
  - 每个样式规则有两个部分：选择器和样式声明



# CSS 样式表特征

- 继承性
  - 大多数 CSS 的样式规则可以被继承
- 层叠性
  - 可以定义多个样式
  - 不冲突时，多个样式表中的样式可层叠为一个
- 优先级
  - 样式定义冲突时，按照不同样式规则的优先级来应用样式



# 样式优先级

- 浏览器缺省设置
- 外部样式表或者内部样式表
  - 就近优先
- 内联样式

低

高



相同的样式，如果重复定义，以最后一次的定义为准



## 测试 CSS 样式优先级

- 【参见 COOKBOOK】
  - 1、为 <h3> 元素定义样式
  - 2、使用内联样式为其定义字体颜色为灰色
  - 3、在内部样式表中，为其定义字体颜色为红色，且字体大小为 30pt
  - 4、在外部样式表 demo.css 中，定义其字体颜色为黄色，字体大小为 10pt，且背景色为银灰色
  - 5、查看效果
  - 6、改变内部样式表和外部样式表的位置，查看效果





# !important 规则

- !important 可以调整样式规则的优先级
- 将 !important 添加在样式规则之后，中间用空格隔开
  - 选择器 {属性:属性值 !important;}
- 谨慎使用

html 文档中： `<p class="s1">段落文本</p>`

样式表中：

```
<style type="text/css">
  p.s1 { color:Red;}
  p.s1 { color:Green;}
</style>
```

```
<style type="text/css">
  p.s1 { color:Red !important;}
  p.s1 { color:Green;}
</style>
```

段落文本

段落文本

# CSS 基础选择器



# 通用选择器

- 通用选择器，显示为一个星号（\*）
  - 可以与任何元素匹配
  - 常用于设置一些默认样式，比如设置整个文档的文本的默认字体和大小

```
*  
{  
    font-size : 9pt;  
    font-family : "Times New Roman";  
}
```



# 元素选择器

- html 文档的元素就是选择器
  - 比如 <p>、<h1> 等

```
html
{
    color : black ;
}
h1
{
    color : blue ;
}
h2
{
    color : silver ;
}
```



## CSS 选择器 ( 1 )

- 【参见 COOKBOOK】
  - 分别使用通用选择器、元素选择器为页面元素设置样式



# 类选择器

- 语法为：`.className { color:red;}`
  - 类名称不能以数字开头
- 所有能够附带class属性的元素都可以使用此样式声明
  - 将元素的 class属性的值设置为样式类名

样式表中：声明一个样式类

```
.myClass
{
    background-color : Pink;
    font-size : 35pt;
}
```

html 文档中：将元素的 class 属性的值设置为样式类的名称

```
<h2 class="myClass">h2文本</h2>
<p class="myClass">段落文本</p>
```



# 类选择器（续1）

- 可以将多个类选择器应用于同一个元素（多类选择器）
  - HTML 元素的 class 属性的值中可能包含一个词列表
  - 各个词之间用空格分隔，每个词都是一个类选择器

样式表中：声明两个样式类

```
.important {
    font-weight : bold;
}
.warning {
    color : red;
}
```

html 文档中：将元素的 class 属性的值设置为多个样式类名的列表

```
<p class="warning">警告</p>
<p class="important">重要</p>
<p class="important warning">哈哈</p>
```



# 类选择器（续2）

- 可以将类选择器和元素选择器结合起来使用，以实现某种元素中不同样式的细分控制（分类选择器）
- 语法为：元素选择器.className { }
  - 先声明一个元素选择器
  - 然后使用一个点号（.）代表将使用类选择器
  - 然后声明一个类的名称
  - 最后使用一对大括号声明样式规则
- 将样式规则与附带 class 属性的某种元素匹配
  - 元素的 class 属性的值为分类选择器中指定的样式类名





# 类选择器（续3）

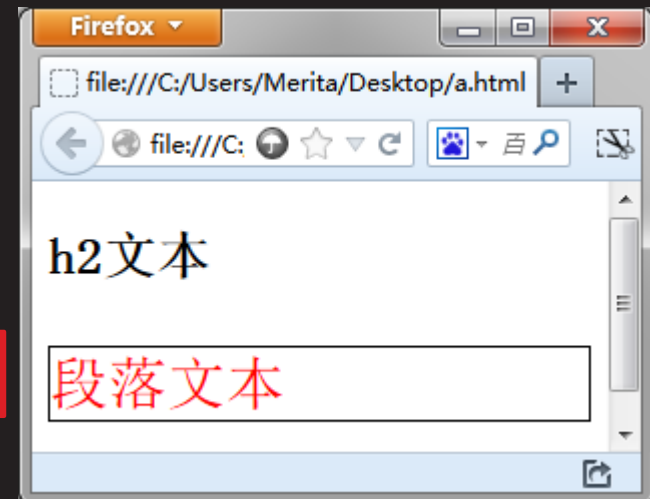
样式表中：定义一个分类选择器

```
p.important
{
    color : red ;
    font-size : 20pt;
    border:1px solid black;
}
```

html 文档中：将元素的 class 属性的值设置为样式类的名称

```
<h2 class="important">h2文本</h2>
<p class="important">段落文本</p>
```

只有 <p> 元素使用样式



## CSS 选择器 ( 2 )

- 【参见 COOKBOOK】
  - 使用类选择为页面元素设置样式



# id 选择器

- id 选择器以一种独立于文档元素的方式来指定样式
- 它仅作用于 id 属性的值
- 语法为：
  - 选择器前面需要有一个 # 号
  - 选择器本身则为文档中某个元素的 id 属性的值



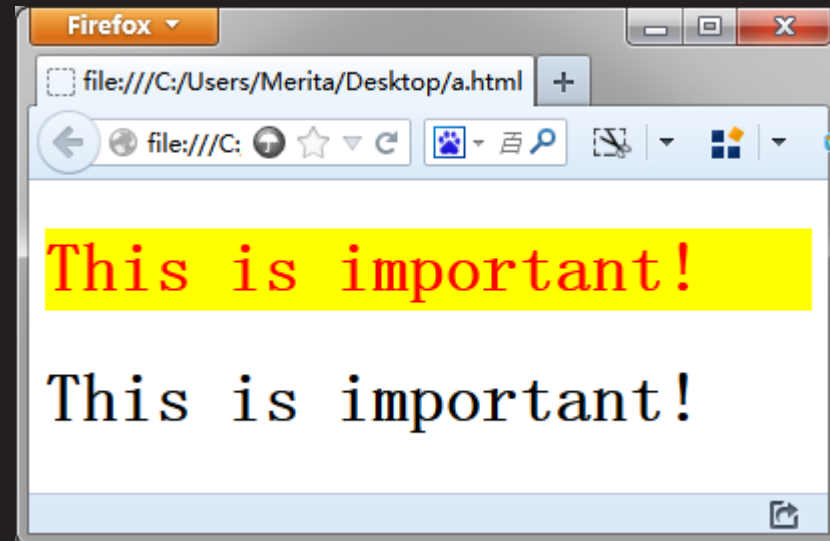
# id 选择器 ( 续1 )

样式表中：定义一个 id 选择器

```
#mostImportant
{
    color:red;
    background:yellow;
}
```

html 文档中：将元素的 id 属性的值设置为选择器的名称

```
<h1 id="mostImportant">This is important!</h1>
<h1 >This is important!</h1>
```



# 群组选择器

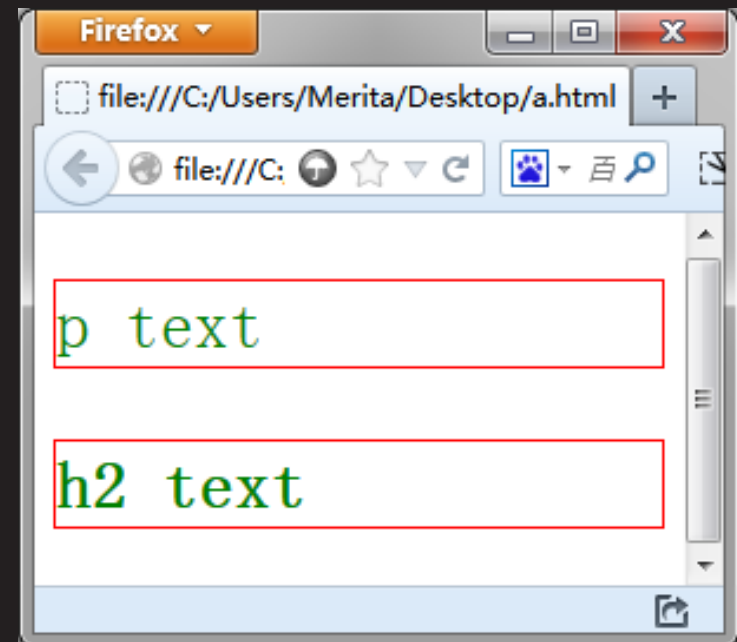
- 选择器声明为以逗号隔开的选择器列表
  - 将一些相同的规则作用于多个元素

样式表中：定义选择器分组

html 文档中：

```
h2, p.important
{
    color:green;
    font-size:20pt;
    border:1px solid red;
}
```

```
<p class="important">p text</p>
<h2>h2 text</h2>
```



## CSS 选择器 ( 3 )

- 【参见 COOKBOOK】
  - 使用 id 选择器和群组选择器为页面元素设置样式



# 后代选择器

- 依据元素在其位置的上下文关系来定义样式
- 又称为包含选择器，用于选择作为某元素后代的元素
  - 选择器一端包括两个或多个用空格分隔的选择器

知识讲解

样式表中：

定义派生选择器

```
h1 span
{
    color:red;
}
```

html 文档中：

<h1>

An <span>important</span> question.

</h1>

<span>other</span>



只有 <h1> 元素中的 <span> 元素中的文本使用该样式

An **important** question.

other



# 子代选择器

- 子代选择器要求选择器之间只能存在父子关系
  - 不希望选择任意的后代元素，而是希望缩小范围，只选择某个元素的子元素
  - 使用一个特殊的符号大于号（>）作为子结合符





# 子代选择器（续1）

样式表中：  
定义子代选择器

```
div>span
{
    color : red;
    font-size:20pt;
}
```

html 文档中：

<div>

<span>子元素</span>

<p>

段落中的文本

<span>不是子元素</span>

</p>

</div>



只有 <div> 的子元素 <span> 元素中的文本使用该样式  
为什么另一个 <span> 元素中的文本没有使用该样式？



## CSS 选择器 ( 4 )

- 【参见 COOKBOOK】
  - 使用后代选择器和子代选择器为页面元素设置样式



# 伪类选择器

- 伪类用于向某些选择器添加特殊的效果
- 使用冒号(:)作为结合符，结合符左边是其他选择器，右边是伪类
  - 选择器：伪类选择器
- CSS 伪类选择器可以分为
  - 链接伪类
  - 动态伪类
  - 目标伪类
  - 元素状态伪类
  - 结构伪类
  - 否定伪类



# 伪类选择器（续1）

- 链接伪类
  - :link , 适用于尚未访问的链接
  - :visited , 适用于访问过的链接
- 动态伪类, 用于呈现用户操作
  - :hover , 适用于鼠标悬停在 HTML 元素时
  - :active , 适用于 HTML 元素被激活时
  - :focus , 适用于 HTML 元素获取焦点时



# 伪类选择器（续2）

样式表中：定义伪类选择器

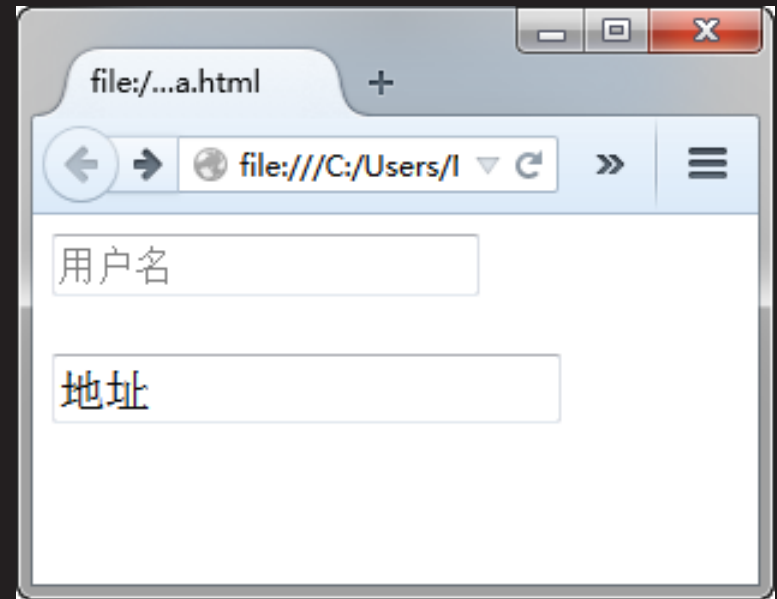
```
a:link {
    color: black;
    font-size: 15pt;
}
a:visited {
    color: pink;
    font-size: 15pt;
}
a:hover {
    font-size: 20pt;
}
a:active {
    color: red;
}
```



# 伪类选择器（续3）

样式表中：定义伪类选择器

```
.txt {
    color : gray;
    font-size : 9pt;
}
.txt:focus {
    color : black;
    font-size: 11pt;
}
```



html 文档中：

```
<input class="txt" type="text" value="用户名" />
<br /> <br />
<input class="txt" type="text" value="地址" />
```



## CSS 选择器 ( 5 )

- 【参见 COOKBOOK】
  - 使用 伪类选择器 为页面元素设置样式



# 选择器优先级

选择器类型	权值
元素选择器	0,0,0,1
类选择器	0,0,1,0
伪类选择器	0,0,1,0
ID选择器	0,1,0,0
内联样式	1,0,0,0

选择器的权值加到一起，大的优先；如果权值相同，后定义的优先





# 尺寸与边框

## 尺寸与边框

### CSS 单位

尺寸单位

颜色单位

### 尺寸属性

尺寸

溢出

哪些 HTML 元素可以设置尺寸属性

### 边框属性

边框

边框倒角

边框阴影

轮廓

# CSS 单位



# 尺寸单位

- %：百分比
- in：英寸
- cm：厘米
- mm：毫米
- pt：磅(1pt等于1/72英寸)
- px：像素(计算机屏幕上的一个点)
- em：1em等于当前的字体尺寸，2em等于当前字体尺寸的两倍，继承父级元素的字体大小
- rem：与em类似，但是相对于根元素设置字体尺寸的倍数



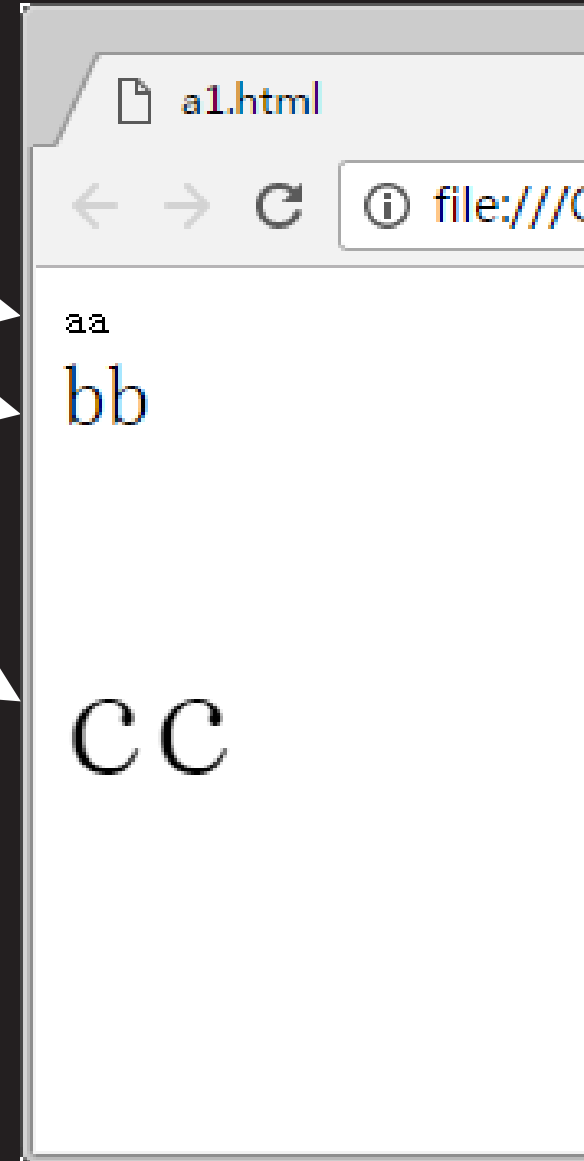
# 尺寸单位 (续1)

样式表中：

```
Html {font-size:10pt;}
div {font-size:2rem;}
p {font-size:2em;}
```

HTML文档中：

```
<body>
  aa
  <div>
    bb
  </div>
  <p>cc<p>
</body>
```



# 颜色单位

- `rgb(x,x,x)` : RGB 值 , 如 `rgb(255,0,0)`
- `rgb(x%,x%,x%)` : RGB百分比值 , 如`rgb(100%,0%,0%)`
- `#rrggbb` : 十六进制数 , 如`#ff0000`
- `#rgb` : 简写的十六进制数 , 如`#f00`
- 表示颜色的英文单词 , 如 `red`



# 尺寸属性



# 尺寸

- 尺寸属性是用于设置元素的宽度和高度
  - 单位一般为像素或百分比
- 宽度属性
  - width
  - max-width
  - min-width
- 高度属性
  - height
  - max-height
  - min-height



# 溢出

- 使用尺寸属性控制框的大小时，如果内容所需的空间大于框本身的空间，会导致内容溢出
- overflow：当内容溢出元素框时如何处理

- visible
- hidden
- scroll
- auto

样式表中：

```
div
{
    width:130px;
    height:50px;
    border:1px solid black;
}
div.hidden {overflow:hidden;}
div.scroll {overflow:scroll;}
div.auto {overflow:auto;}
```



# 溢出 ( 续1 )

html 文档中：

`<div>`

默认为显示，即使超出尺寸，也显示全部内容  
`</div> <br /> <br />`

`<div class="hidden">`

文本内容超出尺寸时，隐藏文本不显示  
`</div> <br />`

`<div class="scroll">`

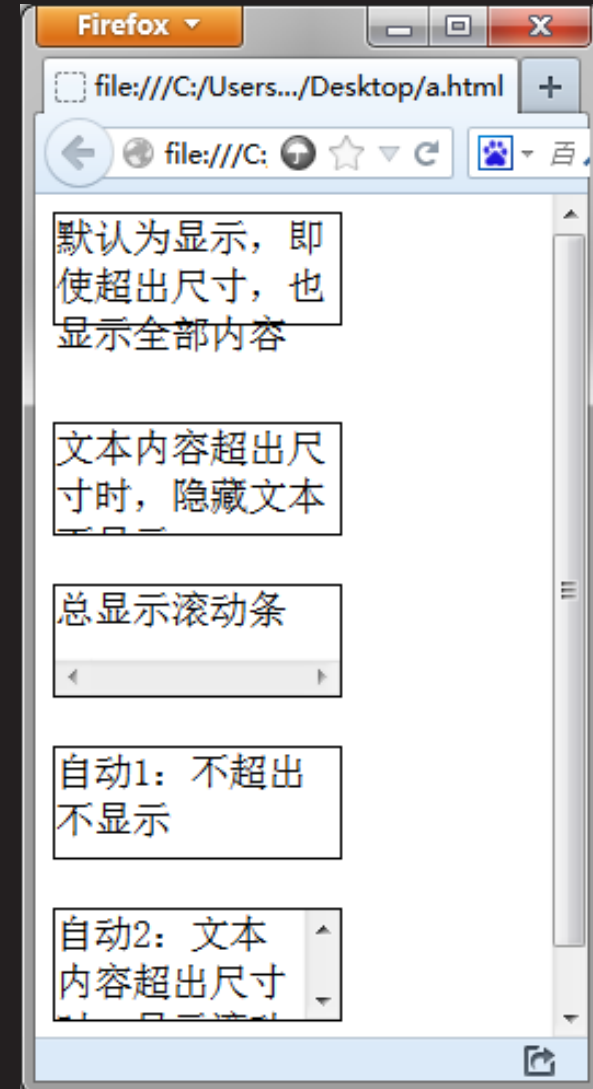
总显示滚动条  
`</div> <br />`

`<div class="auto">`

自动1：不超出不显示  
`</div> <br />`

`<div class="auto">`

自动2：文本内容超出尺寸时，显示滚动条  
`</div> <br />`



# 尺寸属性的使用

- 【参见 COOKBOOK】
  - 使用尺寸属性为页面元素设置样式



# 哪些 HTML 元素可以设置尺寸属性

- 块级对象，如
  - p,
  - div
  - h1,h2,h3,h4,h5,h6
  - ul,ol,li,dl,dt,dd等
- 存在width、height属性的 HTML 元素
  - img
  - table 等



# 边框属性



# 边框

- 简写方式
  - border:width style color;
- 单边定义
  - border-left/right/top/bottom:width style color;
- border-width
  - border-left/right/top/bottom-width
- border-style
  - border-left/right/top/bottom-style
- border-color
  - border-left/right/top/bottom-color

边框颜色，可设置为值 transparent（用于创建有宽度的不可见边框）



# 边框倒角

- border-radius 属性
  - 为简写属性，按顺时针顺序设置四个倒角
  - 取值为绝对值或者百分比
- 单独定义
  - border-top-left-radius : 边框左上角
  - border-top-right-radius : 边框右上角
  - border-bottom-left-radius : 边框左下角
  - border-bottom-right-radius : 边框右下角



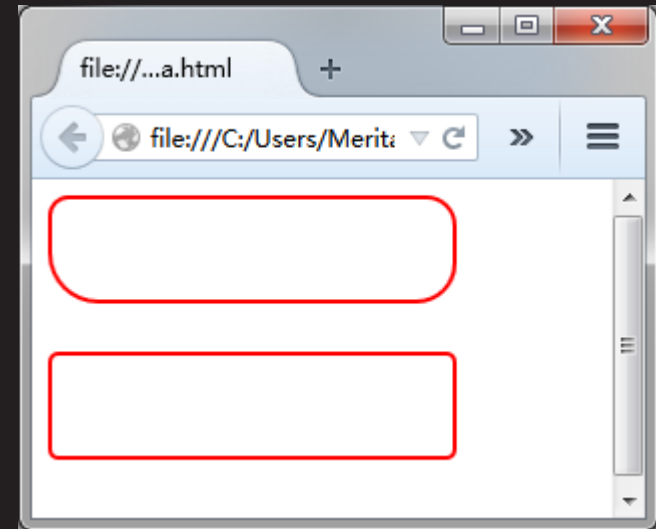
# 边框倒角（续1）

样式表中：

```
div
{
    border:2px solid red;
    width:200px;
    height:50px;
}
#d1
{
    border-radius:10px 15px 20px 25px;
}
#d2
{
    border-radius:5px;
}
```

html 文档中：

```
<div id="d1"></div><br />
<div id="d2"></div><br />
```



# 边框阴影

- box-shadow : 向方框添加一个或多个阴影
  - 取值为多个属性值的列表
- box-shadow: h-shadow v-shadow blur spread color inset;
- 其中
  - h-shadow : 必需, 为水平阴影的位置
  - v-shadow : 必需, 为垂直阴影的位置
  - blur : 可选, 为模糊距离
  - spread : 可选, 为阴影的尺寸
  - color : 可选, 为阴影的颜色
  - inset : 可选, 将外部阴影 (outset) 改为内部阴影





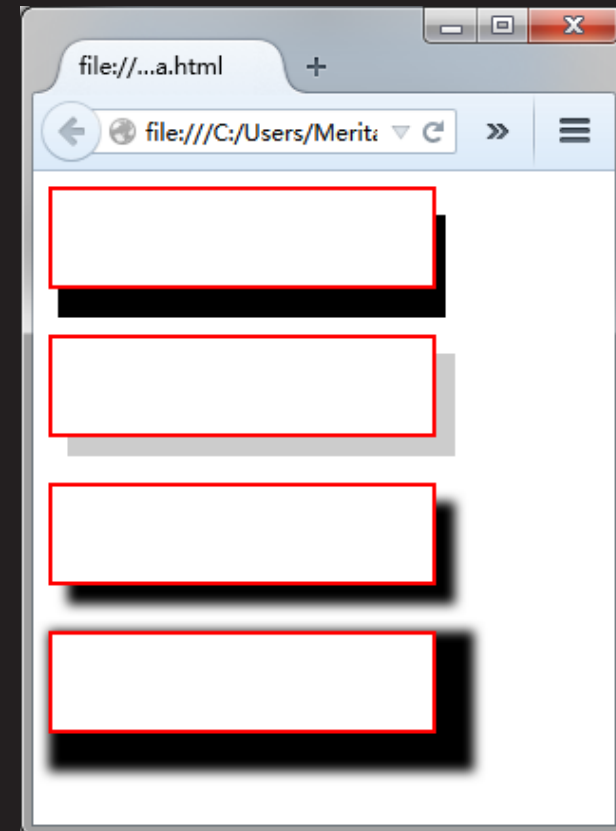
# 边框阴影（续1）

样式表中：

```
div
{
    border:2px solid red;
    width:200px;
    height:50px;
}
#d1 {box-shadow: 5px 15px;}
#d2 {box-shadow: 10px 10px #ccc;}
#d3 {
    box-shadow: 10px 10px 5px #000;}
#d4 {
    box-shadow: 10px 10px 5px 10px #000;}
```

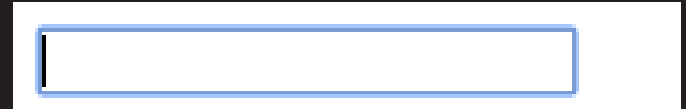
html 文档中：

```
<div id="d1"></div><br />
<div id="d2"></div><br />
<div id="d3"></div><br />
<div id="d4"></div><br />
```

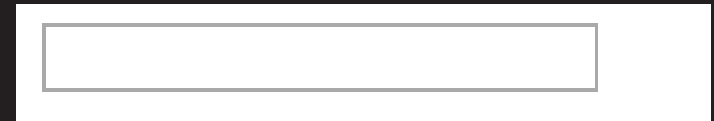


# 轮廓

- 轮廓 ( outline ) 是绘制于元素周围的一条线，位于边框边缘的外围，可起到突出元素的作用
- 简写方式
  - outline:color style width;
- outline-width : 轮廓的宽度
- outline-style : 轮廓的样式
- outline-color : 轮廓的颜色



input {outline:0; } →



# 边框属性的使用

- 【参见 COOKBOOK】
  - 使用边框属性为页面元素设置样式



# 总结和答疑

