

# Web前端开发技术

(*HTML5 + CSS3 + Javascript*

*t)*

## ■ 单元2 网页元素

## 单元2 网页元素

### 目 录

- 2.1 任务1 创建和美化网页文本元素
- 2.2 任务2 创建和美化网页图像元素
- 2.3 任务3 创建网页布局元素
- 2.4 任务4 创建和美化网页导航元素
- 2.5 任务5 添加网页媒体元素
- 2.6 任务6 创建网页表单元素

## 2.3 任务3 创建网页布局元素

### 目 录

- 子任务1 使用table标签排列超链接
- 子任务2 使用div标签创建布局元素

## 子任务1 使用table标签排列网页元素

## ✧使用html中的table标签组实现网页元素的排列



# 子任务1 使用table标签排列网页元素

表格由 `<table>` 标签来定义。每个表格均有若干行（由 `<tr>` 标签对定义），每行被分割为若干单元格（由 `<td>` 标签定义）。`<td>`代表表格数据（`table data`），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等。HTML中创建一个表格，包含2行3列：

```
<table>
  <tr>
    <td> data</td>
    <td> data</td>
    <td> data</td>
  </tr>
  <tr>
    <td> data</td>
    <td> data</td>
    <td> data</td>
  </tr>
</table>
```

A diagram illustrating a 2x3 table structure. It consists of a light gray outer rectangle containing two light blue horizontal bars. Each bar contains three dark blue squares, each with the word "data" in white text, representing the cells of the table.

data	data	data
data	data	data



`table>tr*2>td*3`

```

<table border="1">
  <tr>
    <td>red</td>
    <td>yellow</td>
  </tr>
  <tr>
    <td>blue</td>
    <td>green</td>
  </tr>
</table>

```



red	yellow
blue	green

表头单元格 <th>

班级	学生数	平均成绩
一班	30	89
二班	35	85
三班	32	80

```

<table border="1">
  <tr>
    <th>班级</th>
    <th>学生数</th>
    <th>平均成绩</th>
  </tr>
  <tr>
    <td>一班</td>
    <td>30</td>
    <td>89</td>
  </tr>
  <tr>
    <td>二班</td>
    <td>35</td>
    <td>85</td>
  </tr>
  <tr>
    <td>三班</td>
    <td>32</td>
    <td>80</td>
  </tr>
</table>

```

- 一个表格中，只允许出现一对`<table>`标签。表格内有多少对`<tr></tr>`标签，就表示表格中有多少行。HTML5中不再支持这两者的任何属性。一对`<tr></tr>`标签内有多少对`<td></td>`标签，就表示该行有多少个单元格。HTML5中，`<td>`还含有`colspan`和`rowspan`两个属性，前者是跨列合并，后者是跨行合并。

# 表格的CSS样式

## 表格

border  
width  
height  
border-collapse

### 表格大小

width,height  
属性：宽、高

```
table{  
  width: 500px;  
  height: 200px;  
}
```

### 表格边框

border 属性

```
table, td, th {  
  border: 1px solid #eee;  
}
```

### 表格边框

border-collapse  
属性

```
table {  
  border-collapse: collapse;  
}
```

内容1	内容2
内容1	内容2

内容1	内容2
内容1	内容2



## 奇偶选择器

`:nth-child(odd|even)`

```
tr:nth-child(odd){  
    background-color:#EAF2D3;  
}
```



tr的父元素table的奇数个子元素



Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK
Paris spécialités	Marie Bertrand	France

## 子任务2 使用div标签创建布局元素

使用布局元素div标签实现网页元素的布局，并进行适当的美化



# div标签



将进酒 | 送友人 | 静夜思



## 将进酒

君不见，黄河之水天上来，奔流到海不复回。

君不见，高堂明镜悲白发，朝如青丝暮成雪。

人生得意须尽欢，莫使金樽空对月。

天生我材必有用，千金散尽还复来。

烹羊宰牛且为乐，会须一饮三百杯。

岑夫子<sup>1</sup>，丹丘生<sup>2</sup>，将进酒，杯莫停。

与君歌一曲，请君为我倾耳听。

钟鼓馔玉何足贵，但愿长醉不愿醒。

古来圣贤皆寂寞，唯有饮者留其名。

陈王昔时宴平乐，斗酒十千恣欢谑。

主人何为言少钱，径须沽取对君酌。

五花马，千金裘，呼儿将出换美酒，与尔同销万古愁。

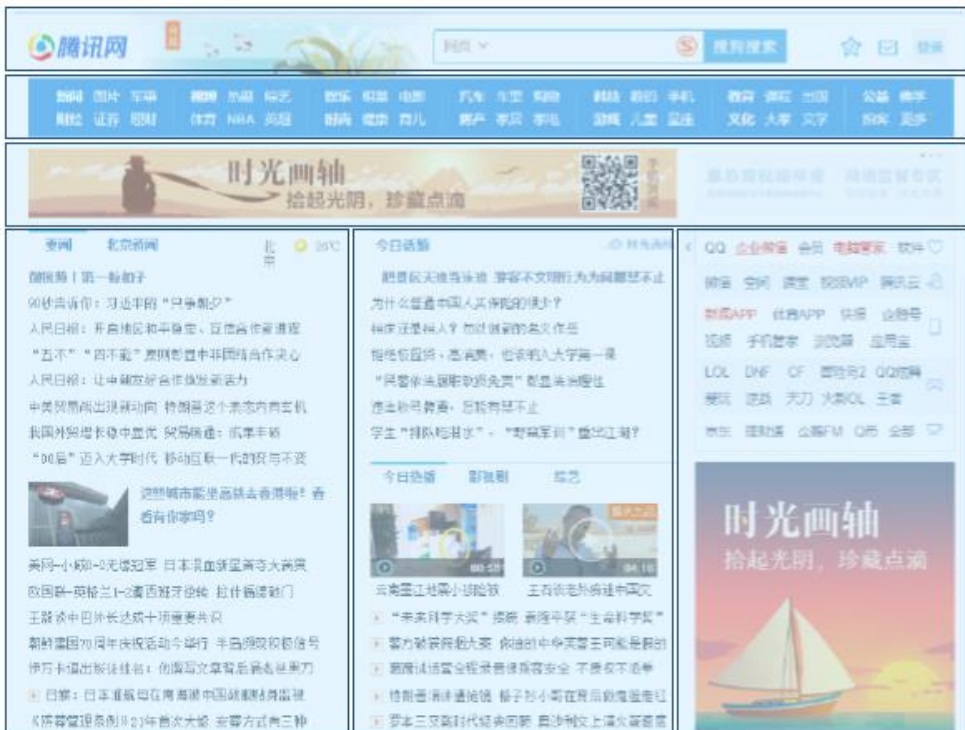
1.岑夫子：人名

2.丹丘生：人名

```
7 <body>
8 <div id="container" align="center">
9   <div id="nav">
10     <p>将进酒 | 送友人 | 静夜思</p>
11   </div>
12   <hr>
13   <div id="content">
14     
15     <h1>将进酒</h1>
16     <p>君不见，黄河之水天上来，奔流到海不复回。 </p>
17     <p>君不见，高堂明镜悲白发，朝如青丝暮成雪。 </p>
18     <p>人生得意须尽欢，莫使金樽空对月。 </p>
19     <p>天生我材必有用，千金散尽还复来。 </p>
20     <p>烹羊宰牛且为乐，会须一饮三百杯。 </p>
21     <p>岑夫子<sup>1</sup>，丹丘生<sup>2</sup>，将进酒，杯莫停。 </p>
22     <p>与君歌一曲，请君为我倾耳听。 </p>
23     <p>钟鼓馔玉何足贵，但愿长醉不愿醒。 </p>
24     <p>古来圣贤皆寂寞，唯有饮者留其名。 </p>
25     <p>陈王昔时宴平乐，斗酒十千恣欢谑。 </p>
26     <p>主人何为言少钱，径须沽取对君酌。 </p>
27     <p>五花马，千金裘，呼儿将出换美酒，与尔同销万古愁。 </p>
28   </div>
29   <hr>
30   <div id="exp">
31     <p>1.岑夫子：人名</p>
32     <p>2.丹丘生：人名</p>
33   </div>
34 </div>
35 </body>
```

# 区域标签div

HTML 中的区域标签: `<div>` `</div>`



```
<div align="center">  
  <h1>web 前端开发</h1>  
  <p>HTML</p>  
  <p>CSS</p>  
  <p>JavaScript</p>  
</div>
```



## Web前端开发

HTML

CSS

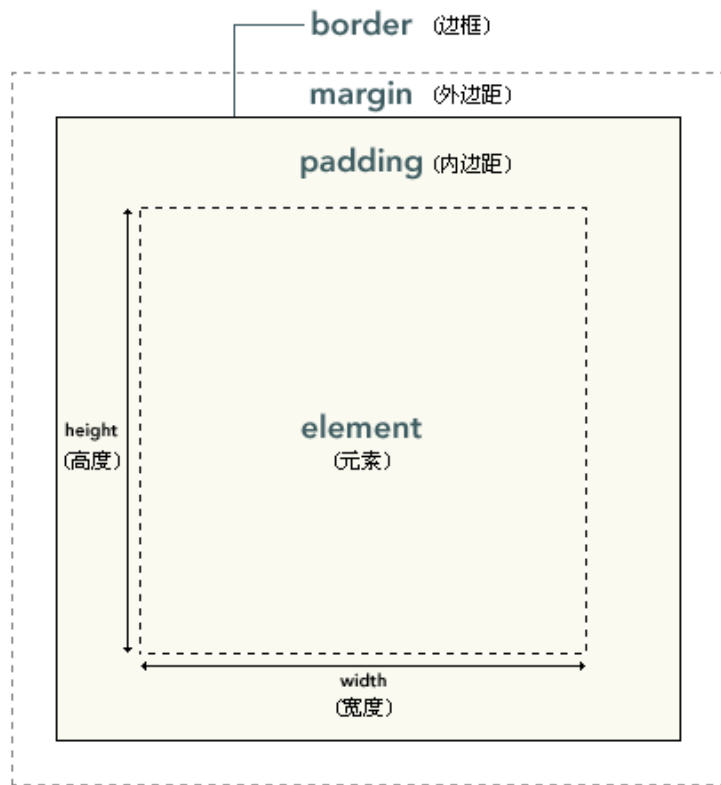
JavaScript

# CSS的盒子模型

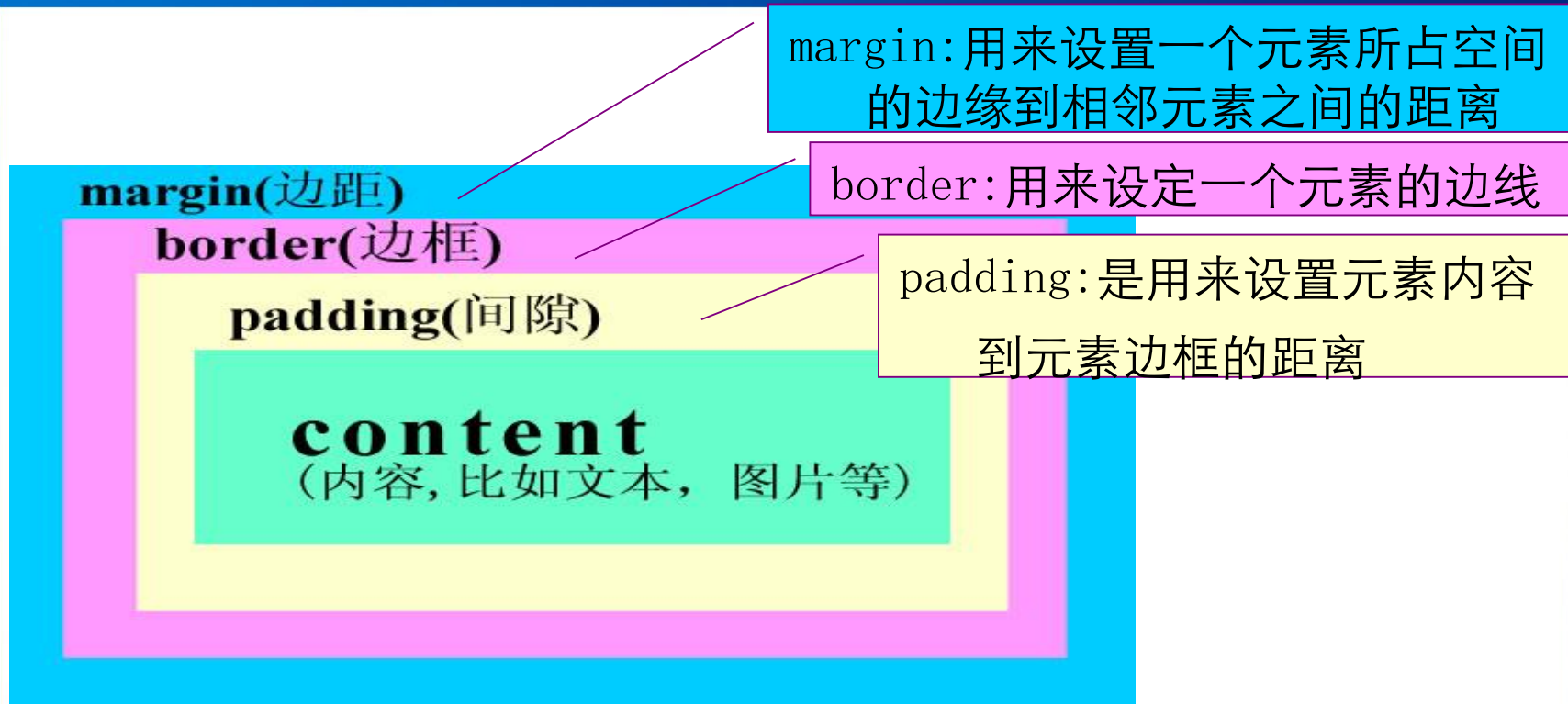
- 盒子模型是CSS的基石之一，它指定元素如何显示以及如何相互交互
- 页面上的每个元素都被浏览器看成是一个矩形的盒子，这个盒子由元素的内容、填充、边框和边界组成。
- 网页就是由许多个盒子通过不同的排列方式（上下排列，并列排列，嵌套排列）堆积而成。

# CSS 盒子 (BOX) 模型

✧ element是实际的内容，具有height和width属性，内容的外围为内边距，用padding属性进行设置。内边距的边缘是边框，用border属性进行设置。边框以外是外边距，用margin属性进行设置，外边距设置不同盒子之间的空隙，外边距默认是透明的。盒子模型也可以设置背景background属性，一般背景属性应用于由内容和内边距、边框组成的区域。



# 盒子模型 (Box model)



# CSS的盒子模型



margin (外边距)

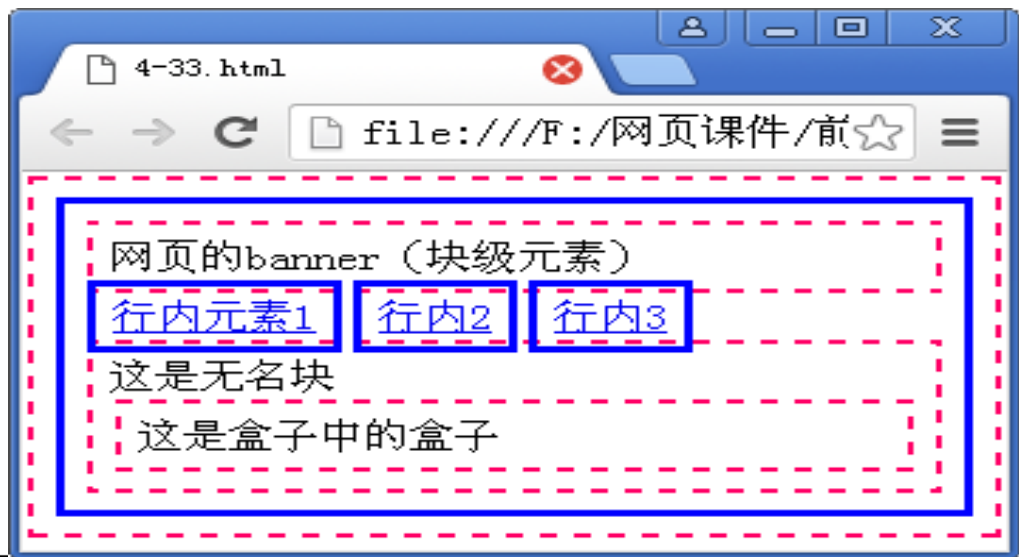
border (边框)

padding (填充距)



# CSS的盒子模型

- 每个HTML元素都是一个装了东西的盒子
- 默认情况下盒子的边框是**无**，背景色是**透明的**，所以在默认情况下看不到盒子



为所有HTML元素添加边框后

# 盒子最终占据的页面宽度

占据的宽度=左外边距+左边框+左填充+width  
+右填充+右边框+右外边距



- ❖ CSS 属性中的 **width** 和 **height** 指的是 **content** 区域的宽和高。
- ❖ CSS 背景属性覆盖的是 **content** 和 **padding** 区域。

# 盒子模型的相关属性

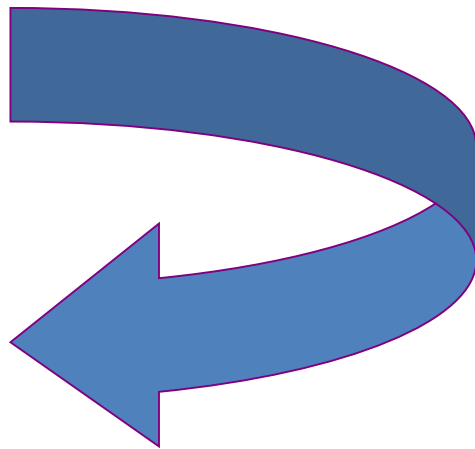
margin: 6px 0 4px 8px;

border: 1px solid red;

padding: 5px;

顺时针方向

上 右 下 左



# 盒子模型的相关属性

CSS的margin属性、padding属性设置的时候，可以为每个边单独设置不一样的值，如：

```
p {  
margin-top:10px;  
margin-bottom:10px;  
margin-right:8px;  
margin-left:5px;  
}
```

属性的单位也支持很多种，比如像素、英寸、毫米、em、百分比等，注意**百分比是相对于其父元素**的width计算的，

# 盒子模型的相关属性

✧CSS的margin属性可以根据情况，分别设置1到4个值，规则如下：  
如果提供全部四个参数值，将按上—右—下—左的顺序作用于四边。

```
body { margin: 36px 24px 36px 25px;}
```

如果只提供一个，将用于全部的四边。

```
body { margin: 11.5%; }
```

如果提供两个，第一个用于上—下，第二个用于左—右。

```
body { margin: 36px 24px;}
```

如果提供三个，第一个用于上，第二个用于左—右，第三个用于下。

```
body { margin: 36px 24px 36px;}
```

# 对盒子模型的总结

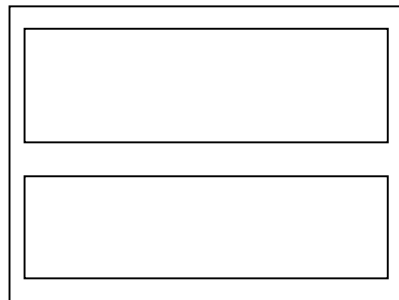
- 边框是实的，我们可以看到实实在在的边框，而填充和外边距都是虚的，我们只能看到他们对元素的影响，边框border默认值为0，即不显示
- 盒子模型中能设置两类颜色，即边框颜色和背景颜色
- 盒子模型可设置三类距离，即外边距margin，填充距padding和边框值border
- 外边距margin可为负，填充padding不可为负

# 各种元素盒子属性的默认值

- 大部分html元素的盒子属性:margin, padding默认值都为0;
- body, p, ul, li, form等少数html元素的margin, padding浏览器默认值不为0, 有时可先设置它们的这些属性为0。
- input元素的边框属性默认不为0。

# 盒子模型的应用

- 制作日历效果



- 制作要点：
  - 3个盒子，外面的盒子设置宽度和边框
  - 设置背景颜色和字体颜色（上下盒子正好相反）
  - 设置水平居中和垂直居中



# chrome浏览器的CSS调试功能

padding位于盒子的边框和内容之间，默认情况下元素的背景图像是以padding的左上角为基准点在盒子中平铺的。

div.box1 210px x 132px

元素占据的大小

元素的行内样式

元素应用的选择器样式

元素的浏览器默认样式

元素继承的样式

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>_</head>
  <body>
    ... <div class="box1">padding位于盒子的边框和内容之间，默认情况下元素的背景图像是以padding的左上角为基准点在盒子中平铺的。</div>
  </body>
</html>
```

html body div.box1

Styles Event Listeners DOM Breakpoints Properties

Filter + .cls

```
element.style {
}

.box1 {
  width: 168px;
  height: 90px;
  border: 6px solid red;
  background: url(images/bac2.jpg) no-repeat -20px -20px;
  text-decoration: underline;
  padding: 15px;
  margin: 5px;
}
```

div { user agent stylesheet display: block; }

Inherited from body padding-1.html:8

```
body {
  line-height: 1.6em;
}
```

margin 5

border 6

padding 15

15 168 x 90 15 6

15 6

6

5

Filter Show all

- background-attachment scroll
- background-clip border-box
- background-color rgba(0, 0, 0, 0)

# 文档流的概念

文档流，就是html文档中，所有标记在浏览器中显示的先后顺序。普通文档流中标签在页面上的位置由标记在html代码中的位置和标签自身的属性决定的，默认情况下，位置靠前的先显示，位置靠后的后显示；根据不同的属性决定，块级元素是**自上而下**，而行内元素是**从左到右**依次进行显示。

# HTML标签的分类

CSS中将html标签分为三种不同的类型：块级元素、行内元素和行内块级元素。

**1.块级元素**，每个块级元素都是从一个新行开始显示，而且其后的元素（不论是否为块级元素）也需另起一行进行显示；块级元素属于盒子模型，可以设置盒子的各项属性；块级元素的默认宽度是其父容器的100%。

html中段落<p>、标题<h1><h2>...、列表<ul><ol><li>、表格<table>、表单<form>、网页主体<body>等标签都称之为块级元素，这些元素显示为一块内容；

**2.行内元素**，也叫内联元素，它们的内容显示在行中，即“行内框”。其主要的特点是：所有的行内元素在一行上，到达父容器的边缘，自动换行；**不可以设置宽度width和高度height**；可以设置行高line-height。

其中，html中表单项标记<label>和<input>、超链接<a>、图像<img>、<span>、<strong>和<em>等都为行内元素。

**3.行内块级元素**，转换行内元素，使得行内元素可以设置宽度和高度属性，是使元素以块级元素的形式呈现在行内。将在一行内水平排列，但每个元素又具有块级元素的特点

# CSS的display属性

✧规定元素应该生成的框的类型，也就是可以将html标记默认的标志类型进行修改。最主要的应用之一是display属性可以使得HTML标记在块级元素、行内元素和行内块级元素之间进行转换。经常使用的属性值：

值	描述
none	此元素不会被显示（隐藏元素）。
<b>block</b>	此元素将显示为块级元素。
<b>inline</b>	默认。此元素会被显示为内联元素。
<b>inline-block</b>	行内块元素，是使元素以块级元素的形式呈现在行内。
list-item	此元素会作为列表显示。
inherit	规定应该从父元素继承 display 属性的值。

## 子任务2 使用div标记创建布局元素

✧<div> 标记和<span>标记

**<div>** 标记，没有特别的含义，属于**块级**元素，主要用来布局网页的，也称之为“布局”标记。

**<span>**标记，没有特别的含义，属于**行内**元素，主要用来表现个别样式。

# 盒子的三种定位形式

- 在标准流(normal flow)下的定位
  - 在浮动(float)属性下的定位
  - 在定位(position)属性下的定位
- 
- 除非设置浮动属性或定位属性，否则所有盒子都是在标准流中定位。顾名思义，标准流中元素盒子的位置由元素在HTML中的位置决定。

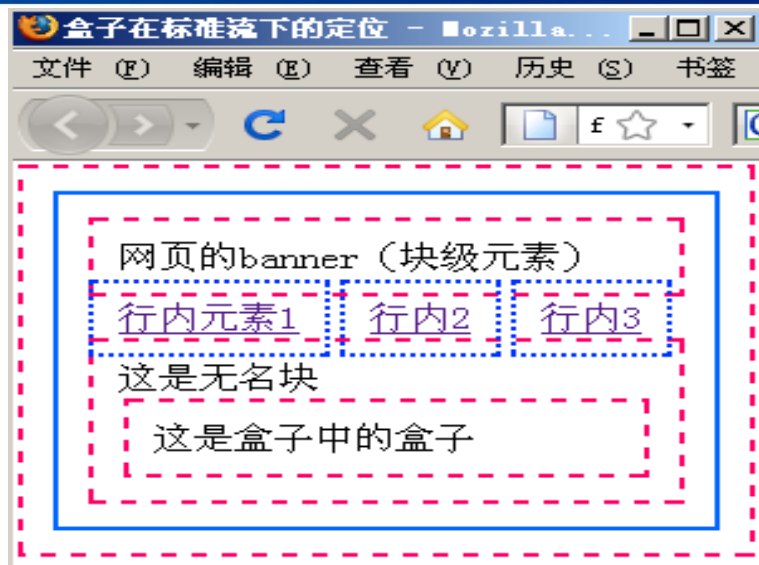
# 标准流

- 标准流下的定位：HTML元素在标准状况下的定位方式，特点如下：
  - 块级元素占满整个一行，在页面中**竖向排列**
  - 行内元素在同一行内**横向排列**
- 元素不会移动到其它地方去，对于嵌套的元素，盒子也是包含的关系

# 标准流下的盒子排列分析

```
<style>
* {
border: 2px dashed #F06;
padding: 10px;
margin: 2px;
}
body{ border: 2px solid blue;}
a{ border: 2px dotted blue;}
</style>
```

```
<div>网页的banner（块级元素）</div>
<a href="#">行内元素1</a><a href="#">行内2</a>
<a href="#">行内3</a>
<div>这是无名块<p>这是盒子中的盒子</p></div>
```





# 盒子模型的应用

- 制作留言评论界面



王小波曾经说过，大多数人在说话，少数人在沉默；大多数人幸福，少数人痛苦。所以，我曾经很热切的希望我能成为大多数，可是我失败了。我是一个与众不同的人，自始至终都是一个与众不同的人，这种与众不同不是我骄傲的资本，而是我前进的动力。...

- 制作要点：
  - 左边的图片是子元素，对它设置负margin-left，使其向左移动到父元素外面
  - 右边的文本区域是父元素

# CSS的层叠性和继承性 - 引例

- 如果元素同时被多个选择器选中，会怎么样？

```
p{ font-size:14px;}
```

```
.red{ color: red; }
```

```
#k2{ background-color: #ff9; }
```

.....

```
<p id="k2" class="red">被选中的元素</p>
```

样式不冲突时

- 显示效果

被选中的元素

# 引例

- 如果元素同时被多个选择器选中，会怎么样？

```
p{ color: yellow;}
```

```
.red{ color: red; }
```

```
#k2{ color: blue; }
```

.....

```
<p id="k2" class="red">被选中的元素</p>
```

- 显示效果

被选中的元素

样式冲突时

# CSS的层叠性

- **层叠性**：当HTML元素被多个选择器同时选中时
  - 1) 如果多个选择器定义的样式不冲突，则元素将应用所有选择器定义的样式。
  - 2) 如果多个选择器定义的样式冲突，则元素将应用**优先级高**的选择器定义的样式。

# CSS选择器的优先级

- CSS规定选择器的优先级从高到低为：

—行内样式 > ID选择器 > 类选择器 > 标签选择器

- 总的原则是：

越特殊的样式，优先级越高。

# !important关键字

- !important: 强制改变选择器的优先级

!important > 行内样式 > ID样式 > 类别样式 > 标记样式

p{color: yellow;}

.red{ color: red !important; }

#k2{ color: blue; }

<p id="k2" class="red">被选中的元素</p>

则显示效果为: 被选中的元素

# CSS的继承性

- 继承性：指子元素会继承父元素定义的某些样式。例如：

```
div{  
    font-size: 24px; color: red; }
```

.....

```
<div>  
    <p>基于<b>TCP/IP</b>的网络编程</p>  
</div>
```

显示效果为：

基于**TCP/IP**的网络编程

# CSS 继承性例题

```
<style>
```

```
body {
```

```
text-align: center;}
```

```
p {
```

```
font-size: 24px;
```

```
text-decoration: underline;}
```

```
em {
```

```
color: #F00;}
```

```
.right{
```

```
text-align: right;}
```

```
</style>
```



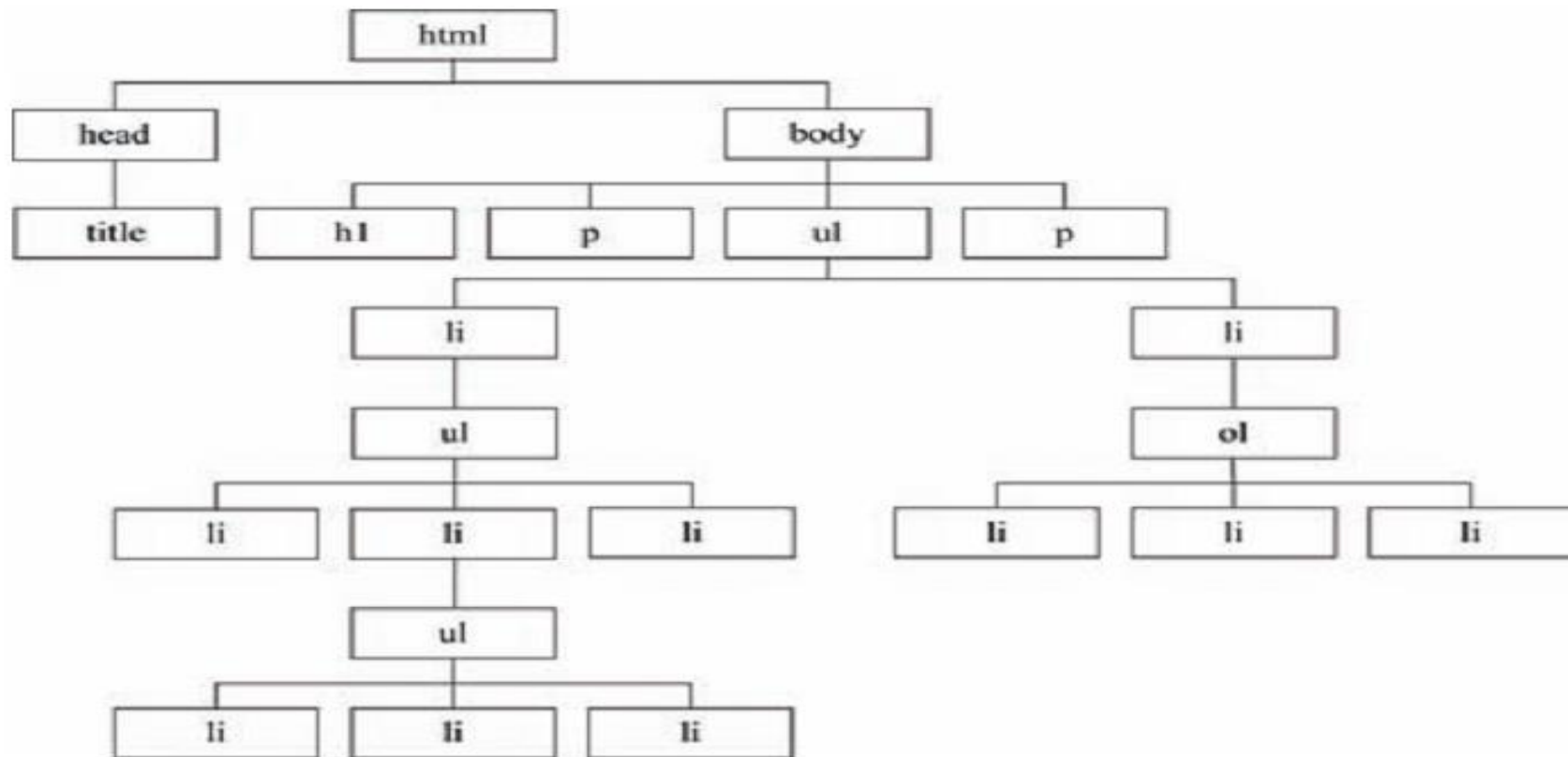
```
<body>
```

```
<p><em>电子商务</em>教研室</p>
```

```
<p class="right"><em>电子商务</em>教  
研室</p></body>
```



# 继承关系树型图 (DOM)



# 要善于利用CSS继承性

- 利用CSS继承性，可大大缩减代码的编写量。

例如，如果网页中大部分文字的字体都是14px，则可以对body元素定义：

```
body{ font-size: 14px; }
```

有些特殊的文本如果要求其他字体大小，则利用类选择器或id选择器对它们重新定义。

# 不是所有的CSS属性都具有继承性

color, font-\*

text-indent

text-align

line-height letter-spacing

border-collapse

opacity

有继承性

无继承性

所有盒子属性:

border, margin, padding

布局属性:float,

display, position等

一般是**CSS**的文本、段落属性具有继承性，而其他属性（如背景属性、布局属性等）则不具有继承性

# 复合选择器

- 复合选择器就是两个或多个基本选择器，通过不同方式组合而成的选择器。主要有：
  - 交集选择器
  - 并集选择器
  - 后代选择器

# 1 交集选择器

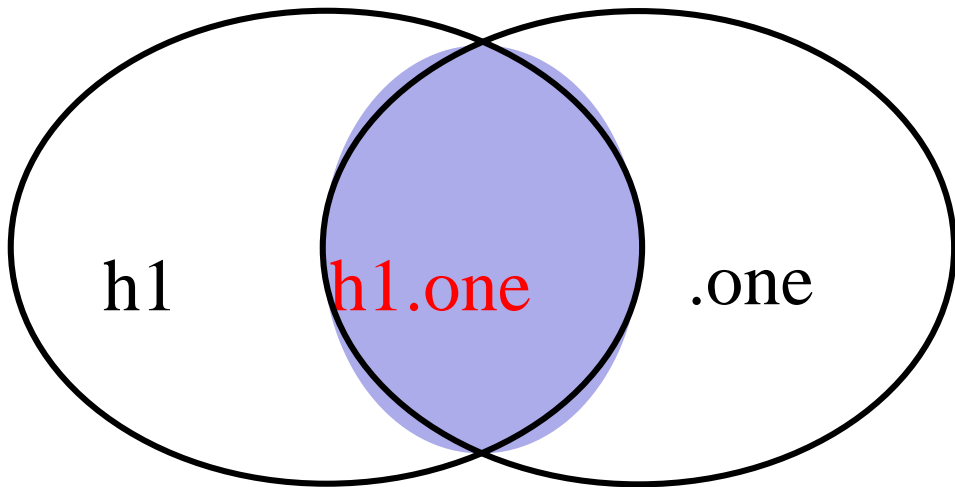
示例：选择所有类名为two的p元素：

`p.two{ ... }`

- 交集选择器：由两个选择器直接连接构成，其结果是选中两者各自作用范围的交集。
- 其中第一个必须是标记选择器，第二个必须是类选择器或id选择器。例如：`h1.one`；`p#intro`。这两个选择器之间不能有空格。

# 交集选择器的作用范围

- 交集选择器将选中同时满足前后二者定义的元素，也就是前者定义的标签类型，并且指定了后者的类别或id的元素。它的作用范围如图所示：



# 交集选择器举例

- 例如由于所有的表单控件都是input标签，如果只希望某个表单控件（如提交按钮）受影响，则可用交集选择器

```
input.red {color: #F00;}
```

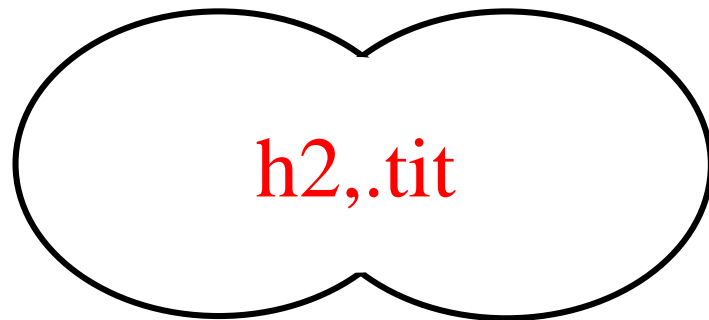
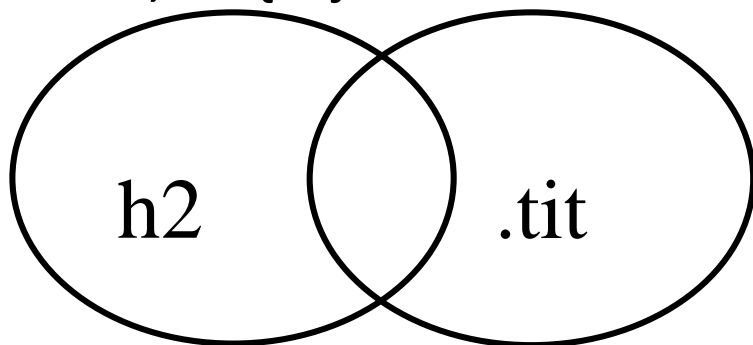
```
<input type="submit" name="Sub" class="red" value="提交" />
```

## 2 并集选择器

- 示例：选中所有h2,h3和类名为tit的元素。

`h2,h3,.tit { ... }`

- 并集选择器：就是对多个选择器进行**集体声明**，多个选择器之间用 **,** 隔开。
- 如h2,.tit{...}的选择范围如图所示：





# 并集选择器举例

```
<style>
h1,h2,h3,p {
font-size: 12px;
color: purple;
}
h2.special, .special, #one {
text-decoration: underline;}
</style>
<body>
<h1>示例文字h1</h1>
<h2 class="special">示例文字h2</h2>
<h3>示例文字h3</h3>
<h4 id="one">示例文字h4</h4>
<p class="special">示例文字p1</p>
```



# 通配选择符 \*

如果要对网页中所有的元素进行集体声明可使用通配选择符 **\***, 这种方式通常也被称为“css的重置”:

```
*{  
    /* 通配选择符 */  
    color: purple; /* 文字颜色 */  
    font-size: 15px; /* 字体大小 */  
}
```

最常见的通配选择符的设置如下:

```
{  
    margin:0;  
    padding:0;  
}
```

### 3 后代选择器

- 示例：选中所有位于#nav元素内的a元素。

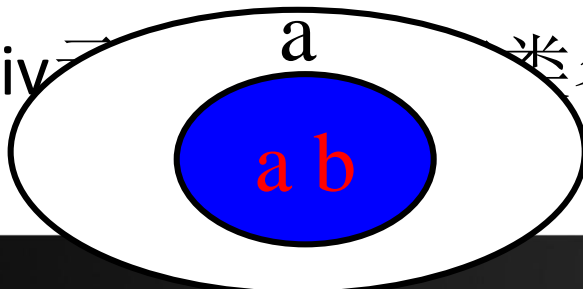
#nav a { ... }

- 后代选择器：选择作为指定元素后代的元素。  
指定元素和其后代元素间用空格隔开。例如：

h1 em{}: 选择位于h1元素中的em元素

.cont \*{}: 选择类名为cont元素的所有后代元素

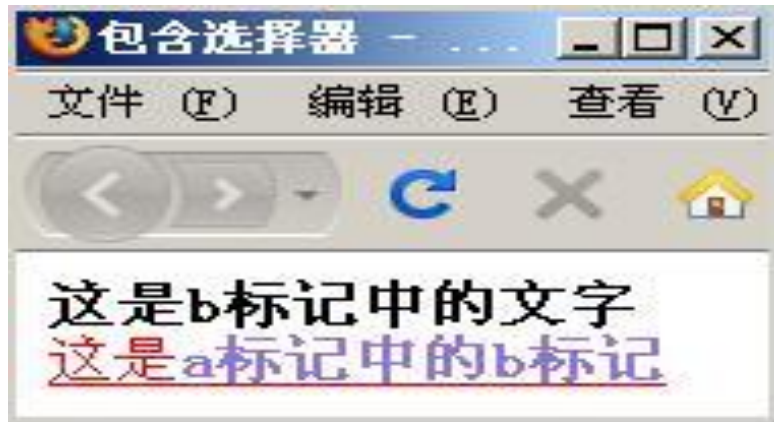
div .tit{}: 选择div元素类名为tit的元素



# 后代选择器例题

```
<style>
a {
font-size: 16px;
color: red; }
a b {
color: mediumpurple; }
</style>
<body>
<b>这是b标记中的文字</b><br />
<a href="#">这是<b>a标记中的b<span>标记</span></b></a>
```

提示: span元素继承了父元素“a b”的颜色样式



# 后代选择器举例

- 后代选择器还能够进行多层嵌套。例如：
  - `.special b { color : red }`
  - `#menu li { padding : 0 6px ; }`
  - `td.top .ban1 strong { font-size : 16px ; }`
  - `#menu a:hover b`
- 经验：选择器的嵌套在CSS的编写中可以大大减少对class或id的声明。因为在构建页面html框架时通常只给外层标签（父标签）定义class或id，内层标签（子标签）能通过嵌套表示的则利用后代选择器来选中。

## 4. 复合选择器的优先级

- 复合选择器的优先级比组成它的单个选择器的优先级都要高。

优先级权重：ID选择器=100，类选择器=10，标记选择器=1

h1{color:red;}	/* 权重=1 */
p em{color:blue;}	/* 权重=2 */
p.note em.dark{color:gray;}	/* 权重=22 */
#main{color:black;}	/* 权重=100 */

- 当权重值一样时，会采用“层叠原则”，一般后定义的会被应用

# 复合选择器优先级计算 例题

```
#aa ul li { color:red }
```

优先级102

```
.aa { color:blue }
```

优先级10

.....

```
<div id="aa">
```

```
<ul>
```

```
<li class="aa">复合选择器</li>
```

```
</ul>
```

```
</div>
```

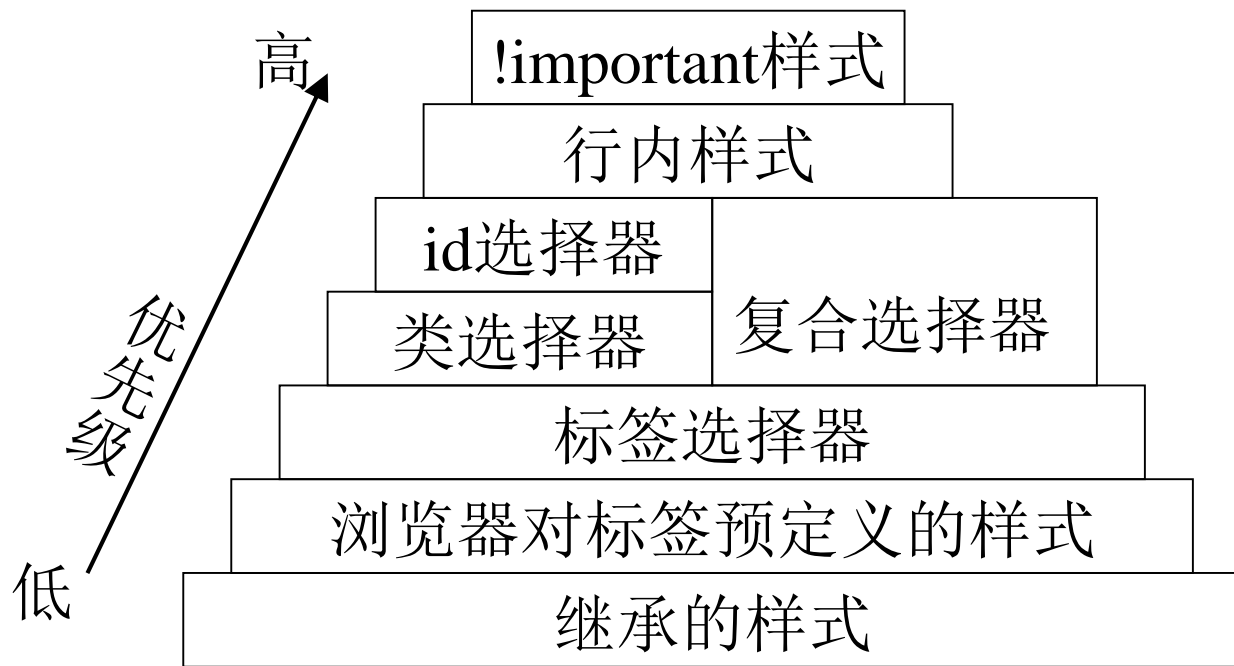
```
<div class="aa">
```

```
复合选择器
```

```
</div>
```

注意：li元素同时被两个选择器选中

# CSS样式的优先级 总图





# 复杂选择器名称的分解

- CSS选择器分解的原则是：先逗号，接着空格，再冒号，最后点号。例如：

#menu a.clas:hover b, .special b.clas { ... }

就可先分解为：

#menu a.clas:hover b      和      .special b.clas



三层后代选择器



两层后代选择器

# 课后测

1、设置围绕表格的边框宽度的HTML代码是:

A.<table size="">

B.<table border="">

C.<table bordersize="">

D.<tableborder="">

2、定义表行的HTML是:

A.<table>

B.<td>

C.<th>

D.<tr>

# 课后测

3、定义表头的HTML是：

A.<table>

B.<td>

C.<tr>

D.<th>

4、表示区域的标签是：

A.table

B.ol

C.div

D.ul

# 判断题

- 1、如果设置table标签的border属性，默认可以看到两层边框线。
- 2、可以使用table-collapse属性将表格边框和单元格边框重合在一起显示。

思考题：

奇偶选择器有哪些应用？

深入学习以下内容：

[http://www.w3school.com.cn/cssref/selector\\_nth-child.asp](http://www.w3school.com.cn/cssref/selector_nth-child.asp)



谢 谢