- HTML英语意思是：Hypertext Marked Language，即超文本标记语言
- 通过HTML可以实现页面之间的跳转
- 通过HTML可以展现多媒体的效果
- 标签　“<标签名字　属性>”
- HTML与XML

  HTML控制显示

  XML存储和交换数据

HTML的基本结构

- 文档头、文档体

- <HTML>
  <HEAD>
  头 部 信 息
  </HEAD>
  <BODY>
  文 档 主 体， 正 文 部 分
  </BODY>
  </HTML>

- 下面是一个最基本的超文本文档的源码：

```
<HTML>
<HEAD>
    <TITLE>一个简单的HTML示例</TITLE>
</HEAD>
<BODY>
 <CENTER>
    <H3>欢迎光临我的主页</H3>
    <BR>
    <HR>
   <FONT SIZE=2>
     这是我第一次做主页，无论怎么样，我都会努力做好！
   </FONT>
  </CENTER>
</BODY>

</HTML>
```

- 单标签

  < 标签名称>　　

- 双标签

  <标签> 内 容</ 标签>　　<EM>强调</EM>

- 标签属性

  < 标签名字 属性1 属性2 属性3 … >

  <HR SIZE=3 ALIGN=LEFT WIDTH="75%">

- 标题
- 换行
- 段落标签<P>
- 水平线段<HR>
- 文字的大小设置
- 文字的字体与样式
- 文字的颜色
- 位置控制
- 综合示例

标题

- HTML中提供了相应的标题标签<Hn>,其中n为标题的等
  HTML总共提供六个等级的标题，n越小，标题字号就越大
- <H1>…</H1> 第一级标题
  <H2>…</H2> 第二级标题
  <H3>…</H3> 第三级标题
  <H4>…</H4> 第四级标题
  <H5>…</H5> 第五级标题
  <H6>…</H6> 第六级标题

- 在HTML语言规范里，每当浏览器窗口被缩小时，浏览器会自动将右边的文字转折至下一行。所以，编写者对于自己需要断行的地方，应加上
标签

段落标签<P>

- 文件段落的开始由<P>来标记，段落的结束由</P>来标记，</P>是可以省略的，因为下一个<P>的开始就意味着上一个<P>的结束

- <P>标签还有一个属性ALING，它用来指名字符显示时的对齐方式，一般值有CENTER、LEFT、RIGHT三种

水平线段<HR>

- 这个标签可以在屏幕上显示一条水平线，用以分割页面中的不同部分。

- <HR>有四个属性：
  size　水平线的宽度
  width 水平线的长，用占屏幕宽度的百分比或象素值来表示
  align 水平线的对齐方式，有LEFT RIGHT CENTER三种
  noshade 线段无阴影属性，为实心线段

文字的大小设置

- 提供设置字号大小的是FONT，FONT有一个属性SIZE，通过指定SIZE属性就能设置字号大小，而SIZE属性的有效值范围为1－7，其中缺省值为3。我们可以SIZE属性值之前加上“＋”、“－”字符，来指定相对于字号初始值的增量或减量。

- HTML4.0提供了定义字体的功能，用FACE属性来完成这个工作。FACE的属性值可以是本机上的任一字体类型，只有对方的电脑中装有相同的字体才可以在他的浏览器中出现你预先设计的风格。

-

文字的字体与样式 (cont.)

- 为了让文字富有变化，或者为了着意强调某一部分，HTML提供了一些标签产生这些效果，现将常用的标签列举如下：
- <B> </B> 粗体 **HTML语言**
- <I> </I> 斜体 *HTML语言*
- <U> </U> 加下划线 <u>HTML语言</u>
- <TT> <TT> 打字机字体 HTML语言
- <BIG> </BIG> 大型字体 HTML语言
- <SMALL> </SMALL> 小型字体 HTML语言
- <BLINK> </BLINK> 闪烁效果 HTML语言
- <EM> </EM> 表示强调，一般为斜体 *HTML语言*
- <STRONG> </STRONG>表示特别强调，一般为粗体 **HTML语言**
- <CITE> </CITE> 用于引证、举例，一般为斜体 *HTML语言*

- 文字颜色设置格式如下：

  …

  这里的颜色值可以是一个十六进制数(用“#”作为前缀)，也可以是以下16种颜色名称

  如：Black = "#000000“ Green = "#008000"

  Red = "#FF0000“ Blue = "#0000FF“

位置控制

- 通过ALIGN属性可以选择文字或图片的对齐方式，LEFT表示向左对齐，RIGHT表示向右对齐，CENTER表示居中。基本语法如下：

  <DIV ALIGN=#>　　#＝left／right／center

- 另外，ALIGN属性也常常用在其它标签中，引起其内容位置的变动。

  如：<P ALIGN=#>
  　　<HR ALIGN=#>　　#＝left／right／center
  <H1 ALIGN=#〉

- 前面我们所讲是单独使用一个标签的方法，在实际的编写中，许多标签和一些属性是结合起来使用的，
  比如：
  <FONT COLOR="#" SIZE=#>文字</FONT>

- 无序号列表
- 序号列表
- 定义性列表

- 无序号列表使用的一对标签是
，每一个列表项前使用<LI>。其结构如下所示：

```
<UL>
        <LI>第一项
        <LI>第二项
        <LI>第三项
</UL>
```

序号列表

- 序号列表和无序号列表的使用方法基本相同，它使用标签<OL></OL>，每一个列表项前使用<LI>。每个项目都有前后顺序之分，多数用数字表示。其结构如下所示：
-
```
<OL>
    <LI>第一项
    <LI>第二项
    <LI>第三项
</OL>
```

定义性列表

- 定义性列表可以用来给每一个列表项再加上一段说明性文字，说明独立于列表项另起一行显示。在应用中，列表项使用标签<DT>标明，说明性文字使用<DD>表示。在定义性列表中，还有一个属性是COMPACT,使用这个属性后，说明文字和列表项将显示在同一行。其结构如下所示：
-
```
<DL>
  <DT>第一项 <DD>叙述第一项的定义
  <DT>第二项 <DD>叙述第二项的定义
  <DT>第三项 <DD>叙述第三项的定义
  </DL>
```

TABLE表格

- 表格的基本结构
- 表格的标题
- 表格的尺寸设置
- 表格内文字的对齐、布局
- 跨多行、多列的表元
- 表格的颜色

表格的基本结构

- **<table>...</table>** 定义表格
  <caption>...</caption> 定义标题
  <tr> 定义表行
  <th> 定义表头
  <td> 定义表元(表格的具体数据)

表格的标题

- 表格标题的位置，可由ALIGN属性来设置，其位置分别由表格上方和表格下方。下面为表格标题位置的设置格式。

  设置标题位于表格上方：

  <caption align=top> ... </caption>

  设置标题位于表格下方：

  <caption align=bottom> ... </caption>

- 表格的大小

一般情况下，表格的总长度和总宽度是根据各行和各列的总和自动调整的，如果我们要直接固定表格的大小，可以使用下列方式：

<table width=n1 height=n2>

width和height属性分别指定表格一个固定的宽度和长度，n1和n2可以用像素来表示，也可以用百分比（与整个屏幕相比的大小比例）来表示

- 边框尺寸设置

  边框是用border属性来体现的，它表示表格的边框边厚度和框线。将border设成不同的值，有不同的效果。

如：

```
<table border=10   width=250>
  <caption>定货单</caption>
  <tr><th>苹果</th><th>香蕉</th><th>葡萄</th>
  <tr><td>200公斤</td><td>200公斤</td><td>100公斤</td>
  </table>
```

- **格间线宽度**

格与格之间的线为格间线，它的宽度可以使用<TABLE>中的CELLSPACING属性加以调节。格式是：

<TABLE　CELLSPACING=#>

#表示要取用的像素值

例：

```
<table border=3　　cellspacing=5>
 <caption>定货单</caption>
 <tr><th>苹果</th><th>香蕉</th><th>葡萄</th>
 <tr><td>200公斤</td><td>200公斤</td><td>100公斤
 </td>
 </table>
```

- **内容与格线之间的宽度**

  我们还可以在<TABLE>中设置CELLPADDING属性，用来规定内容与格线之间的宽度。格式为：

  <TABLE　CELLPADDING=#>

  #表示要取用的像素值

- 例：

```
<table border=3　　cellpadding=5>
  <caption>定货单</caption>
  <tr><th>苹果</th><th>香蕉</th><th>葡萄</th>
  <tr><td>200公斤</td><td>200公斤</td><td>100公斤
  </td>
  </table>
```

表格内文字的对齐/布局

- 表格中数据的排列方式有两种，分别是左右排列和上下排列。左右排列是以ALIGN属性来设置，而上下排列则由VALIGN属性来设置。其中左右排列的位置可分为三种：居左（left)、居右(right)和居中(center)；而上下排列基本上比较常用的有四种：上齐(top)、居中(middle)、下齐(bottom)和基线(baseline)。
- <tr align=#>
  <th align=#>　　#=left, center, right
  <td align=#>
- <tr valign=#>
  <th valign=#>　　#=top，middle，bottom，baseline
  <td valign=#>

- 左右排列

- <table border=1 width="200">
  <tr>
  <th>居左</th><th>居中</th><th>居右</th>
  <tr>
  <td align=left>A</td> <td align=center>B</td> <td align=right>C</td>
  </table>

- 上下排列

- <table border=1 width="250" height="100">
<tr>
<th>上齐</th><th>居中</th> <th>下齐</th><th>基线</th>
<tr>
<td valign=top>A</td> <td valign=middle>B</td> <td valign=bottom>C</td><td valign=baseline>D</td>
</table>

跨多行、多列的表元

- 要创建跨多行、多列的表元，只需在<TH>或<TD>中加入ROWSPAN或COLSPAN属性，这两个属性的值，表明了表元中要跨越的行或列的个数。
- 跨多列的表元 <th colspan=#>　<td colspan=#>
  colspan表示跨越的列数，例如colspan=2表示这一格的宽度为两个列的宽度。
- 跨多行的表元 <th rowspan=#>　<td rowspan=#>
  rowspan所要表示的意义是指跨越的行数，例如rowspan=2就表示这一格跨越表格两个行的高度。
- 跨多列的表元
- 跨多行的表元

表格的颜色

- 在表格中，既可以对整个表格填入底色，也可以对任何一行、一个表元使用背景色。

  表格的背景色彩　<table bgcolor=#>

  行的背景色彩　<tr bgcolor=#>

  表元的背景色彩　<th bgcolor=#>或 <td bgcolor=#>

  #=rrggbb 16进制 RGB数码, 或者是下列预定义色彩名称：

  Black, Olive, Teal, Red, Blue, Maroon, Navy, Gray, Lime, Fuchsia, , Green, Purple, Silver, Yellow, Aqua

文件之间的链接

- 超文本中的链接是其最重要的特性之一，使用者可以从一个页面直接跳转到其他的页面、图象或者服务器。一个链接的基本格式如： **<A HREF="资源地址">链接文字</A>**
- ·标签<A>表示一个链接的开始，</A>表示链接的结束；
  ·属性“HREF”定义了这个链接所指的地方；
  ·通过点击“链接文字”可以到达指定的文件。

  <A HREF=“http://www.swufe.edu.cn”>西南财经大学</A>

- 本地链接
- URL链接
- 目录链接

本地链接

- 以绝对路径表示：

<A HREF="/c:\study\HTML教程\link01.htm">文件的链接</A>

以相对路径表示：

<A HREF="link01.htm">文件的链接</A>

链接上一目录中的文件：

<A HREF="../../Internet/IP地址">IP地址</A>

URL链接

- URL链接的形式是： 协议名：//主机.域名 / 路径 / 文件名
  其中协议包括：

file	本地系统文件
http	WWW服务器
ftp	ftp服务器
telnet	基于TELNET的协议
mailto	电子邮件
news	Usenet新闻组
gopher	GOPHER服务器
wais	WAIS服务器

- 写在HTML文件中，链接其他主机上的文件时，格式如下：

<A HREF="http://www.swufe.edu.cn/default.htm">西南财大</A>
<A HREF="telnet://bbs.swufe.edu.cn">财大bbs</A>

- 直接指到某文件上部、下部或是中央部分

- 制作目录链接方法是：
  ·首先把把某段落设置为链接位置，格式是：〈A NAME=“链接位置名称”></A>
  ·在调用此链接部分的文件，定义连接：<A HREF=“文件名＃链接位置名称”>链接文字</A>
  如果是在一个文件内跳转，文件名可以省略不写。

- 请看例子

多视窗口FRAMES

- 使用Frames结构设计的HTML文件，能够将整个窗口分成几个独立的小窗口，每一个窗口可分别载入不同的文件，令人高兴的是，每个窗口是可以相互沟通的。
- Frames结构的基本格式
- 各窗口的尺寸设置
- 各窗口间相互操作
- FRAME的其它属性

Frames结构的基本格式

- <frameset>
  <frame src="url">
  <frame src="url">
  ...
  </frameset>
- 在有Frames结构的HTML文件中，Frames文件的整体结构为：
- <HTML>
  <HEAD>
  <TITLE></TITLE>
  </HEAD>
  <FRAMESET>
  <FRAME SRC="url">
  <FRAME SRC="url">
  ……
  </FRAMESET>
  </HTML>

各窗口的尺寸设置

- 我们将窗口分割为几块，横向分用ROWS属性，纵向分用COLS属性，每一块的大小可以由这两个属性的值来实现。
- <frameset cols=#>　例：<frameset cols="100,200,300">
  <frameset rows=#>　例：<frameset rows="10%,20%,70%">
- #的值为一对用引号括起来的字符串，字符串中的数字表示每个分窗口所占的尺寸，数字中间用逗号隔开，有几个数字就表示分出了几个窗口。当然，这其中的任何一个数字也可以由“*”来代替，这样表示由浏览器自动设置其大小。

  如：<frameset cols="100,200,*">
  <frameset cols="100,*,*">　将100像素以外的窗口平均分配
  <frameset cols="*,*,*">　将窗口分为三等份

各窗口间相互操作(Frame Target)

- 由Frames分出来的几个窗口的内容并不是静止不变的，往往一个窗口的内容随着另一个窗口的要求而不断变化，这就提高了Frames的利用价值。为了完成各窗口之间的相互操作，我们必须为每一个窗口起一个名字，这个名字用属性Name来定义。

  窗口标识(Frame Name)

  <frame src=url name="窗口名" >

  例如：　<frame src="frame/a.html" name="left">

  定义了窗口名称，还应该有Target来配合使用，Target属性指定了所链接的文件出现在哪一窗口。Target的值可以是name定义的名称，也可以是以下四类值：

  <a href=url target=_blank>　　显示一个新窗口

  <a href=url target=_self>　　显示在同一个窗口

  <a href=url target=_parent>　　显示在Frameset的前一份文件的窗口

  <a href=url target=_top>　　显示在整个浏览器窗口

Frame 的其它属性

- frame frameborder=#> #=yes, no
  各窗口边框的设置，yes表示有边框，no表示没有边框
- <frame marginwidth=# marginheight=#>
  #的值为像素点
  设置各窗口的上下左右边界宽度，如不设置，由浏览器自动决定。
- <frame scrolling=#> #=yes, no, auto

  滚动条设置，yes 表示有，no表示没有，auto表示由浏览器自动设置，#=缺省值是 auto
- <frame noresize>　noresize属性来设置不可变动的框边

- 样式表的概念并不是新创造的，文字处理器和桌面印刷系统长期以来都在使用做某种特殊样式的排版

- 样式表**最重要**的作用是提供了一种能使所有Web页面的样式保持一致的方法

- 许多HTML元素都有外观属性，如果在元素中没有指定相应的值，那么浏览器将使用这些属性的默认值。
- 例如，<h2>元素中包含font-size属性，某个浏览器可能将这个属性的默认值设定为30个点（pt）
- 而通过样式表则可以将该默认值更改为26个，这种改动可以只对某个<h2>元素有效，也可以对这个文档中所有的<h2>元素都有效。

CSS的必要性

- 样式表的某些功能可能需要借助元素的各种属性、字体的样式以及字体的尺寸元素来实现
- 然而，利用样式表可以对这些样式说明进行更精确、更一致的描述
- 目前已经不提倡直接绝大多数用于描述外观的元素属性，而是建议使用样式表。

- 样式表的三个层次按照从低层到高层的顺序依次是:
  - 内置(inline)样式表
  - 文档层(document level)样式表
  - 外部(external)样式表。

样式表的层次

- HTML样式表之所以被称为层叠样式表，是因为文档样式可以三个不同层次上进行定义
- 低层样式表可以取代高层样式表，所以某个元素内容的样式是由样式表的叠加来确定的。

- 内置样式表适用于单个元素的内容
- 文档层样式表适用于整个文档的主体
- 外部样式表则可以应用在多个文档的主体中。

- 在使用时，内置样式表优先于文档样式表，而文档样式表又优先于外部样式表。
- 在出现冲突的情况下，低层次的样式表具有使用上的优先权。

- 与元素和元素属性所面临的情况一样，某种特殊的浏览器可能不能处理样式表中指定的一些属性值。
- 对于这种情况，浏览器要么用一个可选择的值进行替代，要么将简单地忽略所给出的这些属性值。

- 内置样式说明出现在元素中，并且仅适用于那个元素中的内容。
- 这种细粒度的样式应用背离了样式表的主要设计思想——对完整文档的各种元素进行样式统一
- 对内置样式说明应当谨慎使用。

- 文档层样式说明出现在文档的头部分，并且适用于文档的整个主体。
- 这是对文档所有内容的显示外观进行格式统一的一种方法。

- 在网站建设中，往往希望网站的整体风格能够一致，这就是外部样式表的主要应用目的

- 外部样式表并不是它们所适用的文档的一部分。外部样式表单独存储，并且在所有使用它们的文档中都要进行说明。

外部样式表

- 可以用MIME类型text/css将外部样式表编写成一些文本文件，它们可以存储在因特网上的任何一台计算机中，浏览器获取外部样式表就如同获取文档一样。

- 网页首部的<link>元素也可用于指定外部样式表，在<link>元素中，rel属性用于指定被链接的文档与包含该链接的文档之间的关系，而href属性用于指定样式表文档的URL地址

外部样式表的引用

- `<link rel = stylesheet type="text/css" href="themes/AutoTheme/style.css"></link>`
- 指向一个外部样式表的链接必须放在**HTML**基本文档的头部（**header**容器中）。

- 一个样式说明的格式取决于该样式表的层次。在一个元素中内置样式说明是以style属性值的形式出现的，其一般形式如下：

```
style = “property_1:value_1;property_2:value_2;...;property_n:value_n;”
```

- 内置样式说明的作用范围只限于所处元素中的内容部分。

- 文档层样式说明在文档头部以一个<style>元素内容的形式出现，该说明的格式与内置样式表的格式有很大的区别。

- 在<style>元素中内容的一般形式如下：

```
<style type=”text/css”>
<!--
rule_list
-->
</style>
```

- <style>元素中的type属性向浏览器指明了在<style>元素中样式说明的类型。
- 在上面这个例子中，type属性被设置成“text/css“，即层叠样式表。
- 由于还有许多其他类型的样式表，因此样式说明的类型还是必要的。

- 注意，<style>元素中的规则列表是出现在HTML注释中的。

- 由于这些规则列表并不是真正的HTML，在规则列表中所包含的是提供给浏览器的有关外观样式的信息，而不是使用者看到的内容。

- 由于<style>元素的内容是一个HTML注释，所以必须使用一种不同的方法在<style>元素的内容中添加真正的注释

- 在<style>元素中这种注释由“/*”开始至“*/”结束。

- 规则列表中的样式规则均包含两部分：
  - 选择器(selector)
  - 列表
- 前者用于指出受规则影响的一个或多个元素，后者由若干成对的属性/值组成。

- 与元素属性的属性/值不同(在元素符属性的每一对属性/值中，两个部分之间使用等号相分隔，而所形成的列表使用双引号进行界定)；

- 在一个样式规则中，样式属性一值的各部分使用冒号相分隔，而所形成的列表则出现在大括弧之间。

- 一个样式规则的格式如下所示：

```
tag_name_list{property_1:value_1 ;
property_2 : value_2 ; … ; property_n :
value_n}
```

- 其中，**tag_name_list**可以是单个元素名称，或一个用逗号相隔的若干元素名称的列表。

- **tag_name_list{property_1:value_1 ; property_2 : value_2 ; … ; property_n : value_n}**

- 这里所用的词是元素名称，而不是元素，因此并没有包含尖括号。

- 如果某个属性被赋予了若干个值，那么这些值通常使用空格进行分隔，但是某些属性使用逗号分隔这些属性值。

规则列表----rule_list

- 外部样式表的格式与文档样式表的格式相类似，这个外部文件由样式规则的列表构成。

- 通常情况下，一个样式属性在样式表的作用范围内适用于所有的元素，一个内置样式表的作用范围是元素中所包含的内容。

- 对于文档样式表来说，其作用范围是文档的主体，而对于外部样式表来说，其作用范围则是所有使用它的文档的主体。

- 样式类允许在不同地方出现的同一个元素使用来自文档样式表或外部样式表不同样式说明

- 通过在<style>元素中给定一个名称来对样式类进行定义，这个名称用一个句点与元素的名称相连。

- 例如，在某个文档中如果需要设置两种段落样式，譬如normal和special，那么可以在<style>元素的注释中定义这两个类，形式如下：

```
p.normal{property-value list}
p.special{property-value list}
```

- 然后可以有如下的代码：

```
<p class =”normal”>
这个问题是可以用”普通”的风格进行展示
</p>
<p class = “special”>
那个问题是可以用”特殊”的风格进行展示
</p>
```

- 有时候需要一种能作用于若干元素的内容的样式说明类，这可以使用一个普通类实现，在定义这个普通类的时候，它的名称中不包括元素的名称，而用一个以句点开头的通用类的名称替代元素的名称。例如：

```
<style>
.italic {font-style: italic}
</style>
```

- 然后在某个文档的主体中可以包含下面这些代码：

```
<h3 class=”italic”>第3章</h3>
首先，需要注意的是:...
<p class = ”italic”>
但是，还需要提醒的是：...
</p>
```

- HTML元素中的属性可以分成六大类：
  - 字体
  - 颜色及背景
  - 文本
  - 边框与布局
  - 列表
  - 元素符等。

- 属性值可以有多种表示形式。当关键字属性值的可选范围有限并且均进行了预定义，那么就可以使用这些关键字属性值，例如underline和small。

- 属性值度量标准与前述相同。
- 但长度属性值后面必须紧跟着用两个字符的缩写词表示的单位名，在数字和单位名之间不能留有空格。
- 可能出现的单位名包括：px(像素)/ in(英寸)/ cm(厘米)/ mm(毫米)/ pt(点)和百分比，百分比形式的值是在数字后面加上百分号。

- URL属性值的形式与链接中所使用的URL的形式则不同，这些URL属性值的一般形式如下：

```
url(protocol://server/pathname)
```

  - 在url与左括号之间不能留有空格。

属性和属性值

- 在元素中定义属性值之后，该元素中内嵌的所有元素将通过继承获得这些属性值。
- 因此，在<body>元素中使用一个内置样式表设置某个属性值将会有效地影响整个文档中该属性值的设置。
- 除非发生属性值的替代现象，否则文档中的每个元素都将继承<body>中的属性值。

- 在许多情况下，要求某些特殊的字体属性只应用于整个段落中的部分文本。

- 例如，将同一行中的某个字或词组使用不同的字体尺寸或颜色显示出来通常是很有帮助的。

- 元素就是针对此目的设计的。例如，在下例中，“全面”这个词与段落中的其他部分并没有区别显示。

```
<p>
能够对文本进行<span>全面</span>控制是一件很有意
   思的事
</p>
```

- 元素的惟一用法是使用内置样式表更改属性值，例如：

```
<p> 能够对文本进行<span style = "font-size:24pt; font-family:黑体; color:red">全面</span>控制是一件很有意思的事
</p>
```

<Div>元素

- 在网页中，节是很常见的形式，每节由若干段落构成，而且每节都有各自的外观表示样式。

- 虽然在段落中使用样式类。然而，往往在设计中希望不仅是对各个段落，而且可以对网页中的节进行样式设置。

- 这里， 元素可以实现这个要求。因此它的主要用途是指定网页中某节或某区域的外观展示细节。