SQL 2014 https://pan.baidu.com/share/init?surl=tgTztj9VydneTXEgJNU

# 1、VS操作技巧

## 快速注释

### （1）、单行注释

使用 //

### （2）、多行注释

/\*

\*

\* \*\*/

小技巧：先敲/\*\*\*/然后在第一个\*后面敲回车键

### （3）、文档注释

鼠标在类上端按三个斜杠就会自动生成：

/// <summary>

///

/// </summary>

### （4）、代码折叠

使用

#region

#end region

把代码进行折叠

### （5）快速屏蔽多行代码

　　注释多行：先按 Ctrl - K 组合键，再按 Ctrl - C 组合键

　　取消注释多行：先按 Ctrl - K 组合键，再按 Ctrl - U 组合键

## 编译和调试

### （1）、编译

Ctrl+F5

### （2）、断点调试

设置断点：鼠标左键单击要加断点的语句左边

F5：设置完断点之后按F5会启动调试并调到第一个断点，在按一下就跳到下一个断点。

F10：逐步调试（不会进入函数）

F11：逐句调试

# 2、C#函数、关键字、语法

## （1）类型转换

1）、隐型转换

Int price=100;

double price 1 = price;

小的类型转换为大的类型，自动转换

2）、强制转换

double price 1 = 10.1;

Int price3=(int) price 1;

大的类型转换为小的类型，需要用到强制转换

3）、将字符串类型转换为int 或者 double

int age = int.Parse(Console.ReadLine()); //注意如果用户错误输入程序会崩溃

int age;

bool result = int.TryParse(Console.ReadLine(),out age);

//用户错误输入程序不会崩溃，只是抛出异常返回0

4）、将任意类型转换为指定类型

Convert.各种成员

## （2）、字符串的常用处理方法

IndexOf();查找某个字符在字符串中的位置（下标）

LastIndexOf();与上面的区别在于这个是从右往左数

SubString();从字符串中截取字符串

ToLower();将字符串转换成小写形式

ToUpper();将字符串转换成大写形式

Equals();比较两个字符串的值是否相等

Join();连接字符串

Split();分割字符串

Trim();去掉字符串两边空格

## （3）、集合、数组、Hashtable

数组：

**初始化方法：**

方法一：Int [] age = new int[10];

方法二：int [] age;

age = new int[5]{1,2,3,4,5};

方法三：int[] age={1,2,3,4,5}

方法四：Int [] age = new int[5]{1,2,3,4,5};

**计算数组长度：**

Int [] age = new int[5]{1,2,3,4,5};

age.Length

集合（相对于数组来说集合有一个好处就是长度可变）：

**应用命名空间：**using System.Collections;

**创建集合：**ArrayList arrayList = new ArrayList();

**往集合里面添加成员：**

方法一**;**arrayList.Add() //单个添加

arrayList.AddRangz()//多个添加

例如：

Student stu=new Student();

arrayList.Add(stu);

方法二：使用初始化器

ArrayList arrayList = new ArrayList()

{

stu

};

**集合长度：**

arrayList.Count //集合里面有几个成员

**取出集合中的成员（由于集合的形参是个object类型所以取出的值要做类型转换）**

例如：

Student stu1 =( Student) arrayList[0];

**修改集合中的成员**

Student stu1 =( Student) arrayList[0]; //修改stu1等于修改 集合的地0个成员

stu1.age=18;

**删除一个成员**

arrayList.Remove(stu);//通过放入的成员名字来删除

arrayList.RemoveAt(0);//通过下标来删除第0个成员

（4）、Hashtable 类型

创建哈希表：

Hashtable ht = new Hashtable();

Student stu1 = new Student（”1001”,”张三”,15,”101班”）;

Student stu2 = new Student（”1002”,”李四”,16,”102班”）;

Student stu3 = new Student（”1003”,”王五”,17,”103班”）;

增加：

ht.Add(“stu1”,stu1);

ht.Add(“stu1”,stu1);

删除：

ht.Remove(“stu2”);

修改：

Student stu = (Student)ht(“stu2”);

Stu.StuName = “xx”

查：

foreach (string key in ht.Keys)

{

Console.WriteLine(key);

}

（5）、List<T>泛型

优点：1、避免装箱拆箱。2、避免类型转换。

命名空间：using System.Collections.Generic

创建：

List<Student> stuList = new List<Student>()

曾加：

stuList.Add(stu1);

删除：

stuList.Remove(stu1);

修改：

stuList[0].className = “aa”;

查：

for(int i=0;i<stuList.Count;i++)

{

Student s=stuList[i];

Console.WriteLine(s.StuName);

}

（6）、Dictionary 泛型字典集合

创建：

Dictionary<string,Student>dict=new Dictionary<string,Student>();

增加：

dict.Add(“zz”,stu1);

删除：

dict.Remove(“zz”);

查找：

Student stu=dict[“zz”];

# 3、数据库

## （1）、简单的增删改查

--增加

insert into Product (ProductNb, ProductName, ProductImage, Price)

values(1005,'iphone','image/xxx.jpg',9000)

--删除

delete from Product

where ProductNb = 1005

--修改

update Product

set Price=200

where ProductNb = 1004

--查询

select \* from Product

select ProductNb,ProductName from Product

where ProductNb = 1003

## （2）、联表查询

--联表查询

--内联接 inner join

select ProductNb, ProductName, ProductImage, Price, product.typeid , producttype.typeid, typename

from Product inner join ProductType

on(Product.typeid=ProductType.typeid)

select ProductNb, ProductName, ProductImage, Price, typename

from Product , ProductType

where Product.typeid=ProductType.typeid

--使用别名的内联接

select ProductNb, ProductName, ProductImage, Price, typename

from Product a, ProductType b

where a.typeid=b.typeid

--左外联接 left join （以左边为基准，左边有的会全部显示）

select ProductNb, ProductName, ProductImage, Price , typename

from ProductType left join Product

on(Product.typeid=ProductType.typeid)

--左外联接 right join （以右边为基准，左边有的会全部显示）

select ProductNb, ProductName, ProductImage, Price , typename

from ProductType right join Product

on(Product.typeid=ProductType.typeid)

## （3）、聚合函数

--聚合函数（count sum avg max min）

-- count 统计记录条数

--使用指定列查询

select count(Product.Price) as 价格等于600 from Product

where Product.Price=600

--使用任意列查询

select count(\*) from Product

--使用伪列查询（实战中基本上用这个）

select count(1) from Product

--sum（求总和一般只能针对于数字和日期）

select sum(Product.Price) as sum\_price from Product

--avg (求平均值一般只能针对于数字和日期)

select avg(Product.Price) from Product

--max (找出最大值一般只能针对于数字和日期)

select max(Product.Price) from Product

--min (找出最小值一般只能针对于数字和日期)

select min(Product.Price) from Product

（4）、排序和分组

--排序和分组 order by group by

--排序order by [desc]降序 [asc]升序

select \* from Product

order by price desc,typeid desc

--分组 根据产品类型统计每个产品的个数

select ProductType.typename ,count(1) as num

from Product inner join ProductType

on Product.typeid=ProductType.typeid

group by ProductType.typename

order by num

注意，如果同时出现 where、group by、order by 那么order by必须放在最后面 where 应该在greop by前面

## （4）、数据库变量

--数据库变量

--自定义变量

declare @name varchar(50) = '匿名',@id int =10

set @name = '帅哥' --通常赋予固定值 set select 都可以

select @name = '张工',@id = 11 --如果同时给多个变量进行赋值只能用select 不能使用set

select @name=ProductType.typename , @id=ProductType.typeid from ProductType

print @name+'\_'+cast(@id as varchar) --cast类型转换

--查询返回多个值赋给同一个变量set会出错 而select会赋予最后一个返回值给变量

--查询不到信息set会赋予NULL值给变量而select会保持变量之前的值

--系统变量

select @@ERROR --放在语句后面捕获错误类型

--自增

insert into ProductType( typename)

values('我')

select @@identity

--查询当前语言

select @@LANGUAGE

--编译器版本

select @@VERSION

## （5）、数据库事务和程序事务

1）、事务的特性

原子性：不可分割（多个操作）要么一起执行，要么都不执行。

一致性：当数据操作完成，数据保持一致性。

隔离性：每一个事务都是个独立的个体，相互之间是隔离的（不相干）

持久性：事务提交数据会永久保存，永久存储到物理磁盘（数据库文件）

2）、常见的3种的数据库事务

自动提交事务：sqlserver 默认开启而oralce默认关闭；在sqlserver种的insert、delete、update都是自动提交事务。平时 程序sqlconnection关闭的时候也会自动提交。

显示事务：

开始事务 begin transaction

编写事务脚本

commit transaction提交事务 或者 rollback transaction回滚事务

eg:

--转账业务

begin transaction –开始事务

declare @errorsum int = 0 –默认没有错误

update Account

set UserMoney= UserMoney-2000 –如果值是负数会报错

where CardID=1001 --张三

set @errorsum=@errorsum+@@error –只要update报错error就不等于0

update Account

set UserMoney= UserMoney+2000

where CardID=1002 –李四

set @errorsum=@errorsum+@@error –只要update报错error就不等于0

if @errorsum<>0 --其中<>是不等于的意思

begin

print ‘转账失败’

rollback; --回滚

end

else

begin

print ‘转账成功’

commit; --提交

end

隐式事务：sqlserver 默认是关闭状态而oralce是开启状态。隐式事务开启的时候自动提交事务就不会自动提交。（有自带开启事务、没有提交事务）

--开启隐式事务 set implicit\_transaction on

--关闭隐式事务 set implicit\_transaction off

## （6）、触发器

### 1）、增删改查会分别产生两个临时表

inserted 临时表 :在执行insert动作的时候会产生

deleted 临时表 ：在执行delet动作时会产生

注意：在执行update 动作底层会执行delete 和 insert动作

2）、触发器实例

--insert

create trigger insert\_Recorlog

on product

for insert -- for 是受约束限制（比如之前的主表与子表关联），此处也可以使用delete update

as

begin

declare @uerid int --自定义一个用户编号

select @userid=typeid form inserted

insert into logs(logContent,createuserid)

values('添加动作',@userid)

end

--delete删除

create trigger delect\_Recorlog

on ProductType

instead of delete -- instead of 不受约束限制这是和之前for的区别，此处也可以使用delete update

as

begin

declare @uerid int --自定义一个用户编号

select @userid=typeid form inserted

delete from product where typeid=@userid

delete form producttype where typeid=@user

end

注意：触发器第一次创建用create 修改的时候是使用alter

## （7）、用命令行操作数据库

1）、连接SQL：进入数据库的bin路径：mysql -hlocalhost -uroot –proot

查看里面有哪些数据库：show databases

删除指定的数据库：drop database web1

2）、创建一个数据库：create database web;

3）、使用这个数据库：use database; 或者 use web

4）、查看数据库里面所有的表 : show tables;

查看单表admin里面的内容 desc admin

5）、创建表

管理员表admin(ID、登入名、密码)：

create table admin(

id int primary key auto\_increment,

username varchar(20),

password char(32)

);

注意：

primary key 主键

auto\_increment自动增长

单页模板表 board（ID、模块名、内容）

create table board(

id int primary key auto\_increment,

boardname varchar(20),

content text

);

注意：

text 文本

新闻表：news（ID、标题、分类、作者、内容、头图、发布时间、点击率）

create table news(

id int primary key auto\_increment,

title varchar(50),

cate\_id int default 0,

author varchar(30),

content text,

img varchar(50),

intime timestamp default current\_timestamp,

hlts int default 0

);

注意：

时间轴：timestamp

当前时间：current timestamp

产品表product（ID、名称、编号、分类、详细说明、产品分类、产品图、发布时间、点击率）

create table product(

id int primary key auto\_increment,

productname varchar(40),

pro\_no varchar(30),

cate\_id int default 0,

content text,

img varchar(50),

intime timestamp default current\_timestamp,

hlts int default 0

);

分类表category(ID、分类名、排序号)

create table category(

id int primary key auto\_increment,

catename varchar(30),

orderno tinyint unsigned default 5

);

注意

tinyint 是小数字的意思 0-255

友情链接表 flink(ID 、链接名、网址、说明、时间)

create table flink(

id int primary key auto\_increment,

title varchar(20),

link\_url varchar(50),

content varchar(50),

intime timestamp default current\_timestamp

);

留言表questbook(ID、用户名、内容、留言时间)

create table questbook(

id int primary key auto\_increment,

username varchar(30),

content text,

intime timestamp default current\_timestamp

);

6）、删除表admin

drop table admin

# HTML CSS相关学习

学习网站：<https://www.imooc.com/learn/9>

w3school官网：https://www.w3school.com.cn/

## 1、慕课网html入门学习笔记

### （1）、标签的语法

1. 标签由英文尖括号<和>括起来，如<html>就是一个标签。

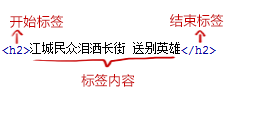
2. html中的标签一般都是成对出现的，分开始标签和结束标签。结束标签比开始标签多了一个/。

如：

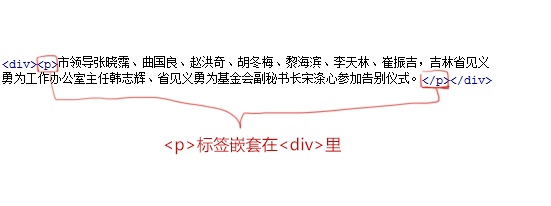
（1） <p></p>

（2） <div></div>

（3） <span></span>



3. 标签与标签之间是可以嵌套的，但先后顺序必须保持一致，如：<div>里嵌套<p>，那么</p>必须放在</div>的前面。如下图所示。



4. HTML标签不区分大小写，<h1>和<H1>是一样的，但建议小写，因为大部分程序员都以小写为准。

### （2）、认识文件的基本结构

一个HTML文件是有自己固定的结构：

<html>

<head>...</head>

<body>...</body>

</html>

代码讲解：

1. <html></html>称为根标签，所有的网页标签都在<html></html>中。

2. <head> 标签用于定义文档的头部，它是所有头部元素的容器。头部元素有<title>、<script>、 <style>、<link>、 <meta>等标签，头部标签在下一小节中会有详细介绍。

3. 在<body>和</body>标签之间的内容是网页的主要内容，如<h1>、<p>、<a>、<img>等网页内容标签，在这里的标签中的内容会在浏览器中显示出来。

### （3）、认识head标签

文档的头部描述了文档的各种属性和信息，包括文档的标题等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在 head 部分：

<head>

<title>...</title>

<meta>

<link>

<style>...</style>

<script>...</script>

</head>

<title>标签：在<title>和</title>标签之间的文字内容是网页的标题信息，它会出现在浏览器的标题栏中。网页的title标签用于告诉用户和搜索引擎这个网页的主要内容是什么，搜索引擎可以通过网页标题，迅速的判断出网页的主题。每个网页的内容都是不同的，每个网页都应该有一个独一无二的title。

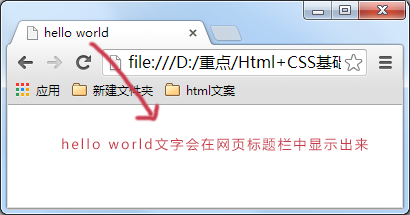
例如

<head>

<title>hello world</title>

</head>

<title>标签的内容“hello world”会在浏览器中的标题栏上显示出来，如下图所示



### （4）、HTML的代码注释

语法：

<!--注释文字 -->

### （5）、语义化，让你的网页更好的被搜索引擎理解

学习html标签过程中，主要注意两个方面的学习：标签的用途、标签在浏览器中的默认样式。

**标签的用途：**我们学习网页制作时，常常会听到一个词，语义化。那么什么叫做语义化呢，说的通俗点就是：明白每个标签的用途（在什么情况下使用此标签合理）比如，网页上的文章的标题就可以用标题标签，网页上的各个栏目的栏目名称也可以使用标题标签。文章中内容的段落就得放在段落标签中，在文章中有想强调的文本，就可以使用 em strong标签表示强调等等。

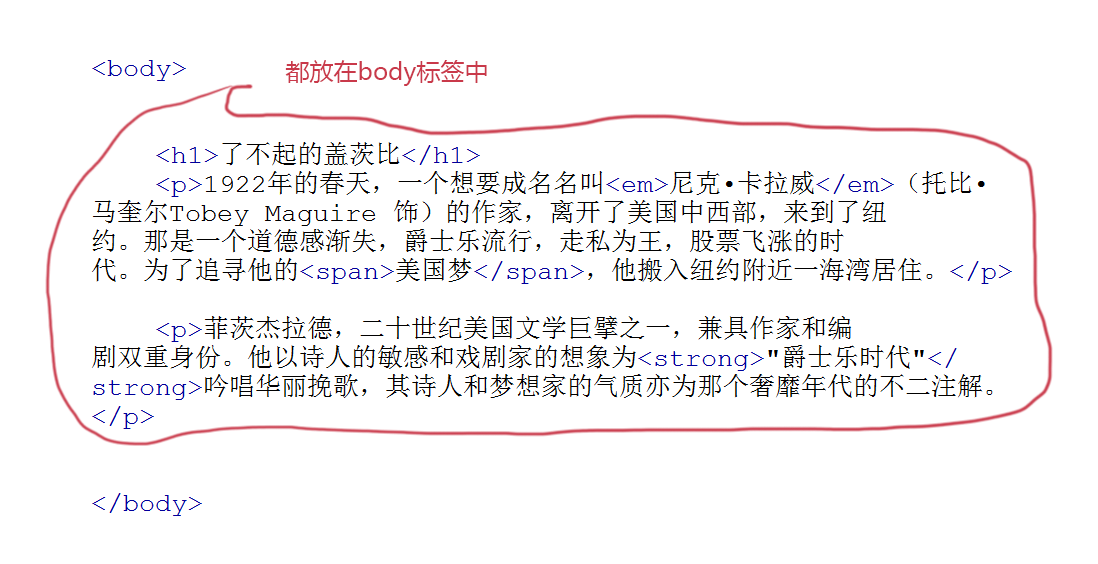
语义化可以给我们带来什么样的好处呢？

1. 更容易被搜索引擎收录。

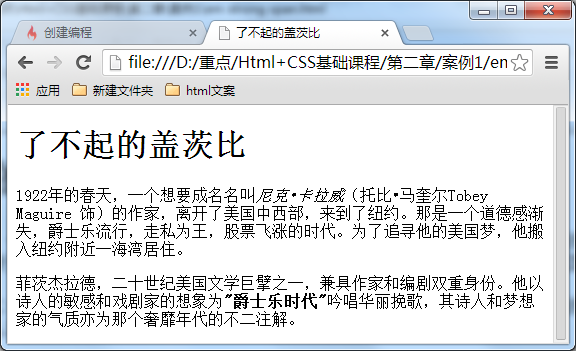
2. 更容易让屏幕阅读器读出网页内容。

### （6）、<body>标签，网页上显示的内容放在这里

网页上要展示出来的页面内容一定要放在body标签中。如下图是一个新闻文章的网页。



在浏览器中的显示效果：



### （7）、<p>标签，添加段落

如果想在网页上显示文章，这时就需要<p>标签了，把文章的段落放到<p>标签中。

语法：

<p>段落文本</p>

### （8）、了解<hx>标签，为你的网页添加标题

文章的段落用<p>标签，那么文章的标题用什么标签呢？在本节我们将使用<hx>标签来制作文章的标题。

标题标签一共有6个，h1、h2、h3、h4、h5、h6分别为一级标题、二级标题、三级标题、四级标题、五级标题、六级标题。并且依据重要性递减。<h1>是最高的等级。

语法：

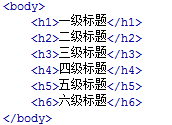
<hx>标题文本</hx> (x为1-6)

文章的标题前面已经说过了，可以使用标题标签，另外网页上的各个栏目的标题也可使用它们。如下图为腾讯网站。

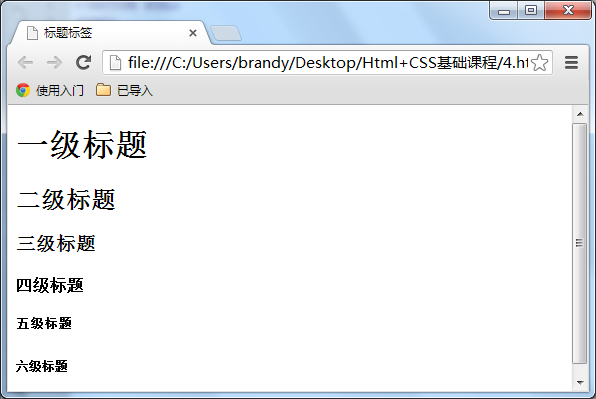


h1-h6标签的默认样式：

标签代码：



在浏览器中显示的样式：



### （9）、加入强调语气，使用<strong>和<em>标签

有了段落又有了标题，现在如果想在一段话中特别强调某几个文字，这时候就可以用到<em>或<strong>标签。

但两者在强调的语气上有区别:<em> 表示强调，<strong> 表示更强烈的强调。**并且在浏览器中<em> 默认用斜体表示，<strong> 用粗体表示**。两个标签相比，目前国内前端程序员更喜欢使用<strong>表示强调。

语法：

<em>需要强调的文本</em>

<strong>需要强调的文本</strong>

如，在网上商城中，某产品的打折后的价格是需要强调的。如下图。

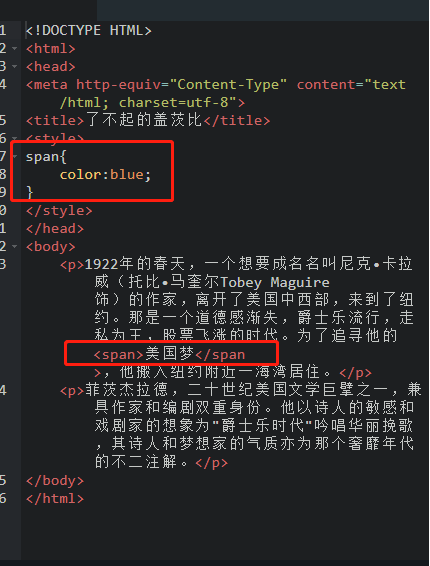


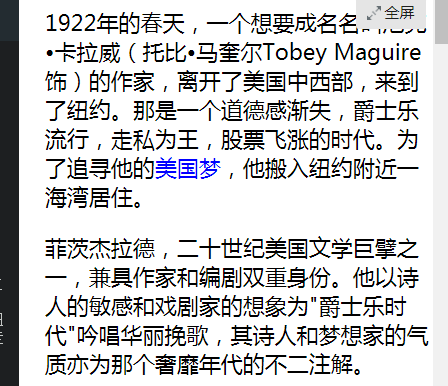
### （10）、使用<span>标签为文字设置单独样式

<span>标签是没有语义的，它的作用就是为了设置单独的样式用的。

如果现在我们想把上一小节的第一段话“美国梦”三个字设置成blue（蓝色），但注意不是为了强调“美国梦”，而只是想为它设置和其它文字不同的样式（并不想让屏幕阅读器对“美国梦”这三个字加重音读出），所以这样情况下就可以用到<span>标签了。

如下面例子：





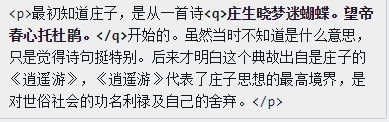
### （11）、<q>标签，短文本引用

想在你的html中加一段引用吗？比如在你的网页的文章里想引用某个作家的一句诗，这样会使你的文章更加出彩，那么<q>标签是你所需要的。

语法：

<q>引用文本</q>

如下面例子：

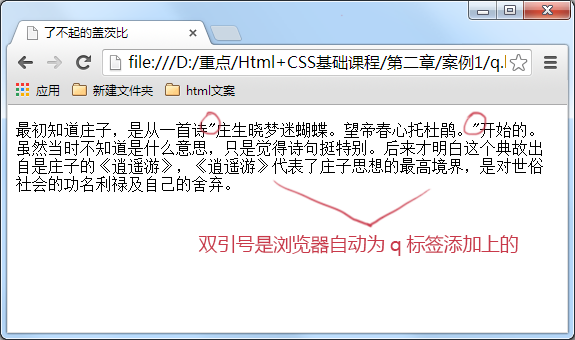


讲解：

1. 在上面的例子中，“庄生晓梦迷蝴蝶。望帝春心托杜鹃。” 这是一句诗歌，出自晚唐诗人李商隐的《锦瑟》 。因为不是作者自己的文字，所以需要使用<q></q>实现引用。

2. 注意要引用的文本不用加双引号，浏览器会对q标签自动添加双引号。

下图是代码显示结果：



注意:这里用<q>标签的真正关键点不是它的默认样式双引号（如果这样我们不如自己在键盘上输入双引号就行了），而是它的语义：引用别人的话。

### （12）、<blockquote>标签，长文本引用

<blockquote>的作用也是引用别人的文本。但它是对长文本的引用，如在文章中引入大段某知名作家的文字，这时需要这个标签。

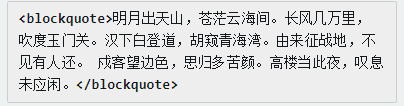
等等，上一节<q>标签不是也是对文本的引用吗？不要忘记<q>标签是对简短文本的引用，比如说引用一句话就用到<q>标签。

如想在我的文章中引用李白《关山月》中的诗句，因为引用文本比较长，所以使用<blockquote>。

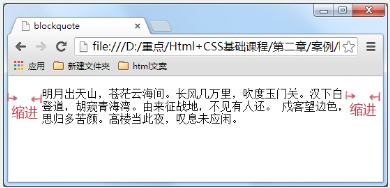
语法:

<blockquote>引用文本</blockquote>

如下面例子：



浏览器对<blockquote>标签的解析是缩进样式。如下图所示：



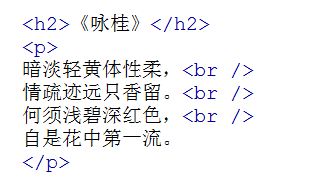
### （13）、使用<br>标签分行显示文本

对于上一小节的例子，我们想让那首诗显示得更美观些，如显示下面效果：



怎么可以让每一句诗词后面加入一个折行呢？那就可以用到<br />标签了，在需要加回车换行的地方加入<br />，<br />标签作用相当于word文档中的回车。

上节的代码改为：



语法：

xhtml1.0写法

<br />

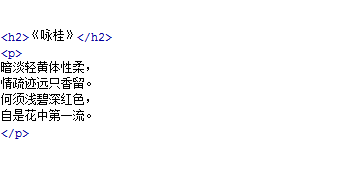
html4.01写法：

<br>

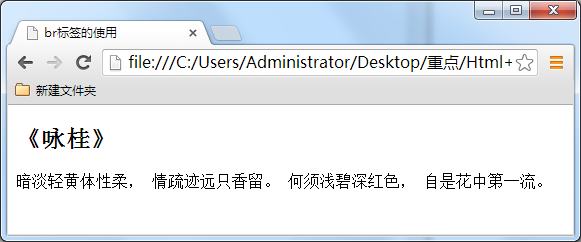
现在一般使用 xhtml1.0 的版本的写法（其它标签也是），这种版本比较规范。

与以前我们学过的标签不一样，<br />标签是一个空标签，没有HTML内容的标签就是空标签，空标签只需要写一个开始标签，这样的标签有<br />、<hr />和<img />。

讲到这里，你是不是有个疑问，想折行还不好说嘛，就像在 word 文件档或记事本中，在想要折行的前面输入回车不就行了吗？很遗憾，在 html 中是忽略回车和空格的，你输入的再多回车和空格也是显示不出来的。如下边的代码。



上面的代码在浏览中显示是没有回车效果的。如下图所示：



总结：在 html 代码中输入回车、空格都是没有作用的。在html文本中想输入回车换行，就必须输入<br />。

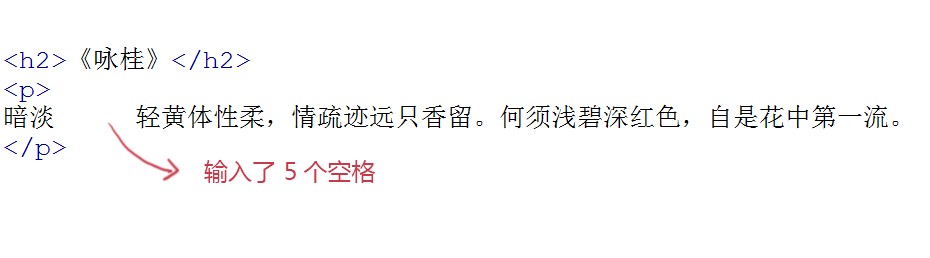
### （14）、网页中添加一些空格

在上一节的例子，我们已经讲解过在html代码中输入空格、回车都是没有作用的。要想输入空格，必须写入&nbsp;。

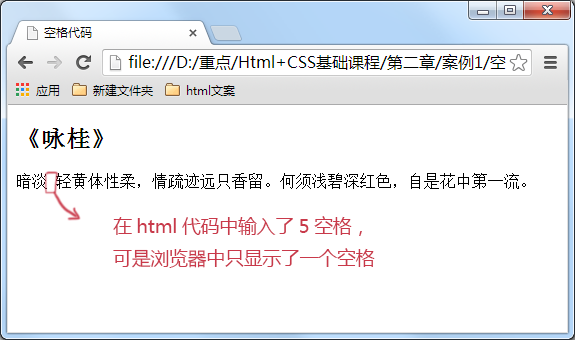
语法：

&nbsp;

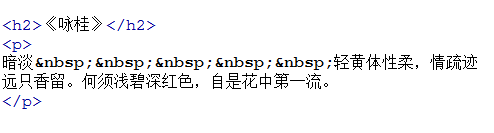
在html代码中输入空格是不起作用的，如下代码。



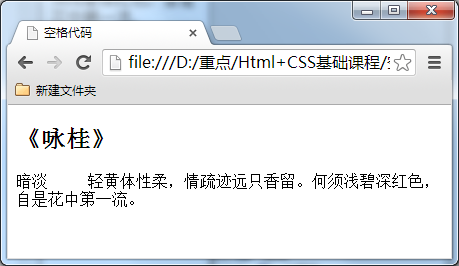
在浏览中显示，还是没有空格效果。



输入空格的正确方法：

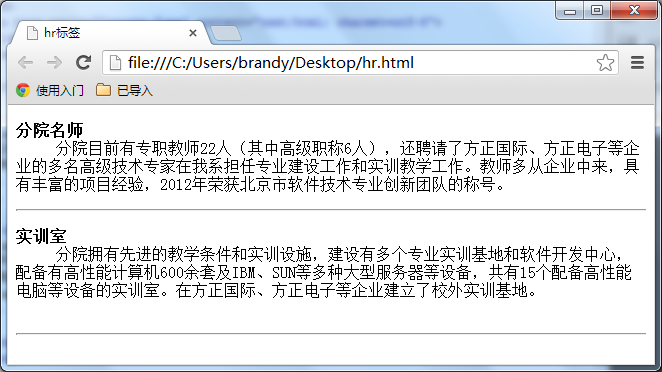


在浏览器中的显示出来的空格效果。如下图所示。



### （15）、认识<hr>标签，添加水平横线

在信息展示时，有时会需要加一些用于分隔的横线，这样会使文章看起来整齐些。如下图所示：



语法：

html4.01版本 <hr>

xhtml1.0版本 <hr />

注意：

1. <hr />标签和<br />标签一样也是一个空标签，所以只有一个开始标签，没有结束标签。

2. **<hr />标签的在浏览器中的默认样式线条比较粗，颜色为灰色**，可能有些人觉得这种样式不美观，没有关系，这些外在样式在我们以后学习了css样式表之后，都可以对其修改。

3. 大家注意，现在一般使用 xhtml1.0 的版本（其它标签也是），这种版本比较规范。

### （16）、<address>标签，为网页加入地址信息

一般网页中会有一些网站的联系地址信息需要在网页中展示出来，这些联系地址信息如公司的地址就可以<address>标签。也可以定义一个地址（比如电子邮件地址）、签名或者文档的作者身份。

语法：

<address>联系地址信息</address>

如：

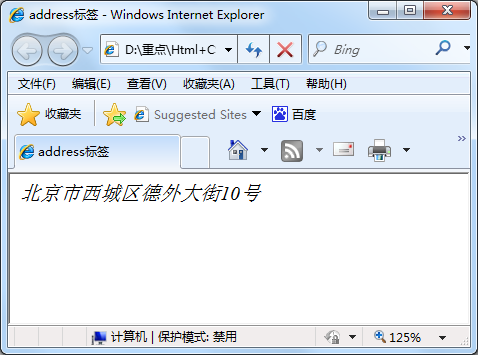
<address>文档编写：lilian 北京市西城区德外大街10号</address>

<address>

本文的作者：<a href="mailto:lilian@imooc.com">lilian</a>

</address>

在浏览器上显示的样式为斜体，如果不喜欢斜体，当然可以，可以在后面的课程中使用 css 样式来修改它<address>标签的默认样式。



### （17）、想加入一行代码吗？使用<code>标签

在介绍语言技术的网站中，避免不了在网页中显示一些计算机专业的编程代码，当代码为一行代码时，你就可以使用<code>标签了，如下面例子：

<code>var i=i+300;</code>

注意：在文章中一般如果要插入多行代码时不能使用<code>标签了。

语法：

<code>代码语言</code>

注：如果是多行代码，可以使用<pre>标签。

使用<pre>标签为你的网页加入大段代码

### （18）、使用<pre>标签为你的网页加入大段代码

在上节中介绍加入一行代码的标签为<code>，但是在大多数情况下是需要加入大段代码的，如下图：



怎么办？不会是每一代码都加入一个<code>标签吧，没有这么复杂，这时候就可以使用<pre>标签。

语法：

<pre>语言代码段</pre>

<pre> 标签的主要作用:预格式化的文本。被包围在 pre 元素中的文本通常会保留空格和换行符。

pre>

var message="欢迎";

for(var i=1;i<=10;i++)

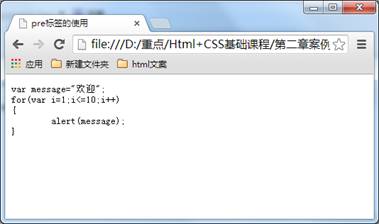
{

alert(message);

}

</pre>

在浏览器中的显示结果为：

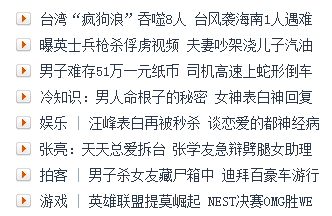


在上面的例子中可以看到代码中的空格，换行符都保留下来。如果用以前的方法，回车需要输入<br>签，空格需要输入&nbsp;

注意：<pre> 标签不只是为显示计算机的源代码时用的，在你需要在网页中预显示格式时都可以使用它，只是<pre>标签的一个常见应用就是用来展示计算机的源代码。

### （19）、使用ul li，添加新闻信息列表

在浏览网页时，你会发现网页上有很多信息的列表，如新闻列表、图片列表，如下图所示。



新闻列表

这些列表就可以使用ul-li标签来完成。ul-li是没有前后顺序的信息列表。

语法：

<ul>

<li>信息</li>

<li>信息</li>

......

</ul>

举例：

<ul>

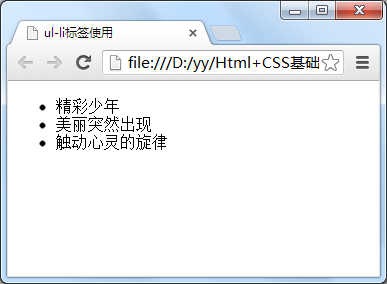
<li>精彩少年</li>

<li>美丽突然出现</li>

<li>触动心灵的旋律</li>

</ul>

ul-li在网页中显示的默认样式一般为：每项li前都自带一个圆点，如下图所示：



### （20）、使用ol，添加图书销售排行榜

如果想在网页中展示有前后顺序的信息列表，怎么办呢？如，当当网上的书籍热卖排行榜，如下图所示。这类信息展示就可以使用<ol>标签来制作有序列表来展示。



语法：

<ol>

<li>信息</li>

<li>信息</li>

......

</ol>

举例：

下面是一个热点课程下载排行榜：

<ol>

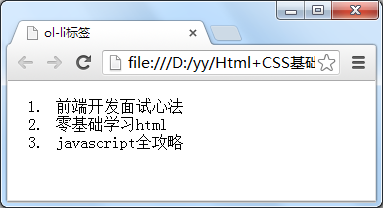
<li>前端开发面试心法 </li>

<li>零基础学习html</li>

<li>JavaScript全攻略</li>

</ol>

<ol>在网页中显示的默认样式一般为：每项<li>前都自带一个序号，序号默认从1开始，如下图所示：



### （21）、div相关的知识

1-1 认识div在排版中的作用：

在网页制作过程过中，可以把一些独立的逻辑部分划分出来，放在一个<div>标签中，这个<div>标签的作用就相当于一个容器。

语法：

<div>…</div>

确定逻辑部分：

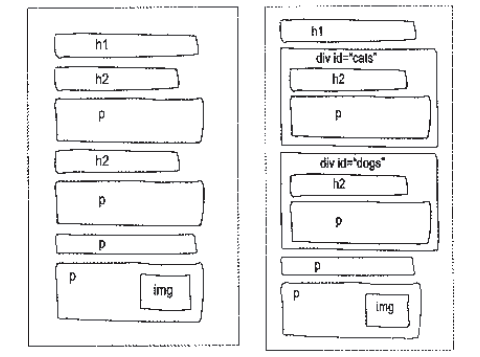
什么是逻辑部分？它是页面上相互关联的一组元素。如网页中的独立的栏目版块，就是一个典型的逻辑部分。如下图所示：图中用红色边框标出的部分就是一个逻辑部分，就可以使用<div>标签作为容器。



1-2 给div命名，使逻辑更加清晰

在上一小节中，我们把一些标签放进<div>里，划分出一个独立的逻辑部分。为了使逻辑更加清晰，我们可以为这一个独立的逻辑部分设置一个名称，用id属性来为<div>提供唯一的名称，这个就像我们每个人都有一个身份证号，这个身份证号是唯一标识我们的身份的，也是必须唯一的。

如下两图进行比较，如果设计师把两个图给你，哪个图你看上去能更快的理解呢？是不是右边的那幅图呢。



语法：

<div id="版块名称">…</div>

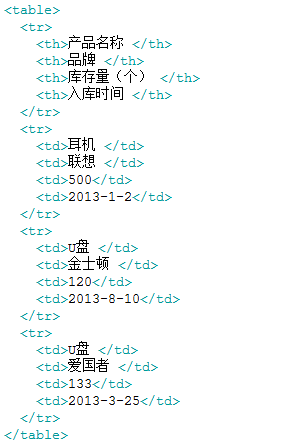
### （22）、table标签

1-1如何写出一个网页表格

有时候我们需要在网页上展示一些数据，如某公司想在网页上展示公司的库存清单。如下表：



想在网页上展示上述表格效果可以使用以下代码：



创建表格的四个元素：

table、tbody、tr、th、td

1、<table>…</table>：整个表格以<table>标记开始、</table>标记结束。

2、<tbody>…</tbody>：如果不加<thead><tbody><tfooter> , table表格加载完后才显示。加上这些表格结构， tbody包含行的内容下载完优先显示，不必等待表格结束后在显示，同时如果表格很长，用tbody分段，可以一部分一部分地显示。（通俗理解table 可以按结构一块块的显示，不在等整个表格加载完后显示。）

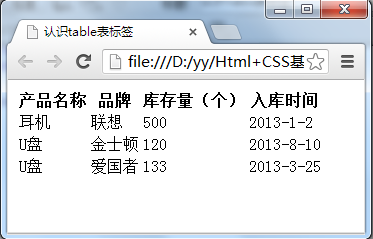
3、<tr>…</tr>：表格的一行，所以有几对tr 表格就有几行。

4、<td>…</td>：表格数据的一个单元格，一行中包含几对<td>...</td>，说明一行中就有几列。

5、<th>…</th>：表格的头部的一个单元格，表格表头。

6、表格中列的个数，取决于一行中数据单元格的个数。

上述代码在浏览器中显示的默认的样式为：



总结：

1、table表格在没有添加css样式之前，在浏览器中显示是没有表格线的

2、表头，也就是th标签中的文本默认为粗体并且居中显示

1-2用css样式，为表格加入边框

Table 表格在没有添加 css 样式之前，是没有边框的。这样不便于我们后期合并单元格知识点的讲解，所以在这一节中我们为表格添加一些样式，为它添加边框。

使用如下代码：

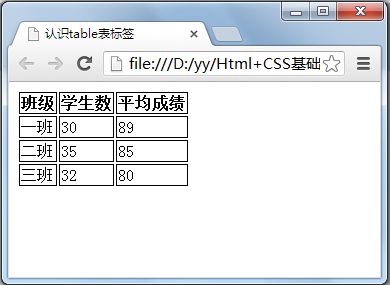
<style type="text/css">

table tr td,th{border:1px solid #000;}

</style>

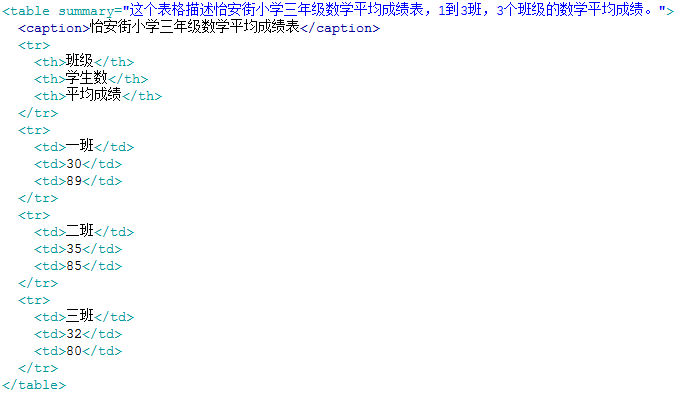
上述代码是用 css 样式代码（后面章节会详细讲解），为th，td单元格添加粗细为一个像素的黑色边框。

结果窗口显示出结果样式：



1-3为表格添加标题caption和摘要summary

表格还是需要添加一些标签进行优化，可以添加标题和摘要。代码如下：



**摘要**

摘要的内容是不会在浏览器中显示出来的。它的作用是增加表格的可读性(语义化)，使搜索引擎更好的读懂表格内容，还可以使屏幕阅读器更好的帮助特殊用户读取表格内容。

语法：<table summary="表格简介文本">

标题

用以描述表格内容，标题的显示位置：表格上方。

语法：

<table>

<caption>标题文本</caption>

<tr>

<td>…</td>

<td>…</td>

…

</tr>

…

</table>

### （23）、使用<a>标签，链接到另一个页面

1-1 链接到的网页在当前浏览器窗口打开

使用<a>标签可实现超链接，它在网页制作中可以说是无处不在，只要有链接的地方，就会有这个标签。

语法：

<a href="目标网址" title="鼠标滑过显示的文本">链接显示的文本</a>

例如：

<a href="http://www.imooc.com" title="点击进入慕课网">click here!</a>

上面例子作用是单击click here!文字，网页链接到http://www.imooc.com这个网页。

title属性的作用，鼠标滑过链接文字时会显示这个属性的文本内容。这个属性在实际网页开发中作用很大，主要方便搜索引擎了解链接地址的内容（语义化更友好）。

注意：还有一个有趣的现象不知道小伙伴们发现了没有，只要为文本加入a标签后，文字的颜色就会自动变为蓝色（被点击过的文本颜色为紫色），颜色很难看吧，不过没有关系后面我们学习了css样子就可以设置过来（a{color:#000}),后面会详细讲解。

1-2 链接到的网页在新建的浏览器窗口中打开

<a>标签在默认情况下，链接的网页是在当前浏览器窗口中打开，有时我们需要在新的浏览器窗口中打开。

如下代码：

<a href="目标网址" target="\_blank">click here!</a>

1-3使用mailto在网页中链接Email地址

<a>标签还有一个作用是可以链接Email地址，使用mailto能让访问者便捷向网站管理者发送电子邮件。我们还可以利用mailto做许多其它事情。下面一一进行讲解，请看详细图示：



注意：如果mailto后面同时有多个参数的话，第一个参数必须以“?”开头，后面的参数每一个都以“&”分隔。

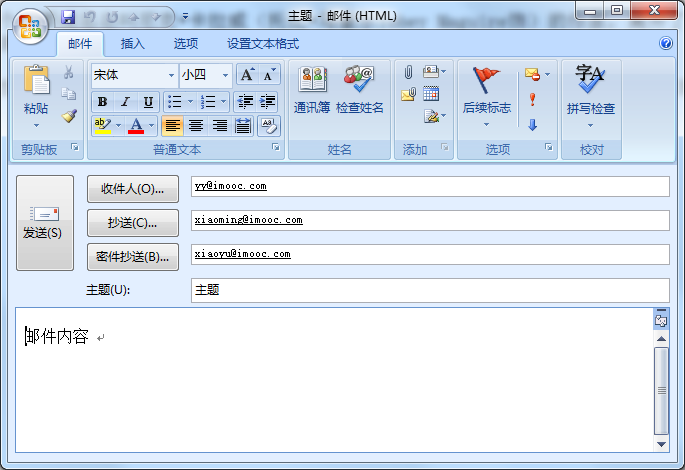
下面是一个完整的实例:

http://img.mukewang.com/52da52200001e00e07930061.jpg

在浏览器中显示的结果：

发送

点击链接会打开电子邮件应用，并自动填写收件人等设置好的信息，如下图：



### （24）、认识<img>标签，为网页插入图片

在网页的制作中为使网页炫丽美观，肯定是缺少不了图片，可以使用<img>标签来插入图片。

语法：

<img src="图片地址" alt="下载失败时的替换文本" title = "提示文本">

举例：

<img src = "myimage.gif" alt = "My Image" title = "My Image" />

讲解：

1、src：标识图像的位置；

2、alt：指定图像的描述性文本，当图像不可见时（下载不成功时），可看到该属性指定的文本；

3、title：提供在图像可见时对图像的描述(鼠标滑过图片时显示的文本)；

4、图像可以是GIF，PNG，JPEG格式的图像文件。

### （25）、使用表单标签，与用户交互

网站怎样与用户进行交互？答案是使用HTML表单(form)。表单是可以把浏览者输入的数据传送到服务器端，这样服务器端程序就可以处理表单传过来的数据。

语法：

<form method="传送方式" action="服务器文件">

讲解：

1.<form> ：<form>标签是成对出现的，以<form>开始，以</form>结束。

2.action ：浏览者输入的数据被传送到的地方,比如一个PHP页面(save.php)。

3.method ： 数据传送的方式（get/post）。

<form method="post" action="save.php">

<label for="username">用户名:</label>

<input type="text" name="username" />

<label for="pass">密码:</label>

<input type="password" name="pass" />

</form>

注意:

1、所有表单控件（文本框、文本域、按钮、单选框、复选框等）都必须放在 <form></form> 标签之间（否则用户输入的信息可提交不到服务器上哦！）。

2、method : post/get 的区别这一部分内容属于后端程序员考虑的问题。

文本输入框、密码输入框

当用户要在表单中键入字母、数字等内容时，就会用到文本输入框。文本框也可以转化为密码输入框。

语法：

<form>

<input type="text/password" name="名称" value="文本" />

</form>

1、type：

当type="text"时，输入框为文本输入框;

当type="password"时, 输入框为密码输入框。

2、name：为文本框命名，以备后台程序ASP 、PHP使用。

3、value：为文本输入框设置默认值。(一般起到提示作用)

举例：

<form>

姓名：

<input type="text" name="myName">

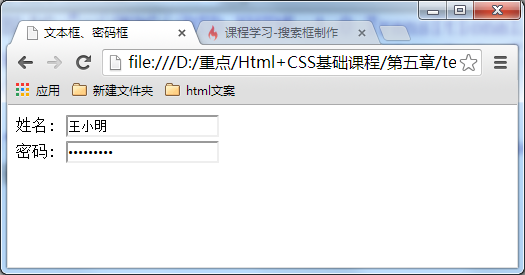
<br/>

密码：

<input type="password" name="pass">

</form>

在浏览器中显示的结果：



### （26）、文本域，支持多行文本输入

当用户需要在表单中输入大段文字时，需要用到文本输入域。

语法：

<textarea rows="行数" cols="列数">文本</textarea>

1、<textarea>标签是成对出现的，以<textarea>开始，以</textarea>结束。

2、cols ：多行输入域的列数。

3、rows ：多行输入域的行数。

4、在<textarea></textarea>标签之间可以输入默认值。

举例：

<form method="post" action="save.php">

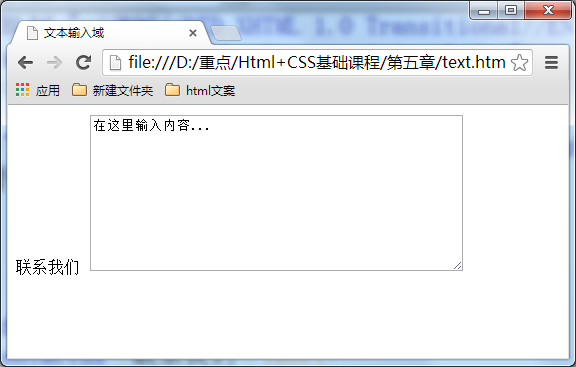
<label>联系我们</label>

<textarea cols="50" rows="10" >在这里输入内容...</textarea>

</form>

注意：代码中的<label>标签在本章5-9中讲解。

在浏览器中显示结果：



注意这两个属性可用css样式的width和height来代替：col用width、row用height来代替。（这两个css样式在以后的章节会讲解）

### （27）、使用单选框、复选框，让用户选择

在使用表单设计调查表时，为了减少用户的操作，使用选择框是一个好主意，html中有两种选择框，即单选框和复选框，两者的区别是单选框中的选项用户只能选择一项，而复选框中用户可以任意选择多项，甚至全选。请看下面的例子:

语法：

<input type="radio" value="值" name="名称" checked="checked"/>

1、type:

当 type="radio" 时，控件为单选框

当 type="checkbox" 时，控件为复选框

2、value：提交数据到服务器的值（后台程序PHP使用）

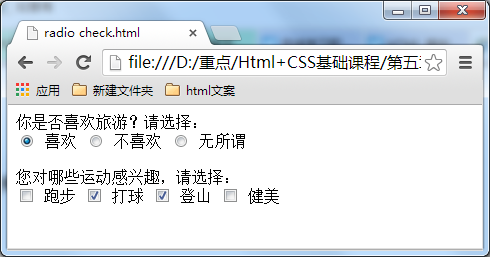
3、name：为控件命名，以备后台程序 ASP、PHP 使用

4、checked：当设置 checked="checked" 时，该选项被默认选中

如下面代码：



在浏览器中显示的结果：

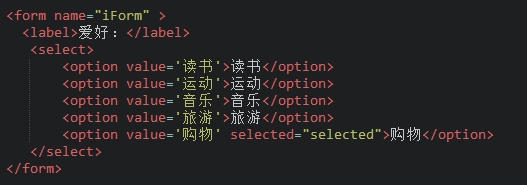


注意:**同一组的单选按钮，name 取值一定要一致**，比如上面例子为同一个名称“radioLove”，这样同一组的单选按钮才可以起到单选的作用。

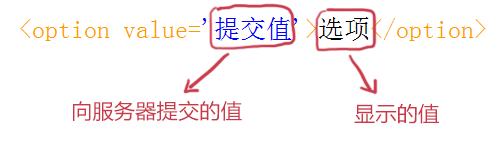
### （28）、使用下拉列表框，节省空间

拉列表在网页中也常会用到，它可以有效的节省网页空间。既可以单选、又可以多选。

如下代码（**单选**）：



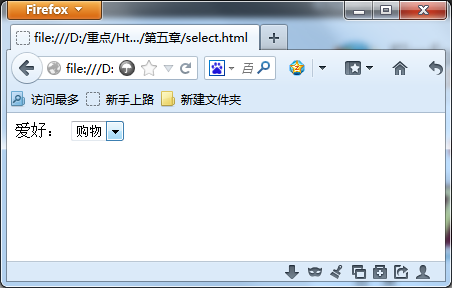
1、value：



2、selected="selected"：

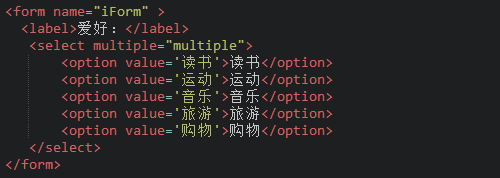
设置selected="selected"属性，则该选项就被默认选中。

在浏览器中显示的结果：

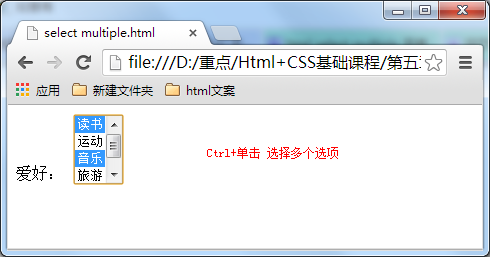


**使用下拉列表框进行多选：**

下拉列表也可以进行多选操作，在<select>标签中设置multiple="multiple"属性，就可以实现多选功能，在 windows 操作系统下，进行多选时按下Ctrl键同时进行单击（在 Mac下使用 Command +单击），可以选择多个选项。如下代码：



在浏览器中显示的结果：



### （29）、使用提交按钮，提交数据

在表单中有两种按钮可以使用，分别为：提交按钮、重置。这一小节讲解提交按钮：当用户需要提交表单信息到服务器时，需要用到提交按钮。

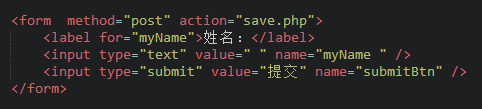
语法：

<input type="submit" value="提交">

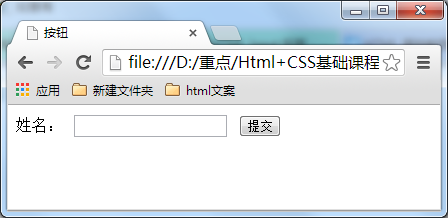
type：只有当type值设置为submit时，按钮才有提交作用

value：按钮上显示的文字

举例：



在浏览器中显示的结果：



### （30）、使用重置按钮，重置表单信息

当用户需要重置表单信息到初始时的状态时，比如用户输入“用户名”后，发现书写有误，可以使用重置按钮使输入框恢复到初始状态。只需要把type设置为"reset"就可以。

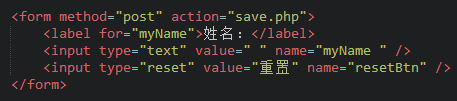
语法：

<input type="reset" value="重置">

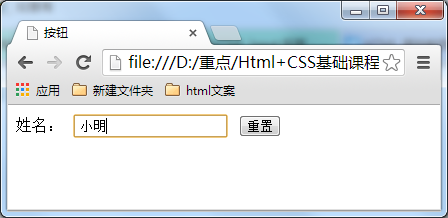
type：只有当type值设置为reset时，按钮才有重置作用

value：按钮上显示的文字

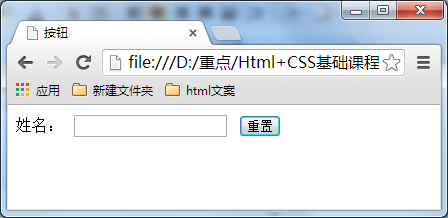
举例：



在浏览器中显示的结果：



输入账号



单击重置按钮

### （32）、form表单中的label标签

小伙伴们，你们在前面学习表单各种控件的时候，有没有发现一个标签--label，这一小节就来揭晓它的作用。

label标签不会向用户呈现任何特殊效果，它的作用是为鼠标用户改进了可用性。如果你在 label 标签内点击文本，就会触发此控件。就是说，当用户单击选中该label标签时，浏览器就会自动将焦点转到和标签相关的表单控件上（就自动选中和该label标签相关连的表单控件上）。

语法：

<label for="控件id名称">

注意：标签的 for 属性中的值应当与相关控件的 id 属性值一定要相同。

例子：

<form>

<label for="male">男</label>

<input type="radio" name="gender" id="male" />

<br />

<label for="female">女</label>

<input type="radio" name="gender" id="female" />

<label for="email">输入你的邮箱地址</label>

<input type="email" id="email" placeholder="Enter email">

</form>

## 2、慕课网html入门学习笔记

### （1）、认识CSS样式

CSS全称为“层叠样式表 (Cascading Style Sheets)”，它主要是用于定义HTML内容在浏览器内的显示样式，如文字大小、颜色、字体加粗等。

如下列代码：

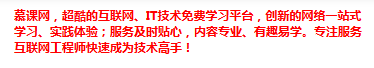
p{

font-size:12px;

color:red;

font-weight:bold; /\*字体加粗\*/

}



使用CSS样式的一个好处是通过定义某个样式，可以让不同网页位置的文字有着统一的字体、字号或者颜色等。

### （2）、CSS样式的优势

为什么使用css样式来设置网页的外观样式呢？右边编辑器是一段文字，我们想把“超酷的互联网”、“服务及时贴心”、“有趣易学”这三个短语的文本颜色设置为红色，这时就 可以通过设置样式来设置，而且只需要编写一条css样式语句。

第一步：把这三个短语用<span></span>括起来。

第二步：写入下列代码：（见右边代码编辑器7-8行）

span{

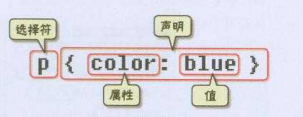
color:red;

}

观察结果窗口文字的颜色是否变为红色了。

### （3）、CSS代码语法

css 样式由选择符和声明组成，而声明又由属性和值组成，如下图所示：



选择符：又称选择器，指明网页中要应用样式规则的元素，如本例中是网页中所有的段（p）的文字将变成蓝色，而其他的元素（如ol）不会受到影响。

声明：在英文大括号“｛｝”中的的就是声明，属性和值之间用英文冒号“：”分隔。当有多条声明时，中间可以英文分号“;”分隔，如下所示：

p{font-size:12px;color:red;}

注意：

1、最后一条声明可以没有分号，但是为了以后修改方便，一般也加上分号。

2、为了使用样式更加容易阅读，可以将每条代码写在一个新行内，如下所示：

p{

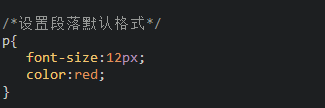
font-size:12px;

color:red;

}

### （4）、CSS注释代码

就像在Html的注释一样，在CSS中也有注释语句：用/\*注释语句\*/来标明（Html中使用<!--注释语句-->)。就像下面代码：



### （5）、CSS样式

CSS样式可以写在哪些地方呢？从CSS 样式代码插入的形式来看基本可以分为以下3种：内联式、嵌入式和外部式三种。这一小节先来讲解内联式。

**内联式：**

内联式css样式表就是把css代码直接写在现有的HTML标签中，如下面代码：

<p style="color:red">这里文字是红色。</p>

注意要写在元素的开始标签里，下面这种写法是错误的：

<p>这里文字是红色。</p style="color:red">

并且css样式代码要写在style=""双引号中，如果有多条css样式代码设置可以写在一起，中间用分号隔开。如下代码：

<p style="color:red;font-size:12px">这里文字是红色。</p>

**嵌入式：**

嵌入式css样式，就是可以把css样式代码写在<style type="text/css"></style>标签之间。如下面代码实现把三个<span>标签中的文字设置为红色：

<style type="text/css">

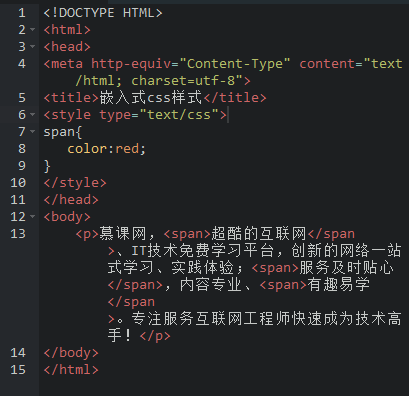
span{

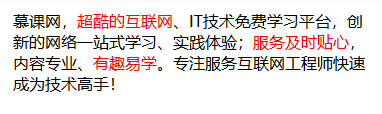
color:red;

}

</style>

嵌入式css样式必须写在<style></style>之间，并且一般情况下嵌入式css样式写在<head></head>之间。如右边编辑器中的代码。





**外部式（外联式）**

外部式css样式(也可称为外联式)就是把css代码写一个单独的外部文件中，这个css样式文件以“.css”为扩展名，在<head>内（不是在<style>标签内）使用<link>标签将css样式文件链接到HTML文件内，如下面代码：

<link href="base.css" rel="stylesheet" type="text/css" />

注意：

1、css样式文件名称以有意义的英文字母命名，如 main.css。

2、rel="stylesheet" type="text/css" 是固定写法不可修改。

3、<link>标签位置一般写在<head>标签之内。

### （6）、三种方法的优先级

如果有一种情况：对于同一个元素我们同时用了三种方法设置css样式，那么哪种方法真正有效呢

比如：

1、使用内联式CSS设置“超酷的互联网”文字为粉色。

2、然后使用嵌入式CSS来设置文字为红色。

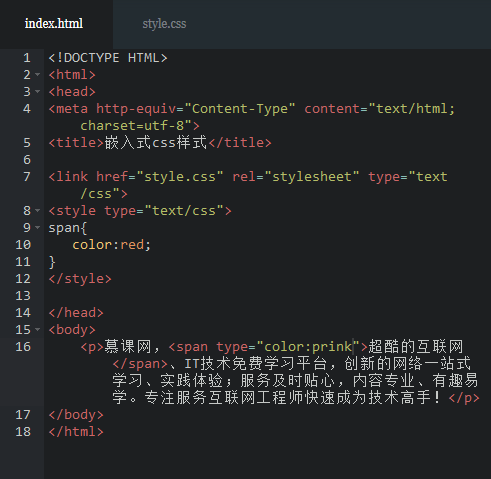
3、最后又使用外部式设置文字为蓝色（style.css文件中设置）。

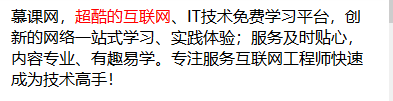
但最终你可以观察到“超酷的互联网”这个短词的文本被设置为了粉色。因为这三种样式是有优先级的，记住他们的优先级：内联式 > 嵌入式 > 外部式

但是嵌入式>外部式有一个前提：嵌入式css样式的位置一定在外部式的后面。如右代码编辑器就是这样，<link href="style.css" ...>代码在<style type="text/css">...</style>代码的前面（实际开发中也是这么写的）。总结谁在后面谁的优先级高。

其实总结来说，就是--就近原则（离被设置元素越近优先级别越高）。

但注意上面所总结的优先级是有一个前提：内联式、嵌入式、外部式样式表中css样式是在的相同权值的情况下，什么是权值呢？在后面的会讲解到。





### （7）、选择器

**1-1什么式选择器**

每一条css样式声明（定义）由两部分组成，形式如下：

选择器{

样式;

}

在{}之前的部分就是“选择器”，“选择器”指明了{}中的“样式”的作用对象，也就是“样式”作用于网页中的哪些元素。

**1-2 标签选择器**

标签选择器其实就是html代码中的标签。比如<html>、<body>、<h1>、<p>、<img>。例如下面代码：

p{font-size:12px;line-height:1.6em;}

上面的css样式代码的作用：为p标签设置12px字号，行间距设置1.6em的样式。

例如：

为标题“勇气”添加样式，把h1标签默认的粗体样式去掉并将字体颜色设置为红色

1、在右边编辑器的第7行，输入h1{

2、在右边编辑器的第8行，输入font-weight:normal;

3、在右边编辑器的第9行，输入color:red;

4、在右边编辑器的第10行，输入}





**1-3类选择器**

类选择器在css样式编码中是最常用到的，如右侧代码编辑器中的代码:可以实现为“胆小如鼠”、“勇气”字体设置为红色。

语法：

. 类选器名称{css样式代码;}

注意：

1、英文圆点开头

2、其中类选器名称可以任意起名（但不要起中文噢）

使用方法：

第一步：使用合适的标签把要修饰的内容标记起来，如下：

<span>胆小如鼠</span>

第二步：使用class="类选择器名称"为标签设置一个类，如下：

<span class="stress">胆小如鼠</span>

第三步：设置类选器css样式，如下：

.stress{color:red;}/\*类前面要加入一个英文圆点\*/

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>认识html标签</title>

<style type="text/css">

.stress{

color:red;

}

.setGreen{

color:green;

}

</style>

</head>

<body>

<h1 class="stress">勇气</h1>

<p class="setGreen">三年级时，我还是一个<span class="stress">胆小如鼠</span>的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个<span class="stress">勇气</span>来回答老师提出的问题。学校举办的活动我也没勇气参加。</p>

<p>到了三年级下学期时，我们班上了一节<span class="setGreen">公开课</span>，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。</p>

<img src="http://img.mukewang.com/52b4113500018cf102000200.jpg" >

</body>

</html>



**1-4 ID选择器**

在很多方面，ID选择器都类似于类选择符，但也有一些重要的区别：

1、为标签设置id="ID名称"，而不是class="类名称"。

2、ID选择符的前面是井号（#）号，而不是英文圆点（.）。

注意：简单来说id供前后台交互使用，class用于前端样式Css使用，但是有个别的两个都要，比如单选框

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>认识html标签</title>

<style type="text/css">

#stress{

color:red;

}

#setGreen{

color:green;

}

</style>

</head>

<body>

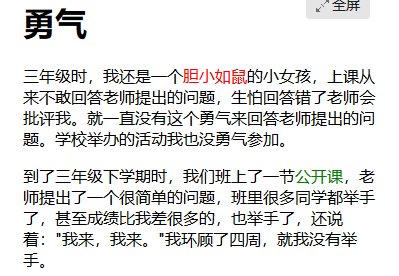
<h1>勇气</h1>

<p>三年级时，我还是一个<span id="stress">胆小如鼠</span>的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。</p>

<p>到了三年级下学期时，我们班上了一节<span id="setGreen">公开课</span>，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。</p>

</body>

</html>



**1-5 类和ID选择器的区别**

学习了类选择器和ID选择器，我们会发现他们之间有很多的相似处，是不是两者可以通用呢？我们不要着急先来总结一下他们的相同点和不同点：

相同点：可以应用于任何元素

不同点：

1、ID选择器只能在文档中使用一次。与类选择器不同，在一个HTML文档中，ID选择器只能使用一次，而且仅一次。而类选择器可以使用多次。

下面代码是正确的：

<p>三年级时，我还是一个<span class="stress">胆小如鼠</span>的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个<span class="stress">勇气</span>来回答老师提出的问题。</p>

而下面代码是错误的：

<p>三年级时，我还是一个<span id="stress">胆小如鼠</span>的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个<span id="stress">勇气</span>来回答老师提出的问题。</p>

2、可以使用类选择器词列表方法为一个元素同时设置多个样式。我们可以为一个元素同时设多个样式，但只可以用类选择器的方法实现，ID选择器是不可以的（不能使用 ID 词列表）。

下面的代码是正确的(完整代码见右侧代码编辑器)

.stress{

color:red;

}

.bigsize{

font-size:25px;

}

<p>到了<span class="stress bigsize">三年级</span>下学期时，我们班上了一节公开课...</p>

上面代码的作用是为“三年级”三个文字设置文本颜色为红色并且字号为25px。

下面的代码是不正确的(完整代码见右侧代码编辑器)

#stressid{

color:red;

}

#bigsizeid{

font-size:25px;

}

<p>到了<span id="stressid bigsizeid">三年级</span>下学期时，我们班上了一节公开课...</p>

上面代码不可以实现为“三年级”三个文字设置文本颜色为红色并且字号为25px的作用。

**1-6 子选择器**

还有一个比较有用的选择器子选择器，即大于符号(>),用于选择指定标签元素的第一代子元素。如右侧代码编辑器中的代码：

.food>li{border:1px solid red;}

这行代码会使class名为food下的子元素li（水果、蔬菜）加入红色实线边框。

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>子选择符</title>

<style type="text/css">

.food>li{border:1px solid red;}/\*添加边框样式（粗细为1px， 颜色为红色的实线）\*/

</style>

</head>

<body>

<p class="first">三年级时，<span>我还是一个<span>胆小如鼠</span>的小女孩</span>，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。</p>

<h1>食物</h1>

<ul class="food">

<li>水果

<ul>

<li>香蕉</li>

<li>苹果</li>

<li>梨</li>

</ul>

</li>

<li>蔬菜

<ul>

<li>白菜</li>

<li>油菜</li>

<li>卷心菜</li>

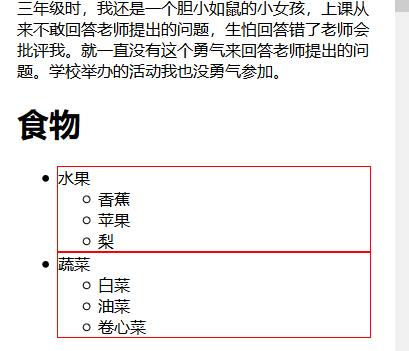
</ul>

</li>

</ul>

</body>

</html>



**1-7 包含(后代)选择器**

包含选择器，即加入空格,用于选择指定标签元素下的后辈元素。如右侧代码编辑器中的代码：

.first span{color:red;}

这行代码会使第一段文字内容中的“胆小如鼠”字体颜色变为红色。

请注意这个选择器与子选择器的区别，子选择器（child selector）仅是指它的直接后代，或者你可以理解为作用于子元素的第一代后代。而后代选择器是作用于所有子后代元素。后代选择器通过空格来进行选择，而子选择器是通过“>”进行选择。

总结：>作用于元素的第一代后代，空格作用于元素的所有后代。

**1-8 通用选择器**

通用选择器是功能最强大的选择器，它使用一个（\*）号指定，它的作用是匹配html中所有标签元素，如下使用下面代码使用html中任意标签元素字体颜色全部设置为红色：

\* {color:red;}

<!DOCTYPE HTML>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>\* 选择符</title>

<style type="text/css">

\* {color:red;}

</style>

</head>

<body>

<h1>勇气</h1>

<p>三年级时，我还是一个胆小如鼠的小女孩，上课从来不敢回答老师提出的问题，生怕回答错了老师会批评我。就一直没有这个勇气来回答老师提出的问题。学校举办的活动我也没勇气参加。</p>

<p>到了三年级下学期时，我们班上了一节公开课，老师提出了一个很简单的问题，班里很多同学都举手了，甚至成绩比我差很多的，也举手了，还说着："我来，我来。"我环顾了四周，就我没有举手。</p>

<img src="http://img.mukewang.com/52b4113500018cf102000200.jpg" >

</body>

</html>

**1-9 伪类选择符**

为什么叫做伪类选择符，它允许给html不存在的标签（标签的某种状态）设置样式，比如说我们给html中一个标签元素的鼠标滑过的状态来设置字体颜色：

a:hover{color:red;}

上面一行代码就是为 a 标签鼠标滑过的状态设置字体颜色变红。这样就会使第一段文字内容中的“胆小如鼠”文字加入鼠标滑过字体颜色变为红色特效。

关于伪选择符：

关于伪类选择符，到目前为止，可以兼容所有浏览器的“伪类选择符”就是 a 标签上使用 :hover 了（其实伪类选择符还有很多，尤其是 css3 中，但是因为不能兼容所有浏览器，本教程只是讲了这一种最常用的）。其实 :hover 可以放在任意的标签上，比如说 p:hover，但是它们的兼容性也是很不好的，所以现在比较常用的还是 a:hover 的组合。

**1-10 分组选择符**

当你想为html中多个标签元素设置同一个样式时，可以使用分组选择符（，），如下代码为右侧代码编辑器中的h1、span标签同时设置字体颜色为红色：

h1,span{color:red;}

它相当于下面两行代码：

h1{color:red;}

span{color:red;}

### （8）、继承

CSS的某些样式是具有继承性的，那么什么是继承呢？继承是一种规则，它允许样式不仅应用于某个特定html标签元素，而且应用于其后代。比如下面代码：如某种颜色应用于p标签，这个颜色设置不仅应用p标签，还应用于p标签中的所有子元素文本，这里子元素为span标签。

p{color:red;}

<p>三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

可见右侧结果窗口中p中的文本与span中的文本都设置为了红色。但注意有一些css样式是不具有继承性的。如border:1px solid red;

p{border:1px solid red;}

<p>三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

在上面例子中它代码的作用只是给p标签设置了边框为1像素、红色、实心边框线，而对于子元素span是没用起到作用的。

### （9）、特殊性

有的时候我们为同一个元素设置了不同的CSS样式代码，那么元素会启用哪一个CSS样式呢？下面我们一起来看一下代码：

p{color:red;}

.first{color:green;}

<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

p和.first都匹配到了p这个标签上，那么会显示哪种颜色呢？green是正确的颜色，那么为什么呢？是因为浏览器是根据权值来判断使用哪种css样式的，权值高的就使用哪种css样式。

下面是权值的规则：

标签的权值为1，类选择符的权值为10，ID选择符的权值最高为100。例如下面的代码：

p{color:red;} /\*权值为1\*/

p span{color:green;} /\*权值为1+1=2\*/

.warning{color:white;} /\*权值为10\*/

p span.warning{color:purple;} /\*权值为1+1+10=12\*/

#footer .note p{color:yellow;} /\*权值为100+10+1=111\*/

注意：还有一个权值比较特殊--继承也有权值但很低，有的文献提出它只有0.1，所以可以理解为继承的权值最低。

### （10）、层叠

我们来思考一个问题：如果在html文件中对于**同一个元素**可以有多个css样式存在并且这多个css样式具有**相同权重值**怎么办？好，这一小节中的层叠帮你解决这个问题。

层叠就是在html文件中对于同一个元素可以有多个css样式存在，当有相同权重的样式存在时，会根据这些css样式的前后顺序来决定，处于最后面的css样式会被应用。

如下面代码:

p{color:red;}

p{color:green;}

<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

最后 p 中的文本会设置为green，这个层叠很好理解，理解为后面的样式会覆盖前面的样式。

所以前面的css样式优先级就不难理解了：

**内联样式表（标签内部）> 嵌入样式表（当前文件中）> 外部样式表（外部文件中）。**

### （11）、重要性

我们在做网页代码的时，有些特殊的情况需要为某些样式设置具有最高权值，怎么办？这时候我们可以使用!important来解决。

如下代码：

p{color:red!important;}

p{color:green;}

<p class="first">三年级时，我还是一个<span>胆小如鼠</span>的小女孩。</p>

这时 p 段落中的文本会显示的red红色。

注意：!important要写在分号的前面

这里注意当网页制作者不设置css样式时，浏览器会按照自己的一套样式来显示网页。并且用户也可以在浏览器中设置自己习惯的样式，比如有的用户习惯把字号设置为大一些，使其查看网页的文本更加清楚。这时注意样式优先级为：浏览器默认的样式 < 网页制作者样式 < 用户自己设置的样式，**但记住!important优先级样式是个例外，权值高于用户自己设置的样式。**

### （12）文字排版

我们可以使用css样式为网页中的文字设置字体、字号、颜色等样式属性

**1）、字体**

为网页中的文字设置字体为宋体：

body{font-family:"宋体";}

这里注意不要设置不常用的字体，因为如果用户本地电脑上如果没有安装你设置的字体，就会显示浏览器默认的字体。（因为用户是否可以看到你设置的字体样式取决于用户本地电脑上是否安装你设置的字体。）

现在一般网页喜欢设置“微软雅黑”，如下代码：

body{font-family:"Microsoft Yahei";}

或

body{font-family:"微软雅黑";}

注意：第一种方法比第二种方法兼容性更好一些。

因为这种字体即美观又可以在客户端安全的显示出来（用户本地一般都是默认安装的）。

**2）、-字号、颜色**

可以使用下面代码设置网页中文字的字号为12像素，并把字体颜色设置为#666(灰色)：

body{font-size:12px;color:#666}

**3）、粗体**

可以使用下面代码实现设置文字以粗体样式显示出来：

p span{font-weight:bold;}

**4）、斜体**

p a{font-style:italic;}

<p>三年级时，我还是一个<a>胆小如鼠</a>的小女孩。</p>

**5）、下划线**

有些情况下想为文字设置为下划线样式，这样可以在视觉上强调文字，可以使用下面代码来实现： p a{text-decoration:underline;}

<p>三年级时，我还是一个<a>胆小如鼠</a>的小女孩。</p>

**6）、-删除线**

删除线使用下面代码就可以实现：

.oldPrice{text-decoration:line-through;}

### （13）、段落排版

**1）、缩进**

实现方法：

p{text-indent:2em;}

<p>1922年的春天，一个想要成名名叫尼克卡拉威（托比?马奎尔Tobey Maguire 饰）的作家，离开了美国中西部，来到了纽约。那是一个道德感渐失，爵士乐流行，走私为王，股票飞涨的时代。为了追寻他的美国梦，他搬入纽约附近一海湾居住。</p>

注意：2em的意思就是文字的2倍大小。

**2）、行间距（行高）**

如下代码实现设置段落行间距为1.5倍：

**p{line-height:1.5em;}**

<p>菲茨杰拉德，二十世纪美国文学巨擘之一，兼具作家和编剧双重身份。他以诗人的敏感和戏剧家的想象为"爵士乐时代"吟唱华丽挽歌，其诗人和梦想家的气质亦为那个奢靡年代的不二注解。</p>

**3）、中文字间距、字母间距**

中文字间隔、字母间隔设置：

如果想在网页排版中设置文字间隔或者字母间隔就可以使用 letter-spacing 来实现，如下面代码：

h1{

letter-spacing:50px;

}

...

<h1>了不起的盖茨比</h1>

注意：这个样式使用在英文单词时，是设置字母与字母之间的间距。

单词间距设置：

如果我想设置英文单词之间的间距呢？可以使用 word-spacing 来实现。如下代码：

h1{

word-spacing:50px;

}

...

<h1>welcome to imooc!</h1>

**4）、对齐**

想为块状元素中的文本、图片设置居中样式吗？可以使用text-align样式代码，如下代码可实现文本居中显示

h1{

text-align:center;

}

<h1>了不起的盖茨比</h1>

同样可以设置居左：

h1{

text-align:left;

}

<h1>了不起的盖茨比</h1>

还可以设置居右：

h1{

text-align:right;

}

<h1>了不起的盖茨比</h1>

### （14）、元素分类

**1）、元素分类**

在讲解CSS布局之前，我们需要提前知道一些知识，在CSS中，html中的标签元素大体被分为三种不同的类型：块状元素、内联元素(又叫行内元素)和内联块状元素。

常用的块状元素有：

<div>、<p>、<h1>...<h6>、<ol>、<ul>、<dl>、<table>、<address>、<blockquote> 、<form>

常用的内联元素有：

<a>、<span>、<br>、<i>、<em>、<strong>、<label>、<q>、<var>、<cite>、<code>

常用的内联块状元素有：

<img>、<input>

**2）、块级元素**

什么是块级元素？在html中<div>、 <p>、<h1>、<form>、<ul> 和 <li>就是块级元素。设置display:block就是将元素显示为块级元素。如下代码就是将内联元素a转换为块状元素，从而使a元素具有块状元素特点。

a{display:block;}

块级元素特点：

1、每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）

2、元素的高度、宽度、行高以及顶和底边距都可设置。

3、元素宽度在不设置的情况下，是它本身父容器的100%（和父元素的宽度一致），除非设定一个宽度。

**3）、内联元素**

在html中，<span>、<a>、<label>、 <strong> 和<em>就是典型的内联元素（行内元素）（inline）元素。当然块状元素也可以通过代码display:inline将元素设置为内联元素。如下代码就是将块状元素div转换为内联元素，从而使 div 元素具有内联元素特点。

div{

display:inline;

}

......

<div>我要变成内联元素</div>

内联元素特点：

1、和其他元素都在一行上；

2、元素的高度、宽度及顶部和底部边距不可设置；

3、元素的宽度就是它包含的文字或图片的宽度，不可改变。

**4）、内联块状元素**

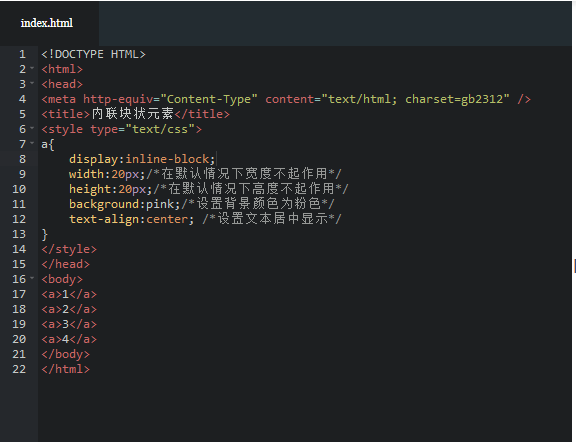
内联块状元素（inline-block）就是同时具备内联元素、块状元素的特点，代码display:inline-block就是将元素设置为内联块状元素。(css2.1新增)，<img>、<input>标签就是这种内联块状标签。

inline-block 元素特点：

1、和其他元素都在一行上；

2、元素的高度、宽度、行高以及顶和底边距都可设置。

在右边的代码编辑器中大家可以看到我们为 a 元素设置了宽和高，但都没有起到作用，原因是a在默认的时候是内联元素，内联元素是不可以设置宽和高的。1.在右边编辑器的第8行，输入：display:inline-block;





### （15）、盒子模型

概念：

Div ul ol p table 等块元素具有盒子模型的特点：边框（border） 外边距（margin）内边距（padding） 宽度 高度 等。

**1）边框1：**

盒子模型的边框就是围绕着内容及补白的线，这条线你可以设置它的粗细、样式和颜色(边框三个属性)。

如下面代码为 div 来设置边框粗细为 2px、样式为实心的、颜色为红色的边框：

div{

border:2px solid red;

}

上面是 border 代码的缩写形式，可以分开写：

div{

border-width:2px;

border-style:solid;

border-color:red;

}

注意：

1、border-style（边框样式）常见样式有：

dashed（虚线）| dotted（点线）| solid（实线）。

2、border-color（边框颜色）中的颜色可设置为十六进制颜色（也可以直接使用red blue 等），如:

border-color:#888;//前面的井号不要忘掉。

3、border-width（边框宽度）中的宽度也可以设置为：

thin | medium | thick（但不是很常用），最常还是用像素（px）。

**2）边框2**

如果有想为 p 标签单独设置下边框，而其它三边都不设置边框样式怎么办呢？css 样式中允许只为一个方向的边框设置样式：

div{border-bottom:1px solid red;}

同样可以使用下面代码实现其它三边(上、右、左)边框的设置：

border-top:1px solid red;

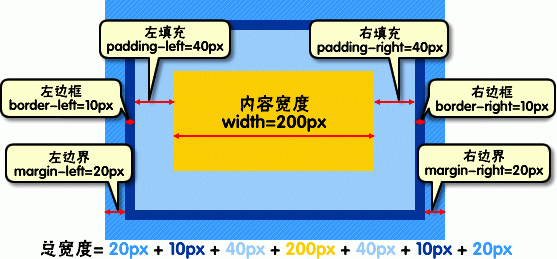
border-right:1px solid red;

border-left:1px solid red;

**3）、宽度和高度**

盒模型宽度和高度和我们平常所说的物体的宽度和高度理解是不一样的，css内定义的宽（width）和高（height），指的是填充以里的内容范围。

因此一个元素实际宽度（盒子的宽度）=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。



元素的高度也是同理。

比如：

css代码：

div{

width:200px;

padding:20px;

border:1px solid red;

margin:10px;

}

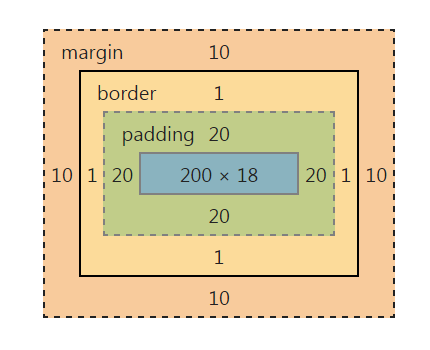
html代码：

<body>

<div>文本内容</div>

</body>

盒子的实际长度为：10px+1px+20px+200px+20px+1px+10px=262px。在chrome浏览器下可查看元素盒模型，如下图：



**4）、填充**

元素内容与边框之间是可以设置距离的，称之为“填充”。填充也可分为上、右、下、左(顺时针)。如下代码：

div{padding:20px 10px 15px 30px;}

顺序一定不要搞混。可以分开写上面代码：

div{

padding-top:20px;

padding-right:10px;

padding-bottom:15px;

padding-left:30px;

}

如果上、右、下、左的填充都为10px;可以这么写

div{padding:10px;}

如果上下填充一样为10px，左右一样为20px，可以这么写：

div{padding:10px 20px;}

**5）、边界**

元素与其它元素之间的距离可以使用边界（margin）来设置。边界也是可分为上、右、下、左。如下代码：

div{margin:20px 10px 15px 30px;}

也可以分开写：

div{

margin-top:20px;

margin-right:10px;

margin-bottom:15px;

margin-left:30px;

}

如果上右下左的边界都为10px;可以这么写：

div{ margin:10px;}

如果上下边界一样为10px，左右一样为20px，可以这么写：

div{ margin:10px 20px;}

总结一下：padding和margin的区别，padding在边框里，margin在边框外。