# CSS基础入门

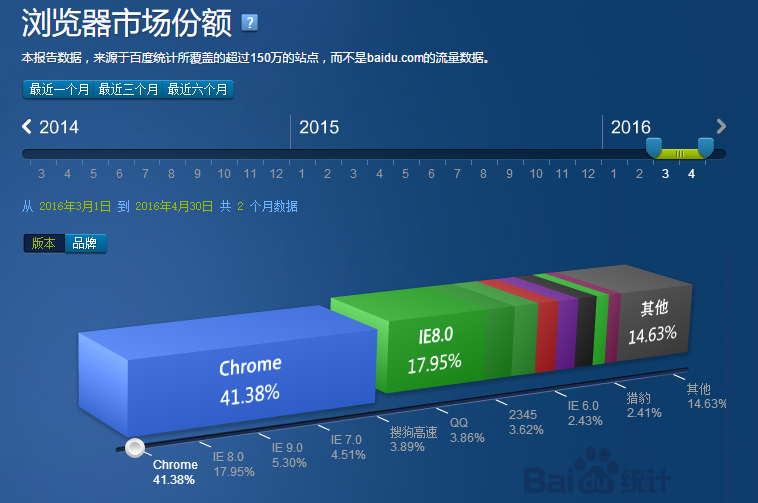
第7天课堂笔记（本课程共8天）

# 

# 浏览器

## 1、浏览器市场份额

谷歌、火狐、IE、苹果、欧朋。



IE8及以下用户还占了很大的份额，制作网页的时候，考虑他们的兼容性问题。

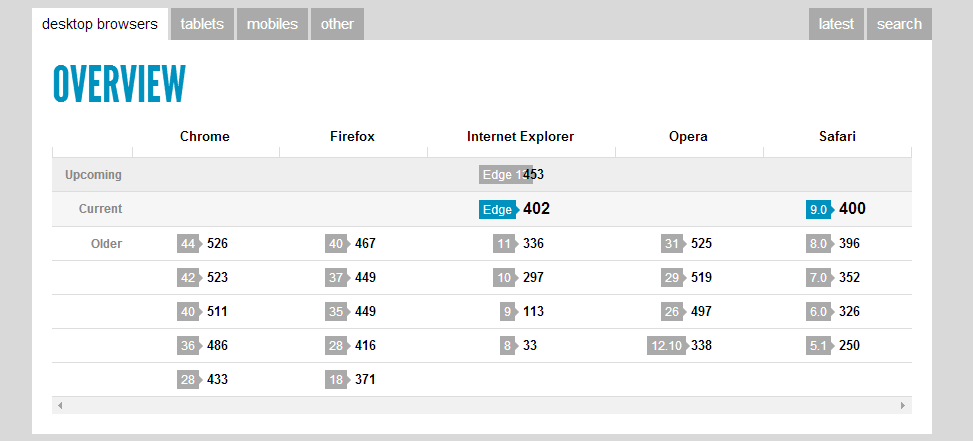
IE版本：机关事业单位。顽固守旧的用户。

## 兼容性

浏览器有版本问题，我们的css版本适应当代的趋势。

html5、css3兼容。

我们的浏览器渲染能力对比。



哆啦A梦测试：



浏览器存在兼容问题，我们在书写网站的时候，必须考虑到，给不同的浏览器写不同的代码。

客户端：前端、IOS、Android等都存在兼容问题。服务器端不存在兼容问题。

为什么学习：

页面兼容到IE8，知道IE6的常见的、臭名昭著的兼容问题。

为了证明你是一个长时间从事前端的人，公司总喜欢用IE6的兼容问题来面试。

# HACK

Hack单词从hacker（黑客）延伸来的。

Hack就是我们针对不同的浏览器去写不同好的html和css。

分为：html的Hack，css的Hack。

## 1、html的Hack

针对不同的浏览器在同一个html文件里，去写不同的html结构。

IE浏览器有很多兼容性问题，给我们提供了一些接口。

Hack：

|  |
| --- |
| 1. <!--[if lte IE 9]> 2. 内容 3. <![endif]--> |

表示Hack符里面的内容只在IE9及以下的浏览器内显示，IE10及以上或者高级浏览器会认为这是注释。

if是如果，lte是less than or equal，小于或者等于，IE浏览器，9代表版本。

endif结束这个Hack。

lte：less than or equal 小于等于

lt：less than 小于

gt：greater than 大于

只在IE8及以下的版本显示，其他的或高级的认为是注释。

|  |
| --- |
| 1. <!--[if lte IE 8]> 2. <![endif]--> |

只在IE10以下的版本显示，其他的或高级的认为是注释。

|  |
| --- |
| 1. <!--[if lt IE 10]> 2. <![endif]--> |

只在IE7以上的版本显示，其他的或高级的认为是注释。

|  |
| --- |
| 1. <!--[if gt IE 7]> 2. <![endif]--> |

固定某一个版本，比如ie6，里面的内容只有IE6认识。

|  |
| --- |
| 1. <!--[if IE 6]> 2. <![endif]--> |

用途：

给低版本浏览器写一些特殊的代码。

只在IE6里加载这个js文件。

|  |
| --- |
| 1. <!--[if IE 6]> 2. <script src="js/png.js"></script> 3. <![endif]--> |

## css的Hack

值的Hack和选择器的Hack。

### 2.1 css值的Hack

/\* IE 6 \*/

.selector { \_color: blue; }

.selector { -color: blue; }

Hack符：-、\_

在属性名的前面加下划线或横线。

表示这个属性只有IE6认识，其他的都不认识这个属性。

|  |
| --- |
| 1. div{ 2. width: 150px; 3. height: 150px; 4. background: red; 5. -background: green; 6. } |

/\* IE 6/7 \*/

Hack符：! $ & \* ( ) = % + @ , . / ` [ ] # ~ ? : < > |

表示这个属性只有IE6/7认识，其他的都不认识这个属性。

|  |
| --- |
| 1. !height: 300px; |

/\* IE 8/9 \*/

.selector { color: blue\0/; }

Hack符：\0/。

需要写在属性值的后面。

只在IE8、9里认识。

|  |
| --- |
| 1. color: gold\0/; |

/\* IE 6/7/8/9/10 \*/

.selector { color: blue\9; }

Hack符：\9.

需要写在属性值的后面，分号前。

|  |
| --- |
| 1. font-size: 30px\9; |

着重记忆：IE6的。

### 2.2 css选择器Hack

/\* IE 6 and below \*/

\* html .selector {}

这种选择器只在IE6里加载，其他的浏览器认为你的选择器是错的。

高级浏览器认为html已经是根标签，\*不是他的祖先元素。

|  |
| --- |
| 1. \* html div{ 2. width: 100px; 3. height: 100px; 4. background: lightgreen; 5. } |

等价：

|  |
| --- |
| 1. div{ 2. \_width: 100px; 3. \_height: 100px; 4. \_background: lightgreen; 5. } |

/\* IE 7 and below \*/

.selector, {}

这种选择器只在IE7及以下版本里加载，其他的浏览器认为你的选择器是错的。

|  |
| --- |
| 1. div,{ 2. width: 100px; 3. height: 100px; 4. background: lightblue; 5. } |

等价：

|  |
| --- |
| 1. div{ 2. !width: 100px; 3. !height: 100px; 4. !background: lightblue; 5. } |

/\* 除了IE 6 \*/

html > body .selector {}

>子级选择器：只选择儿子级，后代其他级别不选。

IE6不认识子级选择器。

除了IE6都能正常加载。

|  |
| --- |
| 1. html>body div{ 2. width: 100px; 3. height: 100px; 4. background: lightblue; 5. } |

# IE6兼容问题

## 选择器兼容

IE6不兼容交集选择器里的类选择器连写。

div.box 可以兼容

div.box.cl 不兼容

|  |
| --- |
| 1. li.item.current{ 2. background: pink; 3. } |

解决方法：使用标签和一个类的交集。

其他的7种选择器完美的兼容：

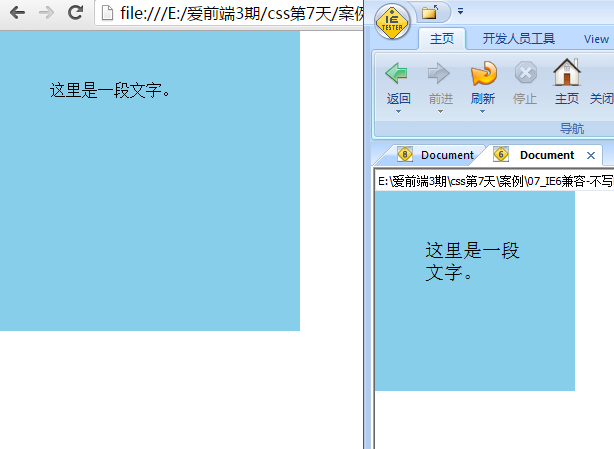
标签、id、类、交集（div.box）、后代、并集、通配符

## 盒模型

### 2.1 DTD

如果不写DTD，IE6里的盒子是内减的，其他浏览器是外扩。

|  |
| --- |
| 1. div{ 2. width: 200px; 3. height: 200px; 4. padding: 50px; 5. background: skyblue; 6. } |

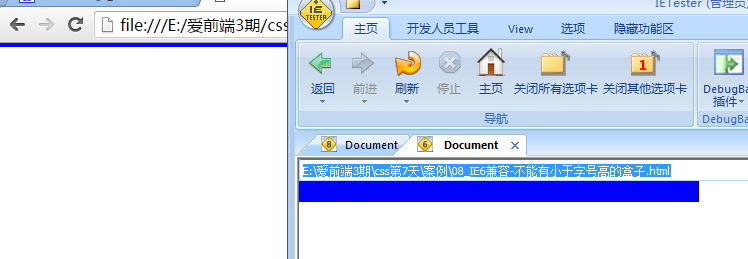


解决方法：写DTD。

### 2.1不能有小于字号的高的盒子

如果盒子高度小于默认字号，不会正常显示。高度会是最小的字号。

|  |
| --- |
| 1. height: 4px; |



解决方法：单独给IE6浏览器，强制给个很小的字号。

|  |
| --- |
| 1. div{ 2. width: 400px; 3. height: 4px; 4. background: blue; 5. \_font-size: 0; 6. } |

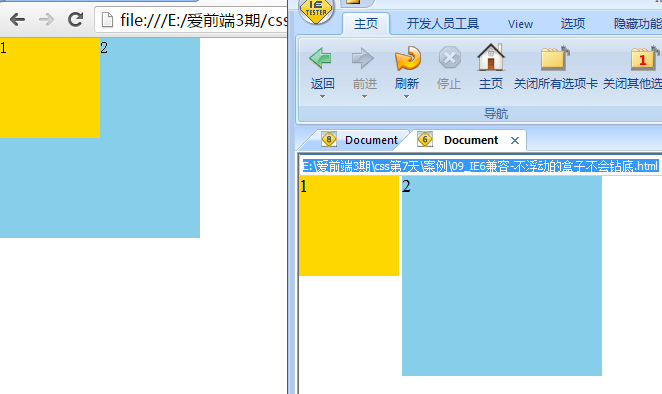
实例：雅虎网站。

## 浮动

### 3.1 不浮动的盒子不会钻到底下

情况：一个盒子浮动，一个盒子不浮动，在IE6里，不浮动的盒子不会钻到浮动盒子的下面占领它原来的标准流位置。

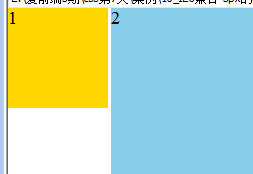
|  |
| --- |
|  |



解决问题：制作压盖效果用定位。同一级的盒子要浮动都浮动。

### 3.2 3px的bug

情况：一个浮动，一个不浮动，IE6里两个盒子之间会出现3px的间距。



解决方法：（不允许这么写）。

必须解决：给左边的浮动盒子加一个-3px的右margin。

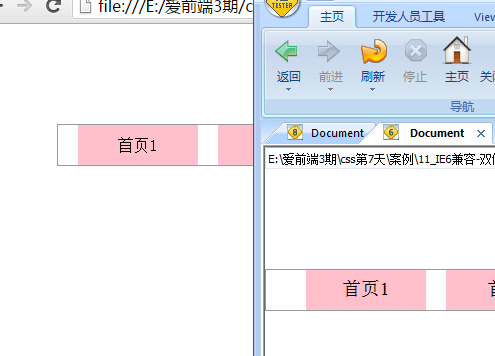
只给IE6浏览器加，注意Hack符。

|  |
| --- |
| 1. .box1{ 2. float: left; 3. width: 100px; 4. height: 100px; 5. background: gold; 6. \_margin-right: -3px; 7. } |

### 3.3 双倍margin问题

情况：一些元素浮动，有一个与浮动方向相同的方向的margin，第一个元素会出现双倍边距的问题。

|  |
| --- |
| 1. ul li{ 2. float: left; 3. width: 120px; 4. height: 40px; 5. background: pink; 6. margin-left: 20px; 7. } |



解决方法：

①不允许父子之间的间距用儿子的margin去踹，用父亲的padding去挤出来，给第一个浮动的元素清除margin，或者给反方向的margin。

|  |
| --- |
| 1. ul li.nomargin{ 2. margin: 0; 3. } |

或者：给margin-right

|  |
| --- |
| 1. ul li{ 2. float: left; 3. width: 120px; 4. height: 40px; 5. background: pink; 6. margin-right: 20px; 7. } |

②（不提倡，不允许父子之间的间距用儿子的margin去踹）给第一个元素的margin减半。

|  |
| --- |
| 1. ul li.first{ 2. \_margin-left: 10px; 3. } |

### 3.4 overflow：hidden失效

IE6不认识overflow：hidden；引申作用，没法帮我们清浮动，自动撑高盒子。

IE浏览器加载机制：

布局：layout

有布局：盒子本身加载的时候，根据自身内容来加载，会强制检测里面的元素，通过里面的元素性质进行自身布局。

没有布局：盒子本身加载的时候，根据祖先元素来加载，不会去管内部的元素。

IE里有很多属性可以触发有布局的机制。触发hasLayout机制，就可以让overflowhidden生效。

有一个属性永远触发hasLayout机制。

|  |
| --- |
| 1. \_zoom: 1; |

## 固定定位不支持

用js来模拟固定定位效果。