

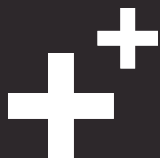
# Python开发

NSD DEVWEB

DAY02

# 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 基础语法
	11:30 ~ 12:00	尺寸与框模型
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	常用样式属性
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



# CSS概述

## CSS 概述

### 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

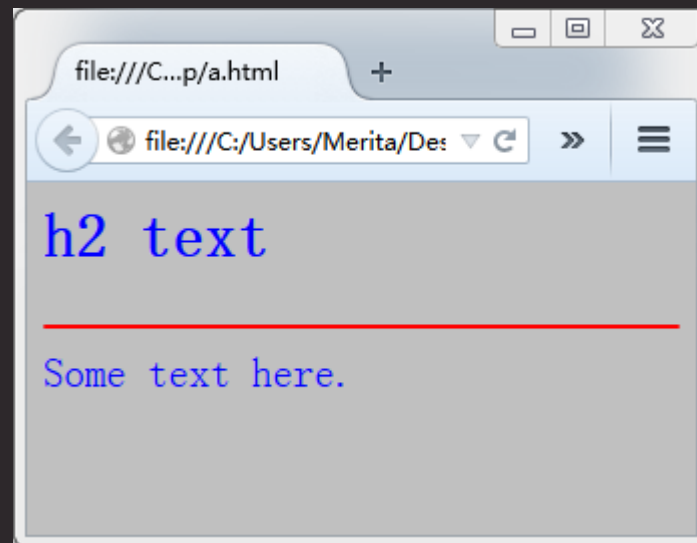
# CSS 概述



# 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>
```

```
<p>段落1</p>
```

```
<p>段落2</p>
```

```
<p>段落3</p>
```

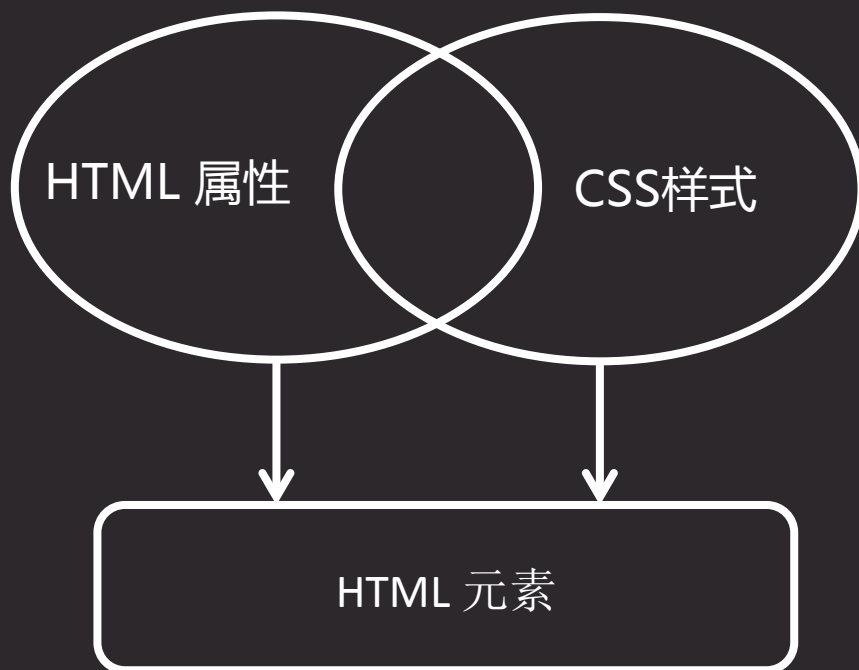
← 表现

← 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>
```

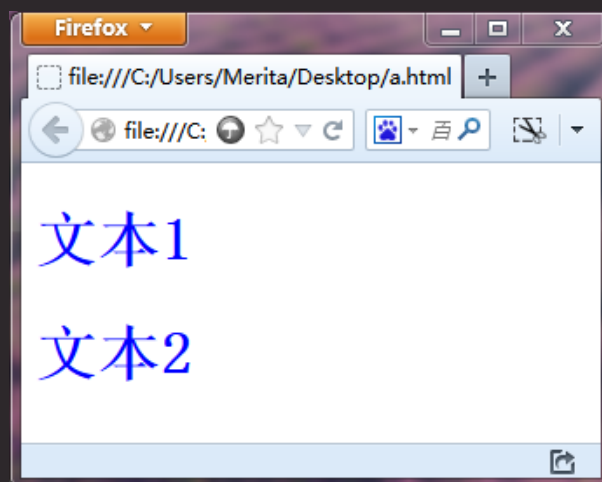


# 内部样式表

- 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

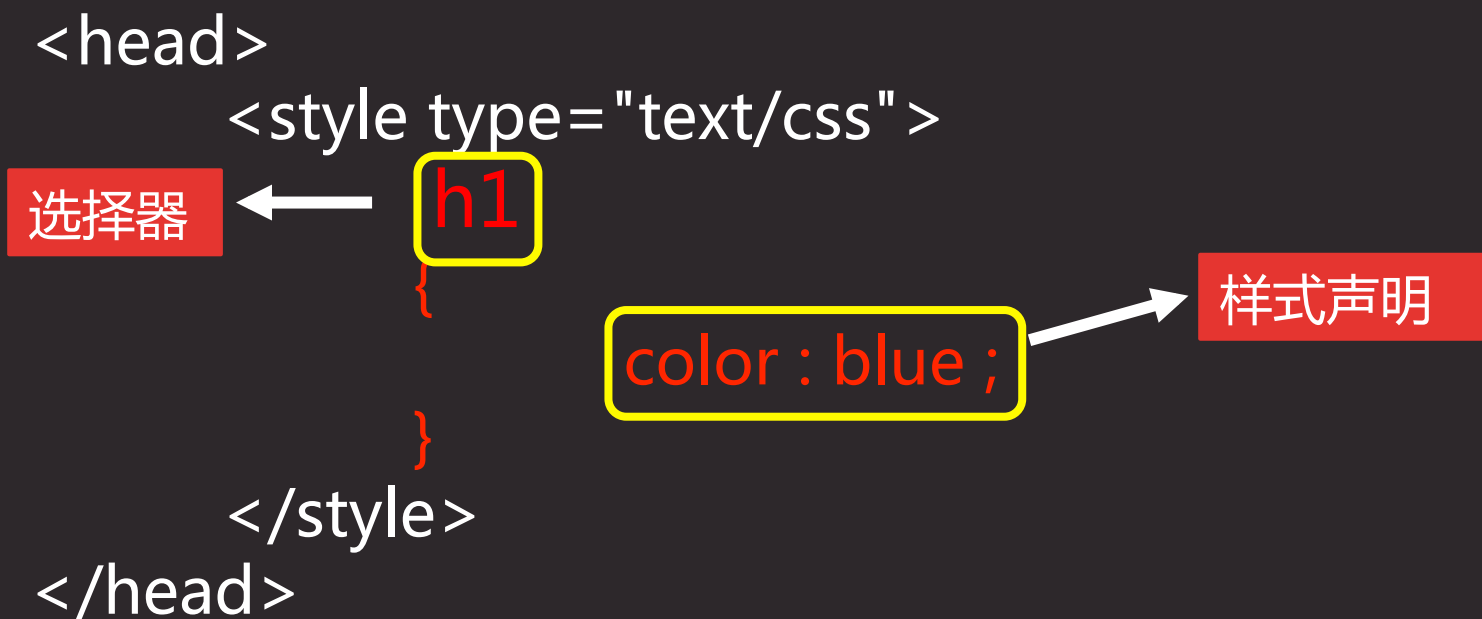
```
<html>
  <head>
    <style type="text/css">
      h1 { color : blue ; }
    </style>
  </head>
  <body>
    <h1>文本1</h1>
    <h1>文本2</h1>
  </body>
</html>
```

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



# 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



CSS 文件：  
myStyle.css

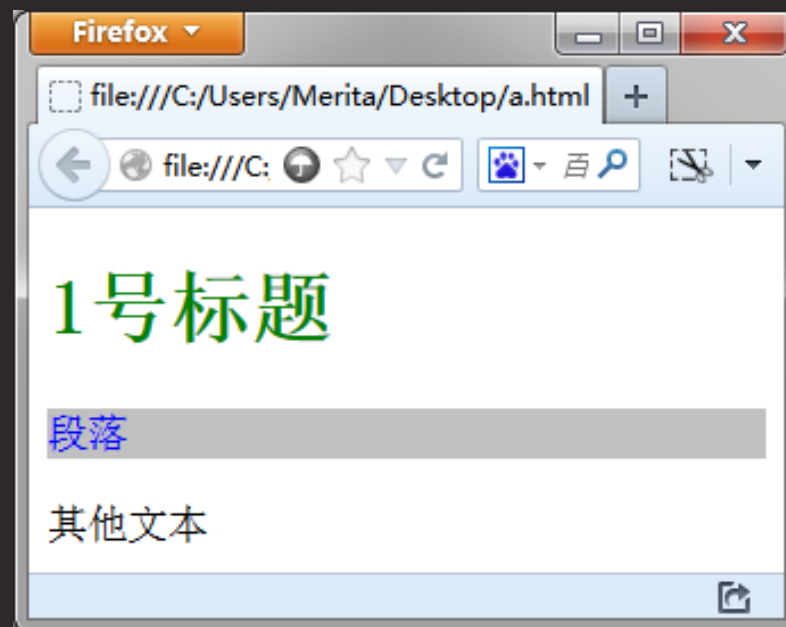


# 外部样式表（续1）

- 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

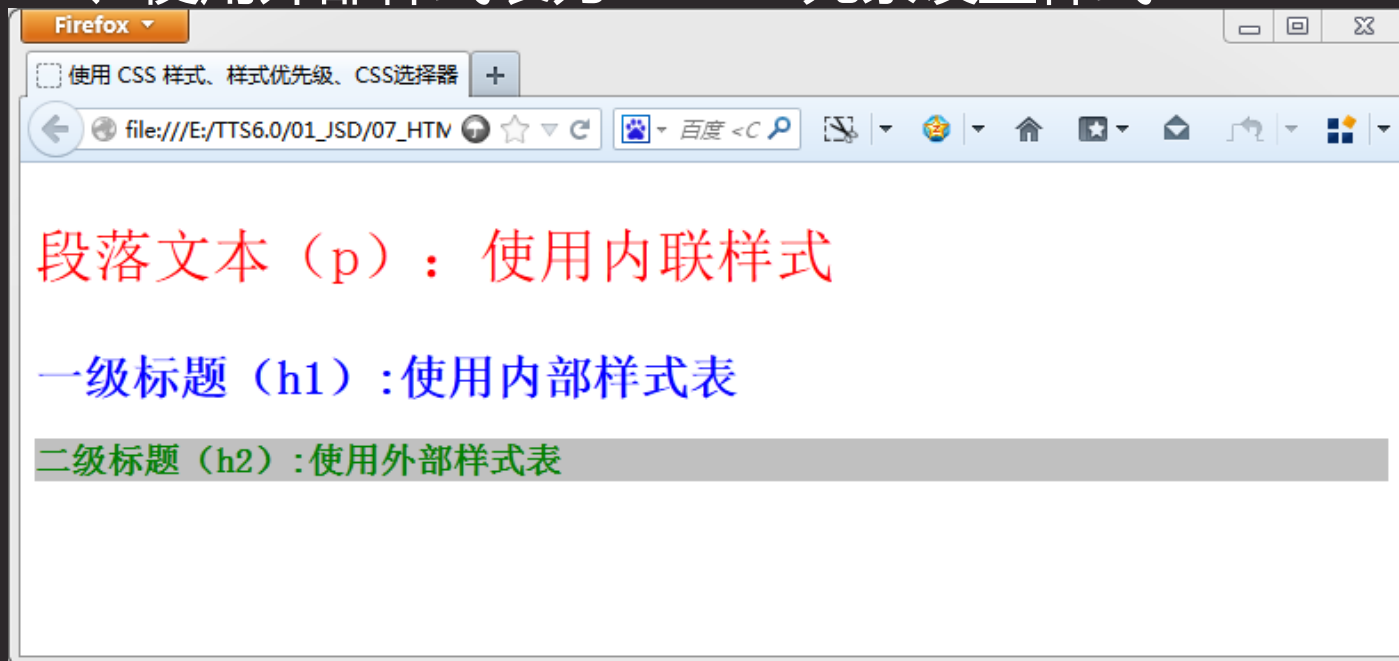
```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="myStyle.css" />
  </head>
  <body>
    <h1>1号标题</h1>
    <p>段落</p>
    <span>其他文本</span>
  </body>
</html>
```

可以实现内容与表现分离  
可以被站点中的所有页面重用



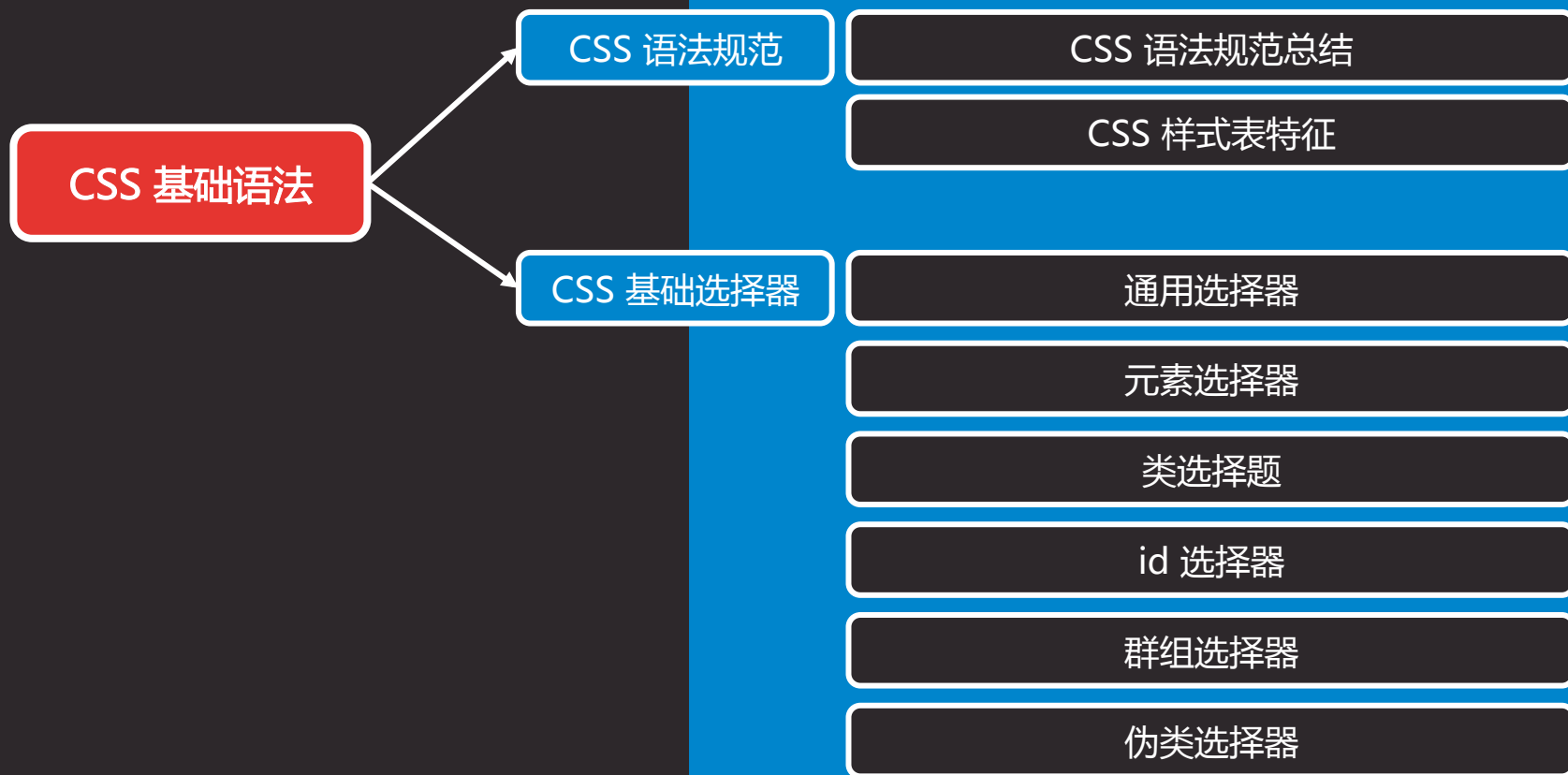
# 使用 CSS 样式

- 【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 基础语法

---



# CSS 语法规范

---

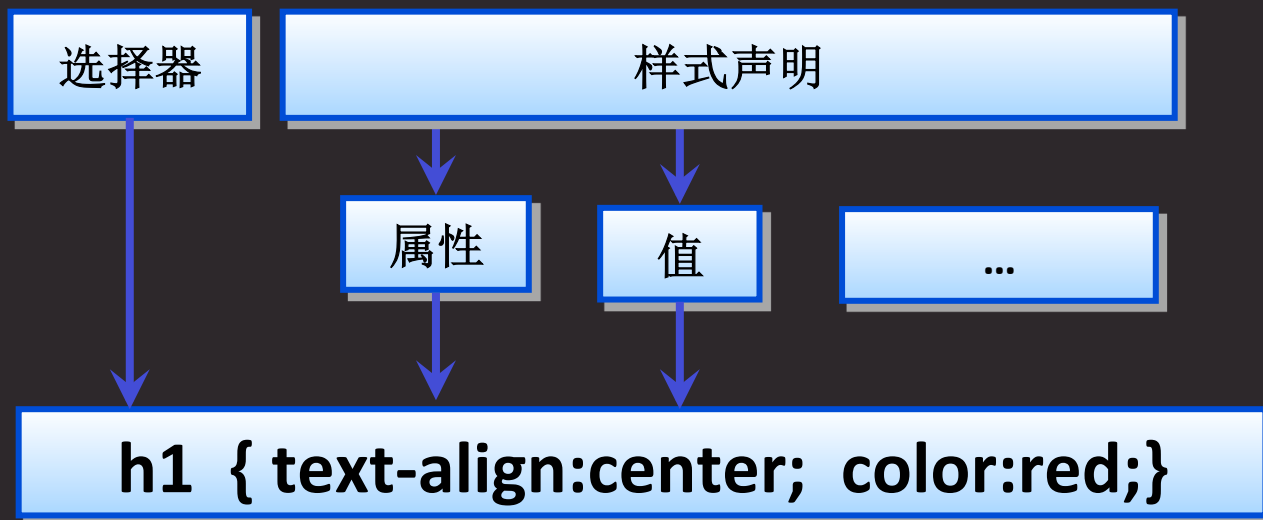
# CSS 语法规则总结

- 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



# 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 ;
}
```



# 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，以实现对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



# 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



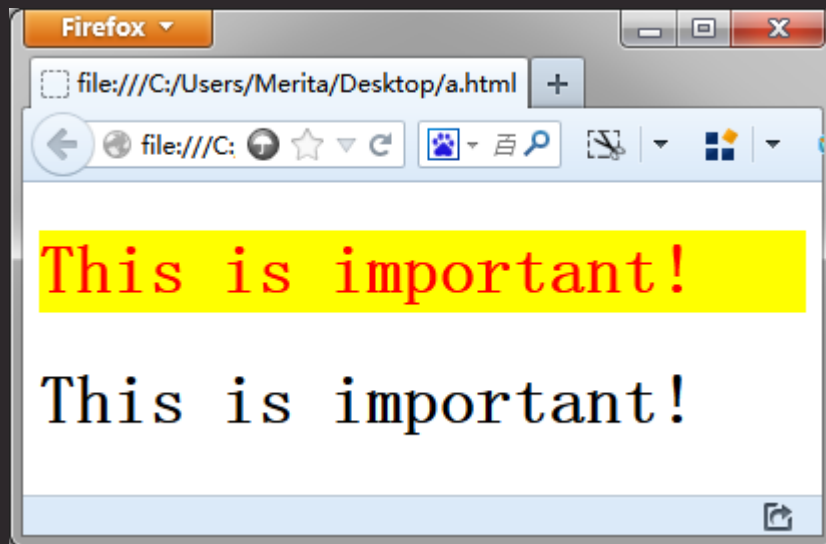
# id 选择器 ( 续1 )

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

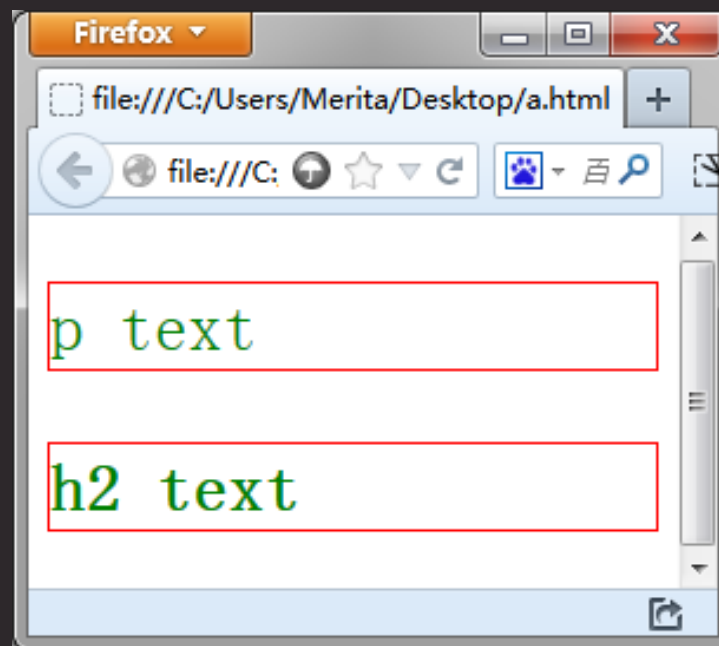
```
h2, p.important  
{
```

```
color:green;  
font-size:20pt;  
border:1px solid red;
```

```
}
```

html 文档中：

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





# 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
- 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1）

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;
}
```

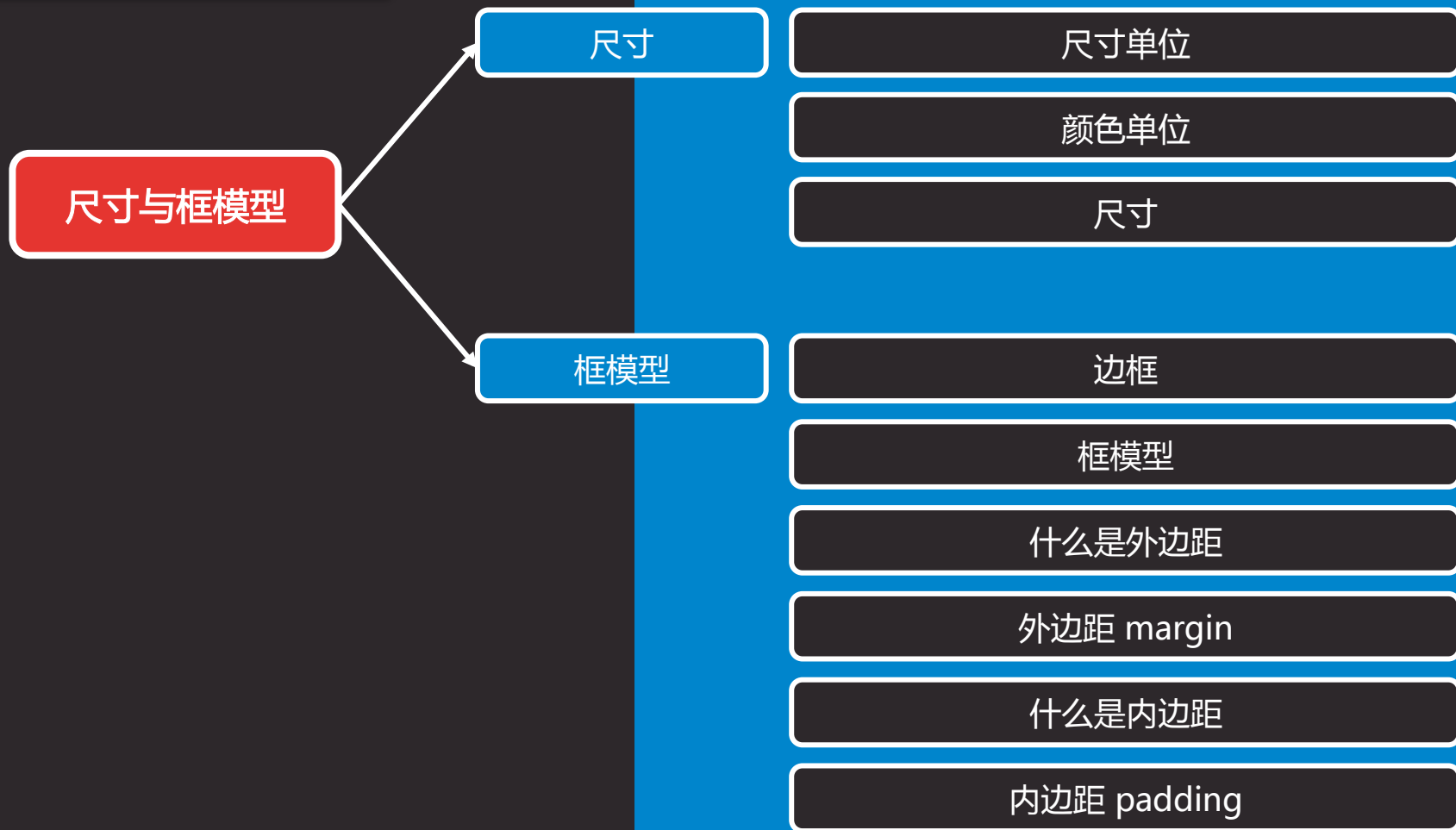


# CSS 基础选择器

- 【参见 COOKBOOK】
  - 使用 CSS 基础选择器为页面元素设置样式



# 尺寸与框模型



# 尺寸



# 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



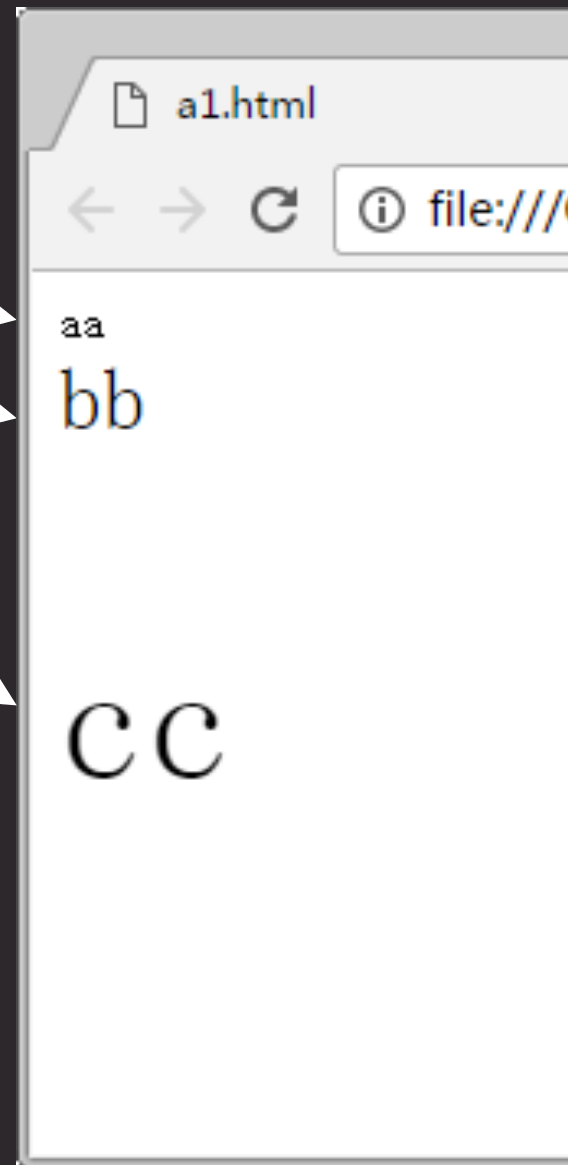
# 尺寸单位 ( 续1 )

样式表中：

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

HTML文档中：

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

- 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



# 框模型



# 边框

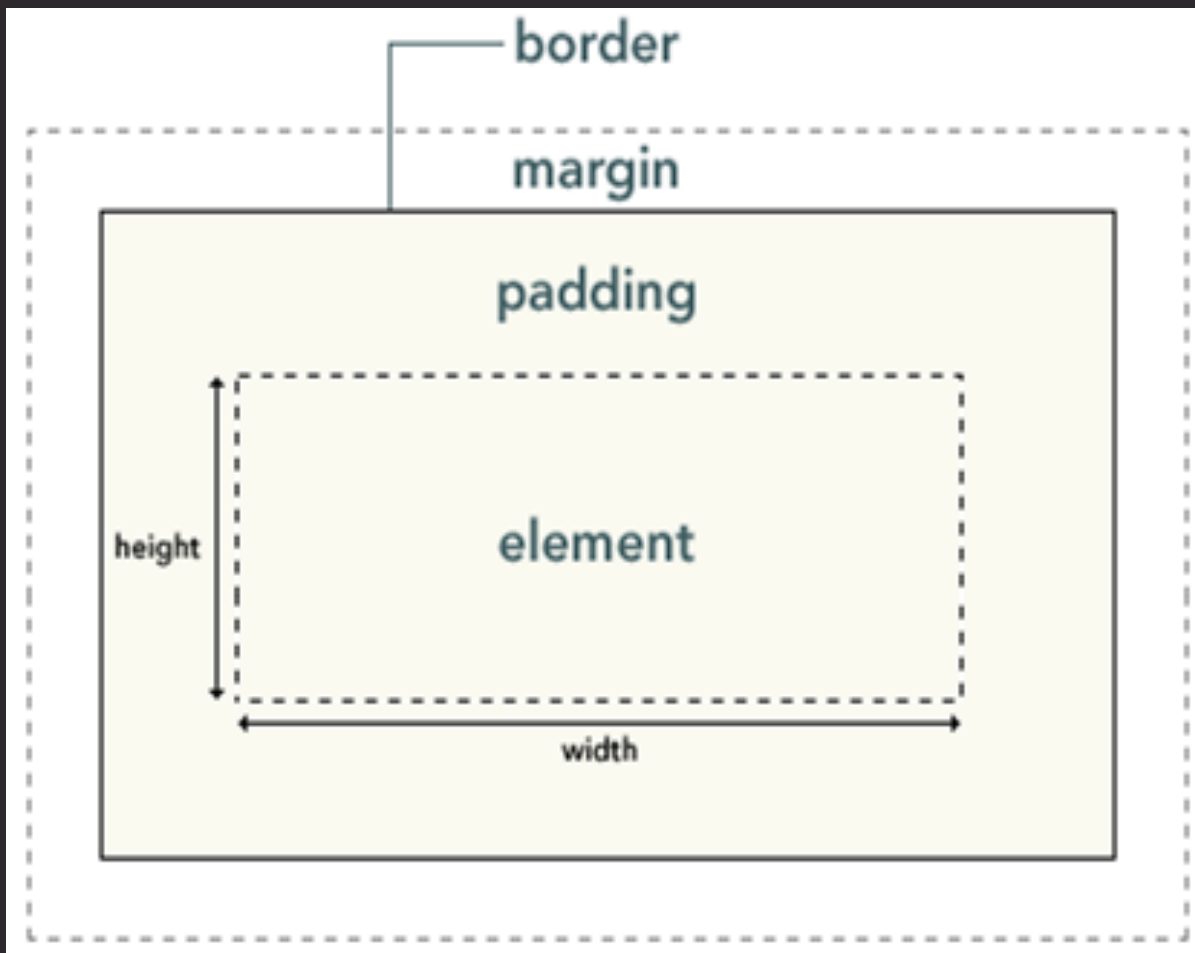
- 简写方式
  - border:width style color;
- 单边定义
  - border-left/right/top/bottom:width style color;
- border-width
  - border-left/right/top/bottom-width
- border-style
  - border-left/right/top/bottom-style
- border-color
  - border-left/right/top/bottom-color

边框颜色，可设置为值 transparent（用于创建有宽度的不可见边框）



# 框模型

- 框模型（Box Model）定义了元素框处理元素内容、内边距、边框和外边距的方式



## 框模型（续1）

- width 和 height 指内容区域的宽度和高度
- 增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸

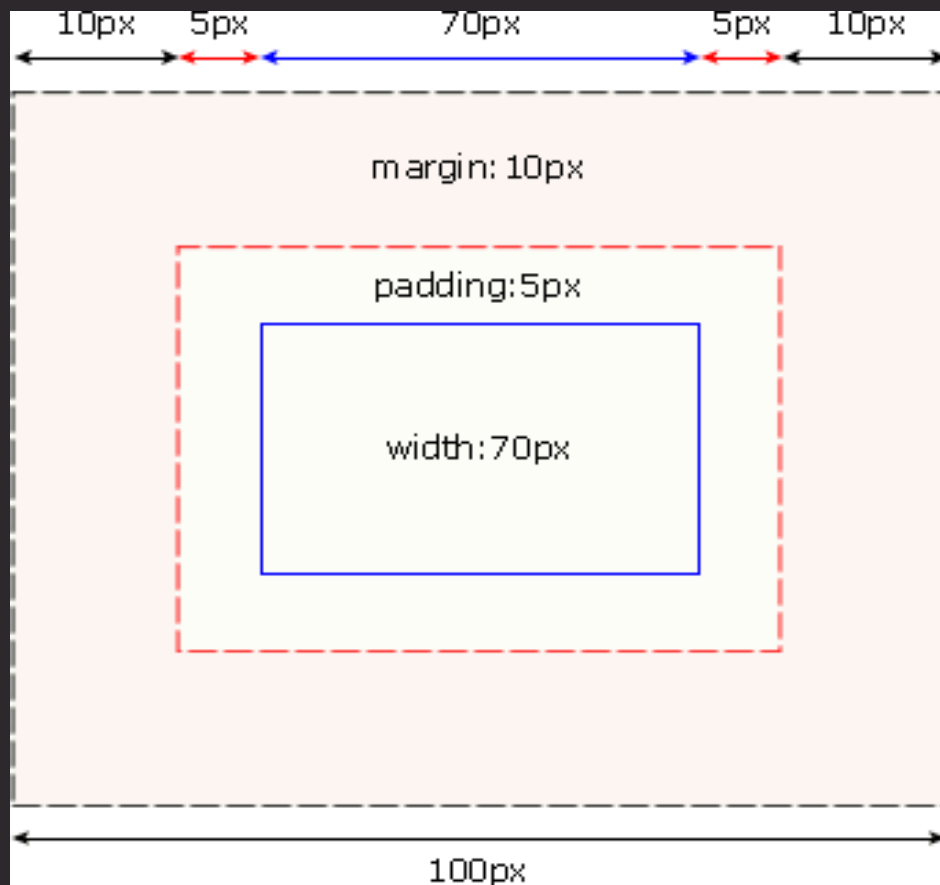
#box

{

width: 70px;  
margin: 10px;  
padding: 5px;

}

对象实际宽度 = 左侧外边距 +  
左侧边框 + 左侧内边距 + 宽度 +  
右侧内边距 + 右侧边框 +  
右侧外边距



## 框模型（续2）

样式表中：

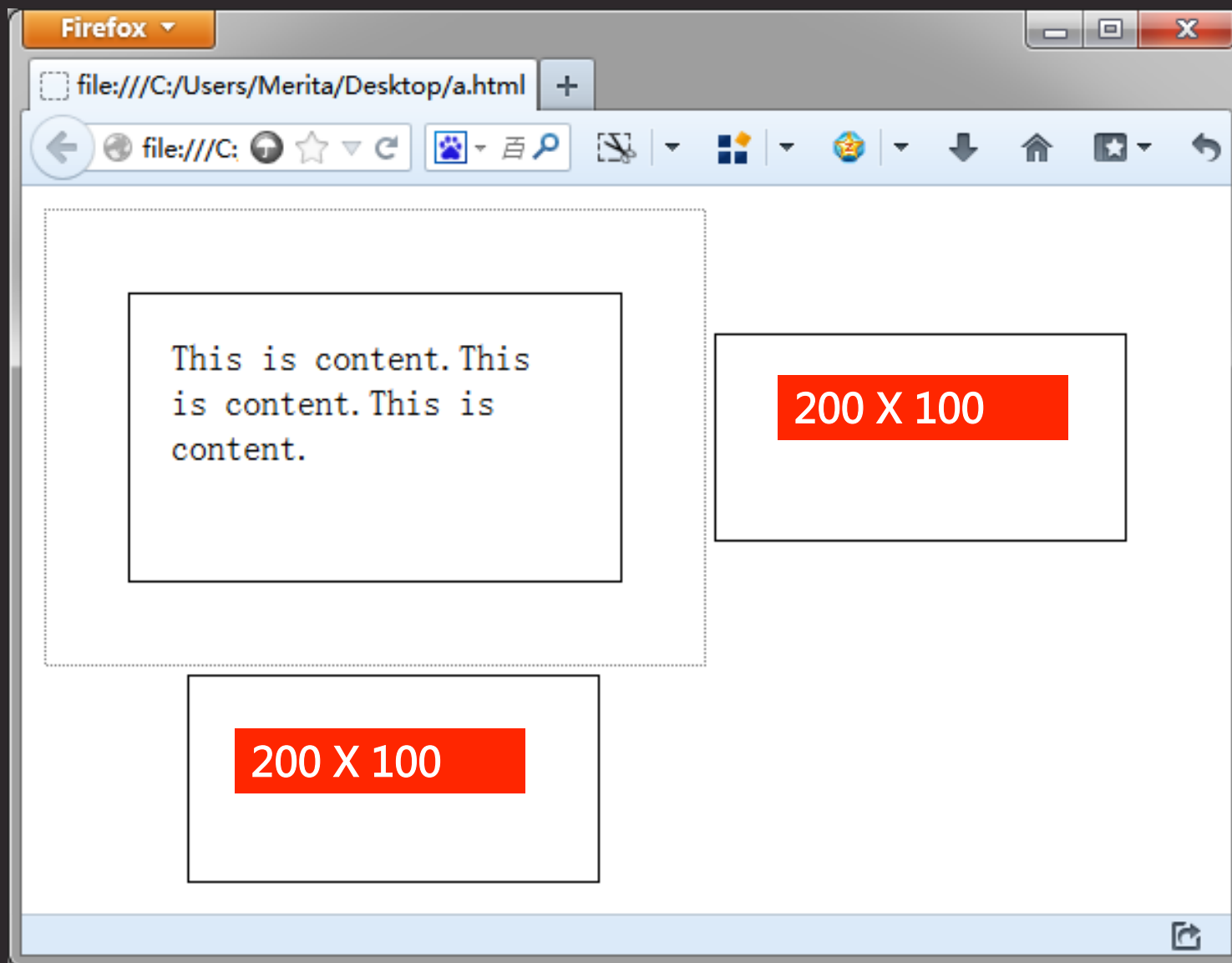
```
div.box
{
    width:200px;
    height:100px;
    padding:20px;
    margin:40px;
    border:1px solid black;
}
```

html 文档中：

```
<div class="box">
    This is content.
    This is content.This is content.
</div>
```



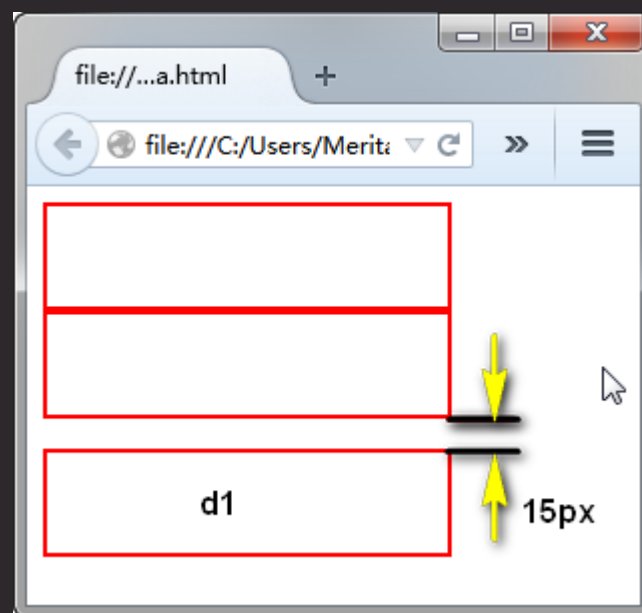
# 框模型（续3）



# 什么是外边距

- 围绕在元素边框周围的空白区域是外边距
  - 会在元素外创建额外的空白
  - 外边距是透明的

```
div
{
    border:2px solid red;
    width:200px;
    height:50px;
}
#d1 {margin-top:15px;}
```

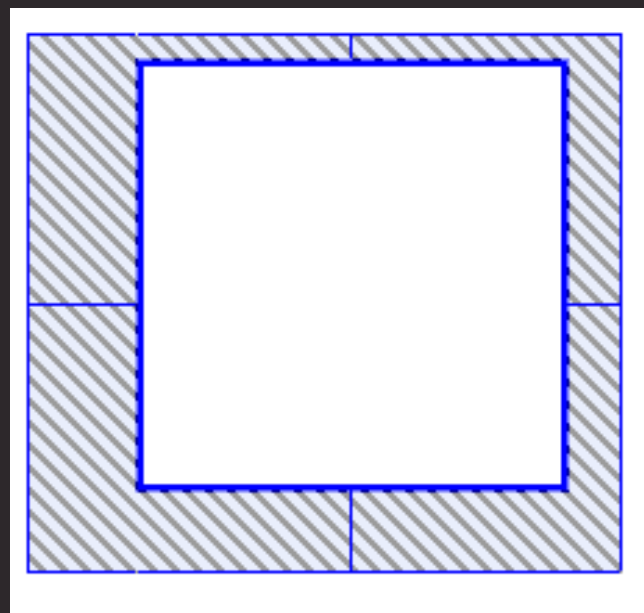




# 外边距 margin

- 外边距：与下一个元素之间的空间量
  - margin : value;
- 单边设置
  - margin-top/right/bottom/left : value;
- 外边距的属性值可能为px(像素)，百分比(%)或自动(auto)，也可以为负值

```
div {
    width:150px;
    height:150px;
    border:2px solid #0f0;
    margin-top:10px;
    margin-right:20px;
    margin-bottom:30px;
    margin-left:40px;
```



## 外边距 margin ( 续1 )

- 默认情况下，以下 HTML 块级元素都存在外边距
  - body
  - h1,h2,h3,h4,h5,h6
  - p...
- 声明 margin 属性，可以覆盖默认样式

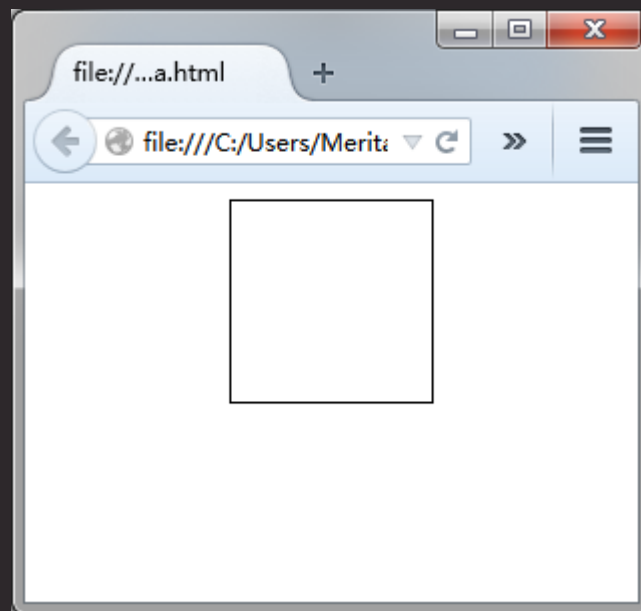
```
body,p,div,h1,h2,h3,h4,h5,h6,pre {  
    margin:0;  
}
```



## 外边距 margin ( 续2 )

- margin 可取值为 auto , 由浏览器计算外边距
  - 实现水平方向居中显示的效果

```
div
{
    width:100px;
    height:100px;
    border:1px solid black;
    margin:auto;
}
```



设置水平方向居中

# 什么是内边距

- 内边距：内容区域和边框之间的空间
  - 会扩大元素边框所占用的区域

```
div
{
    border:2px solid red;
    width:200px;
    height:50px;
}
#d1 {padding:20px;}
```



# 内边距 padding

- 内边距：内容与框线之间的距离
  - padding : value;
- 内边距属性值可以为像素、百分比，但不能为负数
- 单边设置
  - padding-top/right/bottom/left : value;



# 尺寸与框模型

- 使用尺寸、边框、边距属性为页面元素定义样式



# 常用样式属性

## 常用样式属性

### 背景属性

背景色 background-color

背景图片 background-image

### 字体属性

控制字体

字体属性 font

控制文本格式

### 显示方式

显示方式

display 属性

# 背景属性





# 背景色 background-color

- background-color 属性用于为元素设置背景色
  - 接受任何合法的颜色值
  - 可取值为 transparent
- 为元素背景设置一种纯色
  - 会填充元素的内容、内边距和边框区域
  - 如果边框有透明部分，会透过这些透明部分显示出背景色。

```
div
{
    border:2px dashed black;
    width:200px;
    height:50px;
    background-color:#ccc;
}
```



# 背景图片 background-image

- 默认值是 none，表示背景上没有放置任何图像
- 如果需要设置一个背景图像，需要用起始字母 url 附带一个图像的 URL 值
  - 可以是相对 URL 或者绝对 URL

```
body
{
    background-image: url("image/bg_01.gif");
}
```



# 字体属性



# 控制字体

- 指定字体
  - font-family : value1,value2;
- 字体大小
  - font-size : value;
- 字体加粗
  - font-weight : normal/bold/value;
- 字体样式
  - font-style : normal/italic;
- 小型大写字母显示
  - font-variant : normal/small-caps;



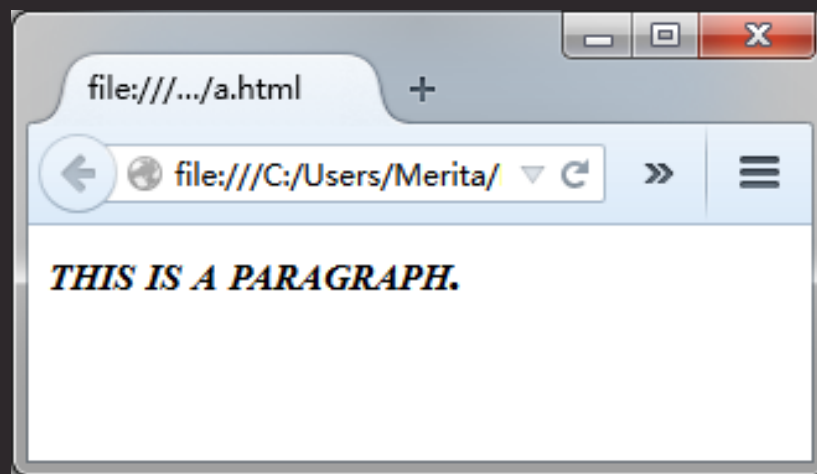
# 控制字体（续1）

样式表中：

```
p {  
    font-family : Times, 'New York', serif ;  
    font-size : 14pt ;  
    font-weight : bold ;  
    font-style : italic;  
    font-variant : small-caps;  
}
```

html 文档中：

<p>this is a paragraph.</p>



# 字体属性 font

- 字体属性 font 用于把所有针对字体的属性设置在一个声明中
- 为简写属性，包含6个值，可以按顺序设置
  - font : font-style font-variant font-weight font-size font-family;
  - 不设置的值，则使用默认设置



# 控制文本格式

- 文本颜色 `color : value;`
- 文本排列 `text-align : left/right/center;`
- 文字修饰 `text-decoration : none/underline;`
- 行高 `line-height : value;`



# 控制文本格式（续1）

```
div {
    color : #FF0000 ;
    text-align : left ;
    text-decoration : underline ;
    height:50px;
    border : 1px solid black;
}
```

div中的文本

```
div {
    color : #FF0000 ;
    text-align : left ;
    text-decoration : underline ;
    height:50px;
    border:1px solid black;
    line-height:50px;
}
```

<div>div中的文本</div>

div中的文本





# 总结和答疑

---