

## 第 1 课 HTML 介绍

你在使用电脑的时候,见过哪些文档格式?

.ppt, .txt, .doc, .pdf

### 1.网页也是一种文档

它是一种以 .html 结尾的文档

### 2.新建一个html文档

1)用记事本创建一个01.txt文本,内容为: hello

2)更改01.txt的文档类型为 html

第一个网页就完成了

### 3.html 用什么打开

各种浏览器

### 4.容易出现的问题:

新建html,双击一直是记事本打开而不是浏览器;

计算机默认隐藏了文件的后缀

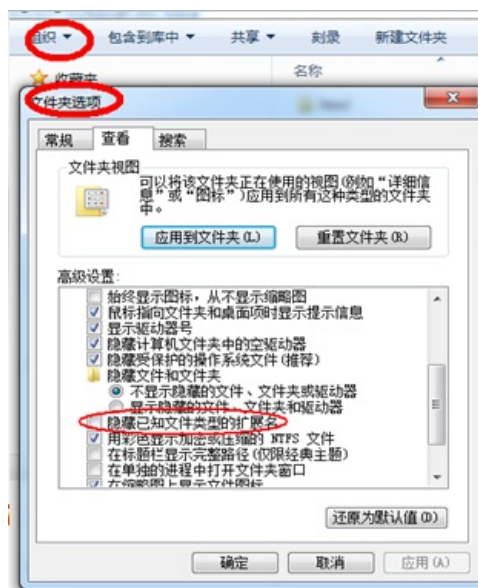
注意:

不要隐藏文件夹的扩展名

组织->文件夹和搜索->查看->隐藏已知类型的扩展名

否则: 新建的文档 a.html.txt

而不是我们想要的 a.html



课后作业:

自行搜索解决为什么有的网页后缀为 htm

## 第 2 课 HTML 开发工具准备

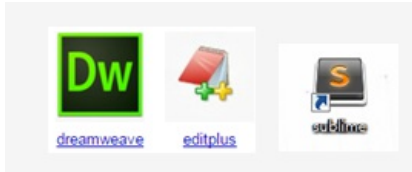
工欲善其事,必先利其器

想学习html,我们需要先把工具准备好

软件的安装:

### 1.开发工具

编辑器:



这些都可以,没有本质区别(这里我们用的是sublime)

语法高亮关键词,方便我们查看即可

记事本也可以编辑

### 2.浏览器

用编辑器编辑之后,我们需要打开和调试

用浏览器打开和调试我们的代码

1)浏览器:用firefox和chrome

这两个浏览器相对来说,最为标准,不要用一些第三方的浏览器,如猎豹和360等

2)firebug 插件安装

### 3.漂亮的字体安装

win7下直接双击就能安装

winxp下,需要放在 C:\Windows\Fonts\

### 4.sublime 安装和使用

打包下载,已经配置好,直接解压就可以使用

注意:

sublime解压要放在全英文的目录下,不然会出一些莫名其妙的问题

包括我们以后安装其它的软件,都要放在英文目录下,因为很多软件都会存在这个问题

**sublime使用(sublime使用技巧.txt)**

```
侧边栏
ctrl+n 新建
ctrl+s 保存
!+tab 生成一个最简单的初始化html5页面

写入
hello sublime!

ctrl+s 保存我们生成页面
ctrl+b 在浏览器中打开这个页面
```

能准确的完成以上步骤,你的准备工作就已经完成了,可以进入下一步的学习

课后作业:

安装 firefox,firebug,及 sublime

## 第 3 课 HTML 组成部分

html既然是一种文档,那么它必然有自己的一种规范

这个规范是如何规定的?

html是由哪几部分组成的

分析以下页面

<http://www.zixue.it>

<http://www.itbool.com>

<http://www.taobao.com>

查看网页源代码

大致总结一下,这几个页面是由哪几部分组成的

dtd,head,body

```
dtd部分
<html>

<head>
头
</head>

<body>
身体
</body>

</html>
```

## 1.dtd的作用即历史

DOCTYPE标签是一个单独的标签

dtd是:

文档类型说明或者是文档定义说明

什么东西都是在发展过程中,都在变化

例:

汽车尾气排放有一定的标准

之前是 欧II,欧III,现在的新标准应该是 欧IV

## html发展历史

```
HTML 4.01 (微小改进) —1999 年 12 月 24 日, W3C 推荐标准

XHTML 1.0 在 2000 年 1 月 26 日成为 W3C 的推荐标准。

HTML 5—2014 年 10 月 28 日, W3C 推荐标准[4] (详见本处参考资料)
```

随便给你一个网页,你让浏览器用什么样的标准来解析它?

用旧的 h4,xh1 还是比较新的 h5 呢,我们需要告诉浏览器用什么标准来解析它

## 2.dtd的作用

就是告诉浏览器,我是用什么标准来写的

```
//用于 HTML 4.0
<!DOCTYPE HTMLPUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
//XHTML 1.0 的 DTD
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
// html5 的 DTD
<!DOCTYPE html>
```

html5众望所归,统一了

不需要写那么复杂的dtd,也不需要考虑各浏览器的兼容,而且html5的dtd非常的简单

taobao 已经使用最新的html5来写了

head头部的meta标签中

description和keywords,用来说明自己页面的关键词,来吸引蜘蛛.

```
<meta name="keywords" content=" 公益php培训, php教程, 免费php教学, PHP学习, php视频下载, 北京php培训" />
<meta name="description" content="免费PHP培训, 每晚教学直播, 屏幕共享。群内助教指导。学习内容: PHP基础,
```

head这部分主要是给机器看

body是人眼看到的部分(文字,图片等)-->我们的主要战场

课后作业:

自行搜索,

了解 html 发展的历史

了解 xhtml2 与 html5 的关系

知道 doctype 声明的意义

## 第 4 课 网页布局之切切豆腐

经过前三课的学习,我们现在已经可以做网页了

准备工作:

我们准备做一个博客网站

<http://www.yanshiba.com>

从大处着眼

将这个网页当作一幅图,画出来

以画画为例,让学员从大处画分成块

网页布局如画画,从大到小,从上到下,从左到右,从外到里

观察 blog 页面,分析结构



做布局,3 个 div,

div 英文 division 区域,块

先做布局.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <div></div>
  <div></div>
  <div></div>
</body>
</html>
```

打开页面,为什么空白?

因为我只是宣布占位,占3个位,至于这个占位多宽,多高,什么颜色,什么线条等

我们并没有说明  
只占位是不够的,我们需要声明宽高背景色

课后作业:  
完成三行的布局

## 第 5 课 css 控制宽高背景

width,height,background

**div**布局,**css**控制

**div** 布局只是完成了城市的规划,至于广场占多少面积,还得有国土资源局来登记。

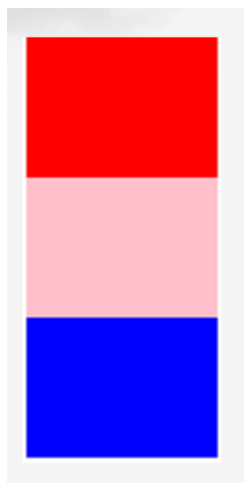
**div** 是地皮,**css** 是房管档案,记住地皮的大小。

写 3 个 **div**,**div**控制宽高背景,全蓝色,看着是一个块  
如何区分每一块?

```
div{
  width: 300px;
  height: 300px;
  background: red;
}
```

**id** 标记每个 **div**, 起个独特的名字, 显示 3 块出来

```
#a{
  width: 300px;
  height: 300px;
  background: red;
}
#b{
  width: 300px;
  height: 300px;
  background: pink;
}
#c{
  width: 300px;
  height: 300px;
  background: blue;
}
```



做网页,虽然我们只学习了这么几课,但是我们知道  
先**div**布局,分好块  
再用**css**控制它的宽高,背景具体显示出来

再往里面塞入文字,大小一调试,一个页面就完成了

### 课后作业:

自行搜索,研究 **css** 有几种引入方式

**div** 除了可以声明 **id** 来控制,还可以声明什么?

如何让 **div** 左右并排显示

## 第 6 课 浮动布局

---

**float:left;**

**float:right;**

博客首页,中间部分左右并排显示

2 个 **div**,分别叫 **right**,**left**不设置宽,设置高和背景,

看是否并排

是否因为 **div** 过大

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
<style>
#right{
  height: 100px;
  background: red;
}
#left{
  height: 100px;
  background: blue;
}
</style>
</head>
<body>
  <div id="right">我是右</div>
  <div id="left">我是左</div>
</body>
</html>
`
```

**div**如果不声明宽,它会使劲的往两边的父元素的位置撑,尽量的宽

把 **div** 变宽变小,仍不能并排

### **div**特性:

**div** 默认占据 1 排

声明了宽,不论多宽多窄,都是独占一行

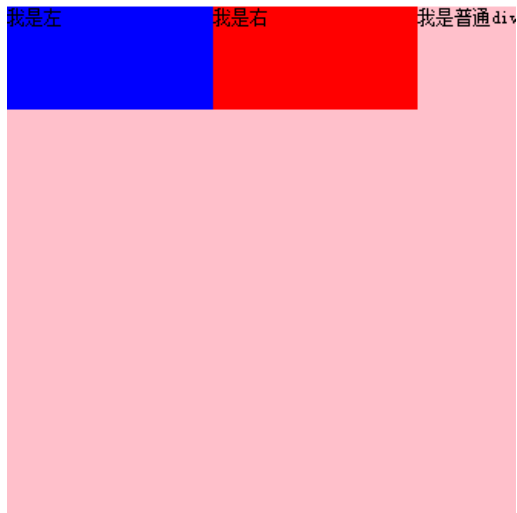
我们如何不让它这么霸道,不独占一行呢?

这个时候,要用到浮动的概念,浮动可以解决这个问题

演示浮动布局

```
#rside{
  width: 200px;
  height: 100px;
  background: red;
  float:left;
}
#lside{
  width: 200px;
  height: 100px;
```

```
background: blue;
float: left;
}
```



浮动布局我们学完之后,对于网页布局,我们就已经入门一小半了

加上一个普通div观察效果

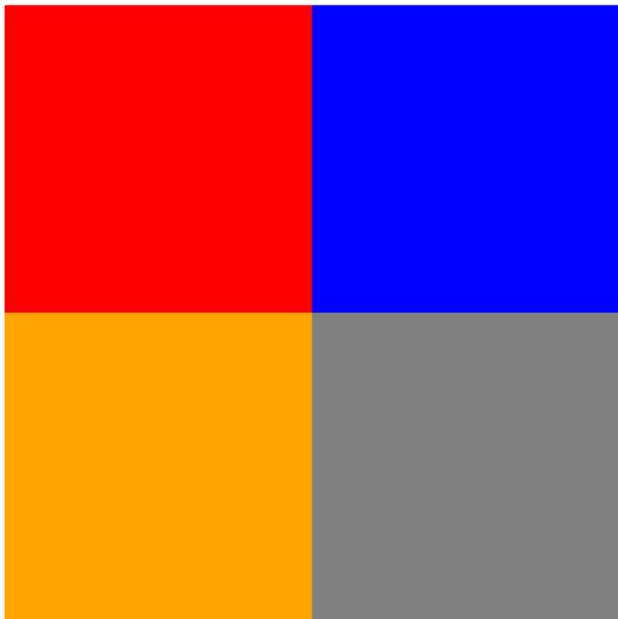
```
<body>
  <div id="right">我是右</div>
  <div id="left">我是左</div>
  <div id="comm">我是普通div</div>
</body>
```

浮动div把普通div给覆盖住了

那如何解决这个问题呢?

课后作业:

完成一个田字格的布局



## 第 7 课 清除浮动

```
clear: left;
```

clear:right;  
clear:both;

为什么要清除浮动？

浮动div会把后面的普通div盖住

如果普通元素中有文字等,也会被覆盖住,

当普通div遇到浮动div时,它不想有div浮动在它的头上

那就需要清除浮动

### 1.clear:left

左浮元素无法覆盖它,那右浮元素可以覆盖吗？

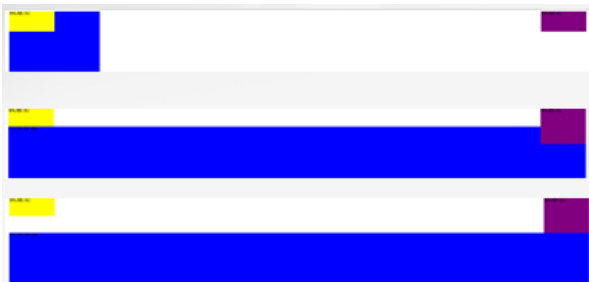
清除左浮动,

清除右浮动

清除 2 侧浮动

```
#lside{
  width: 200px;
  height: 100px;
  background: blue;
  float:left;
}
#rside{
  width: 200px;
  height: 300px;
  background: red;
  float:right;
}
#comm {
  height: 500px;
  background: pink;
  /* clear:left; */
  /* clear:right; */
  clear:both;
}

<div id="lside">我是左</div>
<div id="rside">我是右</div>
<div id="comm">我是普通div</div>
```



课后作业：

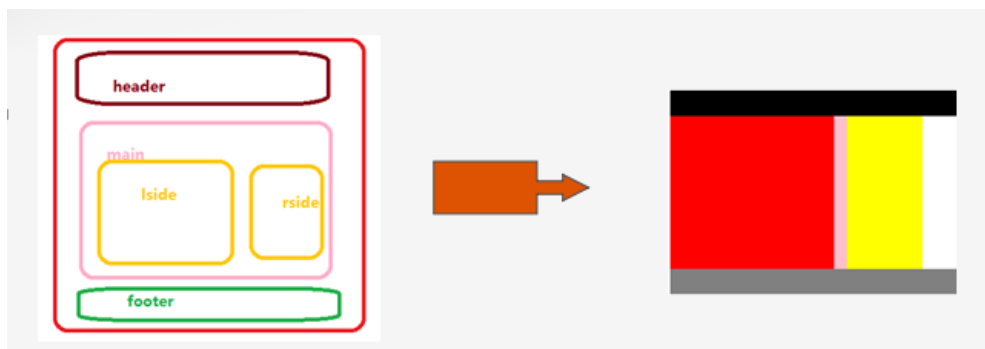
自行设置观察如果不清除浮动,也不设置浮动元素的父元素的高,  
那么父元素的高是多少？

## 第 8 课 实战首页浮动布局

有了刚才几节课的知识,我们现在来做一下首页的浮动布局

完成首页布局,含三行,且中间行为左右块





上节课的课后问题：

自行设置观察如果不清除浮动,也不设置浮动元素的父元素的高,那么父元素的高是多少?

去掉父元素的高

子元素浮动,无法撑起来父元素的高

父元素又没有设置自己的宽高,那么父元素高为零.

父元素的高度为0,则下面的普通元素就会占据原来父元素的位置

如何清除浮动呢?

在main中加一个清除浮动的div

```
#clear{
clear:both;
/*background:red;*/
height:10px;(->0px);
}

<div id="header"></div>
<div id="main">
  <div id="lside"></div>
  <div id="rside"></div>
  <div id="clear"></div>
</div>
<div id="footer"></div>
```

当然还有其它办法能清除浮动,后面会学到其他的办法

课后作业：

完成首页布局

## 第 9 课 新手易犯的错误

同学们刚学习的时候,容易犯的错误

### 1.不加

dtd不能乱用,它代表着浏览器到底用什么标准来解析它

乱用,或者不加dtd,会导致低版本IE,解析效果不一样

html5

ie9以上兼容非常好,且直接感叹号会自动补全

html5,也会自动声明doctype;

### 2.id为数字

id="1"[2,3,4...]

但是:

id是不能为数字

不能以数字开头

php,js,java,c等,它的变量名称都是不能用数字开头的

```
<div id="1"></div>
<div id="a1"></div>
```

### 3.编码不一致

我们提供的sublime编辑器,这个问题不容易出现

如果使用记事本或者editplus等编辑的,可能会出现这个问题

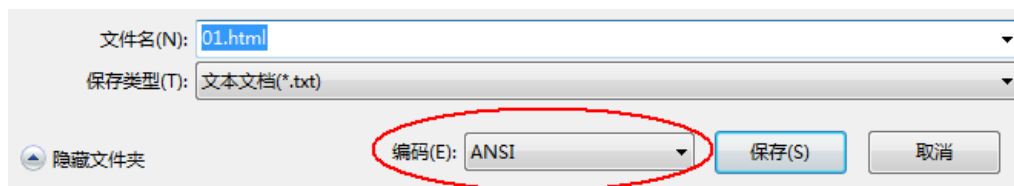
文件存储在我们的磁盘上,本身是有个编码的

文件保存编码要和html页面的charset编码一致

```
<meta charset="UTF-8">
```

用记事本保存的编码和head里面声明的字符集不一致

中文下就会乱码



ANSI在windows下指的是gbk编码的意思

对于计算机而言是不存在乱码的,只不过它按照你要求的解析,解析出来你不认识罢了;

如何才能不乱码:

实际编码和输出的编码保持一致,就不会乱码

用什么编码无所谓,只是要求声明和输出的编码保持一致

我们的sublime默认不支持gbk,都是utf8的,以后大家都使用utf8,这个国际化趋势

课后作业:

把这 3 个错误各犯一遍,并观看错误后果

## 第 10 课 盒模型

我们之前学过div,那如何更好的控制div

此时,我们需要对div的盒模型做一定的了解

什么是盒模型呢?

其实就是把div当成一个块,一个盒子来看

既然是一个盒子,那么生活中的盒子与盒子与div这个盒子,完全能够对应上



以盒子为例,

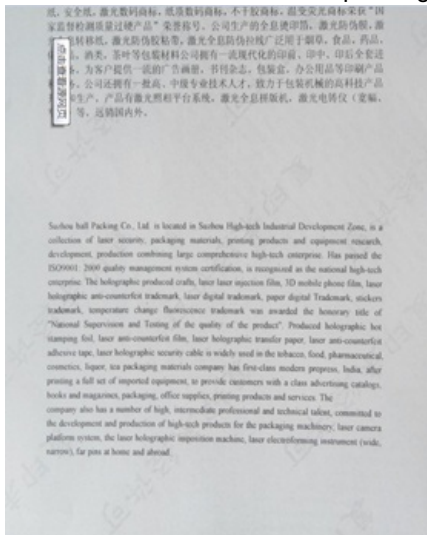
盒子与盒子之间的距离 margin - 外边距

盒子的边框 border - 边框

盒子与物品的距离 padding - 内边距

如一个A4纸,文字从左上角开始写,直接贴在div的边缘处开始写,也不好看

文字 到 纸张 的边处的距离就是padding



默认情况下,所有的 div 都挤在一起,没有美感,

文字也直接贴在 div 的边缘,且没有边框

这些,要靠盒模型来解决.

课后作业:

用纸片模拟盒子,并写出自己的理解.

## 第 11 课 margin 设置

学会设置 4 个方向上的 margin 值

margin-top/right/bottom/left 设置 4 个方向的 margin

margin的设置,就是外边据的设置,盒子到盒子之间的距离

```
div#cont>(div#test1+div#test2+div#test3)
#cont {width:1000px; height:600px;border:1px solid red}
```

```
#test1,#test2,#test3{width:200px;height:200px;
border:1px solid blue;float:left;}
```

设置 test1 margin 值,再设置 4 个方向 margin不同

```
margin:10px;
margin:10px 20px 30px 40px;

margin:10px 20px;
margin:10px 20px 30px;
```

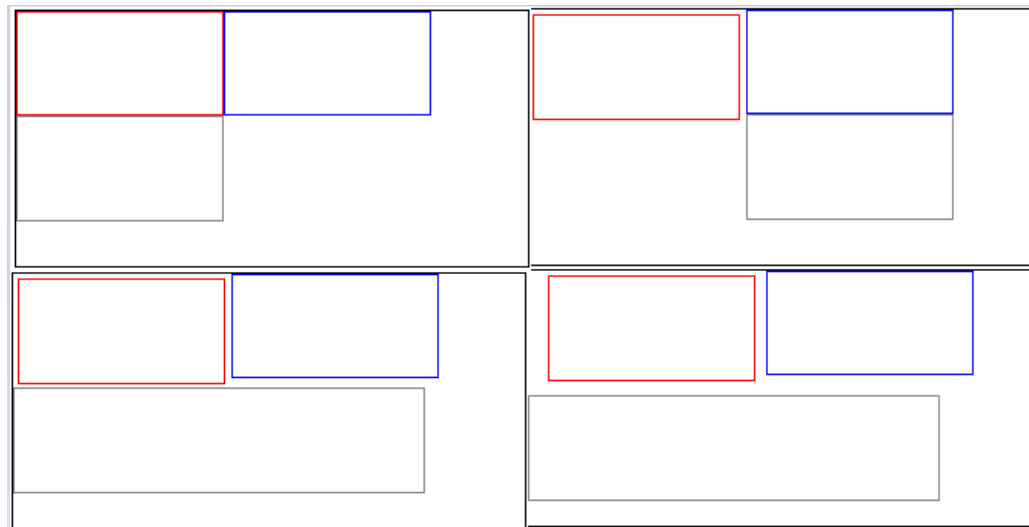
分别设置 margin-top/right/bottom/left 边的值

```
margin-top:10px;
margin-right:20px;
margin-bottom:30px;
margin-left:40px;
```

总结规律:

4 个方向的 margin 分配规律:

从上边开始,顺时针分配,没分配到的,取对边的值



课后作业:

结 4 个方向上设置 margin 的规律

通过 margin 设置,完成田字格中 4 个小单元的间隔.

heade 区域加 margin-bottom:20px;

## 第 12 课 利用 margin 实现水平居中

网页是居中显示的,如何使网页居中显示?

利用margin的知识,可以实现水平居中;

### 1.计算屏幕宽多少

我们的main这个div宽多少,利用margin-left可以实现居中;

首页mian区域居中

```
margin-left:90px;
```

### 2.每个人的屏幕大小不一定是一样的

如何使我们的网页在不论多宽的屏幕上,都居中显示;

```
margin:0px auto;
```

**auto**有自动的意思

浏览器计算了外边距,然后根据这个外边据自动调整水平方向的margin  
最终居中显示(**auto**只能在水平方向使用)

### 3.首页竖直方向居中

```
margin:auto auto;
```

为什么不能竖直居中呢?

因为网页排版参考打字机的排版

打字机纸张的宽度是固定的,但是长度不好说,内容越多,网页越长

课后作业:

利用 **margin auto** 完成首页的居中

## 第 13 课 盒模型之 border

**border** 就是边框

盒子的四条边也可以分别设置不同的颜色等等

边框的设置,比**margin**要稍微复杂一些,需要设置三个参数

**border** 的 3 要素 宽度,形状,颜色,

```
#test1 {  
  width: 400px;  
  height: 200px;  
  background: blue;  
  border: 10px solid red;  
}
```

### 1.观察,加不加边框,**div**的大小变化

盒子的宽高确定后,加上边框,盒子的存放内容没有变化

整个盒子变大了,只是将边框往外挤了;

### 2.盒子**border**三要素:宽,形状,颜色

用**border**分别讲解三要素

```
border:20px solid red;
```

1)宽 : **border-top**[**bottom**,**left**,**right**]-width

2)形状 : **border-style**

**solid** 实线

**dashed** 虚线

**dotted** 点线

**inset** 3D立体

那这么多的参数,我们需要我记下来吗?

我们不需要一一背下来,知识掌握它的大纲,细节我们可以通过查手册

css3手册 -> 边框 -> **border-style**

3)颜色 : **border-color**

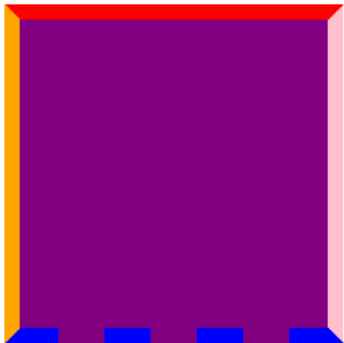
### 3.分别设置4条边的值

```
#test1 {
```

```

width:200px;
height: 200px;
background: purple;
border-top:10px solid red;
border-bottom:10px dashed blue;
border-right:10px solid pink;
border-left:10px solid orange;
}

```



课后作业:

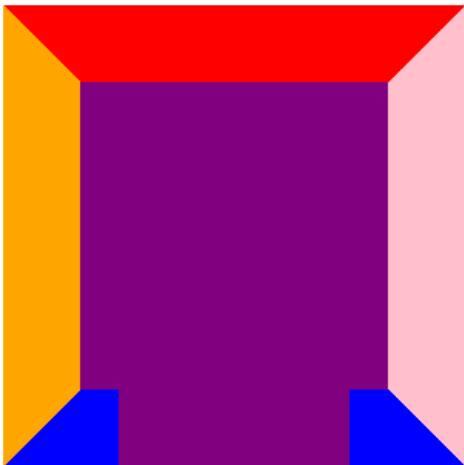
练习 4 条边的设置,  
用 css 画 三角形

## 第 14 课 css 画 3 角形

div是横平竖直的矩形块

怎么画三角形呢?

将上节课的图形,border 设置宽度 50px;



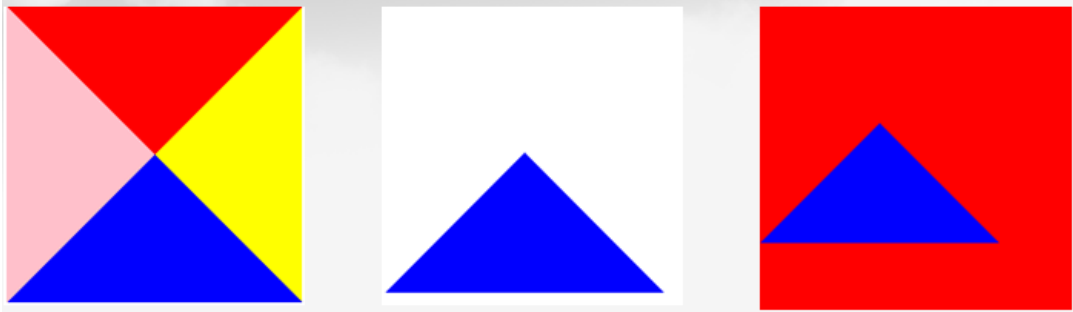
border-width 越大,一个边越接近于三角形

```

#test1 {
width:0px;
height: 0px;
background: purple;
border-top:50px solid red;
border-bottom:50px dashed blue;
border-right:50px solid pink;
border-left:50px solid orange;
}

```

当有背景色的时候怎么办?(transparent -- 透明)



课后作业：

如果有底色,则白色显现,自行搜索,实现边的透明

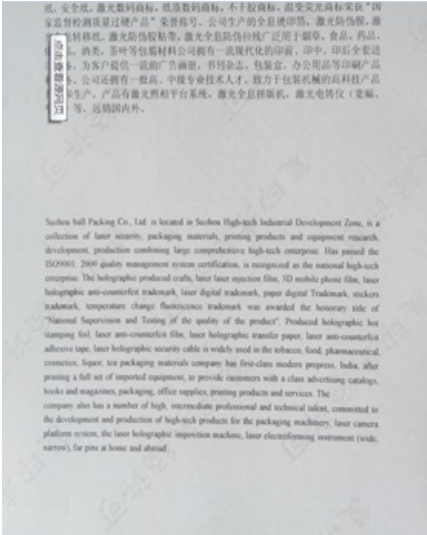
用 css 画圣诞树

画出钝角三角形,和锐角三角形

## 第 15 课 padding 详解

padding内边距,就是盒子的边框往内,到盒子内部文字的距离

1 个盒子,不会放的满满的,比如 A4 纸的打印



### 1.盒子定义宽高,观察写内容的地方

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    div {
      width: 100px;
      height: 110px;
      background: pink;
      /* padding: 20px; */
    }
  </style>
</head>
<body>
  <div>
    aaaaaaaaaa<br >
    aaaaaaaaaa<br >
    aaaaaaaaaa<br >
    aaaaaaaaaa<br >
    aaaaaaaaaa<br >
```

```
aaaaaaa<br>  
</div>  
</body>  
</html>
```

## 2.给div加上padding,观察内容是否会被挤出来

不会的,padding是往外挤的

也就是,一旦我们定义了宽高,写文字的地方是不会变的

再加上border,看效果

```
border:10px solid red;
```

## 3.设置 4 个方向的 padding

跟margin是一样的

```
padding: 10px 20px 30px 40px;
```

## 4.自行探索

如何分别设置 4 个方向的 padding

课后作业:

为 1 个 div 设置 border,padding 及背景色,思考背景将会铺到哪些范围?

# 第 16 课 padding 与背景范围

让学员知道背景会铺到 border 边缘,而不是 padding 边缘

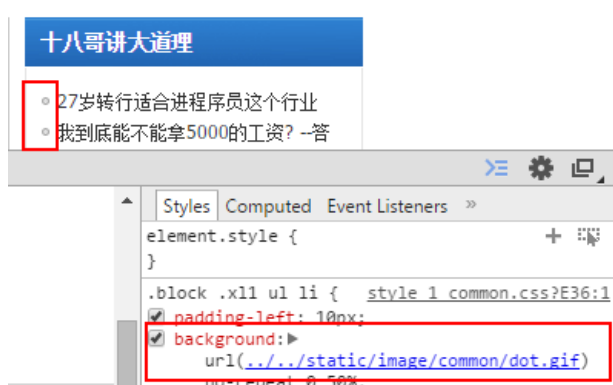
## 1.给上节课的盒子去掉padding再加上padding

分别观察背景的平铺范围

```
div {  
  width: 100px;  
  height: 110px;  
  background: pink;  
  padding: 20px;  
  border: 20px solid blue;  
}
```

背景颜色平铺在哪里?

border的边缘,只不过边框色会覆盖在背景色之上



如图,前面的点就是一个背景图标,背景图没有被文字覆盖住  
因为padding将文字往右侧以挤,背景图标就露出来了

问:

能否将背景平铺在盒子的内容区,就是宽高区域,而不是铺在border边缘?



我们学习的html5支持,指定盒子的背景可以平铺的范围,可以设置平铺到内容区域

更改背景颜色平铺的范围

css3新增属性

background-clip 指定对象的背景图像向外裁剪的区域

border-box(默认) 背景被裁剪到边框盒

padding-box 背景被裁剪到内边距框

content-box 背景被裁剪到内容框



课后作业:

完成左边是背景图片,图片右边是文字这种效果

- 布尔教育的PHP培训有什么特点
- 一个班多少学员?
- 包就业是不是骗人的?
- 我们的包就业为什么是可靠的?
- 

自行搜索:在 css3的新用法中,用哪一个属性,可以让背景不要铺到 padding下方,即只铺在内容区域

## 第 17 课 盒子模型总结

---

观察

一个盒子,在浏览器上,实际占了多大的空间?

画 div,只有宽高,背景,总结面积

设置 padding,观察变化

设置 border,再观察

设置 margin,再次观察

问:能写字的部分是多大?

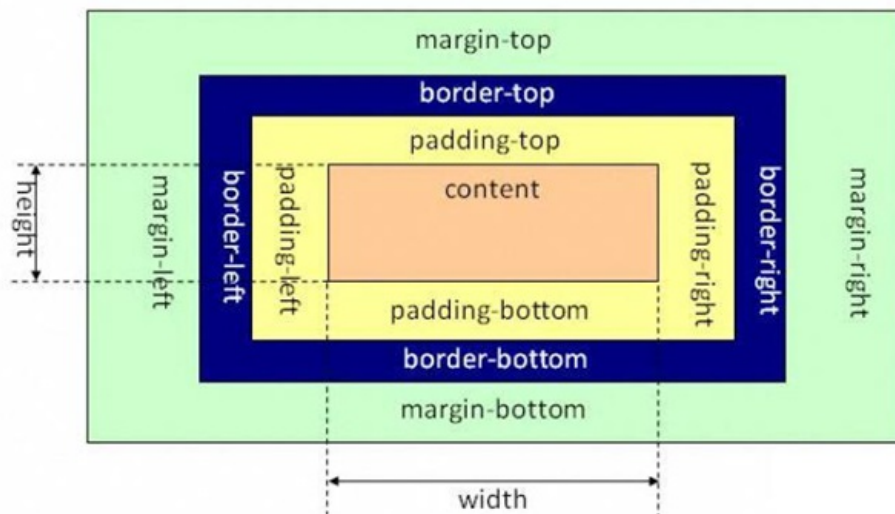
能铺背景色的部分是多大?

实际占地面积多大?

总结:

一旦定义了宽高,实际写文字的地方就固定了,加上padding,border,margin,盒子的实际占地面积变大

## ■ 标准盒子模型



默认盒子模型是我们刚才所讲的,但是...  
凡事就怕"但是"

课后作业:

画图总结盒子模型的 3 要素,并发布到论坛

## 第 18 课 盒子模型再讨论

接着"但是"...

### 1.对盒子模型要进行一个讨论

加上padding之后,按理说,应该是内容被挤出来了

1)w3c的盒子模型标准

padding是往外铺,写文字的大小是固定的

2)ie的旧式的盒子模型详解:

padding和border不会外铺,而是占据内部宽高的位置,往里挤,画图描述

### 2.现在网页设计的趋势:

响应式设计:随着屏幕的大小,可以自适应;

自适应就不能再给定一个准确的宽高.精确到多少像素

而是给定一个比例

t1 {width:50%}

t2 {width:50%}

t1和t2 一左一右,那么不管屏幕多宽,t1和t2都是等宽的

### 3.等比例分div,w3c的盒子模型将要面临挑战

比如:

一个父div,有两个子div,等宽

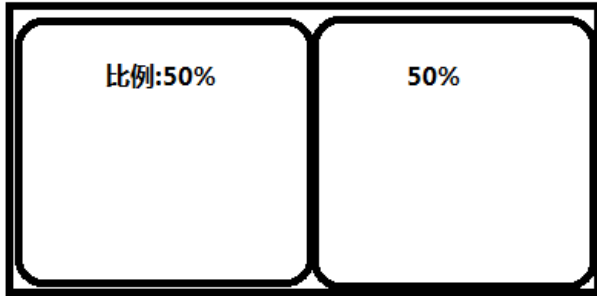
各占(width:50%;).一左一右漂浮,正好将父div全部占据

1)当我们给两个子div加border和padding的时候,会发生什么?

响应式

margin:10px;

border:10px solid blue;



右面的div会被挤下去

2)如果你不想右侧的div被挤下去,如何做?

只能将子div的宽,减少

3)那么减少多少?

少20px,问题是你用比例来写的,难道写50%-20px?

所以当面对形形色色的大小屏幕的时候,你想自适应,用比例,又想加上内边距和边框的,那页面就乱了

4.老式ie浏览器的盒子,占据了优势足够的优势

加上border和padding之后

不是往外挤,而是往内挤,盒子的总的占地面积不变,还是50%

5.所以,w3c意识到,ie的盒子模型是正确的

虽然ie的很多规则不标准,但是ie总算也对了一回

现在,w3c最新的html5的标准里面也做了一个让步

也可以使用老式ie的盒子模型

在响应式布局里面,少不了ie的这种盒子模型

6.那如何使用ie的这种盒子模型呢?

所谓ie的盒子模型,很简单,总的宽高一旦确定,就不会再变化了

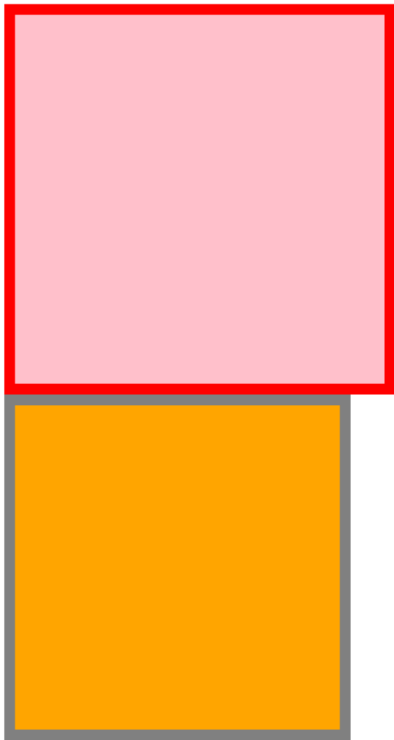
如果有 边框,内边距 都往里挤

padding和border被包含在定义的width和height之内(老式IE盒子)

ie的盒子模型和w3c的盒子模型对比

```
#t1 {
  width: 300px;
  height: 300px;
  border:10px solid red;
  padding: 10px;
  background: pink;
}

#t2 {
  width: 300px;
  height: 300px;
  border: 10px solid gray;
  background: orange;
  padding: 10px;
  box-sizing: border-box;
}
```



### css3新属性

box-sizing 设置或检索对象的盒模型组成模式。

content-box (内容盒)默认值w3c的盒子:盒子的尺寸,宽高是写内容的地方

border-box (边框盒):盒子的尺寸,宽高是包含边框在内的,那写内容的地方变小

在现在的响应式布局上,就需要用 边框盒

课后作业:

搜索居除了 border-box,content-box 外,还有哪种盒子模型

## 第 19 课 内联(行内)元素

---

### 1.内联元素(又叫行内元素)

其实行内元素更能确切描述它的意义

之前我们学习盒模型,一直用的都是div,也没使用其它的标签

div有一个特点:它要独占一行

div为什么要独占一行?

因为div这种元素叫块状元素

### 2.什么是块状元素?

需要跟其它的元素,如行内元素对比,才能更好的理解块状元素

```
<div id="t1">
  鹅鹅鹅<br>
  曲项向天歌<br>
  白毛浮绿水<br>
  红掌拨清波<br>
</div>
```

### 3.我想将 "绿水" 二字变成绿色

#### 1)将t1 设置为绿色

```
#t1{color:green;}
```

全部的诗都变成了绿色

## 2)给"绿水"外面包上div

```
<div id="t1">
  鹅鹅鹅<br >
  曲项向天歌<br >
  白毛浮<div id="green">绿水</div><br >
  红掌拨清波<br >
</div>

#green{color:green;}
```

div是块状元素,独占一行

我们不能让他换行,就需要用到行内元素 span

标签被用来组合文档中的行内元素,以便通过样式来格式化它们

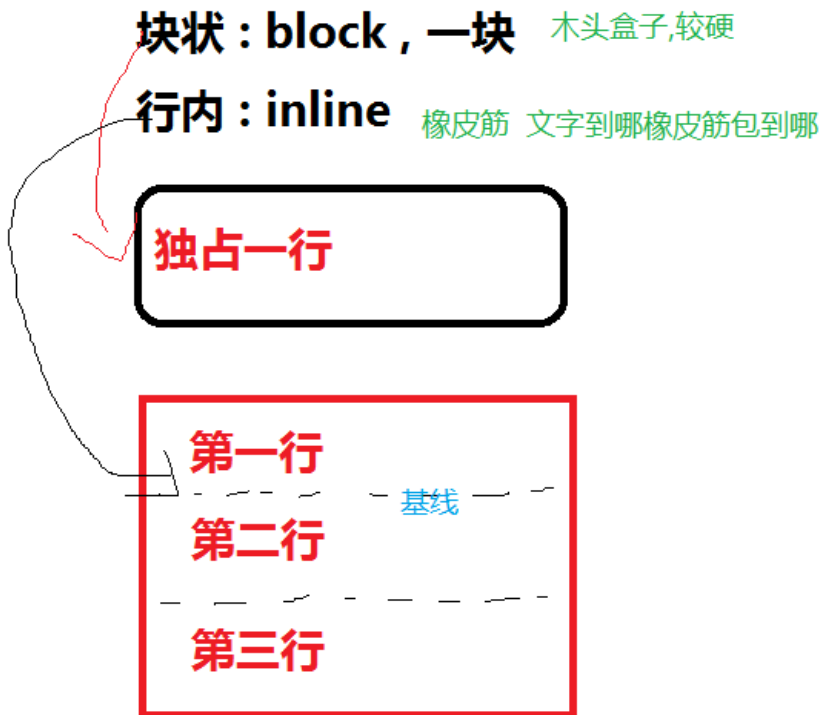
## 3)用 span 包住 "绿水"

span是典型的内联元素

```
<div id="t1">
  鹅鹅鹅<br >
  曲项向天歌<br >
  白毛浮<span id="green">绿水</span><br >
  红掌拨清波<br >
</div>

#green{color:green;}
```

## 4.对比块状与内联的区别(画图)



### 1)块状: block,一块

哪怕它没有这个屏幕宽,这行也被它给占据了

块状元素就像一个盒子,木头盒子,比较硬

### 2)内联: inline

行高：

无法眼睛直观的行高  
浏览器在输出,渲染每一行文字的时候

基线：

内部在维护者一个行的"基线"（Baseline）这种东西；  
就是你在写字的时候,每一行字中间都有一条我们无法看到的'基线'  
行内的影响范围,只在两条基线内,就是行里面,  
span就像橡皮筋一样,将行内的'绿水'给包住,  
但是不会破坏这个行,它只会在这个行内发挥作用,跳不出这个行  
行内元素就像一个橡皮筋,文字到哪,行内元素就包到哪

5.给span加上"竖直"方向的padding和margin

```
#green{
  color:green;
  padding:50px 0px;
  margin: 50px 0px;
}
```

没有作用:跳不出该行

6.给span加上"水平"方向的padding和margin

```
#green{
  color:green;
  padding: 0px 10px ;
  margin: 0px 10px ;
}
```

行就是水平的,在水平方向加margin和padding是可以的

		独占一行	能设宽高	竖直方向的padding和margin
块状元素(block)	是	是	有,可以设置	
行内元素(inline)	否	否	没有,也不可以设置	

div和span是非常典型的块状和内联元素,还有很多其他的内联和块状元素

课后作业：

搜索并总结,哪些元素是块状元素,哪些是内联,发布在论坛

第 20 课 内联与块状的转化

内联和块状各自有它们的特点  
但有时候,它们两者还需要相互转化  
如:  
在做网页,导航标签的时候,两者就需要相互转化

1.鼠标放在导航处,导航背景变色

观看 Blog 的首页导航,为 a 标签设置背景,

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    a:hover {
      background: pink;
    }
  </style>
</head>
</html>
```

```

    }
</style>
</head>
<body>
    <div>
        <a href="#">test1</a>
        <a href="#">test2</a>
    </div>
</body>
</html>

```

但背景较小,设置 width,height,无效.  
 因为a是一个行内元素,文字到哪,它就包到哪

需要将a转成块状元素,才能设置 宽高  
 可以通过display属性使两者相互转换;

```

a {
    display: block;
    width: 50px;
    height: 50px;
}
a:hover {
    background: pink;
}

```

## 2.块状转内联

```

<div>
    aaa <div id="test1">bbb</div> ccc
</div>

```

## 3.display:none;

可令 div 隐藏  
 下面的部分就用到了display+js做到的

[首页公告](#)
[最新帖子](#)
[感谢信](#)

自学php开发的网址导航站  
 (解压密码) rareowhy夫唯vip全套教程+内部  
 rareowhy夫唯vip全套教程+内部资料  
 如何更好的get百度竞价排名后台优化  
 关于缓存方便的知识  
 file\_put\_contents到底能不能自动建立目录  
 十八哥教学中用到的所有手册!  
 <?php \$a=1; \$b=&\$a;  
 学习布尔商城框架时遇到mysql access denie  
 将数组的值进行冒泡排序  
 SEO的优化公式

课后作业:  
 把 a 标签设为块状并设置宽高  
 自行搜索 display 属性有多少种值,并各自有什么特点?

# 第 21 课 css 控制段落

div布局 占位 , css控制显示

画块足够精细,大致布局完成后,再在里面填上相应的文字(画图描述)  
 文字填写进去后,再通过css,把文字的大小,色彩,间距,字体等这些一控制,这个页面就差不多了

## css控制段落和文字

### 1.段落缩进:text-indent

打开163的新闻网页

观察每个段落的首行缩进

作文要求每段首行空两格

查看源码,看是否为'小编'每次都是手敲两个空格

```
<div id="endText" class="end-text">
<p>既不逢五也不逢十，中国今年为啥要大阅兵？是为了炫武力、秀肌肉，还是为了警告、震慑
阅兵的各种猜测和议论就一直不曾停歇。只有搞清楚这些似是而非的问题，才能痛痛快快、明明白白地看懂即将?
<p>
<strong>1、阅兵是为了秀肌肉？</strong>
</p>
<p>阅兵的消息被确认后，国内外的各种解读很快就铺天盖地。其中一股声势较大的论调宣称，中国打破惯例搞误
国博弈中向美国示威。</p>
<p>后来，阅兵准备情况一公布：上万人受阅、动用500多件不同型号装备、近200架多种型号飞机。这些人更坚定
<p>这些人显然没有搞清楚重点。为什么今年要搞阅兵？说白了，很简单，就是纪念抗战胜利70周年，纪念世界反
<p>有人要说了，拉倒吧，纪念形式有那么多，干嘛非要阅兵？答案也很简单，有了阅兵更显隆重啊。再说了，我
<p>更何况，在重大纪念日举办阅兵式也不是中国首开先河，而是国际惯例。俄罗斯每年的“卫国战争胜利日”都
```

是css来控制的

复制 2 段内容及其 div 标签, class 为 test1

```
.test1 {
    text-indent: 20px;
}

<div class="test1"></div>
<div class="test1"></div>
```

知识很多,我们要靠背是背不完的,需要我们掌握一定的规律

总结css的工作:

选中谁{控制什么属性:什么值}

掌握css的设置规律,参数可以参考手册

查看css手册 -> 文本

### 2.文本对齐 text-align (水平对齐方式)

left[默认左边];

right[右边];

center[文本排列到中间]

### 3.文本装饰线text-decoration

underline[文本下划线];

overlinn[文本上的一条线];

line-through[文本穿越线];

text-decoration-style: css3新属性,各浏览器还不兼容

一些新的css3属性,各浏览器的私有试验性质的属性,会在前面加上:

-moz , -ms , -webkit

```
-moz代表firefox浏览器私有属性
-ms代表IE浏览器私有属性
-webkit代表chrome、safari私有属性
```

### 4.文字阴影text-shadow

以往的阴影,需设计人员,用ps做个图片才行

现在用css3即可做到

```
h1 {
```



```
text-shadow:1px 1px 1px red;
}
```

### 课后作业:

把上述 4 种 css 属性测试一遍,  
自行搜索,其他控制段落的属性,并总结其功能  
找一个能使文字倒影的属性 (百度图片倒影,就是此效果的文字倒影)  
展示师兄的效果图

## 第 22 课 文字控制

整段文本的控制  
精细到每个文字的效果

```
font-style  正斜
font-weight 粗细
font-size   大小
line-height 行高
font-family 字体
```

### 准备:

2 个 div,div#test1,div#test2,写同样的内容,分别用上述属性测试效果

### 1.font-style 文本字体样式

normal: 指定文本字体样式为正常的字体

italic: 指定文本字体样式为斜体

### 2.font-weight 字体粗细

bold

normal

字体有更细吗?thin

中文的方块字有好几千,做一套中文字体的代价要比一套英文字体的代价要多很多,所以字体里面就是正常和加粗,没有更细或者更粗.  
只有两种粗细,normal正常和bold 加粗

### 3.font-size 字体大小

### 4.line-height 行高

行高变大,两行字之间的距离就变大

就是内部基线与基线的距离

后面我们做导航的时候,经常要把行高放的很大,以达到文字竖直居中的目的

后面讲: 行高可以达到文字的竖直居中的原理

### 5.font-family 字体系列

'黑体','微软雅黑','宋体','新宋体'

["SimHei"黑体],["SimSong"宋体默认]

### 6.css的简化写法

一行集中控制这些属性

font-style font-weight font-size line-height font-family

注意:

参数必须按照如上的排列顺序

且font-size和font-family是不可忽略的,是必须要写

字体大小和行高都是用尺寸表示,必须用斜线隔开

演示用1行达到5行的效果

```
font-style:italic;
font-weight:bold;
font-size:15px;
line-height:40px;
font-family:'微软雅黑';

font: italic bold 15px/40px '微软雅黑';
```

课后作业:

练习文字控制的各属性,  
自行搜索还有哪些没讲到的文字属性,并总结他们的用法

## 第 23 课 字体深入讲解

`font-family` 可以细致控制字体的种类  
再更深入探讨字体的种类

本节课学完,要掌握3点:

- 1:字体大类
- 2:安全字体
- 3:字体顺序

### 1.字体大类

准备:

<http://paper.people.com.cn>

新闻的大标题,和它的内容,字体是不一样的

人民日报图文数据库 (1946-2015)

确定5年内减贫417万人目标  
**甘肃集中资源精准扶贫**

《人民日报》(2015年07月)

本报兰州7月13日电 (记者林治波、曹树林、银燕) “多亏了咱  
甘肃定西市官营镇农民秦永功终于摘掉了“贫困户”的帽子。  
种植技术,也没有其他技能的秦永功主要靠到外地当搬运工养家

中文字体2大类:

无衬线:黑体,雅黑,文泉驿正黑 sans-serif

有衬线:宋体,新宋, serif (像衬衫一样有棱角)

宋体 [有衬线字体 serif]

黑体 [无衬线体 ( sans-serif ) ]

对比 两个p标签,对比 serif 和 sans-serif

```
#p1 {font-family:'宋体','新宋体',serif};
```

```
#p2 {font-family: '微软雅黑','黑体','文泉驿正黑体',sans-serif;}
```

英文下有:

花体(segor script) 很漂亮,像手写的

中文下做花体的代价太高

等宽字体(Monaco,Courier,Monospace)

不等宽字体(KodchiangUPC)



在editplus(工具->参数设置)中可设置字体

我们编程时,如果用不等宽字体,很可能给我们的对齐带来麻烦

因为良好的缩进和对齐,非常的便于我们调试错误,发现错误.

编程时我们用的都是程序员专用的等宽字体.

## 2.安全字体

```
font-family: '微软雅黑';
```

别人网站设置的字体是微软雅黑,我们打开显示也是微软雅黑字体,是因为我们的计算机上有微软雅黑这个字体

如果是一个国外的用户打开之后,他的机器上并没有微软雅黑这种字体,会报错吗?

也就是,我们设置的字体,客户(浏览器)机器上未必有

但是也不会出错,

电脑会尽量在自己本地有的字体中找个相似的字体显示出来.

网页设计者如果用比较稀奇古怪的字体,用户机器上没有,字体效果也显示不出来

所以要注意,字体的常用和不常用,也就是安全字体

什么是安全字体?

安全字体就是几乎大家都有的字体

## 3.字体顺序

win7的系统'微软雅黑'

xp是'黑体'

linux是'文泉驿正黑体'

```
font-family: '静蕾体','微软雅黑','文泉驿正黑体','黑体',sans-serif;
```

字体顺序: 独特字体,安全字体,字体大类;

课后作业:

搜索衬线字体及无衬线字体的区别,

搜索程序员一般用哪些字体来编程

## 第 24 课 css 设置背景

我们之前学习的是用颜色做背景,我们现在用图片做背景

演示

<http://dxq123.sinaapp.com/index.php> 顶部的背景

准备 1 张小图片,及

```
div#test1{width:500px;height:500px;border:1px solid blue;}
```

background 类似 font

background:red; -> background-color:red;

## 1.背景图

`background-image[url(图片地址)]`

在网站里,引用其它资源,图片,css的时候

只有两个写法, / 当前目录 ../上级目录

不能写 `D:/`或者 `E://`,

因为我们的服务器基本上是linux的;没有D,E盘

即使是windows,也不保证一定有D盘,而且还要将图片再放在你写的D盘E盘中才能发挥作用

要注意,url书写的规范性

## 2.background-repeat

背景图像如何铺排填充

div大,背景图小,但是它也会横向和纵向铺满整个背景

css3新属性

**round:** 背景图像自动缩放直到适应且填满整个容器

**space:** 背景图像以相同的间距平铺且填满整个容器或某个方向

## 3.background-attachment

背景图像是随对象内容滚动还是固定的

**fixed:** 背景图像相对于窗体固定。

**scroll:** 背景图像相对于元素固定 (默认)

```
body {  
    background-image: url(1.png);  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
}
```

## 4.background-position

背景图片的位置默认是在最左上侧

更改背景图片的位置

```
background-position:right top;
```

## 5.使用一行达到效果 (类似font)

```
background-image: url(1.png);  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position:right top;  
  
background:gray url(1.png) no-repeat fixed right top;  
  
极度简写:  
background:gray;
```

课后作业:

练习背景的各种属性

# 第 25 课 大图片背景

大图片做背景的效果:

163邮箱的登录

<http://mail.163.com/>

背景图地址

[http://mimg.127.net/index/163/img/2013/bg\\_v3.png](http://mimg.127.net/index/163/img/2013/bg_v3.png)

这个页面多处的背景,用的是同一副图片

如何做到,多个div,多个元素,公用一个背景图呢?这样做有什么好处呢?

## 1.大图做多个背景的实现原理

position.doc

背景图和div是如何对齐的?

在计算机的世界里,它有一个数轴,这个数轴的原点在左上角



用background-position 实现小图背景,音乐符号

```
div {  
  width: 30px;  
  height: 30px;  
  border: 1px solid red;  
  background-image: url(./bg.png);  
  background-position: -158px -455px;  
}
```

## 2.大图做背景的好处

观察

<http://news.163.com/>

打开一个网页,可以看到有大量的请求

有多少次请求,就等于服务器连接了多少次

对于大网站而言,每天有上亿的访问量

首页每天上亿次的访问量\*页面的请求次数==很大的请求次数

所以对于大型网站而言,它们要想尽一切办法,来减少网页的请求

一个页面多一个请求,那上亿的访问量,就多了上亿的请求

少一个请求,就能节省很大一部分的资源

跟大图作背景的关系:

假设一个页面有5个div,这5个div各有各的背景图

那我们请求这个页面,就连带着需要请求5次这个背景图

用大图作背景图

只请求一次这个图片,这个页面的好多地方的背景图就都有了.节省了大量的服务器请求次数

课后作业:

## 第 26 课 css 选择器

什么叫选择器？

一个页面,可能div10多个,p标签20多个,a标签更多

这些标签,我想控制第一个,第三个div

这个或者那个标签等

如何精确的控制到某一个或控制哪几个元素？

这个时候需要我们用css先选中它

css的工作:

css先选中元素{某个属性:某个值};

如何选中某个元素,或者某几个元素,这就需要选择器的相关知识

准备:div#test1+div.test2\*3+div>p

```
<div id="test1">id test1</div>
<div class="test2">class test2</div>
<div class="test2">class test2</div>
<div class="test2">class test2</div>
<div>
  <p>div下的p标签</p>
</div>
```

### 1.通过id选中元素

```
#id名
```

特点: id选择器前面加#,浏览器就理解为这是个id选择器,谁的id是这个id名,我就选中谁

```
#test1 {
  width: 300px;
  height: 200px;
  border: 1px solid red;
}
```

### 2.类选中某个或某几个元素

```
.test2 {
  width: 300px;
  height: 200px;
  border: 1px solid blue;
}
```

### 3.标签选择器

选择所有div标签,p{},h1{},...

只要是合法的标签都可以用标签选择器

```
div {
  background: blue;
}
```

给body下加上一个p标签

```
p {
  border:1px solid red;
}
```

```
<p>我是body下的p标签</p>
```

#### 4.层级选择器

选中div下的 p 标签

```
div p {  
    background: green;  
}
```

将p标签用span标签包住

```
<div>  
    <span>  
        <p>div下的p标签</p>  
    </span>  
</div>  
  
div p {  
    background: green;  
}
```

#### div p

准确来说,是座位div下的后代p标签

不一定非要是div的二字级别的p标签

那如果只选中div下儿子级别的p可以不可,可以的

css3新增的选择器,就可以精确控制元素的儿子级别的标签

我们在第48课,会专门探讨css3新增的这些更加强大灵活的选择器

这4种,id,类,标签,派生 选择器 用的最为频繁

#### 5.群组选择器

它其实算不是一种选择器,只是一种选择器的简化写法

```
#test1,.test2 {  
    width:300px;  
    height:200px;  
}
```

以上几种选择器,使用频率高达90%

当标签选择器,类选择器,id选择器同时命中一个元素的时候,这三个选择器对这个元素设置有不同的属性

那哪个发挥作用呢?

这个问题称之为,css的优先级问题

课后作业:

总结你所学到的 css 选择器,

搜索学习是否还有其他选择器及其用法

## 第 27 课 css 优先级

准备: div>p#test1.test2 (id和类同时命中这个div)

给p标签加上问题: 我是div下的p标签,看看我的颜色

#### 1.标签选择器

```
p {  
    border: 1px solid blue;  
    color: red;  
}
```

---

## 2.类选择器

```
.test1 {  
    color : orange;  
}
```

观察文字是什么颜色

## 3.id选择器

```
#test1 {  
    color: purple;  
}
```

当多个选择器同时命中一个元素的时候,到底以哪个选择器为准呢?

那就看谁的优先级更高

## 4.分析谁的优先级更高

**p**标签在一个页面中肯定有多个  
标签和类谁命中的元素更多? 类

**id**是唯一的,它只命中一个元素

标签: 极多

类: 多

**id**唯一: 最少

结合刚才的效果,总结选择器优先级的特点:

选择器命中的越精准,越少,优先级越高

## 5.具体div 下的 id

```
div #test1 {color: pink;}
```

优先级比**id**更高,因为更精准的描述了某个**div**下的**id**为**test1**的元素;找的范围缩小了,那找的就更准确了

总结:越具体,优先级越高

提醒:不必死记,通过 **firebug** -> 样式 -> 可查看元素具体使用的属性

记住大的原则,具体调试的时候,如果拿不准,可以通过控制台查看它实际的表现

课后作业:

练习本节课的事例

总结你所学到的 **css** 选择器,

搜索学习是否还有其他选择器及其用法

---

# 第 28 课 **css** 引入方式

之前我们引入**css**都是在**head**标签中写一个**style**标签

引入**css**,还有没有其它引入**css**的办法?

有

## 1.使用HTML标签的STYLE属性

```
<style>  
    div {  
        width: 200px;  
        height: 200px;  
        background: red;  
    }  
</style>
```



## 2.使用 `link` 标签, 引入外部CSS文件

如果我们的css越写越多,整个页面显得非常乱

我们可以将css给单独拿出来,放在一个专门文件里

28.css

`link`标签通常放置在一个网页的头部标签`head`标签内,

可以用于链接外部css文件

```
/*stylesheet 样式表*/  
<link rel="stylesheet" href="./28.css">
```

注意:在css文件中,不需要写`style`标签,直接写css代码即可

在实际开发中以上两种引入css的方式,是最为频繁使用的

## 3.内联样式,直接写在标签的`style`属性里

临时需要测试,也没必要去写了,直接写在标签里面就可以了

```
<div style="width:200px;height:200px;background:red;"></div>
```

## 4.使用`@import`导入CSS文件

在一个css文件里,还可以引入其它的css

28-1.css

```
@import url(6-1.css);  
div {  
    width: 200px;  
    height: 200px;  
    background: green;  
}
```

28-2.css

```
h1 {  
    font-style:italic;  
    font-weight:normal;  
}
```

注意:

在css文件中,引入其它的css,一定要放在第一行

不然会没有效果

课后作业:

写 1 个.html 文件,用 4 种方式引入一段 css 代码

并自行搜索各种方式的特点

# 第 29 课 css 初始化

为什么要将css初始化?

写一个div

我们看这个div离我们的浏览器有一定的距离

它没有死死的贴着浏览器的边

是因为浏览器的一些行为,让你的div距离浏览器的边缘处,有了一定的距离,也就是说浏览器可能有默认的padding或者margin

写3个p标签,并加上border

3个p标签并不是贴在一起的,而是有一定的距离

为什么?

```
<p>p1</p>
<p>p2</p>
<p>p3</p>
```

```
p {
  border:1px solid red;
}
```

因为p标签默认是有margin值的

有就有吧,但是其他浏览器,火狐,chrome,ie

它们对某些标签都有一定的初始值,但是这些初始值有可能不一样

如果初始值不一样,有可能导致,在不同浏览器访问下,有可能页面是变形的

有什么办法能取消初始值的干扰?

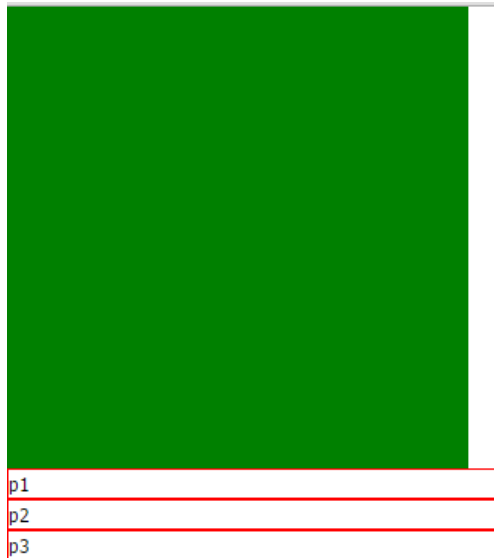
将各种标签的初始值都变为0,大家都在同一起跑线上,这个过程称为css的初始化,它非常的有实际意义

在多浏览器兼容的时候,首先初始化就可以避免各浏览器初始值不一样,所带来的干扰

taobao.com 的css初始化代码 index-min.css

```
blockquote,body,button,dd,dl,dt,fieldset,form,h1,h2,h3,h4,h5,h6,hr,input,legend
,li,ol,p,pre,td,textarea,th,ul{margin:0;padding:0}
body,button,input,select,
textarea{font:12px/1.5 tahoma,arial,'Hiragino Sans GB',\5b8b\4f53,sans-serif}
h1,h2,h3,h4,h5,h6{font-size:100%}address,cite,dfn,em,var{font-style:normal}
code,kbd,pre,samp{font-family:courier new,courier,monospace}
small{font-size:12px}ol,ul{list-style:none}a{text-decoration:none}
a:hover{text-decoration:underline}sup{vertical-align:text-top}
sub{vertical-align:text-bottom}legend{color:#000}fieldset,img{border:0}
button,input,select,textarea{font-size:100%}
table{border-collapse:collapse;border-spacing:0}
```

我们引入淘宝的css代码,再看我们的页面



课后作业:

搜索查找淘宝,腾讯,网易,新浪的 css 初始化代码,并做比较

## 第 30 课 HTML学习思维导图

回顾我们之前的学习: div布局 css控制显示

如果需要填充文字,修饰某几个文字,我们可以用span把他包起来,再进行控制

从目前所学的知识点来说,你已经足够做一个复杂的网站了

只需要用div任意的画块,画到最细,再往里面塞文字,用span包,在css控制它,也就是说如果网站原意,只用div+span两个标签,也能布出一个漂亮的网页

但是只用div+span还不够好

div division 区块,它没什么语言上的意义

什么叫有语言上的意义?

就是一个机器,一个蜘蛛过来了,可以通过你的标签,分析出你实际放置的内容

h1 大标题

p 一段话

table 表格

这些标签称为语义化的标签

相对于div,我们能否只用div+span,不用这些语义化标签,也能做一个漂亮的网页,只是机器过来分析你的网页,没有着重点,不知道哪个是大标题,哪个是文章,哪个是表格

所以,我们想要页面,对于机器而言,对于蜘蛛而言,更优化的话

我们可以再学一些语义化标签

详解 -> 思维导图

课后作业:

沿思维导图,总结笔记.

## 第 31 课 H 标签与 p 标签

开始学习有语义标签

1.h标签详解

打开 word,引入大标题,段落概念

h1-h6,表示各级标题,而非正文

准备: h\$\*7>{标题\$}

用来定义标题,越往下字越小

标题只有h1-h6

没有7号标题,写了是普通文字的大小

2.p标签详解 (paragraph 段落)

h1一般一个网页,只有一个,经常和p标签一起配合使用

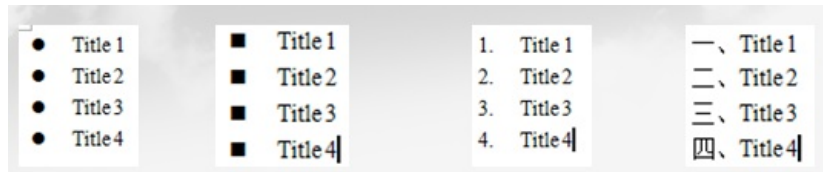
打开163的一篇文章,看源码,h1和p

课后作业:

观察 163,腾讯新闻,分析 h,p 一般出现于什么场合?

## 第 32 课 有序列表和无序列表

在word文档中我们学习什么是有序列表,什么是无序列表.



1.

标签定义无序列表

```
<ul>
  <li>title1</li>
  <li>title2</li>
  <li>title3</li>
</ul>
```

2.

标签定义有序列表

```
<ol>
  <li>title1</li>
  <li>title2</li>
  <li>title3</li>
</ol>
```

3.list-style-type: 设置不同的列表样式

给ul或者是ol设置这个属性

disc 默认。标记是实心圆。

circle 标记是空心圆。

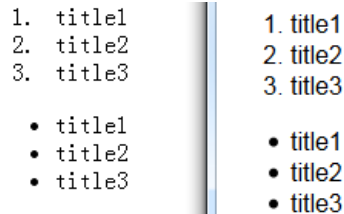
square 标记是实心方块。

decimal 标记是数字。

none 不显示列表样式

4.实际开发中list-style-type 都置为 none,为什么?

因为各浏览器兼容效果不同,火狐和chrome就不同,所以置为none



|           |           |
|-----------|-----------|
| 1. title1 | 1. title1 |
| 2. title2 | 2. title2 |
| 3. title3 | 3. title3 |
| <br>      |           |
| • title1  | • title1  |
| • title2  | • title2  |
| • title3  | • title3  |

这种列表样式在各浏览器的差异,用css就无法改变了

所以在css初始化的时候,已经将这个属性设置为了 none

在实际应用中我们用背景图片来设置样式

#### 十八哥讲大道理

- 从中国好声音谈人生的高开低走与
- 27岁转行适合程序员这个行业
- 我到底能不能拿5000的工资? -答
- 你们当中,必有失败者
- 十八哥-凡人也
- 你应该用3个小时学会编程

分析 [www.zixue.it](http://www.zixue.it) 首页 哪里使用的 ul ,ol

课后作业:

用图片背景实现 li 的小图标效果,来达到各浏览器兼容的目的

## 第 33 课 整齐的表格

表格用来放置格式化的数据:

工资表,课程表,值班表...

div+css画表格,是非常麻烦的

画图分析一个: 个人信息登记表

分析表格,行,单元格

画 table

画 3 行 tr

每 tr 中画 4 个 td  
给表格加上表框 -> td 加上 border  
去掉单元格之间的距离 -> 手册找 -> border-collapse  
注意: 它是 table 的属性

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

表头的声明: th

```
th,td { border:1px solid black }
```

如何画下面的表格:

| 时间\日期 | 1  | 2  | 3  | 4 | 5 |
|-------|----|----|----|---|---|
| 上午    | 语文 | 数学 | 英语 |   |   |
|       | 政治 | 地理 | 化学 |   |   |
|       |    |    |    |   |   |
| 下午    |    |    |    |   |   |
|       |    |    |    |   |   |

表格讲解什么是:跨列合并,合并单元格

td 的 css 属性

1. 跨列合并

一行少了一个单元格(横跨两列,水平防线)

colspan 属性规定单元格可横跨的列数

```
<td colspan="2"></td>
```

2. 合并单元格 (竖直方向)

rowspan 属性规定单元格可横跨的行数

课后作业:

| 时间\日期 | 1  | 2  | 3  | 4 | 5 |
|-------|----|----|----|---|---|
| 上午    | 语文 | 地理 | 代数 |   |   |
|       | 语文 | 地理 | 代数 |   |   |
|       |    |    |    |   |   |
| 下午    |    |    |    |   |   |
|       |    |    |    |   |   |

## 第 34 课 HTML5 新增的几个语义标签

header footer nav[导航] article aside[侧边栏] time

不用语义标签,只用 div+span 都可以做网页  
有语义标签能让机器更加识别它,更有意义

1. header, footer, nav[导航]

www.itbool.com 分析 这三个标签

```
<header>我是头部</header>
<nav>我是导航</nav>
<footer>我是版权 &copy; itbool.com</footer>
```

视觉效果和div差不多

区别在于对机器而言,机器过来采集我的网页的时候  
搜索引擎看到比较好理解

## 2.aside,article,time

```
<aside>
    侧边的小块信息
</aside>
<article>
    <h1>我是标题</h1>
    <time>2015年8月8日</time>
    <p>p1</p>
    <p>p2</p>
    <p>p3</p>
</article>
```

html5新增的这几个标签,主要体现的意义是在语义效果上  
使用效果也没有比div多出什么新花样  
还是用来占位的

课后作业:

自行搜索,新增的几个标签用在什么场合

## 第 35 课 img 图片标签

---

目前,我们可以布局,可以写文字,如果要引入图片,该如何做

### 1.img 引入一张图片

准备 div>img

### 2.src属性

src: 规定显示图像的 URL (当前目录,上级目录,其他本地目录,和远程图片)

#### 1)相对URL

指向站点内的文件 当前目录 ./ 或者上级目录 ../

#### 2)绝对URL - 指向其他站点

比如 src="<http://ww3.sinaimg.cn/bmiddle/62f4cb1fgw1evq6gsagsaj20c8085dgo.jpg>"

### 3.alt 属性: 规定图像的替代文本

做过SEO的知道:

alt可用于这个图片的重要搜索说明

搜索引擎越来越智能,不止能根据alt搜,还可以根据文字信息分析图片信息  
甚至可以做图片识别

### 5.title: 鼠标放上去,显示此文本

注意:

从语法的角度上,可以引用其它网站的图片资源,但是对方的网站未必允许我们引用

baidu贴吧的图片是防盗链的

课后作业:

写 html 引用站内图片,站外图标

## 第 36 课 图片是内联还是块状？

准备 `img*3` 三张图片并排显示

1. `img` 设置宽高

2. 图片有空隙

设置 `margin, padding` 为 0 不管用

```
img {  
  width: 100px;  
  height: 100px;  
  padding: 0px;  
  margin: 0px;  
}
```

`img` 是特殊的内联元素, 是内联替换元素

3. 为什么图片可以设置宽高？

图片和文字不同, `h1` 是 16 像素, 写几个字就能计算出这几个字的占地面积;

而图片, 需要浏览器需要等图片加载完毕之后, 再根据图片的实际大小分配它的占地面积, 无法提前算出应该预留的位置

`img` 图片加载完, 实际多大, 替换多大的位置, 所以叫内联替换(置换)元素

对于替换元素, 我们可以设置宽高

4. 如何去掉图片之间的缝隙

转块状 -> 设置 `margin, padding` 为 0 -> `float`

```
img {  
  display: block;  
}
```

建议 `css` 初始化的时候, 直接将 `img` 标签转为块状, 以免受图片的几像素的缝隙的影响

课后作业:

搜索解决除了 `img`, 还有哪些元素是替换元素

参考 <http://www.w3.org/TR/CSS2/visudet.html#inline-replaced-width>

## 第 37 课 超链接标签

谁也离不开的 -> `a` 标签

目前互联网有 400 多亿个网页

如果网页是相互孤立的？

上网的时候, 没有费力的去找网页, 还能一逛就一天 -> 就是超链接在帮忙

各个网页怎么联系起来的 -> 就是通过 `a` 标签, 超连接

`a` 标签作用很大, 但是使用很简单

1. `a` 标签的 `href`

可以连接本地网页

也可以连接互联网上的绝对地址

```
<a href="/1.html">第一个网页</a>  
  
<a href="http://www.zixue.it">学习论坛</a>
```

2. 在新标签中打开连接

`target` 属性

`target="_blank"`

3.title属性 鼠标放上去时,显示的文字

4.href="#" 是什么意思?

公司套模版时可见

没有什么实际意义,就是一个空连接,点击没有效果

课后作业:

搜索解决,a 链接的 href 为#代表什么意思?

## 第 38 课 锚点

---

a连接,可以连接到具体的网页

锚点更精细,可以具体定位到网页的某个位置

效果: <http://www.itbool.com/> -> 常见问题 就是用的锚点

观察地址栏的 #q2 #q3 ...

1.旧式写法

地址栏 38.html#q3

```
<br >
<br >
<br >
<br >
<a name="q3">
<h2>住宿情况</h2>
<br >
<br >
<br >
<h2>学习周期</h2>
```

这样写可以是html5对旧写法的兼容,但是现在不要这样写

2.html5用id当锚点

地址栏: 38.html#q1

```
<br >
<br >
<br >
<br >
<h2 id="q3">住宿情况</h2>
<br >
<br >
<br >
<h2 id="q4">学习周期</h2>
```

只要指定id就可以当锚点用

课后作业:

在 1 个页面内设置 3 个锚点,并创建 3 个链接连向页面的各部分

## 第 39 课 css 伪类

---

我们每天都在时候,当鼠标放在导航处,变色 -> 就是css伪类做的

类 : class

.test{} //选中 class="test"的元素

注意:a标签里面要留#或者其它内容(否则个别伪类不能使用)

```
<a href="adfgs"></a>
```



```
<a href="#"></a>
```

伪类: 选中的元素的某个状态

**a:link** 正常的标签的状态

**a:visited** 选中的是 鼠标点击后的标签的状态

**a:hover** 选中的是 鼠标经过标签时的状态

**a:active** 选中的是 鼠标点击瞬间(还未松鼠标)的状态

注意:

**a:visited**

判断一个页面是否访问过,浏览器会看浏览历史中有没有这个网页;当访问的页面为 **not found** 时,不会加入到浏览历史中

**a:link**可以简写为**a**;

提醒顺序 **lha**,顺序不能乱写,可以不写全,但是顺序不能乱

**a:active**一般不写;

课后作业:

练习4个伪类

## 第 40 课 字符实体

什么是字符实体?

说白了,就是一个字符的外号

在内容中写一个 大于号 <

```
<div>
a
>
</div>
```

能显示出来,但是不建议这样写,因为在html中,<>是用来表示一个标签的开始和结尾的,内容中写>< 可能会干扰它的解析  
可以换一种方式来写 -> 字符实体

|             |             |
|-------------|-------------|
| 空格 &nbsp;   | < 小于号 &lt;  |
| > 大于号 &gt;  | & 和号 &amp;  |
| " 引号 &quot; | ¥ 人民币 &yen; |
| © 版权 &copy; | ® &reg;     |

手册 -> html手册

也可以用实体编号表示

```
¥ 日圆 &yen; &#165;
```

课后作业:

找出并写出 20 个实体

## 第 41 课 首页实战之布局

实战环节

后面我们学习php也是做博客

我们提前为自己的博客提前做一份素材

模仿别人的网站:模仿它的效果而不是代码

比如模仿导航 -> 直接审查元素看就可以了

直接冲着页面的效果,而不要右键查看源代码分析



```

#rside {
    width: 275px;
    height: 500px;
    background: gray;
    float: right;
}

footer {
    height: 85px;
    background: #222;
}

</style>
</head>
<body>
    <header>
        <h1>燕十八的blog</h1>
        <p>成风行,无惧!</p>
    </header>
    <nav>
        <ul>
            <li><a href="">导航1</a></li>
            <li><a href="">导航2</a></li>
            <li><a href="">导航3</a></li>
            <li><a href="">导航4</a></li>
        </ul>
    </nav>
    <div id="main">
        <div id="lside"></div>
        <div id="rside"></div>
    </div>
    <footer>Copyright &copy; 2015 · GeneratePress · WordPress</footer>
</body>
</html>

```



main加上居中显示

注意:在布局时给各个大块都加上背景,方便我们的调试

课后作业:

跟老师完成布局

## 第 42 课 首页实战导航制作

nav ul 1100px \* 60px; 居中显示

```
nav ul li float: left;
nav a { padding:0 20px;}
nav a:hover {background:3F3F3F} 鼠标放上去,问题显示
nav a display:block;
```

参考代码:

```
nav {
  height: 60px;
  background: #222;
}

nav ul {
  width: 1100px;
  height: 60px;
  margin: 0px auto;
}

nav li {
  float: left;
}

#nav1 {
  background: #3f3f3f;
}

nav a {
  color: white;
  display: block;
  width: 74px;
  height: 60px;
  font-size: 15px;
  line-height: 60px;
  padding-left: 20px;
}

nav a:hover {
  background: #3f3f3f;
}
```



课后作业:

完成导航区

## 第 43 课 首页实战之 main 区域(1)

带领学员完成 main 区右侧

注意:近期评论模块,由于使用的第三方插件  
所以跟其他的有所不同,可以忽略.

```
aside {padding:40}
aside h4 {}
aside a,a:visited{}
aside a:hover{color}
aside li{padding-bottom}
aside input表单
```

参考代码:

```
<div id="rside">
  <aside>
    <form action="">
      <input type="text" name="" type="search" />
    </form>
  </aside>
  <aside>
    <h4>分类目录</h4>
    <ul>
      <li><a href="#">我是标题</a>&nbsp;(1)</li>
      <li><a href="#">我是标题</a>&nbsp;(2)</li>
      <li><a href="#">我是标题</a>&nbsp;(3)</li>
      <li><a href="#">我是标题</a>&nbsp;(4)</li>
    </ul>
  </aside>

  <aside>
    <h4>分类目录</h4>
    <ul>
      <li><a href="#">我是标题</a>&nbsp;(1)</li>
      <li><a href="#">我是标题</a>&nbsp;(2)</li>
      <li><a href="#">我是标题</a>&nbsp;(3)</li>
      <li><a href="#">我是标题</a>&nbsp;(4)</li>
    </ul>
  </aside>

  <aside>
    <h4>分类目录</h4>
    <ul>
      <li><a href="#">我是标题</a>&nbsp;(1)</li>
      <li><a href="#">我是标题</a>&nbsp;(2)</li>
      <li><a href="#">我是标题</a>&nbsp;(3)</li>
      <li><a href="#">我是标题</a>&nbsp;(4)</li>
    </ul>
  </aside>
</div>
```

## rside.css

```
aside {
  padding: 40px;
  background: white;
  margin-bottom: 20px;
}

aside h4 {
  font: normal 20px '微软雅黑', '文泉驿正黑体', '黑体', sans-serif;
  margin-bottom: 30px;
}

aside a, a:visited {
  color: #1e73be;
  font: '新宋', '宋体', serif;
}

aside a:hover {
  color: black;
  text-decoration: none;
}

aside li {
  width: 195px;
  height: 21px;
  padding-bottom: 5px;
}
```

```
aside input{
    border:1px solid #ccc;
    width: 163px;
    height: 23px;
    padding: 10px;
}
```



## 分类目录

[我是标题 \(1\)](#)

[我是标题 \(2\)](#)

[我是标题 \(3\)](#)

[我是标题 \(4\)](#)

课后作业：

独立完成 main 区右侧

## 第 44 课 首页实战之 main 区域(2)

带领学员完成 main 区左侧

参考代码：

```
<div id="lside">
  <article>
    <h2><a href="#">午夜的街头</a></h2>
    <div class="arthead">
      <time>2015-01-28</time>
      <span>by 十八</span>
      <span><a href="#">闲谈随笔</a></span>
      <span>
        <a href="#">午后</a>|
        <a href="#">时光</a>
      </span>
      <span><a href="#">9条评论</a></span>
    </div>
    <p>
      有时突然想说我爱你，<br >
      有时突然慌张忧伤。<br >
      <br >
      像迷路的孩子，<br >
      东张西望。<br >
      <br >
      伸出双手想抓住，时光像泥鳅一样。
    </p>
  </article>
</div>
```

lside.css

```
article {
  margin-bottom: 20px;
  padding: 40px;
  background: white;
}

article h2 a, article h2 a:visited {
  color: #1e73be;
  font: 30px/36px '微软雅黑', '文泉驿正黑体', '黑体', sans-serif;
}

article h2 a:hover {
  color: black;
  text-decoration: none;
}

.arthead {
  width: 725px;
  height: 18px;
  margin-top: 6px;
  margin-bottom: 40px;
}

.arthead a, .arthead a:visited {
  color: rgb(102, 102, 102);
}

.arthead a:hover {
  color: #1e73be;
  text-decoration: none;
}

article p {
  font: 15px/23px '新宋体', '宋体', serif;
}
```

## 午夜的街头

2015-01-28 by 十八 闲谈随笔 午后 | 时光 9条评论

有时突然想说我爱你，  
有时突然慌张忧伤。

像迷路的孩子，  
东张西望。

伸出双手想抓住，时光像泥鳅一样。

课后作业：

独立完成 main 区左侧

## 第 45 课 首页实战之分页代码

参考代码

pager 放在 lside 最下面

```
<div id="pager">
```

```
Pages:&nbsp;
<a href="#">1</a>
<a href="#">2</a>
    3
<a href="#">4</a>
<a href="#">5</a>
<a href="#">6</a>
<a href="#">Next</a>
</div>
```

## pager.css

```
#pager {
    width: 805px;
    height: 25px;
    font: 17px/23px '微软雅黑', '黑体', sans-serif;
}

#pager a, #pager a:visited {
    color: rgb(30, 115, 190);
}

#pager a:hover {
    color: black;
}
```

Pages: 1 2 3 4 5 6 Next

课后作业：  
完成分页代码

## 第 46 课 完成头尾部

### header-footer.css

```
header{
    height: 76px;
    width: 1100px;
    padding: 40px;
    margin: 0px auto;
}

header h1 {
    color: rgb(34, 34, 34);
    font: 45px/54px 'Open Sans', '微软雅黑', '文泉驿正黑体', '黑体', sans-serif;
}

header p {
    width: 1100px;
    height: 22px;
    color: rgb(153, 153, 153);
}

footer {
    height: 65px;
    background: #222;
    text-align: center;
    color: white;
    font: 17px/65px '微软雅黑', '黑体', sans-serif;
}
```



课后作业:

完成首页

## 第 47 课 利用行高完成文字居中

---

之前我们用行高完成文字竖直居中,现在讲一下它的原理

回头看导航文字如何竖直居中 -> line-height 去点看还居中否

准备: `div>{中华人民}`


文字水平居中 `text-align`

竖直居中,没有这个属性,我们需要通过 行高 来变通实现

画图 -> 基线

字是放在行中间的

```
div {  
  width: 200px;  
  height: 80px;  
  background: pink;  
  text-align: center;  
  line-height: 80px;  
}
```



课后作业:

理解行高的原理

## 第 48 课 CSS3 的强大选择器

---

html5的正式标准在2014年已经确定.

html5和css3并不是一回事

而css3的最新是在 -> 2013年才到草案 -> 还未出正式

草案到 -> 正式应用还需要几年

所以我们有时候用的css3的新特性会不管用这是正常的

因为这些浏览器的厂商处于变动和实验中.

对于css3不要过度追求特别新的东西,可能过一段时间会发现不管用了

我给大家罗列出来的css的新属性,是各个浏览器兼容非常强的

css3新的选择器

1.关系选择器(层次选择器):

后代元素 `E F` 匹配 作为所有作为 `E`元素 后代的 `F`元素

儿子元素 `par>son` 匹配 `par` 元素的子级 `son` 元素

相邻元素 `E+F` 匹配紧贴着 `E` 元素的 `F` 元素

兄弟元素 `E~F` 匹配 `E` 后面所有同级别 `F` 元素

准备测试代码

```
<div>  
<ul>  
<li>第一层</li>  
<li>  
  <ul>  
    <li>春</li>  
    <li>夏</li>
```

```
        <li>秋</li>
        <li>冬</li>
    </ul>
</li>
</ul>
</div>
<p>p1</p>
<span><p>span下的p</p></span>
<p>p2</p>
```

1)后代元素 **E F 匹配** 作为所有作为 **E**元素 后代的 **F**元素

```
div ul {
    border: 1px solid blue;
}
```

2)儿子元素 **par>son** 匹配 **par** 元素的子级 **son** 元素

```
div>ul {
    border:10px solid yellow;
}
```

3)相邻元素 **E+F** 匹配紧贴着 **E** 元素的 第一个 **F** 元素

```
div+p {
    border: 1px solid red;
}
```

4)兄弟元素 **E~F** 匹配 **E** 后面所有同级别 **F** 元素

```
div~p {
    border: 1px solid orange;
}
```

注意:+ 必须是"大哥+二哥", ~ 还能是"大哥~三弟"、"二哥~四妹"

2.属性选择器(用在表单里比较多)

**E[属性名]** 含有某属性的元素

**E[属性名=值]** 某属性=某值的元素

准备:

```
<form action="">
    <input type="text" name="username">
    <input type="text" name="email">
</form>
```

1)标签选中input

```
input{
    border:10px solid red;
}
```

2)**E[属性名]** 含有某属性的元素

```
input[name] {
    background: red;
}
```

3)**E[属性名=值]** 某属性=某值的元素

```
input[name=username] {  
    background: yellow;  
}
```

jq为了降低学习成本,它的选择器比较像css的属性选择器

## 2.伪类选择器

**E:first-child** 匹配作为第 1 个子元素的元素 E

**E:last-child** 匹配作为最后一个子元素的元素 E

**E:nth-child(N)** 匹配作为第 N 个子元素的元素 E

准备代码

```
<ul>  
    <li>张三</li>  
    <li>李四</li>  
    <li>王五</li>  
    <li>赵六</li>  
</ul>
```

1)**E:first-child** 匹配作为第 1 个子元素的元素 E

```
li:first-child {  
    border: 1px solid blue;  
}
```

2)**E:last-child** 匹配作为最后一个子元素的元素 E

```
li:last-child {  
    border: 1px solid red;  
}
```

3)**E:nth-child(N)** 匹配作为第 N 个子元素的元素 E

```
li:nth-child(3) {  
    background: gray;  
}
```

## 3.伪对象选择器

人为的伪造出来的对象,本身没有这个对象

**E::before** 设置 E 元素前面的内容

**E::after** 设置 E 元素后面的内容

准备代码:

```
<p>我是第一段</p>  
<p>我是第二段</p>  
<p>我是第三段</p>
```

1)想在p标签之前加一个笑脸

但是加在p里面,会干扰文本本身的内容

^\_^我是第1段

1.**E::before** 设置 E 元素前面的内容

在所有的p标签之内,在p标签的文本之前

```
p {  
    border: 1px solid blue;  
}
```

```
p::before {
  content: '^_^';
}
```

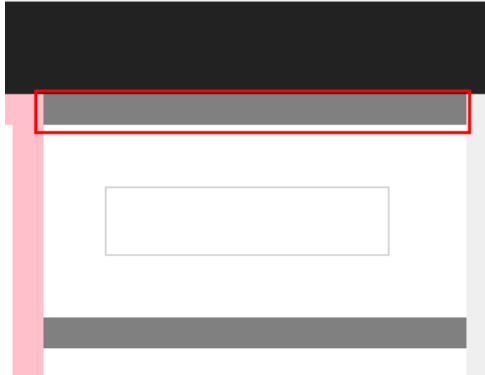
## 2.E::after 设置 E 元素后面的内容

```
p::after {
  content: '!!!';
}
```

## 3.css3选择器 实战 blog首页

### 1)添加 main区域的上边距

给第一个aside添加上边距 -> 伪类选择器



```
aside:first-child,article:first-child {
  margin-top: 20px;
}
```

### 2)清除浮动

旧式方法:

额外加一个清除浮动div

用伪对象构造一个对象来清除浮动

```
#pager {
  width: 805px;
  height: 25px;
  font: 17px/23px '微软雅黑','黑体',sans-serif;
  margin: 20px 0px;
}

#main::after {
  content: '';
  display: block;
  width: 0px;
  height: 0px;
  clear: both;
}
```

### 3)min-height:400px;[min-width]

刚开始可以给父元素一个最小高,内容再少,也是400px,内容多会自动撑开父元素

max-height [max-width]

测试 min-height ,给定一个div,div里面防止超过div大小的内容

查看加不加min-height 的区别

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<title>Document</title>
<style>
  div{
    min-height: 100px;
    width: 200px;
    background: red;
  }
</style>
</head>
<body>
  <div>
    人有消极,积极,勇敢,怯懦,勤劳,懒惰,岂是一个模型?
    择勇者使之战,择懦者使之守,各有用途.

    而激情者看到别人不如他有激情,则认为大家都在反对他,或者不配合他.
    心有怨恨.

    综上,我害怕有激情的人.
    我们凡俗夫子,也不必像千里马一跃十步,
    能像老黄牛一样,平平淡淡,功在不舍,也会做出自己的事业.
  </div>
</body>
</html>

```

#### 4.再加一个aside

去掉aside的最后一个下边距

```

aside:last-child {
  margin-bottom: 0px;
}

```

#### 5.去掉最后一个article的下边距

看这样可以吗?

```

article:last-child {
  border: 1px solid blue;
  margin-bottom: 0px;
}

```

没有效果

我们选中的是作为最后一个孩子出现的article

看article的父亲,aside的最后一个孩子是谁? -> pager

将 pager 这个div去掉,看是否生效

那如何选中最后一个article

article元素 且是 最后一个article类型的元素 -> last-of-type

```

article:last-of-type {
  border: 1px solid blue;
  margin-bottom: 0px;
}

```

#### 6.最后一个aside效果并不好,不应该去掉下边距

至此,首页就完成了

将css选择器引用到自己的项目中来

学东西光理解是不管用的,要会实际运用

课后作业:

独立首页细节

## 第 49 课 图标字体

# 让Nginx支持pathinfo

2015-01-26 by 十八  Linux服务器  Nginx | pathinfo  条评论

观察小图标,既不是背景图,也不是图片.其实是个字

twitter团队开发的

常用图标效果,用图片或者背景图来做(不方便)

所以他们就做了一种独特的字,是个特殊的字体

1.打开官网,复制直接引用字体,看是否可行

<http://fontawesome.github.io/Font-Awesome/>

打开 -> icons

```
<i class="fa fa-hourglass-start"></i>
```

首先这是一种字体,我们机器上没有这种字体

但是你要求来你网站看的人,都有这种字体吗?不是的

css允许我们引入外部字体

页面 -> 引入外部 awesome字体

页面内设置: font-family:awesome;

页面内部调用 -> 气泡 -> 浏览器显示这个气泡字体



显示awesome字体:

1)css,引入外部awesome字体

2)写图标文字

3)声明文字使用awesome字体 font-family

下载awesome字体 -> 解压 -> 只需要fonts目录

css如何引入外部文字:

字体有多种格式,它用src引入了不同格式的文字,为了保证各种浏览器都能适应,可以使用这套文字

```
@font-face {  
  font-family: 'FontAwesome';  
  src: url('../fonts/fontawesome-webfont.eot?v=4.3.0');  
  src: url('../fonts/fontawesome-webfont.eot?#iefix&v=4.3.0')  
  format('embedded-opentype'),  
  url('../fonts/fontawesome-webfont.woff2?v=4.3.0') format('woff2'),  
  url('../fonts/fontawesome-webfont.woff?v=4.3.0') format('woff'),  
  url('../fonts/fontawesome-webfont.ttf?v=4.3.0') format('truetype'),  
  url('../fonts/fontawesome-webfont.svg?v=4.3.0') format('svg');  
}
```

```
url('../fonts/fontawesome-webfont.woff?v=4.3.0') format('woff'),
url('../fonts/fontawesome-webfont.ttf?v=4.3.0') format('truetype'),
url('../fonts/fontawesome-webfont.svg?v=4.3.0#fontawesomeregular')
format('svg');
font-weight: normal;
font-style: normal;
}
```

字符的unicode码

<http://fontawesome.github.io/Font-Awesome/cheatsheet/>

每一个字符,都有独特的unicode码

使用字符的unicode码来引用这个字符

```
<style>
  div {
    font-family: 'FontAwesome';
  }
</style>
</head>
<body>
  <div>&#xf042;</div>
</body>
```

利用伪对象选择器,在a标签[9条评论]前加上小图标

注意:在css中写,跟标签中写法不同,同样是16进制

标签中是 &#xf

css中是 \f

```
<style>
  div {
    font-family: 'FontAwesome';
  }

  a::before {
    font-family: 'FontAwesome';
    content: "\f042";
  }
</style>
</head>
<body>
  <div>&#xf042;</div>
  <a href="#">9条评论</a>
</body>
</html>
```

推特工程师对这400多个图片,早已经写好了400多个类

在元素的::before 加上这个图标文字

<http://fontawesome.github.io/Font-Awesome/get-started/>

在head标签中直接引入这个css即可

看一个这个类干了些什么 -> 类名::before{content:"};

所有的图标:

<http://fontawesome.github.io/Font-Awesome/icons/>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
</head>
```

```
<body>
  <i class="fa fa-hand-paper-o">我是图标</i>
</body>
</html>
```

并不一定要用*i*标签,只要在标签中加入这个类就可以使用

```
<body>
  <i class="fa fa-hand-paper-o">我是小手</i>
  <a href="#" class="fa fa-comment">9条评论</a>
</body>
```

awesome字体实际运用到blog中

```
.catlink::before {
  font-family: 'FontAwesome';
  padding: 0px 4px;
  content: "\f07c";
}

.comment::before {
  font-family: 'FontAwesome';
  padding: 0px 4px;
  content: "\f075";
}

<span><a href="#" class="catlink">闲谈随笔</a></span>
<span><a href="#" class="comment">9条评论</a></span>
```

课后作业:

- 1.在页面上画出 10 个图标
- 2.查看 fontawesome 的官网,看是否有更简单的调用图标的办法

## 第 50 课 色彩的表示

知道色彩的三种表示方式

之前我们在学习中用过 orange,white,还有#EEE,这种形式,  
计算机如何表示色彩,我们来总结一下

```
black;white;red;green 英文单词
#34F21D; 十六进制
rgb(34,21,67);
```

色彩是怎么组成的?

三原色:

红 (Red) [0-255]

绿 (Green) [0-255]

蓝 (Blue) [0-255]

一共可以组成:  $255^3$  种色彩,不过这还是不能模拟出自然界中所有的色彩,对人的眼睛已经足够了  
通过这三种色彩浓淡(深浅)的搭配,可以产生多种多样的颜色

```
rgb(0,0,0);
rgb(255,255,255);
rgb(127,127,127);
rgb(255,0,0)
```

井号的写法,是感觉rgb这种写法比较麻烦,转成16进制表示  
十六进制表示:



rgb(100,34,72);将rgb每个位上的数转成十六进制  
前面加上#表示  
rgb和十六进制表示同样的颜色

```
#t1{
  background:rgb(100,23,188);
}

#t2{
  background:#6417BC;
}

#333333 可以简写为#333
```

英文单词表示是为了便于人来写  
background:purple;  
审查元素 -> computed  
计算机还是转成rgb来认识和显示的  
rgb(128, 0, 128);

课后作业:

做一张常用色彩表,便于自己查阅,比如

|        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|
| 000000 | 000020 | 000040 | 000060 | 000080 | 0000a0 | 0000c0 | 0000ff |
| 800000 | 800020 | 800040 | 800060 | 800080 | 8000a0 | 8000c0 | 8000ff |
| ff0000 | ff0020 | ff0040 | ff0060 | ff0080 | ff00a0 | ff00c0 | ff00ff |

## 第 51 课 尺寸的表示

在计算机内如何表示一个尺寸的大小?  
前面我们用到最多的是px,像素(绝对表示法)  
像素就是在屏幕上划分很多个小格,每一个小格就是1像素  
除了像素还有哪些?  
100%(百分比,相对表示法),em等

准备:div#test1>div#test2>p, p 中加些文字.

1.让test2始终是test1的一半大小

1)绝对表示法,像素px

```
#test1 {
  width: 1000px;
  height: 400px;
  background: gray;
}

#test2 {
  width: 500px;
  height: 200px;
  background: red;
}
```

2)当test1改变大小

绝对表示的test2将不是test1的一半

如何让test2始终是test1的一半大小

只能用百分比(相对表示法)

```
#test1 {
```

```
width: 1000px;
height: 400px;
background: gray;
}

#test2 {
  /*
    50%是相对于test2的容器,也就是它的父元素
  */
  width: 50%;
  height: 50%;
  background: red;
}
```

### 3)相对的单位,em

给test2和p都加上文字

```
<div id="test1">
  <div id="test2">我是test2下的文字
    <p>我是test2下的p标签的文字</p>
  </div>
</div>
```

test2设置字体为12px,p标签设置字体为1em

观察test2和p的文字

```
#test2 {
  width: 500px;
  height: 200px;
  background: red;
  font-size: 12px;
}

p {
  font-size: 1em;
}
```

什么是em?

网页的排版,参考了很多以前老式打字机的排版

它是一个打字排版的术语,代表一个"M"字母的宽度

(英文不同于中文,它的各个字母宽度是不同的)

在css中,代表一个父元素中,一个M的宽度

简化为:父元素中,字体的大小,就是1em

所以1em是不固定大小的,它取决于父元素字体的大小

将p标签的font-size改为1.5em

控制台 查看-> computer 计算出的字体大小

课后作业:

自行搜索,是否还有其他表示尺寸的单位,并研究其用法

## 第 52 课 css 画圆角

<http://www.itbool.com/>

用到了圆角,如何做?

css 圆角 属性

border-radius

画圆需要: 圆心,半径

在div中,画圆弧需要沿着div的四个角画,所以我们只要指定半径就可以了

## 1.div加上border-radius

```
div {  
  width: 300px;  
  height: 300px;  
  border: 1px solid blue;  
  border-radius: 10px;  
}
```

## 2.给div的四个角分别指定圆角

计算机的原点是在左上角

```
border-radius: 10px 20px 30px 40px;
```

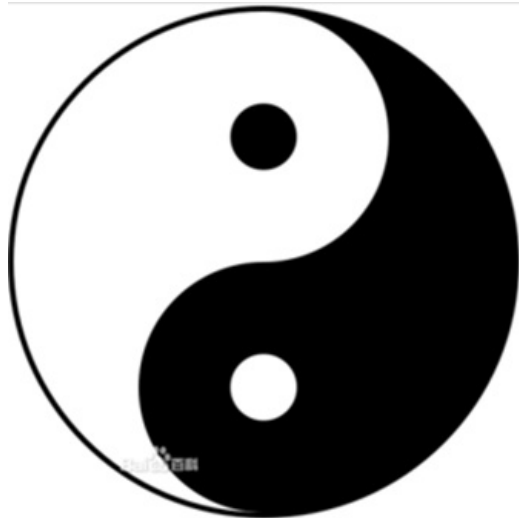
## 3.画直径为100px的圆

```
div {  
  width: 100px;  
  height: 100px;  
  border: 1px solid blue;  
  border-radius: 50px;  
}
```

## 4.画个直径为100px的半圆

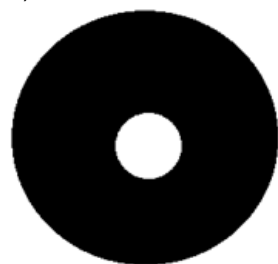
```
div {  
  width: 100px;  
  height: 50px;  
  border: 1px solid blue;  
  border-radius: 50px 50px 0px 0px;  
}
```

## 5.div+css画太极 (用一个div)



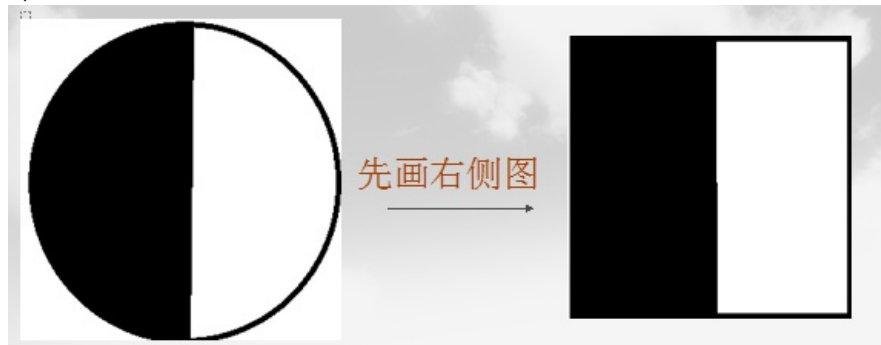
分析太极图的结构

1)画如下图



```
#t1 {
  width: 100px;
  height: 100px;
  border: 100px solid black;
  background: white;
  border-radius: 150px;
}
```

2)画如下图



右侧:

```
#t2 {
  width: 300px;
  height: 600px;
  background: white;
  border: 1px solid black;
  border-left: 300px solid black;
}
```

左侧:

```
#t2 {
  width: 300px;
  height: 600px;
  background: white;
  border: 1px solid black;
  border-left: 300px solid black;
  border-radius: 300px;
}
```

3)利用伪对象,人为的在t2前加上一个t1  
在利用margin-left移动div

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    #t2 {
      width: 300px;
      height: 600px;
      background: white;
      border: 1px solid black;
      border-left: 300px solid black;
      border-radius: 300px;
    }

    #t2::before{
      content: '';
      display: block;
      width: 100px;
```

```

        height: 100px;
        border: 100px solid black;
        background: white;
        border-radius: 150px;
        margin-left: -150px;
    }

    #t2::after {
        content: '';
        display: block;
        width: 100px;
        height: 100px;
        border: 100px solid white;
        background: black;
        border-radius: 150px;
        margin-left: -150px;
    }
}
</style>
</head>
<body>
    <div id="t2"></div>
</body>
</html>

```

课后作业:

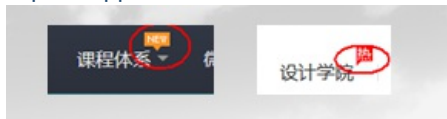
利用圆角优化页面

## 第 53 课 相对定位与绝对定位

观察小图标,是如何悬浮在文字的斜上方的?

<http://study.163.com/>

<http://ke.qq.com/>



如何更自由的控制元素的位置,有 2 种定位方式

相对定位

`position: relative;`

准备: `div#t1+div#t2+div#t3`



1) 让t1这个div往下走10px

可以用margin往下挤

```

#t1 {
    background: red;
    margin-top: 10px;
}

```

t2,t3也往下走了10px,类似第一拍桌子往后,后面的也要往后移动

2) 只让t1往下走10px;t2,t3相对于自己原来的位置不动

那就是t1相对于自己原来的位置向下移动10px,其它元素的位置不变

可以用相对定位来完成 -> 相对于自己原来的位置

`position: relative;` -> 告诉元素相对于自己的位置移动

那怎么动?动多少? -> `top, right, bottom, left` [相对于自己的上下左右四条边移动多少]

```
#t1 {
  background: red;
  position: relative;
  top: 30px;
  left: 20px;
}
```



相对定位和margin挤的区别

用margin挤,挣个布局都会收到干扰

而相对定位只改变元素自己的位置,其它的元素位置不会受到干扰

绝对定位

position: absolute;

准备: div#comm+div#t1+div#t2+div#t3

绝对定位是浏览器先假设这些绝对定位的元素不存在

当浏览器渲染完其他的元素之后

再渲染这些绝对定位的元素,将这些绝对定位的div直接从半空中投射过去 -> 投射到哪里?它的父元素的位置上

所以这些绝对定位的元素有可能会覆盖在其他普通元素之上

```
div {
  width: 300px;
  height: 300px;
}

#t1 {
  background: red;
  position: absolute;
}
```

为什么没有投射在普通元素上?

我们要说明这个绝对div距离它父元素是多少

```
#t1 {
  background: red;
  position: absolute;
  top: 0px;
  left: 0px;
}
```

给3个div设置不同的到父div的位置

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
<style>
  div {
    width: 300px;
    height: 300px;
  }

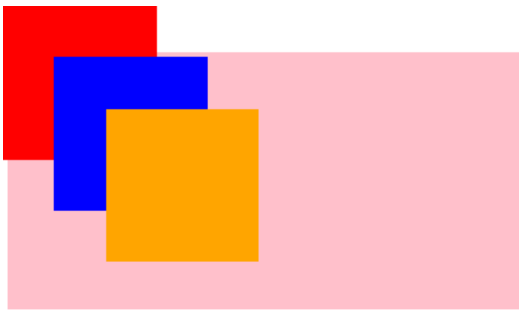
  #t1 {
    background: red;
    position: absolute;
    top: 0px;
    left: 0px;
  }
  #t2 {
    background: yellow;
    position: absolute;
    top: 30px;
    left: 20px;
  }
  #t3 {
    background: pink;
    position: absolute;
    top: 60px;
    left: 40px;
  }
</style>
```

```
    }

    #t2 {
      background: blue;
      position: absolute;
      top:100px;
      left:100px;
    }

    #t3 {
      background: orange;
      position: absolute;
      top:200px;
      left:200px;
    }

    #comm {
      width: 1000px;
      height: 500px;
      background: pink;
    }
  }
</style>
</head>
<body>
  <h1>绝对定位</h1>
  <div id="comm"></div>
  <div id="t1"></div>
  <div id="t2"></div>
  <div id="t3"></div>
</body>
</html>
```

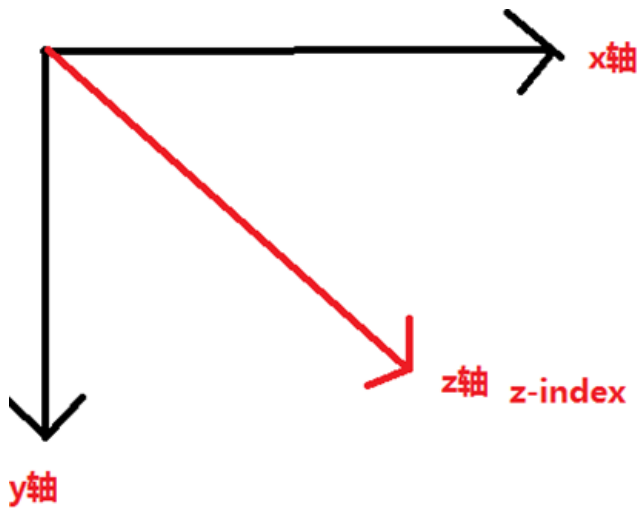


分析3个绝对div覆盖规律

那如何将红色的部分显示在最前面？

明白：

计算机的数轴：



飞来四架飞机,谁在上方?

用 **z-index**.

**z**轴表示屏幕到我们视觉方向的距离;

数值越大,则离我们越近,

则会覆盖在最上方

```
#t1 {  
  background: red;  
  position: absolute;  
  top: 0px;  
  left: 0px;  
  z-index: 1;  
}
```

首页导航实战 -> 小图标

```
nav img {  
  position: absolute;  
  top: 0px;  
  left: 0px;  
}
```

为什么在页面的左上角

如果一个元素需要用绝对定位

需要它的父亲也有**position**属性

如果父亲没有**position**属性,就会找它的爷爷...增爷爷...body

那我们可以加一个相对定位,**top, left**等都不写

让父元素相对自己的位置变化,实际又没有没变

**position: relative;**

```
nav ul li:nth-child(3) {  
  position: relative;  
}  
nav img {  
  position: absolute;  
  top: 0px;  
  right: 0px;  
}
```

课后作业:

完成导航上漂浮 new 小图标





## 5.爱好:多选框 checkbox

```
爱好:<br >
篮球:<input type="checkbox" name="hobby" id="" value="basketball"><br >
足球:<input type="checkbox" name="hobby" id="" value="foottball"><br >
乒乓球:<input type="checkbox" name="hobby" id="" value="pingpang"><br >
```

## 6.学历:下拉框

```
<select name="xueli" id="">
  <option value="daxue">大学</option>
  <option value="gaozhong">高中</option>
  <option value="chuzhong">初中</option>
</select>
```

## 7.简介:文本域 textarea

```
<textarea name="intro" id="" cols="30" rows="10"></textarea>
```

## 8.头像:文件域

```
头像:
<input type="file" name="pic">
```

选择文件这个改不掉,别的网站用的不一样,它是用其他的input模拟出来的,一点用js触发的选择文件的效果

## 9.隐藏域:IP

```
<input type="hidden" name="ip" value="192.168.1.100">
```

## 10.action:提交到哪

```
<input type="submit" value="提交">
```

action可以提交到[www.baidu.com](http://www.baidu.com)

百度如果可以接收我们的信息,存储到数据库,一个动态的网页就完成了

## 11.method:什么方法提交

默认是get传参,参数显示在url地址栏上,我们的个人信息都暴漏了  
所以要用post传参是不在地址栏上显示的

## 12 enctype:是否有文件

如果有上传文件,要在form 处加上:

enctype="multipart/form-data" [这是定死的,死记住]

## 13.恢复原状

表单被填写乱的时候,想全部清空重写可用

```
<input type="reset" value="恢复原状">
```

课后作业:

完成 1 个具有所有类型 input 的表单

# 第 56 课 HTML5 新增 input

这几个新增类型兼容不是特别好,先了解,最好不要用

## 1.选择日期 date

<http://www.my97.net/dp/demo/index.htm> -> 功能及示例



这个效果是用js做出来的,现在可以用h5的新input标签一行就能做出来(火狐不兼容)

```
<input type="date" name="birth">
```

## 2.时间 time

```
<input type="time" name="clock" id="">
```

## 3.星期 week

```
<input type="week" name="" id="">
```

## 4.范围(range)

```
<input type="range" name="age" min="18" max="60">
```

火狐看以上几个表单项的兼容

## 5.搜索

### 1)老式写法,需要删除默认值才能再输入

```
<input type="text" name="search" value="请输入关键词">
```

### 2)h5搜索框的新属性 placeholder

当我们输入内容时,需要将value的值删除才行,  
以前使用js,当点击这个搜索框时,里面的内容重置为空  
现在可以用

placeholder 属性,做搜索框默认文字的效果,  
当输入内容时,默认文字自动删除

```
<input type="text" name="search" placeholder="请输入关键词">
```

博客搜索框加上 placeholder属性

课后作业:

测试这几种新增 input 在不同浏览器下的各自表现

## 第 57 课 结课作业

我们的结课作业就是做这个blog首页

可以参考十八哥的博客首页,也可以参考别人的blog

<http://www.ending.com/>

<http://dxq123.sinaapp.com/index.php>

我们模版是自己写的,后面**php**也是自己写的,数据库是自己建的,我们可以发布在往上,找工作的时候很有说服力