# 账号：luffy-python 密码：luffy-book

# 一、前端介绍

静态页面：web 1.0，仅仅是网页，不能跟用户交互。

web2.0 目前技术：HTML + css +JavaScript

HTML是网页的结构，CSS是网页的外观，而JavaScript是页面的行为

透明颜色设置：

rgba（r,g,b,透明度）

transparent（透明的，直接在颜色设置写）

# 二、HTML

<http://www.w3school.com.cn/tags/html_ref_standardattributes.asp>

一个元素的类名可以有多个，用空格隔开

class=‘class1 class2’

### html全局属性



#### style属性

有width、height、border、background-color、font-family、color 以及 font-size、text-align等大部分css可以设置的都可以经过style来设置。

style 属性规定元素的行内样式（inline style）

style 属性将覆盖任何全局的样式设定，例如在 <style> 标签或在外部样式表中规定的样式。

<div style="width:200px;height:200px"> style里面的不同内容用;隔开，内容赋值用‘：’

HTML 样式实例 - 背景颜色

background-color 属性为元素定义了背景颜色：

<html>

<body style="background-color:yellow">

<h2 style="background-color:red">This is a heading</h2>

<p style="background-color:green">This is a paragraph.</p>

</body>

</html>

HTML 样式实例 - 字体、颜色和尺寸

font-family、color 以及 font-size 属性分别定义元素中文本的字体系列、颜色和字体尺寸：

<html>

<body>

<h1 style="font-family:verdana">A heading</h1>

<p style="font-family:arial;color:red;font-size:20px;">A paragraph.</p>

</body>

</html>

style 属性淘汰了旧的 <font> 标签。

HTML 样式实例 - 文本对齐

text-align 属性规定了元素中文本的水平对齐方式：

<html>

<body>

<h1 style="text-align:center">This is a heading</h1>

<p>The heading above is aligned to the center of this page.</p>

</body>

</html>

HTML作为一门标记语言，是通过各种各样的标签来标记网页内容的。我们学习HTML主要就是学习的HTML标签。

### 那什么是标签呢？

1. 在HTML中规定标签使用英文的的尖括号即`<`和`>`包起来，如`<html>`、`<p>`都是标签。

2. HTML中标签\*\*通常\*\*都是成对出现的，分为开始标签和结束标签，结束标签比开始标签多了一个`/`，如`<p>标签内容</p>`和`<div>标签内容</div>`。开始标签和结束标签之间的就是标签的内容。

3. 标签之间是可以嵌套的。例如：div标签里面嵌套p标签的话，那么`</p>`必须放在`</div>`的前面。

4. HTML标签不区分大小写，`<h1>`和`<H1>`是一样的，但是我们通常建议使用小写，因为大部分程序员都以小写为准。

## （一）HTML结构

<! DOCTYPE HTML> 不是标签，是声明

<!DOCTYPE HTML>

<html>

<head>...</head>

<body>...</body>

</html>

让我们来解释一下上面的代码：

首先，<!DOCTYPE HTML>是文档声明，必须写在HTML文档的第一行，位于<html>标签之前，表明该文档是HTML5文档。

1. <html></html> 称为根标签，所有的网页标签都在<html></html>中。
2. <head></head> 标签用于定义文档的头部，它是所有头部元素的容器。常见的头部元素有<title>、<script>、<style>、<link>和<meta>等标签，头部标签在下一节中会有详细介绍。
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<body>标签中的内容（图中淡绿色部分内容）最终会在浏览器中显示出来。

HTML文档包含了HTML标签及文本内容，不同的标签在浏览器上会显示出不同的效果，所以我们需要记住最常见的标签的特性。



### （1）注释

HTML中注释的格式:

<!--这里是注释的内容-->

注意： 注释中可以直接使用回车换行。

并且我们习惯用注释的标签把HTML代码包裹起来。如：

<!-- xx部分 开始 -->

这里放你xx部分的HTML代码

<!-- xx部分 结束 -->

HTML注释的注意事项：

1. HTML注释不支持嵌套。
2. HTML注释不能写在HTML标签中。

### （2）head标签

#### title、meta、link、style、script

我们首先来介绍一下head标签的主要内容和作用，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、编码方式及URL等信息,这些信息大部分是用于提供索引,辩认或其他方面的应用（移动端）的等。 以下标签是可以用在head标签中的：title、meta、link、style、script

<head lang='en'>

<title>标题信息</title>

<meta charset='utf-8'>

<link>

<style type='text/css'></style>

<script type='text/javascript'></script>

</head>

#### 1.title标签

title标签

<title>标签：在<title>和</title>标签之间的文字内容是网页的标题信息，它会显示在浏览器标签页的标题栏中。可以把它看成是一个网页的标题。主要用来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，迅速的判断出当前网页的主题。

我们接下来做一个小练习，创建一个带有我们自定义标题内容的网页：

<!DOCTYPE HTML>

<html>

<head>

<title>路飞学城</title>

</head>

<body></body>

</html>

将上面的文件另存为demo.html，然后用浏览器打开，就可以看到下面的内容。



#### 2.meta标签

Meta标签介绍：

元素可提供有关页面的原信息（mata-information）,针对搜索引擎和更新频度的描述和关键词。

标签位于文档的头部，不包含任何内容。

提供的信息是用户不可见的。 meta标签的组成：meta标签共有两个属性，它们分别是http-equiv属性和name属性，不同的属性又有不同的参数值，这些不同的参数值就实现了不同的网页功能。

常用的meta标签：

##### ①http-equiv属性

它用来向浏览器传达一些有用的信息，帮助浏览器正确地显示网页内容，与之对应的属性值为content，content中的内容其实就是各个参数的变量值。

<!--重定向： 2秒后跳转到对应的网址，注意分号，-->

<meta http-equiv="refresh" content="2;URL=http://www.luffycity.com">

<!--指定文档的内容类型和编码类型 -->

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

<!--告诉IE浏览器以最高级模式渲染当前网页-->

<meta http-equiv="x-ua-compatible" content="IE=edge">

##### ②name属性

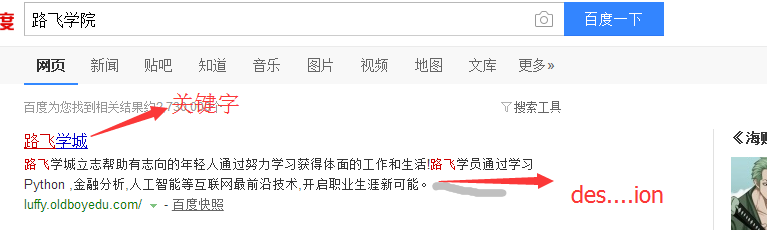
主要用于页面的关键字和描述，是写给搜索引擎看的，SEO优化。关键字可以有多个用 ‘,’号隔开，与之对应的属性值为content，content中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。

<meta name="keywords" content="meta总结,html meta,meta属性,meta跳转">

<meta name="description" content="路飞学城">

关键字是搜索关键字

description是搜索后，搜索页面标题下显示的内容



##### ③其他属性自行掌握

#### 3.link属性

link链接外部样式表、定义网站图标等

type定义内部样式表

script定义内部脚本文件或者链接外部脚本文件

#### 其他标签

<!--标题-->

<title>路飞学城</title>

<!--icon图标（网站的图标）-->

<link rel="icon" href="fav.ico">

<!--定义内部样式表-->

<style></style>

<!--引入外部样式表文件-->

<link rel="stylesheet" href="mystyle.css">

<!--定义JavaScript代码或引入JavaScript文件-->

<script src="myscript.js"></script>

### （3）body标签

#### 1.标题标签 h1~h6

<h1> - <h6> 标签可定义标题。<h1> 定义最大的标题。<h6> 定义最小的标题。 由于 h 元素拥有确切的语义，因此请您慎重地选择恰当的标签层级来构建文档的结构。因此，请不要利用标题标签来改变同一行中的字体大小。相反，我们应当使用css来定义来达到漂亮的显示效果。 标题标签通常用来制作文章或网站的标题。

h1~h6标签的默认样式：

<!DOCTYPE HTML>

<html>

<head lang='en'>

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

<title>路飞学城</title>

</head>

<body>

<h1>一级标题</h1><h2>二级标题</h2>

<h3>三级标题</h3>

<h4>四级标题</h4>

<h5>五级标题</h5>

<h6>六级标题</h6>

</body>

</html>

请看上面代码 <h1>和<h2>书写在一行上展示，但是在浏览器的效果却是换行了，大家现在预习一下9.1.9标签分类知识点。

文本样式标签主要用来对HTML页面中的文本进行修饰，比如加粗、斜体、线条样式等...

#### 11.文本修饰标签b/i/u/s/sup/sub/em/strong

1. `<b></b>`：加粗

2. `<i></i>`：斜体

3. `<u></u>`：下划线

4. `<s></s>`：删除线

5. `<sup></sup>`：上标

6. `<sub></sub>`：下标

7.现在如果想在一段文字中特别强调某几个字，这时候就可以用到`<em>`或`<strong>`标签。

这两个标签都是表示强调，但是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。

在浏览器中`<em>`默认会用斜体表示，`<strong>`会用粗体来表示。两个标签相比，我们通常会推荐大家使用`<strong>`表示强调

#### 2.段落标签p

段落标签 p

<p>,paragraph的简写。定义段落

<body>

<p>路飞学城立志帮助有志向的年轻人通过努力学习获得体面的工作和生活!路飞学员通过学习Python ,金融分析,人工智能等互联网最前沿技术,开启职业生涯新可能</p>

<p>路飞学城立志帮助有志向的年轻人通过努力学习获得体面的工作和生活!路飞学员通过学习Python ,金融分析,人工智能等互联网最前沿技术,开启职业生涯新可能</p>

</body>

浏览器展示特点：

1. 跟普通文本一样，但我们可以通过css来设置当前段落的样式
2. 是否又独占一行呢？ 答案是的 块级元素

#### 3.超链接标签 a

a标签会在一行显示，属于行内标签。

超链接标签 a

超级链接<a>标记代表一个链接点，是英文anchor（锚点）的简写。它的作用是把当前位置的文本或图片连接到其他的页面、文本或图像

<body>

<h1>

<!-- a链接 超链接

target:\_blank 在新的网站打开链接的资源地址

\_self 在当前网站打开链接的资源地址

title: 鼠标悬停时显示的标题

-->

##### 1.超链接网址、下载包、邮箱

<a href="http://www.baidu.com" target="\_blank" title="路飞学城">路飞学城</a>

<a href="a.zip">下载包</a>

<a href="mailto:zhaoxu@tedu.cn">联系我们</a>

<!-- 返回页面顶部的内容 -->

##### 2.回到顶部、回到某个区域

<a href="#">跳转到顶部</a> #井号代表页面顶部

<!-- 返回某个id -->

<p style=’height:2000px’id=’p1’>名字叫p1的区域</p> 设置区域高度为区域命名

<a href="#p1">跳转到p1</a> #id 跳转至该id处

##### 3.连接JavaScript

<!-- javascript:是表示在触发<a>默认动作时，执行一段JavaScript代码，而 javascript:; 表示什么都不执行，这样点击<a>时就没有任何反应。其实就是取消a标签的默认动作 -->

<a href="javascript:alert(1)">内容</a>

<a href="javascript:;">内容</a>

</h1>

</body>

target:\_blank 在新的网站打开链接的资源地址

target：\_self 在当前网站打开链接的资源地址

title: 表示鼠标悬停时显示的标题

链接其他表现形式：

1. 目标文档为下载资源 例如：href属性值，指定的文件名称，就是下载操作(rar、zip等)
2. 电子邮件链接 前提：计算机中必须安装邮件客户端，并且配置好了邮件相关信息。 例如：<a href="mailto:zhaoxu@tedu.cn">联系我们</a>
3. 返回页面顶部的空链接或具体id值的标签 例如：<a href="#">内容</a>或<a href="#id值">内容</a>
4. javascript:是表示在触发<a>默认动作时，执行一段JavaScript代码。 例如：<ahref="javascript:alert()">内容</a>
5. javascript:;表示什么都不执行，这样点击<a>时就没有任何反应 例如：<a href="javascrip:;">内容</a

#### 4.列表标签 ul，ol

网站页面上一些列表相关的内容比如说物品列表、人名列表等等都可以使用列表标签来展示。通常后面跟<li>标签一起用，每条li表示列表的内容

<ul>:unordered lists的缩写 无序列表 <ol>:ordered listsde的缩写 有序列表

<!-- 无序列表 type可以定义无序列表的样式-->

<ul type="circle">

<li>我的账户</li>

<li>我的订单</li>

<li>我的优惠券</li>

<li>我的收藏</li>

<li>退出</li>

</ul>

<!-- 有序列表 type可以定义有序列表的样式 -->

<ol type="a">

<li>我的账户</li>

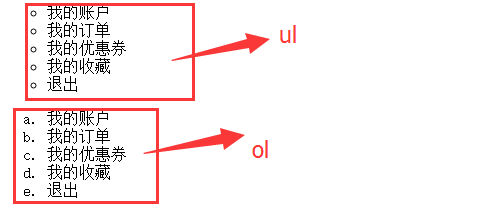
<li>我的订单</li>

<li>我的优惠券</li>

<li>我的收藏</li>

<li>退出</li>

</ol>



ol标签的属性：

type：列表标识的类型

* 1：数字
* a：小写字母
* A：大写字母
* i：小写罗马字符
* I：大写罗马字符

列表标识的起始编号

* 默认为1

ul标签的属性： type：列表标识的类型

* disc：实心圆(默认值)
* circle：空心圆
* square：实心矩形
* none：不显示标识

#### 5.盒子标签 div

<div>可定义文档的分区 division的缩写 译：区 <div> 标签可以把文档分割为独立的、不同的部分,请看下面代码我们将他们进行分区

<!DOCTYPE html>

<html>

<head lang="en">

<meta charset="utf-8" >

<title>常用标签一</title>

</head>

<body>

<div id="wrap"> 为分区起名字，id不可以重复

<div class="para">

<p style="height: 1000px" id="p1">段落</p>

</div>

<div class="anchor">

我是普通的文本

<h1>

<a href="http://www.baidu.com" target="\_blank" title="路飞学城">路飞学城</a>

<a href="a.zip">下载包</a>

<a href="mailto:zhaoxu@tedu.cn">联系我们</a>

<a href="#">跳转到顶部</a>

<a href="#p1">跳转到p1</a>

<a href="javascript:alert(1)">内容</a>

<a href="javascript:;">内容</a>

</h1>

</div>

<!-- <h2>路飞学城</h2>

<h3>路飞学城</h3>

<h4>路飞学城</h4>

<h5>路飞学城</h4>

<h6>路飞学城</h6> -->

<div class="para">

<!-- 定义段落 通常指文章一段内容 -->

<p>路飞学城立志帮助有志向的年轻人通过努力学习获得体面的工作和生活!路飞学员通过学习Python ,金融分析,人工智能等互联网最前沿技术,开启职业生涯新可能</p>

<p>路飞学城立志帮助有志向的年轻人通过努力学习获得体面的工作和生活!路飞学员通过学习Python ,金融分析,人工智能等互联网最前沿技术,开启职业生涯新可能</p>

<p>路飞学城立志帮助有志向的年轻人通过努力学习获得体面的工作和生活!路飞学员通过学习Python ,金融分析,人工智能等互联网最前沿技术,开启职业生涯新可能</p>

</div>

<div class="lists"> 为分区归类，起类名，类名相同的分区是一类

<!-- 无序列表 -->

<ul type="circle">

<li>我的账户</li>

<li>我的订单</li>

<li>我的优惠券</li>

<li>我的收藏</li>

<li>退出</li>

</ul>

<!-- 有序列表 -->

<ol type="a">

<li>我的账户</li>

<li>我的订单</li>

<li>我的优惠券</li>

<li>我的收藏</li>

<li>退出</li>

</ol>

</div>

</div>

</body>

</html>

分析上面代码可以下面的层次结构

<div id='wrap'>

<div class='para'></div>

<div class='anchor'></div>

<div class='para'></div>

<div class='lists'></div>

</div>

我们将文档放在一个父级的区（div）中，它里面包含四块区（div）域，浏览器查看效果,你会发现每小块区域都是独占一行的，所以div是块级元素。另外，每块区域表示独立的一块，id属性和class属性其实很简单，你可以看成给这个区域起个名字。id是唯一的，一个页面中不能有两个重复的id，跟身份证号码一样，class可以设置同样的属性值，并且可以设置多个，例如class=’para n1‘

#### 6.图片标签 <img/>

一个网页除了有文字，还会有图片。我们使用<img/>标签在网页中插入图片。

语法：<img src="图片地址" alt="图片加载失败时显示的内容" title = "提示信息" />

注意：

1. src设置的图片地址可以是本地的地址也可以是一个网络地址。
2. 图片的格式可以是png、jpg和gif。
3. alt属性的值会在图片加载失败时显示在网页上。
4. 还可以为图片设置宽度(width)和高度(height)，不设置就显示图片默认的宽度和高度
5. <div>
6. <span>与行内元素展示的标签<span>
7. <span>与行内元素展示的标签<span>
8. <img src="./machine-right.png" alt="金融量化分析" style="width:200px;height:200px">
9. <img src="./finance-right.png" alt="人工智能实战" style="width: 200px;height: 200px">
10. </div>

浏览器查看效果：行内块元素

1. 与行内元素在一行内显示
2. 可以设置宽度和高度
3. span标签可以单独摘出某块内容，结合css设置相应的样式

<p>路飞学城立志帮助有志向的年轻人通过努力学习获得体面的工作和生活!路飞学员通过学习Python ,<span>金融分析</span>,<span>人工智能</span>等互联网最前沿技术,开启职业生涯新可能</p>

好的，同学们，此时大家是不是对于以上标签有所认知了呢？ 这样吧！我来出一个小练习，你来做！ 练习：将img标签中的图片独占一行展示，并设置设置相应标题，在图片下方，描述该图片内容的信息。

#### 7.三种元素

块级元素：独占一行，可设宽高，如果不设宽度，则默认为浏览器宽度：

head、h1、title、p、ul、ol、li、div、form

行内元素：在一行内展示，不能设置宽高，宽高根据内容填充

a、span

行内块元素：在一行内展示，可设置宽高。

img

#### 8.其他标签

##### 8.1换行标签 <br>

<br>标签用来将内容换行，其在HTML网页上的效果相当于我们平时使用word编辑文档时使用回车换行。

##### 8.2分割线 <hr>

<hr>标签用来在HTML页面中创建水平分隔线，通常用来分隔内容

##### 8.3特殊符号

浏览器在显示的时候会移除源代码中多余的空格和空行。 所有连续的空格或空行都会被算作一个空格。需要注意的是，HTML代码中的所有连续的空行（换行）也被显示为一个空格。

举个例子：

<!DOCTYPE HTML>

<html>

<head>

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

<title>路

| **内容** | **代码** |
| --- | --- |
| 空格 | &nbsp; |
| > | &gt; |
| < | &lt; |
| & | &amp; |
| ¥ | &yen; |
| 版权 | &copy; |
| 注册 | &reg; |

飞学城</title>

</head>

<body>

<p>帮助有志向的年轻人

通过努力学习获得

体面的 工作 和 生活！</p>

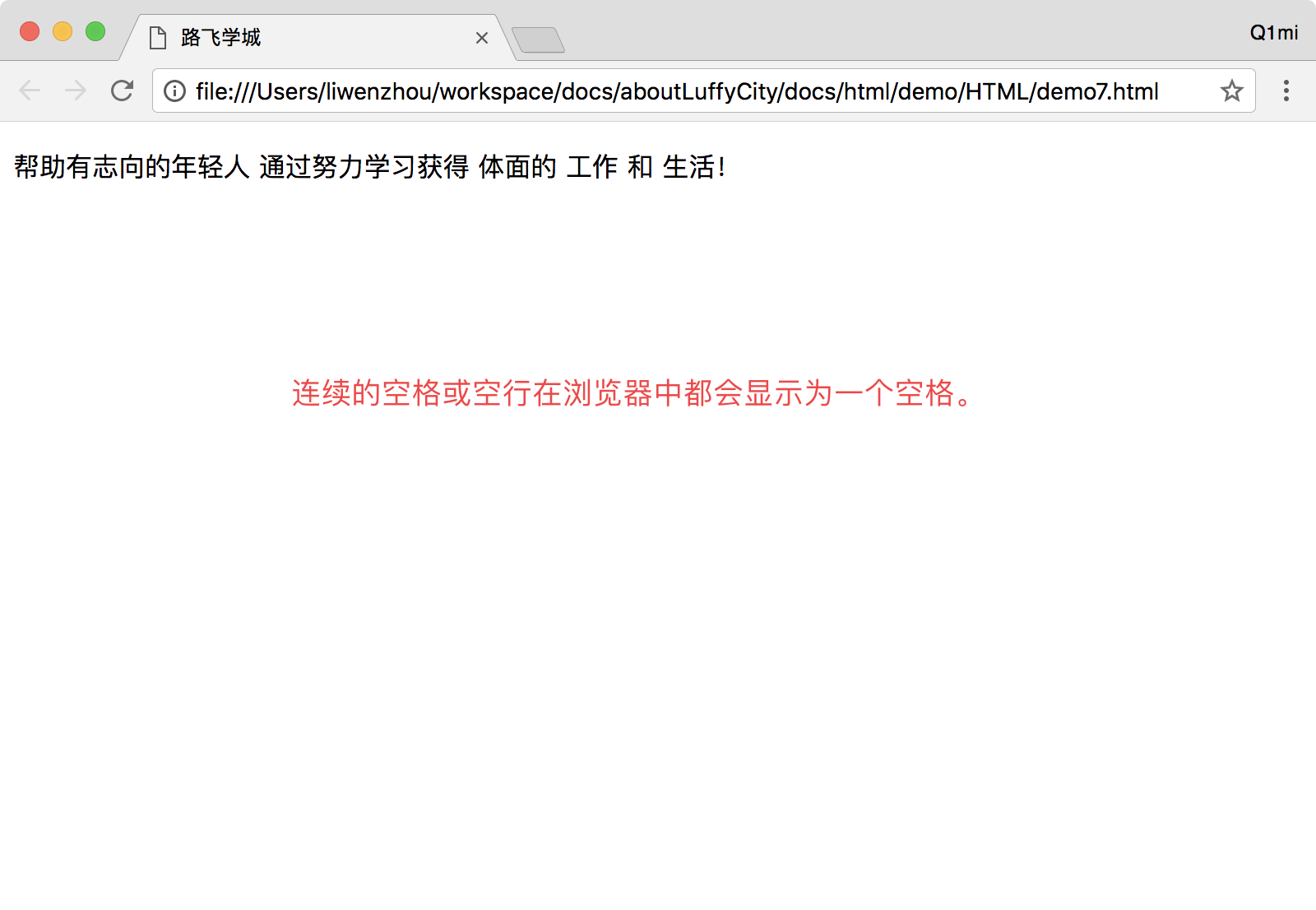
</body>

</html>

上面的代码在浏览器中最终的显示效果：

8.4 span标签

将想要特殊显示的字用span标签标示，用css控制。



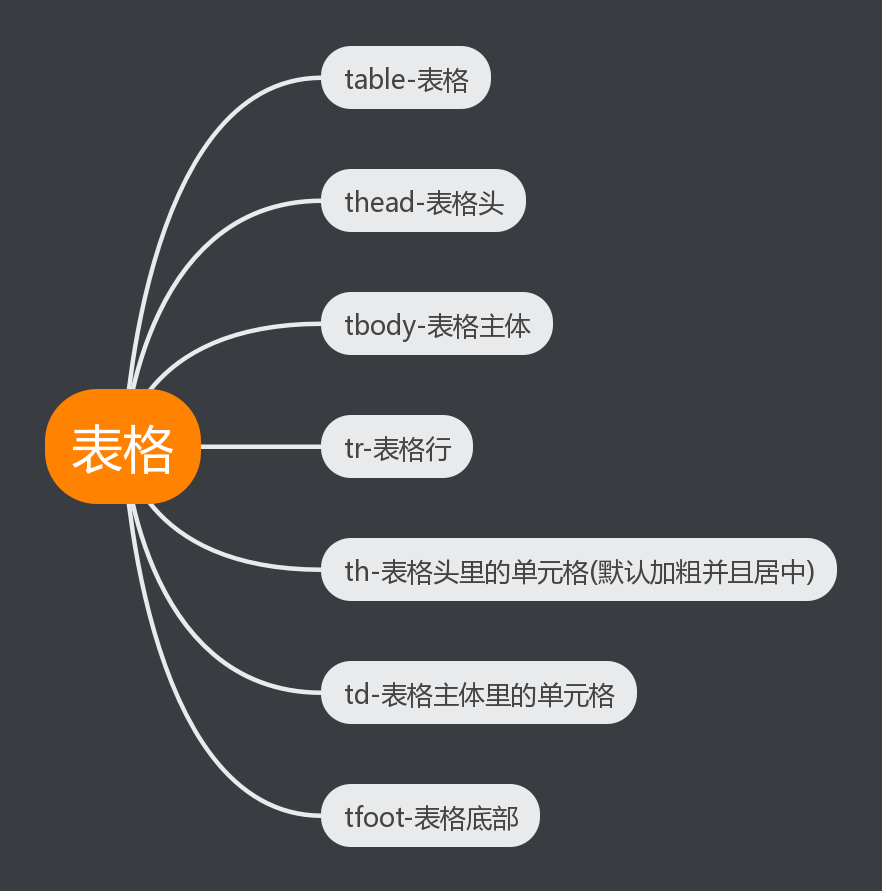
#### 9.特殊字符

在上一个实例中，我们演示了HTML中输入空格、回车都是没有作用的。要想输入空格，需要用特殊符号 -- &nbsp;。

常用的特殊字符：

html特殊字符对照表：<http://tool.chinaz.com/Tools/HtmlChar.aspx>

### （4）表格标签



#### 1.基本概述

表格由<table> 标签来定义。每个表格均有若干行（由 <tr> 标签定义），每行被分割为若干单元格（由<td>标签定义）。字母 td 指表格数据（table data），即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等等

<div class="table">

<table>

<!--表格头-->

<thead>

<!--表格行-->

<tr>

<!--表格列，【注意】这里使用的是th-->

<th></th>

</tr>

</thead>

<!--表格主体-->

<tbody>

<!--表格行-->

<tr>

<!--表格列，【注意】这里使用的是td-->

<td></td>

</tr>

<tr>

<td></td>

</tr>

</tbody>

<!--表格底部-->

<tfoot>

<tr>

<td></td>

</tr>

</tfoot>

</table>

</div>

#### 2.对表格添加边框。

这是整体添加。线宽为1，实心线。

#### 3. 表格行和列的合并

在需要合并的第一个空格处写

<td rowspan=”3”>….内容…</td> 合并三行

<td colspan=”4”>……内容…</td>合并4列

rowspan 合并行(竖着合并)

colspan 合并列(横着合并)

<body>

<div class="table">

<table border="1" cellspacing="0">

<!--表格头-->

<thead>

<!--表格行-->

<tr>

<!--表格列，【注意】这里使用的是th-->

<th></th>

<th>星期一</th>

<th>星期二</th>

<th>星期三</th>

<th>星期四</th>

<th>星期五</th>

</tr>

</thead>

<!--表格主体-->

<tbody>

<!--表格行-->

<tr>

<td rowspan="3">上午</td>

<!--表格列，【注意】这里使用的是td-->

<td>语文</td>

<td>数学</td>

<td>英文</td>

<td>生物</td>

<td>化学</td>

</tr>

<tr>

<!--表格列，【注意】这里使用的是td-->

<td>语文</td>

<td>数学</td>

<td>英文</td>

<td>生物</td>

<td>化学</td>

</tr>

<tr>

<!--表格列，【注意】这里使用的是td-->

<td>语文</td>

<td>数学</td>

<td>英文</td>

<td>生物</td>

<td>化学</td>

</tr>

<tr>

<td rowspan ="2">下午</td>

<!--表格列，【注意】这里使用的是td-->

<td>语文</td>

<td>数学</td>

<td>英文</td>

<td>生物</td>

<td>化学</td>

</tr>

<tr>

<!--表格列，【注意】这里使用的是td-->

<td>语文</td>

<td>数学</td>

<td>英文</td>

<td>生物</td>

<td>化学</td>

</tr>

</tbody>

<!--表格底部-->

<tfoot>

<tr>

<td colspan="6">课程表</td>

</tr>

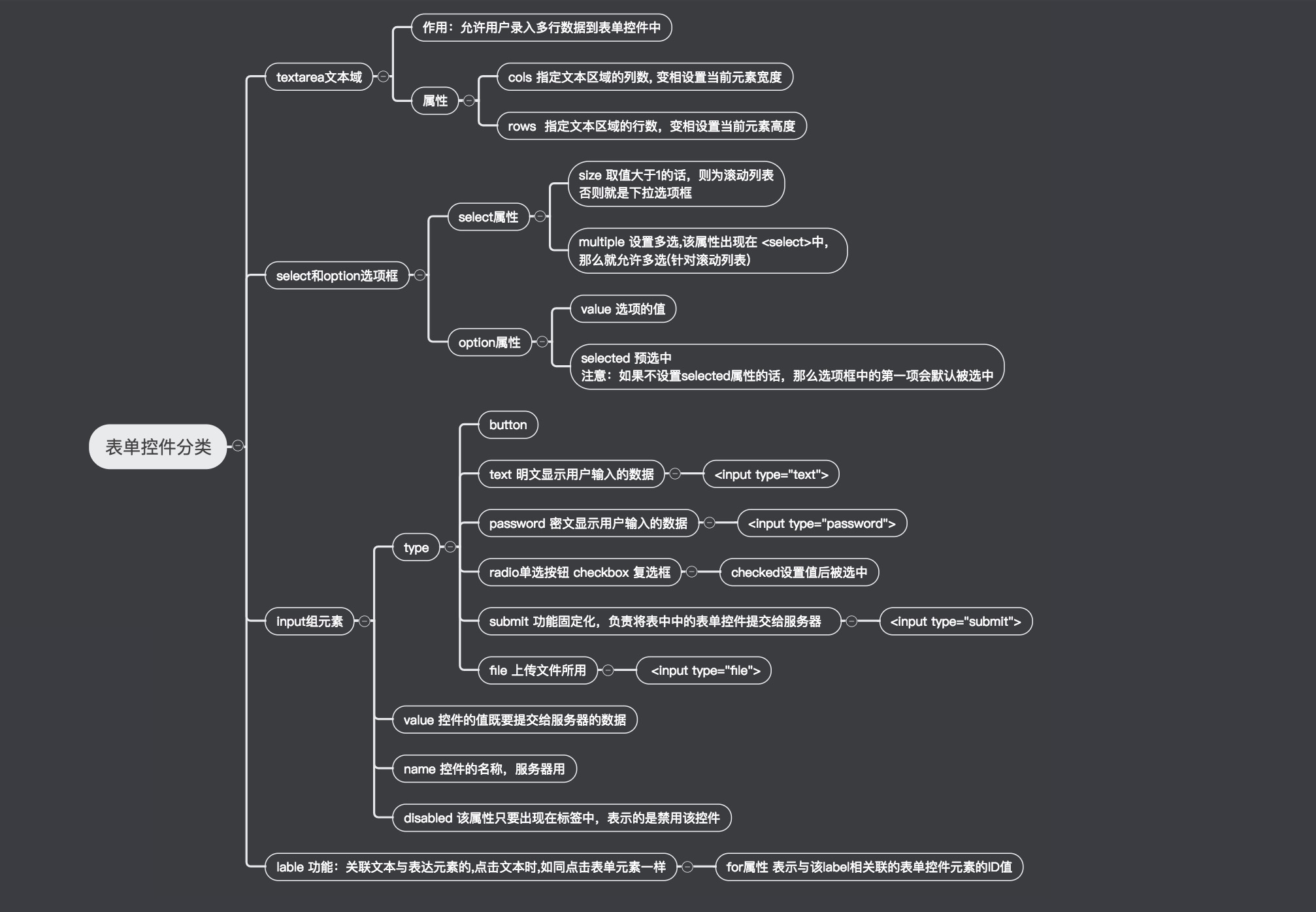
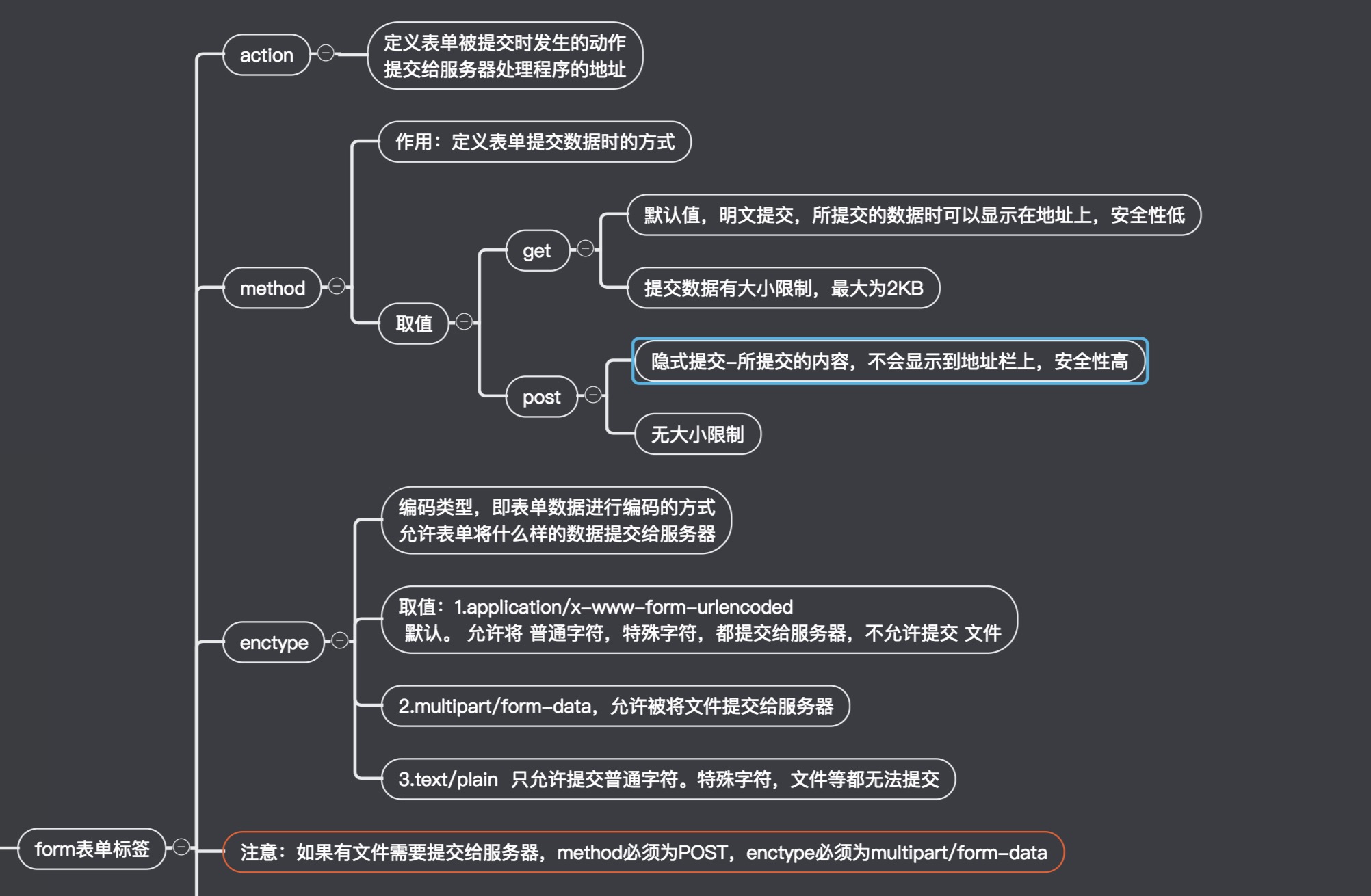
</tfoot>

</table>

</div>

</body>

### （5）表单form



#### 1.form的两个属性

acction 是提交的服务器地址。method看前面图



#### 2.label标签：

行内元素。

<label for “user”>用户名</label>

#### 3.input type= text/password

设置input边框的外线：outline:none

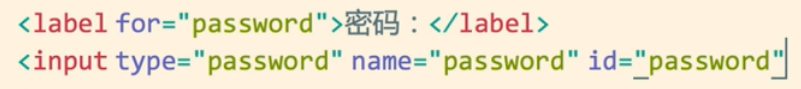
行内块

<input type=”text”name=”username”id=”user”placeholder=”请输入用户名”value=”张三”>

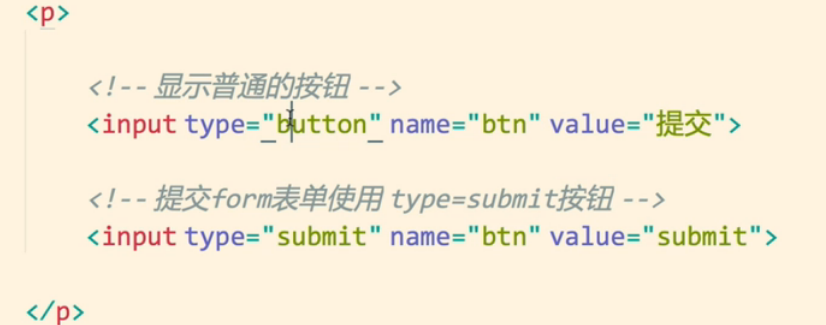
placeholder 为灰色提示语句。value 设置默认值。如果value有默认值，会覆盖掉placeholder

#### 4.input内部的按钮。

input还有“date”类型。type=date 出现日期选框



name是为输入值起的变量名。



#### 禁用按钮：

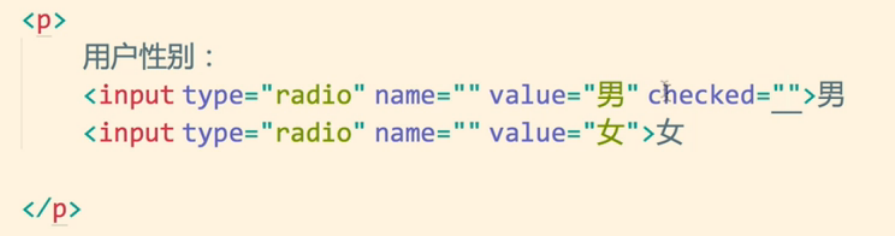
disabled=””

#### 5.还有一个button标签

<button>我是按钮</button>

单独的按钮控件，与form没有关系。

#### 6.单选按钮 radio



checked 为默认选中

value值为被选中后提交的数据。

两个选框name一样时，会有互斥效果，即同一时间只能选中一个按钮

如果name不同，即为两个不相关的单选按钮，可以同时选中。

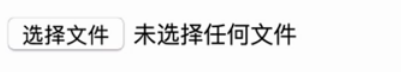
#### 7.复选框



也可以添加checked，默认选中。

#### 8.file



会出现选择文件的按钮



注意上传文件method一定是post

#### 9.重置按钮 input type=“reset”

#### 10.文本域 textarea



#### 11.下拉列表 select option

闭合标签。size 大于1的时候为滚动列表。默认为下拉列表。看前面的总图。

如果multiple出现在滚动列表中，允许多选。selected为预选中。





#### 12.submit

input type=submit

### （6）标签分类及嵌套

#### 1.分类

常用的块状元素：块级元素可以嵌套块级元素。

<div> <p> <h1>~<h6> <ol> <ul> <table><form> <li>

常用的行内元素

<a> <span> <br> <i> <em> <strong> <label>

常用的行内块状元素：

<img> <input>

**块级元素特点**：display:block;

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。**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。

**行内元素特点**：display:inline;

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。**

**行内块状元素的特点**：display:inline-block;

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**

注意

我们可以通过display属性对块级元素、行内元素、行内块元素进行转换，为后面页面布局做好了准备。

#### 2.嵌套规则：

1. 块元素可以包含内联元素或某些块元素，但内联元素却不能包含块元素，它只能包含其它的内联元素，例如：

<div><div></div><h1></h1><p><p></div> ✔️

<a href=”#”><span></span></a> ✔️

<span><div></div></span> ❌

1. 块元素不能放在p标签里面

<p><ol><li></li></ol></p> ️

<p><div></div></p> ❌

1. 有几个特殊的块级元素只能包含内嵌元素，不能再包含块级元素，这几个特殊的标签是：

h1、h2、h3、h4、h5、h6、p

1. li元素可以嵌入ul，ol，div等标签

<ul>

<li>

<ul>

<li>

<div>

</div>

</li>

<li>

<ol>

<li></li>

<li></li>

<li></li>

<li></li>

</ol>

</li>

</ul>

</li>

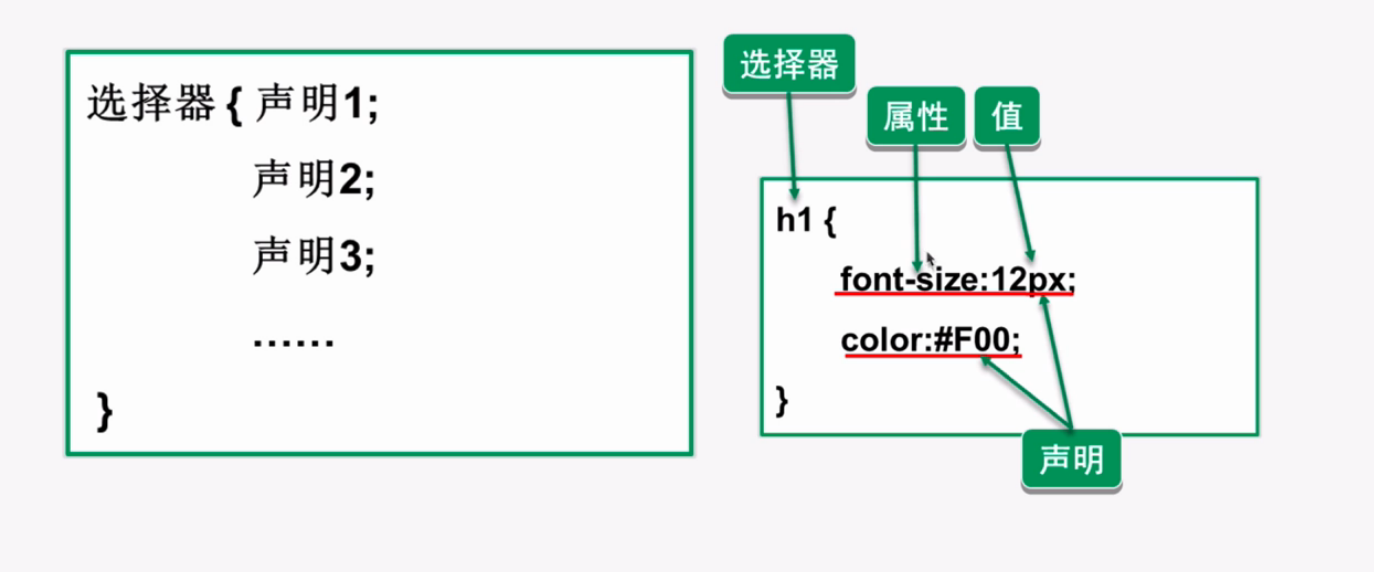
</ul>

# 三、css

css注释/\* \*/

## （一）css特点及优势

css：重叠样式表



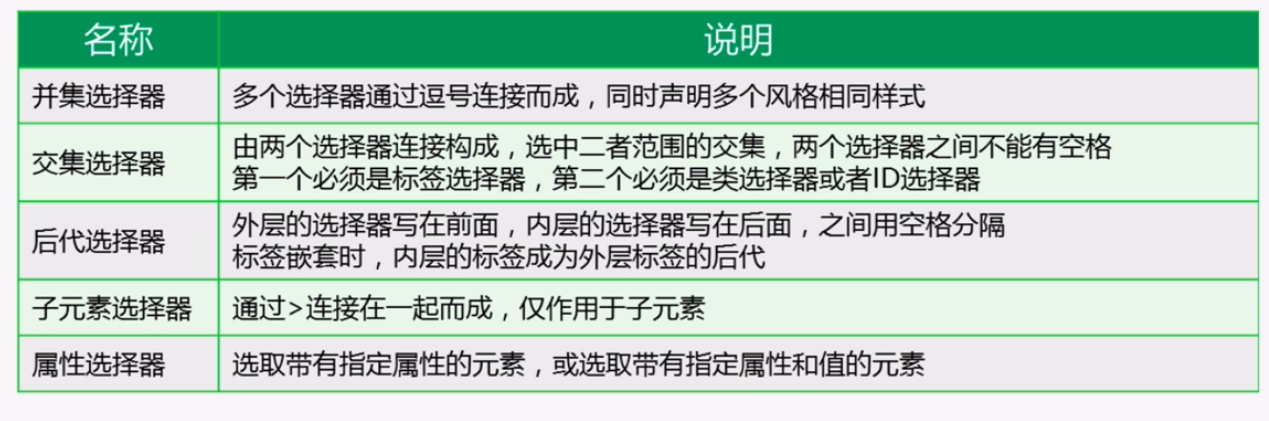
### 1.选择器：

标签选择器

类选择器

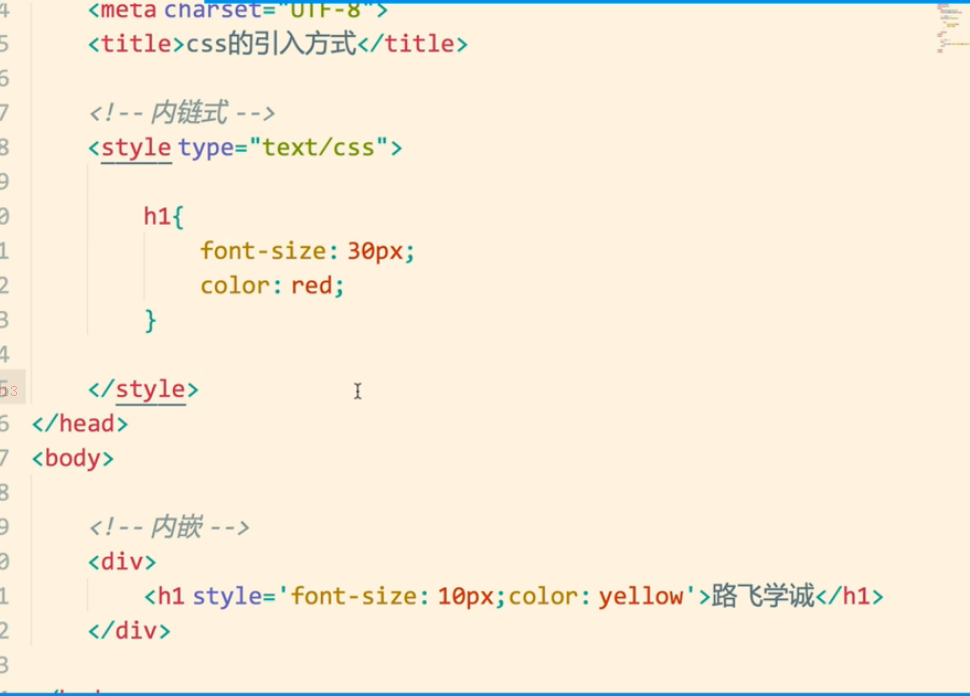
ID选择器

高级选择器：



### 2.样式

#### 内链式与内嵌式



**内嵌式的优先级要大于内链式，内链式优先级大于外链式**

#### 外链式

##### ①链接式



将格式设置写在一个css文件中，通过link来链接。

link标签需要在head里面。

##### ②导入式



通过style标签导入。

③区别



## （二）选择器

### 1.基础选择器

#### ①通用选择器

使用\*选择所有元素：对全局进行设置。

\* {

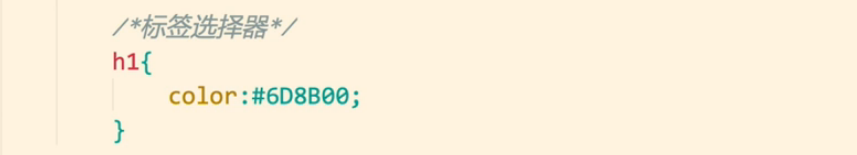
padding：0； /\*据上端多少\*/

margin：0；/\*据左端多少\*/

color: black;

}

#### ②标签选择器：通过标签来选择。



#### ③ID选择器

通过元素的ID值选择元素：

示例：

#i1 {

color: red;

margin：0 auto /\*整个模块居中显示\*/

}

将id值为i1的元素字体颜色设置为红色。

#### ④类选择器

通过样式类选择元素：

示例：

.c1 {

color: red;

}

将所有样式类含.c1的元素设置字体颜色为红色。

### 2.高级选择器

#### 1.群组选择器（并集）



#### 2.交集选择器

选择器之间不能有空格

p.conne

选择类名叫conne的p标签

p#hello

选择id为hello的p标签

#### 3.子代选择器 >

选择嵌套的第一层

.content > .hello

选择content下一层的hello

div>p {

color: red;

}

从div的直接子元素中找到p标签，设置字体颜色为红色。

#### 4. 后代选择器

因为HTML元素可以嵌套，所以我们可以从某个元素的后代查找特定元素，并设置样式：

div p {

color: red;

}

从div的所有后代中找p标签，设置字体颜色为红色。

#### 5. 毗邻选择器

div+p {

color: red;

}

找到所有紧挨在div后面的第一个p标签，设置字体颜色为红色。

#### 6.弟弟选择器

div~p {

color: red;

}

找到所有div标签后面同级的p标签，设置字体颜色为红色。

### 3.属性选择器

除了HTML元素的id属性和class属性外，还可以根据HTML元素的特定属性选择元素。

#### （1）根据属性查找

找到存在title属性的选择器

**[title]**

找到所有title属性等于hello的元素：

**[title="hello"]**

找到所有title属性以hello开头的元素：

**[title^="hello"]**

找到所有title属性以hello结尾的元素：

**[title$="hello"]**

找到所有title属性中包含（字符串包含）hello的元素：

**[title\*="hello"]**

找到所有title属性(有多个值或值以空格分割)中有一个值为hello的元素：

**[title~="hello"]**

选中同时具有 href 和 title 属性a标签：

**a[href][title]**

#### （3）表单常用

input[type="text"] {

backgroundcolor: red;

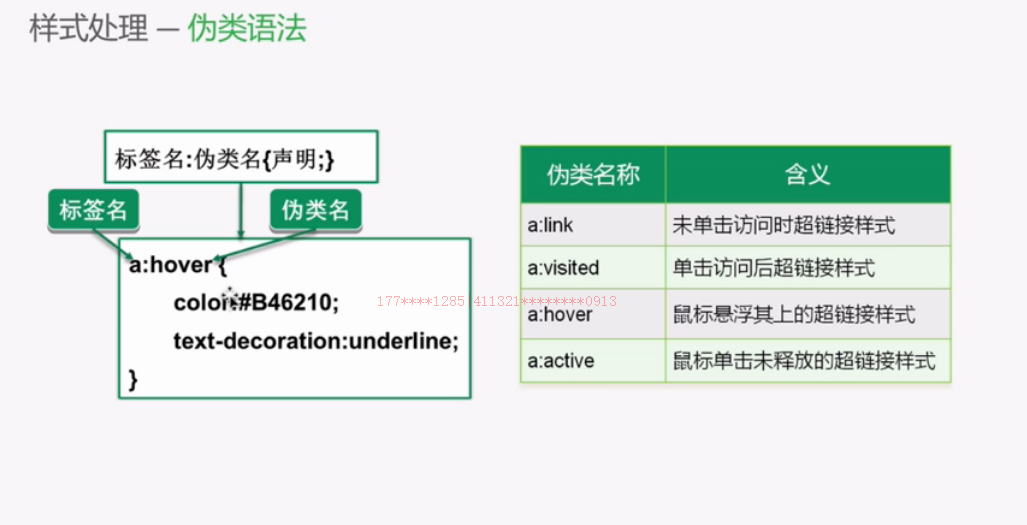
}

### 4.选择器优先级

ID选择器 > 类选择器 >标签选择器

### 5.伪类选择器的使用

#### 1.a的伪类选择器





#### 2.input伪类选择器

input输入框获取焦点时样式：即当鼠标在输入框上时

input:focus {

outline: none;

background-color: #eee;

}

#### 3.first-child、last-child、nth-child(n)

nth-child(n):当n为具体数值时，表示选中第几个，当为n时，表示为选中所有。

nth-child(2n):选中偶数

nth-child（2n-1）：选中奇数。

可以以此类推，选中有规律的。

div li:fisrst-child：选中第一个li

div li:last-child:选中最后一个li

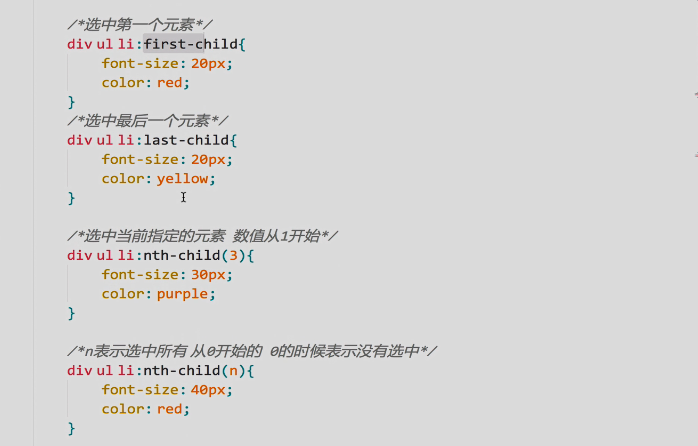
div li:nth-child(5)：选中第五个li

div li : nth-child(n)：选中全部li

div li : nth-child(2n):选中全部偶数li n从0开始

div li :nth-child(2n-1):选中全部奇数li

div li：nth-child（3n-1）：每隔三个选中一个li



### 6.伪元素选择器

#### （1）first-letter

用于为文本的首字母设置特殊样式。

例如：

p:first-letter {

font-size: 48px;

}

#### （2）before

用于在元素的内容前面插入新内容。

例如：

p:before {

content: "\*";

color: red;

}

在所有p标签的内容前面加上一个红色的\*。

#### （3）after

用于在元素的内容后面插入新内容。

例如：

p:after {

content: "\*";

color: red;

}

在所有p标签的内容后面加上一个蓝色的?。

## css继承

可以继承的一些属性：color、font-\*、text-\*、line-\* 文本元素。

有一些盒子元素，定位的元素（浮动，绝对定位，固定定位）不能继承。

## css层叠性、权重

权重：id、class、标签名 权重依次减小。

对于选中同一个标签的显示：按选中时使用的id多的显示，id数量相同，比较clss数量，类推。当权重一样时，后设置的属性覆盖前设置的属性。

继承来的属性权重为0，权重最低。如果权重都为0，则以描述的最近（距离该标签最近）的继承为准。

如：下列三种情况都选中p，但是最终颜色是red，因为选中p用了两个id，权重最大。

#id1 #id2 .c1 p{

color:red

}

#id1 p{

color:green

}

.c1 p{

color:yellow；

}

## !important

加在分号前面，设置权重为最大。尽量少用。

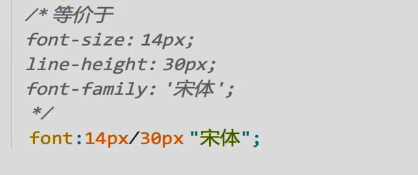
但是不影响继承来的权重，继承的权重就算是加!important，权重仍然是0

## （三）样式处理



### 1.字体样式

简写：



浏览器默认字体是16px=1em

font-style：斜体 oblique、italic；

font-weight: normal、bold、bolder、lighter <integer> 。

字体颜色：color

font-family

字体系列。

font-family可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。浏览器会使用它可识别的第一个值。

简单实例：

body {

font-family: "Microsoft Yahei", "微软雅黑", "Arial", sans-serif

}备用字体。如果用户没有第一个字体，浏览器会自动使用第二个，一次类推。

如果设置成inherit，则表示继承父元素的字体。

font-weight

字重（字体粗细）。

取值范围：

| **值** | **描述** |
| --- | --- |
| normal | 默认值，标准粗细 |
| bord | 粗体 |
| border | 更粗 |
| lighter | 更细 |
| 100~900 | 设置具体粗细，400等同于normal，而700等同于bold |
| inherit | 继承父元素字体的粗细值 |

font-size

字体大小。

p {

font-size: 14px;

}

如果设置成inherit表示继承父元素的字体大小值。

color

设置内容的字体颜色。

支持三种颜色值：

* 十六进制值 如: ＃FF0000
* 一个RGB值 如: RGB(255,0,0)
* 颜色的名称 如: red

p {

color: red;

}

### 2.文本样式

#### 鼠标形状设置



#### text-align

文本对齐

| **值** | **描述** |
| --- | --- |
| left | 左边对齐 默认值 |
| right | 右对齐 |
| center | 居中对齐 |
| justify | 两端对齐 |

text-align 属性规定元素中的文本的水平对齐方式。

#### line-height

行高。

单行文本：当line-height大于盒子高度时，文字会垂直居中。

对于多行文本垂直居中，需要计算盒子的高度，和每行的行高，进行平均。或者设置padding。

#### text-decoration

后面还可以跟颜色。text-decoration:underline blue

文字装饰。

| **值** | **描述** |
| --- | --- |
| none | 默认。定义标准的文本。 |
| underline | 定义文本下的一条线。 |
| overline | 定义文本上的一条线。 |
| line-through | 定义穿过文本的一条线。 |
| inherit | 继承父元素的text-decoration属性的值。 |
| unset |  |

text-shadow



向右偏移为正，向下偏移为正。

<**style type="text/css"**>  
 .**box1**{  
 **width**: 500**px**;  
 **height**: 200**px**;  
 **background-color**: **red**;  
 **color**: **#420377**;  
 **text-decoration**: **underline**;   
 */\*设置鼠标的形状\*/* **cursor**: **pointer**;   
 */\*设置首行缩进\*/* **text-indent**: 20**px**;  
 **font-size**: 25**px**;   
   
 */\*justify:两端对齐\*/* **text-align**: **justify**;  
  
 **text-shadow**: 0**px** 0**px** 1**px #fff**;  
  
 */\*行高\*/  
   
 /\*当行高=盒子的高度时，那么会实现文本垂直方向居中\*/* **line-height**: 200**px**;  
 **text-align**: **center**;  
 }  
</**style**>

#### text-indent：

首行缩进，单位以em为准。

### 3.背景样式



#### 背景属性

#### background-image

常用背景相关属性：

| **属性** | **描述** |
| --- | --- |
| background-color | 规定要使用的背景颜色。 |
| background-image | 规定要使用的背景图像。 |
| background-size | 规定背景图片的尺寸。 |
| background-repeat | 规定如何重复背景图像。 |
| background-attachment | 规定背景图像是否固定或者随着页面的其余部分滚动。 |
| background-position | 规定背景图像的位置。 |
| inherit | 规定应该从父元素继承background属性的设置。 |

background-attachment:fixed 背景固定。

background-repeat取值范围：

| **值** | **描述** |
| --- | --- |
| repeat | 默认。背景图像将在垂直方向和水平方向重复。 |
| repeat-x | 背景图像将在水平方向重复。 |
| repeat-y | 背景图像将在垂直方向重复。 |
| no-repeat | 背景图像将仅显示一次。 |
| inherit | 规定应该从父元素继承background-repeat属性的设置。 |

background-attachment取值范围：

| **值** | **描述** |
| --- | --- |
| scroll | 默认值。背景图像会随着页面其余部分的滚动而移动。 |
| fixed | 当页面的其余部分滚动时，背景图像不会移动。 |
| inherit | 规定应该从父元素继承background-attachment属性的设置。 |

background-position取值范围：

| **值** | **描述** |
| --- | --- |
| top left  top center  top right  center left  center center  center right  bottom left  bottom center  bottom right | 如果只设置了一个关键词，那么第二个值就是"center"。 默认值：0% 0%。 |
| x% y% | 第一个值是水平位置，第二个值是垂直位置。 左上角是 0% 0%。右下角是 100% 100%。 如果只设置了一个值，另一个值就是50%。 |
| xpos ypos | 第一个值是水平位置，第二个值是垂直位置。  数值可以为负值，负值为向左、向上偏移。  左上角是 0 0。单位是像素 (0px 0px) 或任何其他的 CSS 单位。 如果只设置了一个值，另一个值就是50%。 可以混合使用%和position值。 |

示例：

body {

background-color: red;

backgraound-image: url(xx.png);

background-size: 300px 300px;

background-repeat: no-repeat;

background-position: center

}

简写：

body {

background: red url(xx.png) no-repeat fixed center/300px 300px;

}

#### 示例

<**style type="text/css"**>  
 **#img**{  
 **width**: 990**px**;  
 **height**: 980**px**;  
  
   
 */\*设置背景颜色\*/* **background-color**: **yellowgreen**;  
  
 */\*设置背景图像\*/* **background-image**: **url**(**./images/nvshen.jpg**); **默认情况下为平铺  
  
 background-repeat**: **no-repeat**; **取消平铺效果 repeat-x 水平方向平铺 repeat-y 垂直方向平铺  
  
 background-position**: -50**px** -20**px**; **图片偏移** */\*简写\*/* **background**: **url**(**./images/banner.jpg**) **no-repeat** 10**px** 20**px**;  
 **background**: **red**;  
  
 **padding-top**: 10**px**;  
 **border**: 5**px solid black**;  
  
 }

### 4.微调图片上下位置 vertical-align

img

{

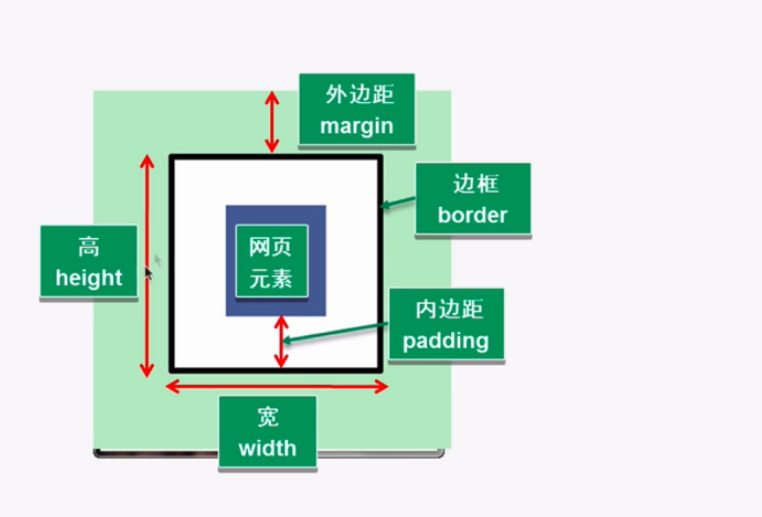
vertical-align:text-top;

}

微调图片的上下位置，可以用 负值。

## （四）盒模型

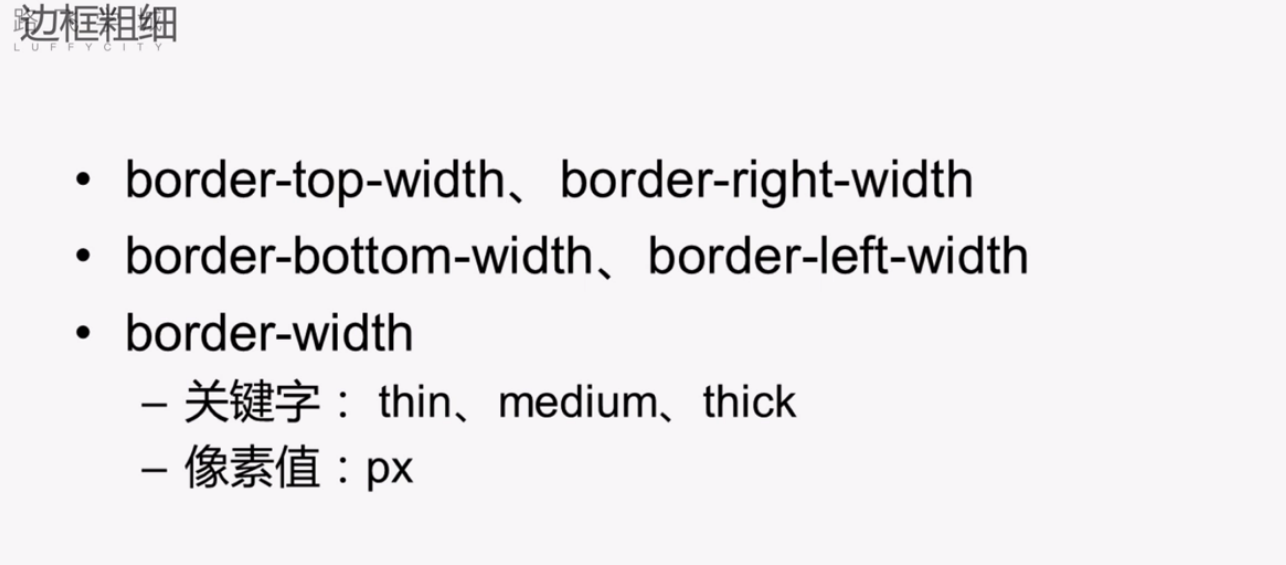
盒子套盒子。



#### 1.border

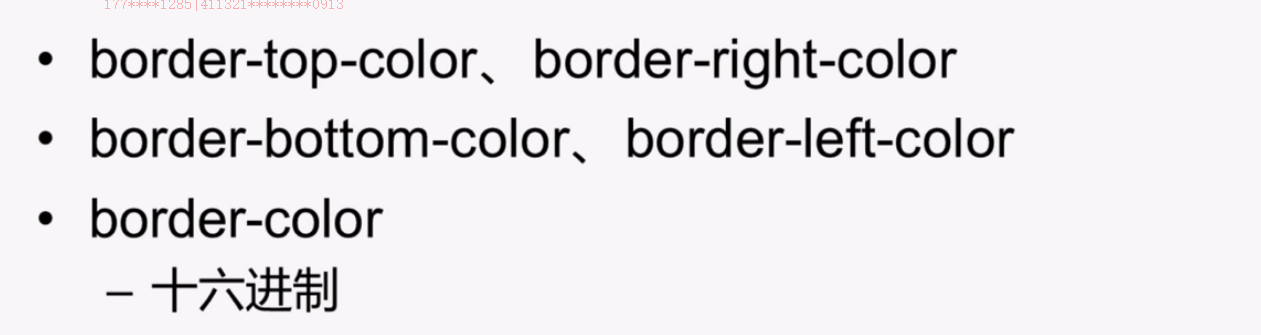
##### 边框样式

##### 边框粗细



##### 边框颜色

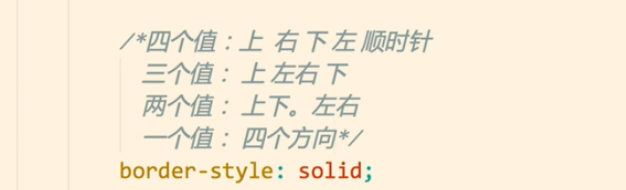
bolder-color



##### 简写



从上面面开始，按照以顺时针方向设置线型。



可以倒换顺序。

#### 2.margin 外边距

margin： 外边距 控制元素与元素之间的距离 1）margin 的4个方向 2）margin会改变实际大小，背景色不会渲染到margin区域 这个区域会以空白显示，但是也属于盒子的一部分 3)margin是添自身的,如果哪个想要改变自己的位置，就给谁添加margin html的部分标签自带margin padding p body ul

##### 边距

margin-left、top、right、bottom 距离外盒上下左右的距离。

**margin**: 20**px** 30**px** 40**px** 50**px**;  
**margin**: 20**px** 40**px**;  
  
**margin-bottom**: 50**px**;

**margin的塌陷：当一个盒子的下边距与另一个盒子的上边距相遇时，边距会以大的为准，不会叠加**

**包括相邻的盒子和嵌套的盒子。**

**嵌套的盒子与相邻的盒子。对于外边距合并，嵌套的盒子看做是一体的，与相邻的边距。**

**相邻盒子：左右、上下**

**对于左右边距，则不存在塌陷问题。**

**解决的方法：1.给父级盒子添加个边框，就可以相互分开。**

**2.使用浮动或绝对定位**

**3.给父级添加padding**

**4.父级overflow hidden()**  
  
**margin**: 0 **auto**; **使盒子元素水平居中显示  
margin**: 0 **auto**;  
**border**:1**px solid green**;

水平居中：margin:0 auto 水平居中盒子。要有明确的width。

#### 3.padding内边距

padding:内边距 控制内容到边框的距离 1.只写一个padding代表四个方向，也可以单独指定某一个方向 2.元素设置了padding值是额外家在原来大小之上的width+padding。如果不想改变实际大小，那就在width-padding对应方向的值 3.padding是添加给父级的，改变的是父级包含的内容的位置，自身位置不变 4.简写方法 1）一个值：4个方向 2）两个值：上下 左右 3）三个值：上 左右 下 4）四个值：上 右 下 左（顺时针方向） 注意：padding不支持负值

padding-left/top/right/bottom

**内边距：内容区域到边框的距离  
 内边距会扩大元素所在的区域  
  
 注意：为元素设置内边距 只能影响自己。不会影响其他的元素**

#### 清楚默认样式

\*{

padding:0;

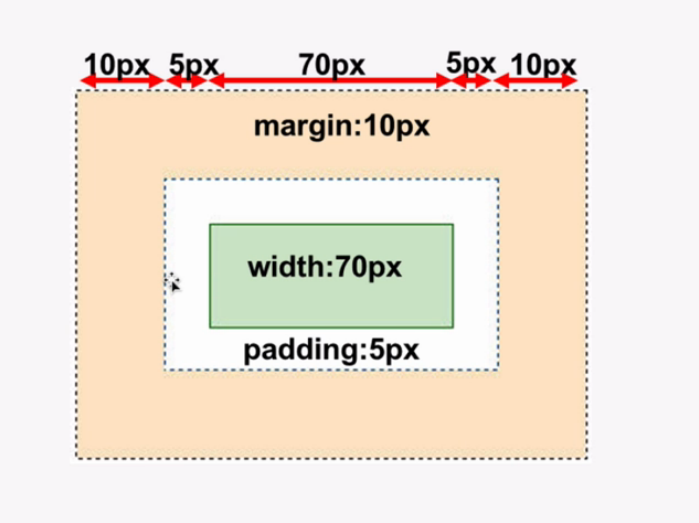
margin:0;

}

缺点：\*效率比较慢。

可以在网上搜索reset css ，将其作为一个css文件，引入到程序中。

#### 4.盒子模型尺寸计算



## 标准文档流

### 标准文档流下的微观现象

#### 1.空白折叠现象。

文档流中的多的空格在显示中会被忽略。使用空格可以用&….表示

所有的换行会被转换成一个空格显示。

#### 2.高矮不齐，底端对齐。

在同一行中，元素高矮不齐时，底端是对齐的。

可以用text-height来上下居中。

text-align：center 文字水平居中。

#### 3.自动换行。

## （五）display的用法

父级：display：table

子元素：display：table-cell

子级元素按照单元格的形式在父级元素中排列。



### 1.隐藏：

**div**{  
 **width**: 100**px**;  
 **height**: 100**px**;  
 **background-color**: **red**;  
  **/\*控制元素 隐藏。不占位置\*/ ①两种隐藏方式，第一种在不占浏览器位置，完全没有   
 /\*display: none;\*/**  
 **/\*控制元素隐藏。占位置 影响 后面盒子的布局\*/ ②第二种，仅仅是不显示，位置还留着  
 /\*visibility: hidden;\*/** **/\*display: inline-block;\*/** 设置为行内块显示，在一行内显示，可设置宽高。 */\*将 块级元素 转换为行内元素\*/* **display**: **inline**;

}

行内块元素转为行内元素无效

行内元素可以转为行内块元素。

## （六）定位处理

### 1.浮动 float



#### **浮动产生的效果**

<!-- left 左浮动 左侧为起始。从左往右依次排列   
 right 右浮动 右侧为起始。从右往左依次排列  
   
 文档流： 可见元素在文档中的显示位置  
   
   
  
 浮动可以使元素按指定位置排列，直到遇到父元素的边界或另一个元素的边界停止，如果没有排列完，在另起一行排列。

块级元素也在同行显示。

#### 浮动特性

\*{

paddint:0

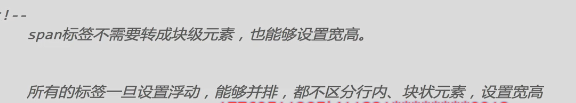
margin:0 清楚默认边距

}

浮动在文档流上面，不占位置。

<!-- 文档流： 可见元素在文档中的排列位置  
   
 浮动效果  
 1.元素的脱标

浮动可以使元素脱离标准文档流 ，不占位置，下方的元素未浮动的元素可以占用该位置。



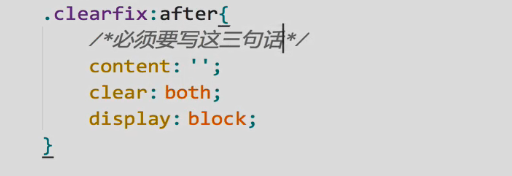
2.浮动会使元素提升层级  
 3.浮动可以使块元素在一行排列，不设置宽高的时候，可以使元素适应内容  
 4.浮动可以使行内元素支持宽高

5.浮动的字围效果：当浮动的盒子遮挡了另外一个盒子（盒子2）时，盒子2中的文字会自动缩进，不会被浮动的盒子遮挡。

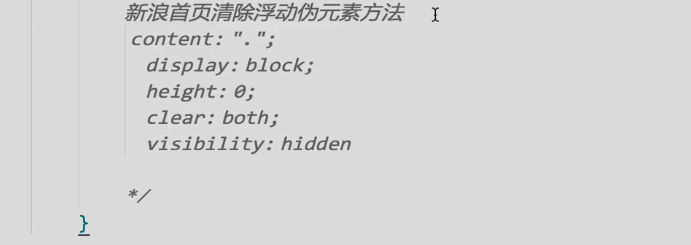
6.浮动的紧凑效果：一个浮动元素，如果不设置宽高，则自动按照元素内部的内容大小进行自适应大小。与行内元素较像

#### 浮动的问题及解决

<!-- 浮动产生的问题：  
父元素不设置高度时，子元素设置了浮动，不会撑开父元素的高度。因为子元素 不占位置   
  
解决办法  
1.给父盒子设置固定高度。不灵活

2.内墙法：给浮动元素最后一个加一个空的div，且该元素不浮动，设置clear:both  
 意思为：清除别人对我的浮动影响。  
结构冗余  
  
3.

或者：



.clearfix为未设置高度的祖先元素类名。  
4.使用给父元素添加overflow:hidden

#### 清除浮动

clear:





### 2.滚动overflow



水平方向 overflow-x

垂直方向 overflow-y

所有方向 overflow

### 3.定位position

大的盒子用定位，盒子里面的小盒子用定位。

相对定位和绝对定位都能是元素脱离文档流，但是绝对定位可以完全无视元素的初始位置，而相对定位却需要注意元素的初始位置。

对于相对定位：虽然元素位置有所移动，但是元素原位置仍然保留，不会被其他元素占用。除非其他元素设置为float，才会占用该位置。

如：如果设置为相对定位的元素（A）在其他元素前面，则A仍然有独占一行的特性，所有A的位置不会被其他元素填充，仍然独占一行，即使A被移到他处，A的原位置仍然被保留。

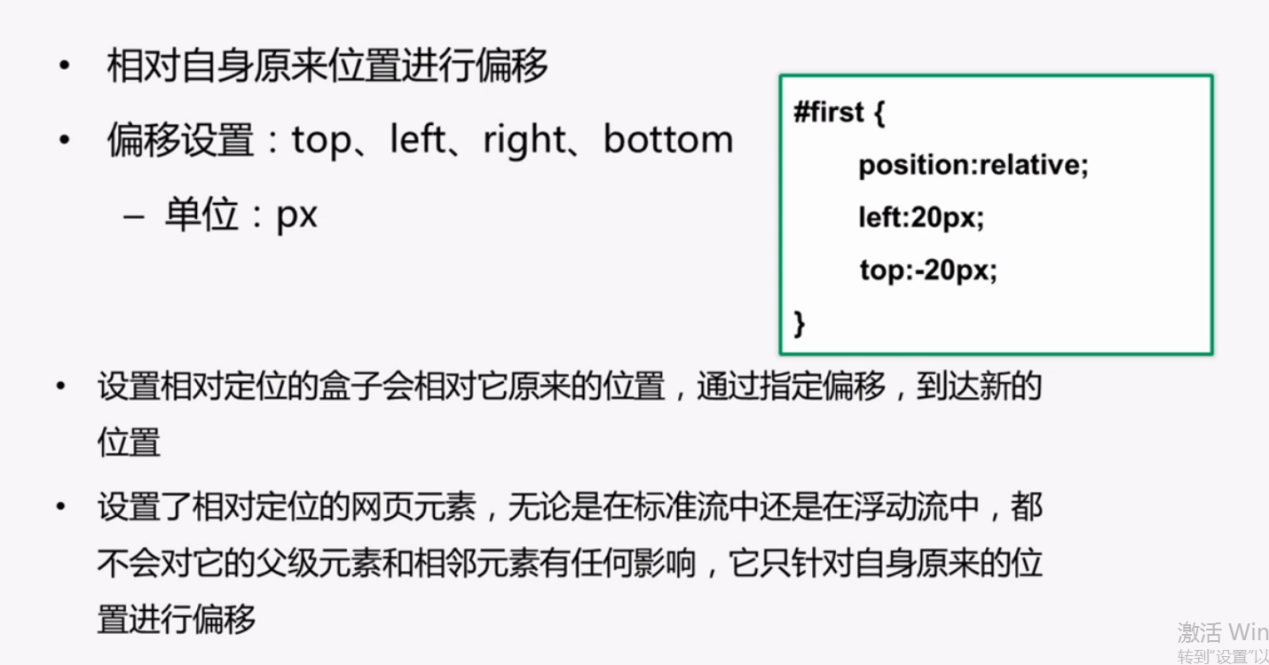
绝对定位：则是完全脱离文档流，原位置也不会被保留，会被其他元素占用。

#### 相对定位

常用于 **微调** 盒子位置

相对于自身现在位置定位。

数值可以设置为%。



1.不设置偏移量的时候 对元素没有任何影响  
2.可以提升层级关系 ：设置定位的将会在上面显示，两个都设置定位的，则后面设置的要在前面设置的上面。

3.不脱标、形影分离、老家留坑。

设置偏移量时，盒子显示位置虽然改变，但是原有的位置继续保留，会影响页面布局。

常用于微调盒子位置。

#### 绝对定位

position:absolute

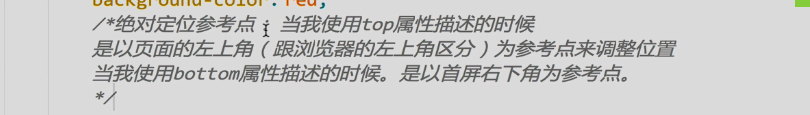
absolute：绝对定位

1.完全脱标，原来的位置不被保留。如果不设置top等数值，则还在原来的位置显示该盒子。但是下面的盒子也会占领该盒子的位置，因此形成遮盖效果。

2.提升层级，做遮盖效果

3.设置绝对定位后，不再区分块级元素和行内元素，都能设置宽高。

相对于页面的参考点：



在父级元素没有定位时，以浏览器的左上角为基准  
有父级的情况下，父级设置相对定位，子级设置绝对定位 是以父盒子进行定位  
  
“父相子绝”、“父绝子绝”、“父固子绝”  
父相：只要是祖辈设置为相对定位就可以，不一定非得是父辈。

绝对定位的盒子无视父辈的padding。

绝对定位盒子的居中：



1.提升层级关系  
2.脱离文档流

float 和 position 联合使用

#### 固定定位

相对于显示器的位置是不变的，随着页面滚动而滚动。

### 4. z-index的使用

数越大层级越高，越在上面

1.数值越高，层级越高，越在上层。

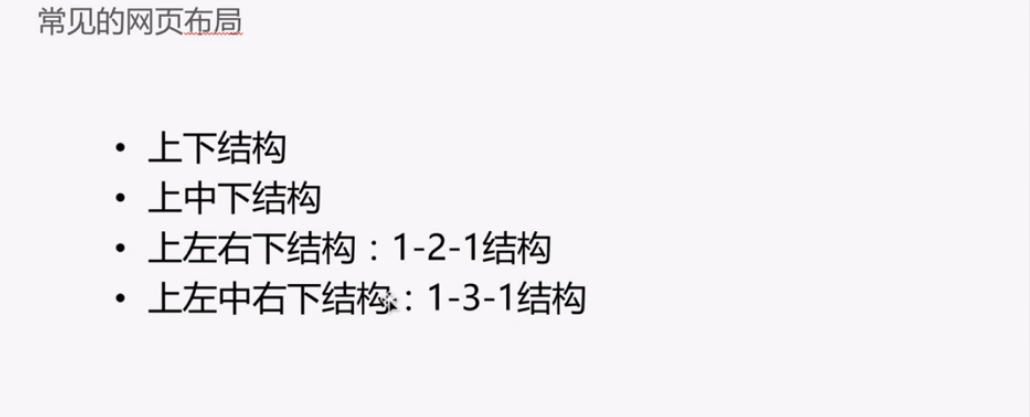
2.只有定位了的，才能使用z-index，绝对定位、固定定位、相对定位可以使用，浮动不能使用。

3.z-inde没有单位，默认值为0

4.如果z-index的值一样，谁写在html的后面，谁层级高。定位的元素永远压着未定位的元素。

5.从父现象：父级层级低，则后背元素层级高也没用。

## （七）网页布局



# 四、技巧

1.将图片、链接居中：在上一级盒子使用 text-align：center

2.图片或链接与盒子底部对不齐：可能是盒子有行距的问题，将盒子行距line-height：0

3.盒子超链接的字上下不居中时，可以将盒子的line-height的高度设置和父盒height一样。

4.画圆圈：border:solid white 1px;  
border-radius: 50%;

border-radius:1px 圆角

5.调节上下左右位置

对于块级元素，上下左右调整使用margin和padding

对于非块级元素，上下位置调节，需要调节父盒的line-height

左右调节需要用text-align

6.嵌套

把非块级元素设置为块级元素后，也可以嵌套块级元素

7.盒阴影：box-shadow

8.对于相对定位和绝对定位

相对定位和绝对定位都能是元素脱离文档流，但是绝对定位可以完全无视元素的初始位置，而相对定位却需要注意元素的初始位置。

**对于相对定位**：虽然元素位置有所移动，但是元素原位置仍然保留，不会被其他元素占用。除非该相对定位元素设置为float，其他元素才会占用该位置。

对于行内元素、行内块元素升为块级元素的，可能不会占用该位置，需要用clear:both，来清除自带。来确保元素独占一行。

如：如果设置为相对定位的元素（A）在其他元素前面，则A仍然有独占一行的特性，所有A的位置不会被其他元素填充，仍然独占一行，即使A被移到他处，A的原位置仍然被保留。

**绝对定位**：则是完全脱离文档流，原位置也不会被保留，会被其他元素占用。

9.下划线无法设置粗细，可以用元素的下边框来代替。可以设置粗细。并利用line-height来调整距离文字的位置。