1.1HTML中Head头

笔记本： 任务六文本笔记
创建时间： 2018/4/25 星期三 下午 4:18 **更新时间：** 2018/4/25 星期三 下午 5:54
作者： 18291893776@139.com
URL： file:///E:/python%E7%B3%BB%E7%BB%9F%E7%8F%AD%E8%AF%BE%E4%BB%B6/%E4%BB%BB%E5%8A%A1%E5%85%AD/...

1.1HTML中Head头

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>网页标题</title>
</head>
<body>
网页显示内容
</body>
</html>
```

HEAD标签里面负责对网页进行一些设置以及定义标题，

设置包括定义网页的编码格式，外链css样式文件和javascript文件等，

设置的内容不会显示在网页上，标题的内容会显示在标题栏

```
设置网页编码：<meta charset="utf-8"/>
关键字：<meta name="Keywords" content="关键字" />
描述： <meta name="Description" content="简介、描述" />
网页标题：<title>本网页标题</title>
导入CSS文件：<link type="text/css" rel="stylesheet" href="**.css"/>
CSS代码：<style type="text/css">嵌入css样式代码</style>
JS文件或代码： <script >。。。</script>
... ...
```

1.2HTML标题

通过 <h1>、<h2>、<h3>、<h4>、<h5>、<h6>,标签可以在网页上定义6种级别的标题。

6种级别的标题表示文档的6级目录层级关系，比如说：<h1>用作主标题（最重要的），其后是 <h2>（次重要的），

再其次是 <h3>，以此类推。搜索引擎会使用标题将网页的结构和内容编制索引，所以网页上使用标题是很重要的。

```
<h1>这是一级标题</h1>
<h2>这是二级标题</h2>
<h3>这是三级标题</h3>
```

1.3HTML链接

2.1Css基本语法及页面引用

css基本语法

css的定义方法是：

选择器 { 属性:值; 属性:值; 属性:值;}

选择器是将样式和页面元素关联起来的名称，属性是希望设置的样式属性每个属性有一个或多个值。代码示例：

```
div{ width:100px; height:100px; color:red }
```

css页面引入方法：

1、外联式：通过link标签，链接到外部样式表到页面中。

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

2、嵌入式：通过style标签，在网页上创建嵌入的样式表。

```
<style type="text/css">
div{ width:100px; height:100px; color:red }
.....
</style>
```

3、内联式：通过标签的style属性，在标签上直接写样式。

```
<div style="width:100px; height:100px; color:red ">
.....
</div>
```

2.2 Css选择器

常用的选择器有如下几种：

1、标签选择器

标签选择器，此种选择器影响范围大，建议尽量应用在层级选择器中。

举例：

```
*{margin:0;padding:0}
div{color:red}
<div>....</div> <!-- 对应以上两条样式 -->
<div class="box">....</div> <!-- 对应以上两条样式 -->
```

2、id选择器

通过id名来选择元素，元素的id名称不能重复，所以一个样式设置项只能对应于页面上一个元素，不能复用，id名一般给程序使用，所以不推荐使用id作为选择器。

举例：

```
#box{color:red}
<div id="box">....</div> <!-- 对应以上一条样式，其它元素不允许应用此样式 -->
```

<a>标签可以在网页上定义一个链接地址，通过href属性定义跳转的地址，通过title属性定义鼠标悬停时弹出的提示文字框。

```
<a href="#"></a> <!-- # 表示链接到页面顶部 -->
<a href="http://www.itxdl.cn/" title="跳转的it兄弟连网站">兄弟连</a>
<a href="2.html">测试页面2</a>
```

锚点:定义页面内滚动跳转

页面内定义了“id”，可以通过a标签链接到它的页面滚动位置，前提是页面要足够高，有滚动条，且元素不能在页面顶部

```
<a href="#mao1">标题一</a>
……
……
<h3 id="mao1">跳转到的标题</h3>
```

属性：

1. href必须，指的是链接跳转地址
2. target: 表示链接的打开方式： _blank 新窗口
3. title属性定义鼠标悬停时弹出的提示文字框

1.4 HTML列表

有序列表

在网页上定义一个有编号的内容列表可以用<ol>、<li>配合使用来实现，代码如下：

```
<ol>
<li>列表文字一</li>
<li>列表文字二</li>
<li>列表文字三</li>
</ol>
```

在网页上生成的列表，每条项目上会按1、2、3编号，有序列表在实际开发中较少使用。

其中type类型值：A a I i 1 start属性表示起始值

无序列表

在网页上定义一个无编号的内容列表可以用<ul>、<li>配合使用来实现，代码如下：

```
<ul>
<li>列表文字一</li>
<li>列表文字二</li>
<li>列表文字三</li>
</ul>
```

在网页上生成的列表，每条项目上会有一个小图标，这个小图标在不同浏览器上显示效果不同，所以一般会用样式去掉默认的小图标，如果需要图标，可以用样式自定义图标，从而达到在不同浏览器上显示的效果相同,实际开发中一般用这种列表。

定义列表

定义列表通常用于术语的定义。<dl>标签表示列表的整体。<dt>标签定义术语的题目。<dd>标签是术语的解释。一个<dl>中可以有多个题目和解释，代码如下：

3、类选择器

通过类名来选择元素，一个类可应用于多个元素，一个元素上也可以使用多个类，应用灵活，可复用，是css中应用最多的一种选择器。

举例：

```
.red{color:red}
.big{font-size:20px}
.mt10{margin-top:10px}
<div class="red">....</div>
<h1 class="red big mt10">....</h1>
<p class="red mt10">....</p>
```

4、层级选择器

主要应用在选择父元素下的子元素，或者子元素下面的子元素，可与标签元素结合使用，减少命名，同时也可以通过层级，防止命名冲突。

举例：

```
.box span{color:red}
.box .red{color:pink}
.red{color:red}
<div class="box">
<span>....</span>
<a href="#" class="red">....</a>
</div>
<h3 class="red">....</h3>
```

5、组选择器

多个选择器，如果有同样的样式设置，可以使用组选择器。也成为 并列选择

举例：

```
.box1,.box2,.box3{width:100px;height:100px}
.box1{background:red}
.box2{background:pink}
.box2{background:gold}
<div class="box1">....</div>
<div class="box2">....</div>
<div class="box3">....</div>
```

6、伪类及伪元素选择器

常用的伪类选择器有hover，表示鼠标悬浮在元素上时的状态，伪元素选择器有before和after,它们可以通过样式在元素中插入内容。

```
.box1:hover{color:red}
<div class="box1">....</div>
a:hover {color: #FF00FF; text-decoration: underline} /* 鼠标在该元素上时 */
a:before{content:"Hello";} /*在每个<a>元素之前插入内容*/
a:after{content:"world";} /*在每个<a>元素之后插入内容*/
```

2.3 Css颜色,文本字体

css颜色表示法

1. 颜色名表示，比如：red 红色，gold 金色
2. 16进制数值表示，比如：#ff0000 表示红色，这种可以简写成 #f00
3. RGB颜色: 红(R)、绿(G)、蓝(B)三个颜色通道的变化 background-color: rgba(200,100,0);
4. RGBA颜色: 红(R)、绿(G)、蓝(B)、透明度(A) background-color: rgba(0,0,0,0.5);

```
<h3>前端三大块</h3>
<dl>
<dt>html</dt>
<dd>负责页面的结构</dd>
<dt>css</dt>
<dd>负责页面的表现</dd>
<dt>javascript</dt>
<dd>负责页面的行为</dd>
</dl>
```

1.5 HTML表格

table常用标签

1、table标签：声明一个表格

2、tr标签：定义表格中的一行

3、td和th标签：定义一行中的一个单元格，td代表普通单元格，th表示表头单元格

table常用属性：

1、border 定义表格的边框

2、cellpadding 定义单元格内内容与边框的距离

3、cellspacing 定义单元格与单元格之间的距离

4、align 设置单元格中内容的水平对齐方式,设置值有：left | center | right

5、valign 设置单元格中内容的垂直对齐方式 top | middle | bottom

6、colspan 设置单元格水平合并

7、rowspan 设置单元格垂直合并

1.6 HTML表单

表单用于搜集不同类型的用户输入，表单由不同类型的标签组成，实现一个特定功能的表单区域（比如：注册），

首先应该用<form>标签来定义表单区域整体，在此标签中再使用不同的表单控件来实现不同类型的信息输入，

具体实现及注释可参照以下伪代码：

```
<!-- form定义一个表单区域,action属性定义表单数据提交的地址，method属性定义提交的方式。 -->
<form action="http://www..." method="get">
<!-- label标签定义表单控件的文字标注，input类型为text定义了
一个单行文本输入框 -->
<p>
<label>姓名：</label><input type="text" name="username" />
</p>
<!-- input类型为password定义了一个密码输入框 -->
<p>
<label>密码：</label><input type="password" name="password" />
</p>
<!-- input类型为radio定义了单选框 -->
```

css文本设置

常用的应用文本的css样式：

color 设置文字的颜色，如： color:red;
font-size 设置文字的大小，如：font-size:12px;
font-family 设置文字的字体，如：font-family:'微软雅黑';
font-style 设置字体是否倾斜，如：font-style:'normal'; 设置不倾斜，font-style:'italic';设置文字倾斜
font-weight 设置文字是否加粗，如：font-weight:bold; 设置加粗 font-weight:normal 设置不加粗
font 同时设置文字的几个属性，写的顺序有兼容问题，建议按照如下顺序写：
font：是否加粗 字号/行高 字体；如： font:normal 12px/36px '微软雅黑';
line-height 设置文字的行高，如：line-height:24px;
text-decoration 设置文字的下划线，如：text-decoration:none; 将文字下划线去掉
text-indent 设置文字首行缩进，如：text-indent:24px; 设置文字首行缩进24px
text-align 设置文字水平对齐方式，如text-align:center 设置文字水平居中

2.4 边框,背景,边距,溢出

css边框属性

border:宽度 样式 颜色;
border-color;
border-style; 边框样式：solid实现，dotted点状线，dashed虚线
border-width:
border-left-color;
border-left-style;
border-left-width:
CSS3的样式
border-radius：圆角处理
box-shadow: 设置或检索对象阴影

背景属性：background

*background-color: 背景颜色
*background-image: 背景图片
*background-repeat：是否重复，如何重复?(平铺)
*background-position：定位
background-attachment： 是否固定背景，
scroll:默认值。背景图像是随对象内容滚动
fixed:背景图像固定
css3的属性
*background-size: 背景大小，如 background-size:100px 140px;
多层背景：
background:url(test1.jpg) no-repeat scroll 10px 20px,url(test2.jpg) no-repeat scroll 50px 60px,url(test3.jpg) no-repeat scroll 90px 100px;
background-origin:content-box,content-box,content-box;
background-clip:padding-box,padding-box,padding-box;
background-size:50px 60px,50px 60px,50px 60px;

元素溢出

当子元素的尺寸超过父元素的尺寸时，需要设置父元素显示溢出的子元素的方式，设置的方法是通过overflow属性来设置。

overflow的设置项：
1、visible 默认值。内容不会被修剪，会呈现在元素框之外。

```
<p>
<label>性别：</label>
<input type="radio" name="gender" value="0"
/> 男
<input type="radio" name="gender" value="1"
/> 女
</p>
<!-- input类型为checkbox定义了单选框 -->
<p>
<label>爱好：</label>
<input type="checkbox" name="like"
value="sing" /> 唱歌
<input type="checkbox" name="like" value="run"
/> 跑步
<input type="checkbox" name="like"
value="swiming" /> 游泳
</p>
<!-- input类型为file定义上传照片或文件等资源 -->
<p>
<label>照片：</label>
<input type="file" name="person_pic">
</p>
<!-- textarea定义多行文本输入 -->
<p>
<label>个人描述：</label>
<textarea name="about"></textarea>
</p>
<!-- select定义下拉列表选择 -->
<p>
<label>籍贯：</label>
<select name="site">
<option value="0">北京</option>
<option value="1">上海</option>
<option value="2">广州</option>
<option value="3">深圳</option>
</select>
</p>
<!-- input类型为submit定义提交按钮
还可以用图片控件代替submit按钮提交，一般会导致提
交两次，不建议使用。如：
<input type="image" src="xxx.gif">
-->
<p>
<input type="submit" name="" value="提交">
<!-- input类型为reset定义重置按钮 -->
<input type="reset" name="" value="重置">
</p>
</form>
```

input表单项中的属性，可以提供

*type属性:表示表单项的类型:值如下:
text:单行文本框
password:密码输入框
checkbox:多选框 注意要提供value值
radio:单选框 注意要提供value值
file:文件上传选择框
button:普通按钮
submit:提交按钮
image:图片提交按钮
reset:重置按钮, 还原到开始\(第一次打开时\)的效果
hidden:主表单隐藏域,要是和表单一块提交的信息,但是不需要用户修改
*name属性:表单项名,用于存储内容值的
*value属性:输入的值\(默认指定值\)
size属性:输入框的宽度值

2、hidden 内容会被修剪，并且其余内容是不可见的，此属性还有清除浮动、清除margin-top塌陷的功能。
3、scroll 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。
4、auto 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。

5、inherit 规定应该从父元素继承 overflow 属性的值。

CSS边距:

*内补白（内补丁）

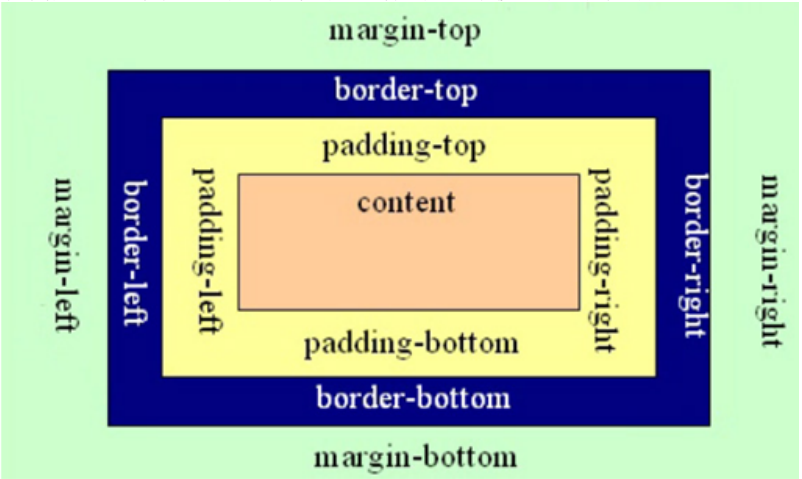
padding： 检索或设置对象四边的内部边距,如padding:10px; padding:5px 10px;
padding-top： 检索或设置对象顶边的内部边距
padding-right： 检索或设置对象右边的内部边距
padding-bottom：检索或设置对象下边的内部边距
padding-left： 检索或设置对象左边的内部边距

*外补白（外补丁）

margin： 检索或设置对象四边的外延边距,如 margin:10px; margin:5px auto;
margin-top： 检索或设置对象顶边的外延边距
margin-right： 检索或设置对象右边的外延边距
margin-bottom： 检索或设置对象下边的外延边距
margin-left： 检索或设置对象左边的外延边距

2.5 盒子

盒子模型解释
元素在页面中显示成一个方块，类似一个盒子，CSS盒子模型就是使用现实中盒子来做比喻，帮助我们设置元素对应的样式。盒子模型示意图如下：



把元素叫做盒子，设置对应的样式分别为：盒子的边框(border)、盒子内的内容和边框之间的间距(padding)、盒子与盒子之间的间距(margin)。

设置边框
设置一边的边框，比如顶部边框，可以按如下设置：

```
border-top-color:red;     /* 设置顶部边框颜色为红色 */
border-top-width:10px;    /* 设置顶部边框粗细为10px */
border-top-style:solid;   /* 设置顶部边框的线性为实线，常用的有：
solid(实线)
  dashed(虚线)  dotted(点线); */
```

上面三句可以简写成一句：

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

设置其它三个边的方法和上面一样，把上面的'top'换成'left'就是设置左边，换成'right'就是设置右边，换成'bottom'就是设置底边。

maxlength属性:输入框的输入内容的最大长度
readonly属性:对输入框只读属性
*disabled属性:禁用属性
*checked属性:对选择框指定默认选项
src和alt是为图片按钮设置的
注意：reset重置按钮是将表单数据恢复到第一次打开时的状态，并不是清空
image图片按钮，默认具有提交表单功能。

---

3. JavaScript

3.1JavaScript嵌入页面的方式

1、行间事件（主要用于事件）

```
<input type="button" name=""
onclick="alert(&amp;apos;ok！&amp;apos;);">
```

2、页面script标签嵌入

```
<script type="text/javascript">
var a = ‘你好’;
alert(a);
</script>
```

3、外部引入

```
<script type="text/javascript" src="js/index.js">
</script>
```

javascript语句与注释

1、一条javascript语句应该以“;”结尾

```
<script type="text/javascript">
var a = 123;
var b = &amp;apos;str&amp;apos;;
function fn(){
alert(a);
};
fn();
</script>
```

2、javascript注释

```
<script type="text/javascript">
// 单行注释
var a = 123;
/*
多行注释
1、...
2、...
*/
var b = &amp;apos;str&amp;apos;;
</script>
```

3.2js变量

JavaScript 是一种弱类型语言，javascript的变量类型由它的值来决定。 定义变量需要用关键字 'var'

```
var a = 123;
```

四个边如果设置一样，可以将四个边的设置合并成一句：

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

设置内间距padding

设置盒子四边的内间距，可设置如下：

```
padding-top: 20px;      /* 设置顶部内间距20px */
padding-left:30px;      /* 设置左边内间距30px */
padding-right:40px;     /* 设置右边内间距40px */
padding-bottom:50px;    /* 设置底部内间距50px */
```

上面的设置可以简写如下：

```
padding: 20px 40px 50px 30px; /* 四个值按照顺时针方向，分别设置的是 上 右 下 左
四个方向的内边距值。 */
```

padding后面还可以跟3个值，2个值和1个值，它们分别设置的项目如下：

```
padding: 20px 40px 50px; /* 设置顶部内边距为20px，左右内边距为40px，底部内边距为50px */
padding: 20px 40px; /* 设置上下内边距为20px，左右内边距为40px*/
padding: 20px; /* 设置四边内边距为20px */
```

设置外间距margin

外边距的设置方法和padding的设置方法相同，将上面设置项中的 'padding'换成'margin'就是外边距设置方法。

盒子模型的尺寸

按照下面代码制作页面：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>盒子的真实尺寸</title>
    <style type="text/css">
        .box01{width:50px;height:50px;background-color:gold;}
        .box02{width:50px;height:50px;background-color:gold;border:50px
         solid #000}
        .box03{width:50px;height:50px;background-color:gold;border:50px
         solid #000;padding: 50px;}
    </style>
</head>
<body>
    <div class="box01">1</div>
    <br />
    <div class="box02">2</div>
    <br />
    <div class="box03">3</div>
</body>
</html>
```

页面显示效果如下：

var b = 'asd';
//同时定义多个变量可以用","隔开，公用一个‘var’关键字
var c = 45,d='qwe',f='68';

变量、函数、属性、函数参数命名规范

字母数字下划线($)
首字母不能为数字
严格区分大小写
不能使用关键字

基本数据类型：

typeof函数获取一个变量的类型：

* boolean - 如果变量是 Boolean 类型的
* number - 如果变量是 Number 类型的 (整数、浮点数)
* string - 如果变量是 String 类型的 （采用""、''）
* object - 如果变量是一种引用类型或 Null 类型的 如：new Array()/ new String()...
* function -- 函数类型
* undefined - 如果变量是 Undefined 类型的

3.3 js数据类型转换

使用：Number（）、parseInt() 和parseFloat（） 做类型转换

Number()强转一个数值(包含整数和浮点数)。
*parseInt()强转整数，
*parseFloat（）强转浮点数

函数isNaN()检测参数是否不是一个数字。

isNaN() is not a number

ECMAScript 中可用的 3 种强制类型转换如下：

Boolean(value) - 把给定的值转换成 Boolean 型；
Number(value) - 把给定的值转换成数字（可以是整数或浮点数）；
String(value) - 把给定的值转换成字符串；

3.4js运算符

算 字 赋 比 逻 位 它

算术运算符

+ - * / ++ --

字符串连接

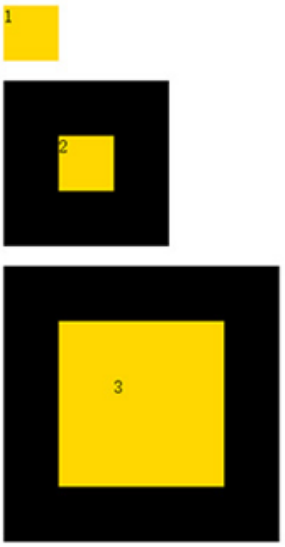
+

赋值运算

= += -= %=

比较运算符

< > >= <= == === != !==



通过上面的页面得出结论：盒子的width和height设置的是盒子内容的宽和高，不是盒子本身的宽和高，盒子的真实尺寸计算公式如下：

- 盒子宽度 = width + padding左右 + border左右
- 盒子高度 = height + padding上下 + border上下

思考题：
1.在布局中，如果我想增大内容和边框的距离，又不想改变盒子显示的尺寸，应该怎么做？

课堂练习
请制作图中所示的标题：

margin相关技巧
1、设置元素水平居中： margin:x auto;
2、margin负值让元素位移及边框合并

外边距合并

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。解决方法如下：

1、使用这种特性
2、设置一边的外边距，一般设置margin-top
3、将元素浮动或者定位

margin-top 塌陷

在两个盒子嵌套时候，内部的盒子设置的margin-top会加到外边的盒子上，导致内部的盒子margin-top设置失败，解决方法如下：

1、外部盒子设置一个边框
2、外部盒子设置 overflow:hidden
3、使用伪元素类：

```
.clearfix:before{
    content: &amp;apos;&amp;apos;;
    display:table;
}
```

2.6 块元素,内联元素,内联块元素

元素就是标签，布局中常用的有三种标签，块元素、内联元素、内联块元素，了解这三种元素的特性，才能熟练的进行页面布局。

块元素
块元素，也可以称为行元素，布局中常用的标签如：div、p、ul、li、h1~h6、dl、dt、dd等等都是块元素，它在布局中的行为：

- 支持全部的样式
- 如果没有设置宽度，默认的宽度为父级宽度100%
- 盒子占据一行、即使设置了宽度

逻辑运算符

&& !! !

位运算

^ & | << >>

其它运算符

? : 三元运算符
delete：用于删除对象中属性的 如：delete o.name; //删除o对象中的name属性
void： void 运算符对任何值返回 undefined。没有返回值的函数真正返回的都是 undefined。
var iNum1=1, iNum2=2, iNum3=3;// 逗号运算符 用逗号运算符可以在一条语句中执行多个运算。

3.5 js流程控制

流程控制用于基于不同的条件来执行不同的动作。

If 语句

if... else ...

if ... else if ... else...

可以单分支,双分支,也可以多分支,需要注意 else if中间必须要有空格

```
if (condition){
当条件为 true 时执行的代码
}else{
当条件不为 true 时执行的代码
}
```

switch 语句

多分支语句： switch（）{。 case ：。。。。。}

switch 语句用于基于不同的条件来执行不同的动作。

```
switch(n){
case 1:
执行代码块 1
break;
case 2:
执行代码块 2
break;
default:
与 case 1 和 case 2 不同时执行的代码
}
```

3.6 js循环

程序中进行有规律的重复性操作，需要用到循环语句。

break 和 continue 语句对循环中的代码执行提供了更严格的控制。

for循环

```
for(var i=0;i<len;i++){
......
}
```

while循环

```
var i=0;
while(i<8){
```

内联元素
内联元素，也可以称为行内元素，布局中常用的标签如：a、span、em、b、strong、i等等都是内联元素，它们在布局中的行为：

- 支持部分样式（不支持宽、高、margin上下、padding上下）
- 宽高由内容决定
- 盒子并在一行
- 代码换行，盒子之间会产生间距
- 子元素是内联元素，父元素可以用text-align属性设置子元素水平对齐方式，用line-height属性值设置垂直对齐方式

解决内联元素间隙的方法
1、去掉内联元素之间的换行
2、将内联元素的父级设置font-size为0，内联元素自身再设置font-size

内联块元素
内联块元素，也叫行内块元素，是新增的元素类型，现有元素没有归于此类别的，img和input元素的行为类似这种元素，但是也归类于内联元素，我们可以用display属性将块元素或者内联元素转化成这种元素。它们在布局中表现的行为：

- 支持全部样式
- 如果没有设置宽高，宽高由内容决定
- 盒子并在一行
- 代码换行，盒子会产生间距
- 子元素是内联块元素，父元素可以用text-align属性设置子元素水平对齐方式，用line-height属性值设置子元素垂直对齐方式

这三种元素，可以通过display属性来相互转化，不过实际开发中，块元素用得比较多，所以我们经常把内联元素转化为块元素，少量转化为内联块，而要使用内联元素时，直接使用内联元素，而不用块元素转化了。

display属性
display属性是用来设置元素的类型及隐藏的，常用的属性有：
1、none 元素隐藏且不占位置
2、block 元素以块元素显示
3、inline 元素以内联元素显示
4、inline-block 元素以内联块元素显示

2.7 浮动

文档流
文档流，是指盒子按照html标签编写的顺序依次从上到下，从左到右排列，块元素占一行，行内元素在一行之内从左到右排列，先写的先排列，后写的排在后面，每个盒子都占据自己的位置。

浮动特性

1、浮动元素有左浮动(float:left)和右浮动(float:right)两种

2、浮动的元素会向左或向右浮动，碰到父元素边界、浮动元素、未浮动的元素才停下来

3、相邻浮动的块元素可以并在一行，超出父级宽度就换行

4、浮动让行内元素或块元素自动转化为行内块元素

5、浮动元素后面没有浮动的元素会占据浮动元素的位置，没有浮动的元素内的文字会避开浮动的元素，形成文字饶图的效果

6、父元素内整体浮动的元素无法撑开父元素，需要清除浮动

7、浮动元素之间没有垂直margin的合并

清除浮动

- 父级上增加属性overflow：hidden
- 在最后一个子元素的后面加一个空的div，给它样式属性 clear:both（不推荐）
- 使用成熟的清浮动样式类，clearfix
- `.clearfix:after,.clearfix:before{ content: "";display: table;}`
- `.clearfix:after{ clear:both;}`
- `.clearfix{zoom:1;}`

清除浮动的使用方法：

```
.....
i++;
}
```

for-in 语句

for-in 语句是严格的迭代语句，用于枚举对象的属性。

```
var a = [10,20,30,40,50];
//迭代的是数组的下标。
for(i in a){
document.write(a[i]);
}
//输出： 1020304050
```

3.7 js元素获取与操作

可以使用内置对象document上的getElementById方法来获取页面上设置了id属性的元素，获取到的是一个html对象，然后将它赋值给一个变量，比如：

```
<script type="text/javascript">
var oDiv =
document.getElementById(&apos;div1&apos;);
</script>
....
<div id="div1">这是一个div元素</div>
```

上面的语句，如果把javascript写在元素的上面，就会出错，因为页面上从上往下加载执行的，javascript去页面上获取元素div1的时候，元素div1还没有加载，解决方法有两种：

第一种方法：将**javascript**放到页面最下边

```
....
<div id="div1">这是一个div元素</div>
....
<script type="text/javascript">
var oDiv =
document.getElementById(&apos;div1&apos;);
</script>
</body>
```

第二种方法：将**javascript**语句放到**window.onload**触发的函数里面,获取元素的语句会在页面加载完后才执行，就不会出错了。

```
<script type="text/javascript">
window.onload = function(){
var oDiv =
document.getElementById(&apos;div1&apos;);
}
</script>
....
<div id="div1">这是一个div元素</div>
```

样式操作

标签对象.style.css属性名="值" //改变标签对象的样式。

示例：id.style.color="red";

注意：属性名相当于变量名,所以css属性名中含有双拼词的(font-size)的减号要去掉，将后面的首字母大写。fontSize

文本操作

标签对象.innerHTML="内容"；//在标签对象内放置指定内容

表单中值的操作

```
.con2{... overflow:hidden}
或者
<div class="con2 clearfix">
```

2.8 定位

关于定位

我们可以使用css的position属性来设置元素的定位类型，postion的设置项如下：

- relative 生成相对定位元素，元素所占据的文档流的位置不变，元素本身相对文档流的位置进行偏移
- absolute 生成绝对定位元素，元素脱离文档流，不占据文档流的位置，可以理解为漂浮在文档流的上方，相对于上一个设置了相对或者绝对或者固定定位的父级元素来进行定位，如果找不到，则相对于body元素进行定位。
- fixed 生成固定定位元素，元素脱离文档流，不占据文档流的位置，可以理解为漂浮在文档流的上方，相对于浏览器窗口进行定位。
- static 默认值，没有定位，元素出现在正常的文档流中，相当于取消定位属性或者不设置定位属性
- inherit 从父元素继承 position 属性的值

定位元素特性

绝对定位和固定定位的块元素和行内元素会自动转化为行内块元素

定位元素层级

定位元素是浮动的正常的文档流之上的，可以用z-index属性来设置元素的层级

典型定位布局
1、固定在顶部的菜单
2、水平垂直居中的弹框
3、固定的侧边的工具栏

4、固定在底部的按钮

---

4 jquery选择器

4.1 jquery选择器

jquery用法思想一

选择某个网页元素，然后对它进行某种操作

jquery选择器

jquery选择器可以快速地选择元素，选择规则和css样式相同

```
$(&apos;li&apos;) //选择所有的li元素
$(&apos;#myId&apos;) //选择id为myId的网页元素
$(&apos;.myClass&apos;) // 选择class为myClass的元素
$(&apos;input[name=username]&apos;) // 选择name属性等于username的input元素
$(&apos;#ul1 li span&apos;) //空格层级获取 选择id为为ul1元素下的所有li下的span元素
$(&apos;#id,.class&apos;);//逗号并列获取
```

对选择集进行修饰过滤(类似**CSS**伪类)

```
$(&apos;#ul1 li:first&apos;) //选择id为ul1元素下的第一个li
$(&apos;#ul1 li:odd&apos;) //选择id为ul1元素下的li的奇数行
```

标签对象.value； //获取标签对象的value值

标签对象.value=”值“；//设置标签对象的value值

3.8 js定时器

通过使用 JavaScript，我们有能力作到在一个设定的时间间隔之后来执行代码，而不是在函数被调用后立即执行。我们称之为计时事件。

定时器在**javascript**中的作用
1、制作动画
2、异步操作

3、函数缓冲与节流

定时器类型及语法

setInterval() - 间隔指定的毫秒数不停地执行指定的代码。 setTimeout() - 暂停指定的毫秒数后执行指定的代码 setInterval() 和 setTimeout() 是 Window对象的两个方法。

```
/*
定时器：
setTimeout 只执行一次的定时器
clearTimeout 关闭只执行一次的定时器
setInterval 反复执行的定时器
clearInterval 关闭反复执行的定时器
*/
var time1 = setTimeout(myalert,2000);
var time2 = setInterval(myalert,2000);
/*
clearTimeout(time1);
clearInterval(time2);
*/
function myalert(){
alert(&apos;ok!&apos;);
}
```

3.9 js函数

*第一种是使用function语句定义函数

```
function abc(){
alert(&apos;abc&apos;);
}
```

*第二种是在表达式中定义函数

```
var 函数名 = function\(参数1，参数2，...\){函数体};
//例如：
//定义
var add = function\(a,b\){
return a+b;
}
//调用函数
document.write\(add\(50,20\)\);
```

第三种是使用Function()构造函数来定义函数（不常用）

```
var 函数名 = new Function\(“参数1”，” 参数2”，” 参数3”.....” 函数体”\);
如：
var 函数名 = new Function\(” x”,” y”,” var z=x+y;return z;” \);
```

arguments 对象

$('#ul1 li:eq(2)') //选择id为ul1元素下的第3个li

对选择集进行函数过滤

```
$(&apos;div&apos;).has(&apos;p&apos;); // 选择包含p元素的div元素
$(&apos;div&apos;).not(&apos;.myClass&apos;); //选择class不等于myClass的div元素
$(&apos;div&apos;).filter(&apos;.myClass&apos;); //选择class等于myClass的div元素
$(&apos;div&apos;).first(); //选择第1个div元素
$(&apos;div&apos;).eq(5); //选择第6个div元素
```

选择集转移

```
$(&apos;div&apos;).prev(&apos;p&apos;); //选择div元素前面的第一个p元素
$(&apos;div&apos;).next(&apos;p&apos;); //选择div元素后面的第一个p元素
$(&apos;div&apos;).children(); //选择div的所有子元素
$(&apos;div&apos;).siblings(); //选择div的同级元素
$(&apos;div&apos;).parent(); //选择div的父元素
$(&apos;#abc&apos;).parents(); //选择id为abc的所有的先辈元素

$(&apos;div&apos;).find(&apos;.myClass&apos;); //选择div内的class等于myClass的元素
```

4.2 jQuery元素操作

通过jQuery可以操作控制元素的样式,文本,属性等

jquery样式操作

css操作行内样式

```
// 获取div的样式
$("div").css("width");
$("div").css("color");
//设置div的样式
$("div").css("width","30px");
$("div").css("height","30px");
$("div").css({fontSize:"30px",color:"red"});
```

特别注意

选择器获取的多个元素，获取信息获取的是第一个，比如：
$("div").css("width")，获取的是第一个div的width。

类名class操作

操作样式类名

```
$("#div1").addClass("divClass2") //为id为div1的对象追加样式divClass2
$("#div1").removeClass("divClass") //移除id为div1的对象的class名为divClass的样式
$("#div1").removeClass("divClass divClass2") //移除多个样式
$("#div1").toggleClass("anotherClass") //重复切换anotherClass样式
```

文本操作

1、html() 取出或设置html内容

```
// 取出html内容
var $htm = $(&apos;#div1&apos;).html();
// 设置html内容
$(&apos;#div1&apos;).html(&apos;
<
span
```

在函数代码中，使用特殊对象 arguments，开发者无需明确指出参数名，就能访问它们。
例如，在函数 sayHi() 中，第一个参数是 message。用 arguments[0]
也可以访问这个值，即第一个参数的值（第一个参数位于位置 0，
第二个参数位于位置 1，依此类推）。

关于变量和参数问题：

函数外面定义的变量是全局变量，函数内可以直接使用。
在函数内部没有使用var定义的=变量则为全局变量，
*在函数内使用var关键字定义的变量是局部变量，即出了函数外边无法获取。
js函数定义的参数没有默认值（目前只有最新的火狐浏览器支持）

3.10 js对象

1.使用原始的方式创建内置对象
```
var myObject = new Object();
myObject.name = “lijie”;
myObject.age = 20;
myObject.say = function(){...}
```
2.直接创建自定义对象
```
var 对象名 = {属性名1：属性值，属性名2：属性值2，…….}
```
*3.使用自定义构造函数创建对象
```
function pen(name,color,price){
//对象的name属性
this.name = name;
//对象的color属性
this.color = color;
//对象的piece属性
this.price = price;
//对象的say方法
this.say = function(){};
}
var pen = new pen(“铅笔”,”红色”,20);
pen.say();
```

this关键字

this单词本身就是 这个 的意思
在对象的方法中使用,代表着当前这个对象
意味着当对象调用这个方法时,方法中的this就代表着这个对象

遍历

```
for(var i in window){
document.write(i+”----”+window[i]);
}
```
这种语句可以遍历对象中的所有属性或数组中的所有元素。

关于类型

测试类型:
```
1.typeof() //global对象的其中一个方法，typeof()
2.对象.constructor; //查看当前对象的构造函数是谁
if(arr.constructor==Array){
alert("数组"); //数组推荐用这种方法，因为typeof得到是object
}
```

3.11 js数组

数组就是一组数据的集合，javascript中，数组里面的数据可以是不同类型的。

>
添加文字
<
/span
>
');

2、text() 取出或设置text内容

```
// 取出文本内容
var $htm = $(&apos;#div1&apos;).text();
// 设置文本内容
$(&apos;#div1&apos;).text(&apos;
<
span
>
添加文字
<
/span
>
&apos;);
```

属性操作

1、attr() 取出或设置某个属性的值

```
// 取出图片的地址
var $src = $(&apos;#img1&apos;).attr(&apos;src&apos;);
// 设置图片的地址和alt属性
$(&apos;#img1&apos;).attr({ src: "test.jpg", alt: "Test Image" });
//也可以用户设置class属性
$(&apos;#abc&apos;).attr(&apos;class&apos;,&apos;all&apos;)
//也可以自定义 属性
$(&apos;#abc&apos;).attr(&apos;love&apos;,&apos;iloveyou&apos;)
```

2、removeattr()删除属性

```
$(&apos;#abc&apos;).removeattr(&apos;class&apos;)
$(&apos;#abc&apos;).removeattr(&apos;love&apos;)
```

4.3 相关尺寸

获取元素相对于文档的偏移量

```
var pos = $(&apos;#small&apos;).offset();
// console.log(pos.left);
// console.log(pos.top);
```

获取当前元素相对于父级元素的偏移量

```
var l = $(&apos;#small&apos;).position().left;
var t = $(&apos;#small&apos;).position().top;
// console.log(l,t);
```

获取文档滚动距离

```
var st = $(window).scrollTop();
var sl = $(window).scrollLeft();
```

设置文档滚动距离

```
// $(window).scrollTop(100);
// $(window).scrollLeft(200)
```

获取元素的宽度和高度

定义数组的方法

```
//对象的实例创建
var aList = new Array(1,2,3);
//直接量创建
var aList2 = [1,2,3,&apos;asd&apos;];
```

操作数组中数据的方法

1、获取数组的长度：aList.length;

```
var aList = [1,2,3,4];
alert(aList.length); // 弹出4
```

2、用下标操作数组的某个数据：aList[0];

```
var aList = [1,2,3,4];
alert(aList[0]); // 弹出1
```

3、push() 和 pop() 从数组最后增加成员或删除成员

```
var aList = [1,2,3,4];
aList.push(5);
alert(aList); //弹出1,2,3,4,5
aList.pop();
alert(aList); // 弹出1,2,3,4
```

4、unshift()和 shift() 从数组前面增加成员或删除成员

```
var aList = [1,2,3,4];
aList.unshift(5);
alert(aList); //弹出5,1,2,3,4
aList.shift();
alert(aList); // 弹出1,2,3,4
```

5、splice() 在数组中增加或删除成员

```
var aList = [1,2,3,4];
aList.splice(2,1,7,8,9); //从第2个元素开始，删除1个元素，然后在此位置增加&apos;7,8,9&apos;三个元素
alert(aList); //弹出 1,2,7,8,9,4
```

多维数组

多维数组指的是数组的成员也是数组的数组。

```
var aList = [[1,2,3],
[&apos;a&apos;,&apos;b&apos;,&apos;c&apos;]];
alert(aList[0][1]); //弹出2;
```

3.12 js数学

```
//四舍五入
var res = Math.round(5.921);
//获取最大值
var res = Math.max(10,23,523,43,65,46,32,32);
//获取最小值
var res =
Math.min(12312,324,32,42,3,23,412,4332,21,3,-1);
//获取绝对值
var res = Math.abs(-100);
//退一取整
var res = Math.floor(1.9);
//进一取整
var res = Math.ceil(1.1);
//幂运算 用来获取x的y次方 2的3次方
var res = Math.pow(2,3);
//开方运算 返回一个数的平方根
var res = Math.sqrt(9);
random() 返回 0 ~ 1 之间的随机数。
```

random 返回 **0 ~ 1** 之间的随机数。

random 获取一个随机数 返回0-1之间的随机小数 有可能到0 ,但是不会取到1

```
var w = $(&apos;#big&apos;).width();
var h = $(&apos;#big&apos;).height();
设置元素的宽度和高度
$(&apos;#big&apos;).width(400);
$(&apos;#big&apos;).height(400);
// console.log(w,h);
获取可视区域的宽度和高度
var cw = $(window).width();
var ch = $(window).height();
获取文档的宽度和高度
var cw = $(document).width();
var ch = $(document).height();
console.log(cw,ch);
```

4.4 关于事件

事件绑定

1.基本绑定

```
$(element).click(function(){})
$(element).dblclick(function(){})
。。。
```

加载完毕事件

```
$(document).ready(function(){})
$(function(){})
```

2.方法绑定

```
$(element).bind(&apos;click&apos;, function(){})//绑定事件
$(element).unbind();//解除事件绑定
```

3.动态绑定

```
$(element).live(&apos;click&apos;, function(){})
```

需注意，live方法在高版本的jquery中移出了,在使用时请注意版本

事件触发

当我们想要去触发某个元素的事件时可以使用 trigger,注意需指定元素的事件类型

```
$(element).trigger(&apos;click&apos;)
```

事件冒泡和默认行为

事件冒泡

当触发一个元素的事件时,会自动触发该元素的父级和先辈级的同类型事件,造成事件并发,导致页面混乱,我们称为事件冒泡

此时我们可以在元素的事件处理函数中 返回一个false 来进行阻止,注意这个方法仅限于在jquery中使用

默认行为

在页面中有些元素是具备默认行为的,例如a链接的单击,表单的提交,都会进行跳转或提交,这些我们成为默认行为

```
//random 获取一个随机数 返回0-1之间的随机小数 有可能到0 ,但是不会取到1
var res = Math.random();
//0-9随机数 小数
var res = Math.random()*10;
//0-9随机整数 (9-0 +1) +0
var res = Math.floor(Math.random()*10);
//1-10随机整数(10-1 +1) +1
var res = Math.floor(Math.random()*10)+1;
//0-10随机整数(10-0 +1) +0
var res = Math.floor(Math.random()*11)+0;
//5-15随机整数(15-5 +1) +5
var res = Math.floor(Math.random()*11)+5;
//3-99 (99-3 +1) +3
//封装函数
function rand(m,n){
return Math.floor(Math.random()*(n-m+1))+m;
}
var res = rand(20,30);
console.log(res);
```

3.13 js正则

正则表达式使用单个字符串来描述、匹配一系列符合某个句法规则的字符串,

正则表达式通常被用来检索、替换那些符合某个模式的文本

声明方式

new RegExp();
/hehe/

普通字符

转义字符

\w \W \d \D \s \S
var reg = /\w/;//单个的字母数字下划线
var reg = /\W/;//单个的非字母数字下划线
var reg = /\d/;//单个数字字符
var reg = /\D/;//单个非数字字符
var reg = /\s/;//单个空白字符
var reg = /\S/;//单个的非空白字符

元字符

. * + ? {} [] () | ^ $
var reg = /./;//除了换行外的其它任意字符
var reg = /z*/;//匹配0次或多次
var reg = /z+/;//匹配至少1次或多次
var reg = /\w+?/;//禁止贪婪
var reg = /\w{5}/;//限制匹配的次数
var reg = /\w{5,12}/;//限制匹配5到12次
var reg = /[a-z_A-Z0-9]+/;//字符范围
var reg = /\d+(\w+)\w+/;//子组
var reg = /abc|def|123/;//或
var reg = /^\$/;//限制开始
var reg = /a$/;//限制结尾

常用方法

test（） exec（）
var res = reg.test(str);//返回布尔类型的值,存在true不存在就false
var res1 = reg.exec(str);//返回值 如果匹配到返回数组,如果不存在 返回 null

但是在绑定上事件后,它首先会先执行事件,再去执行默认行为,而有时我们只想让其触发事件,但不执行默认行为时,

我们可以在该元素的事件中 返回一个**false**来进行阻止默认行为

```
<
a href="http://www.baidu.com"
>
百度
<
/a
>
$(&amp;apos;a&amp;apos;).click(function(){
//阻止默认行为
return false;
})
```

获得当前鼠标的位置和按键

我们有鼠标和键盘按键的事件,在触发事件时如果我们想要获取鼠标的位置或键盘按键信息时,

首先需要在当前的事件中传递一个 事件对象 **e**

```
$(element).click(function(e){
//能够获取鼠标的x轴和y轴坐标,坐标位置相对于浏览器窗口
var x = e.clientX;
var y = e.clientY;
//能够获取鼠标的x轴和y轴坐标,坐标位置相对于文档
var _x = e.pageX;
var _y = e.pageY
})
$(element).keydown(function(e){
//可以打印e对象,或者直接使用该对象中的keyCode属性来获取按键信息
var key = e.keyCode;
console.log(key);
})
```

4.5jquery元素节点操作

创建节点

```
var $div = $(&amp;apos;<div>&amp;apos;);
var $div2 = $(&amp;apos;<div>这是一个div元素</div>&amp;apos;);
```

插入节点

1、append()和appendTo()：在现存元素的内部，从后面插入元素

```
var $span = $(&amp;apos;<span>这是一个span元素</span>&amp;apos;);
$(&amp;apos;#div1&amp;apos;).append($span);
.....
<div id="div1"></div>
```

2、prepend()和prependTo()：在现存元素的内部，从前面插入元素

3、after()和insertAfter()：在现存元素的外部，从后面插入元素

4、before()和insertBefore()：在现存元素的外部，从前面插入元素

删除节点

```
$(element).remove() 删除当前元素
$(element).empty()清空
```

克隆节点

```
$(element).clone(true)
```

json

json是 JavaScript Object Notation 的首字母缩写，单词的意思是javascript对象表示法，这里说的json指的是类似于javascript对象的一种数据格式，目前这种数据格式比较流行，逐渐替换掉了传统的xml数据格式。

javascript对象字面量：

字符串方法 match()

---

5 ajax

ajax是什么?

AJAX 是与服务器交换数据并更新部分网页的艺术，在不重新加载整个页面的情况下。

AJAX 指异步 JavaScript 及 XML（Asynchronous JavaScript And XML）。

AJAX 是一种在 2005 年由 Google 推广开来的编程模式。

AJAX 不是一种新的编程语言，而是一种使用现有标准的新方法。

通过 AJAX，你可以创建更好、更快以及更友好的 WEB 应用程序。

AJAX 基于 JavaScript 和 HTTP 请求（HTTP requests）。

通过 **HTTP** 请求加载远程数据

jQuery 底层对 **AJAX** 实现进行了封装.使得我们在进行**ajax**操作时,不必像原生**js**中那么复杂

$.get, $.post, $.ajax() 返回其创建的 **XMLHttpRequest** 对象。多数情况下我们不需要去操作返回的对象

如何使用ajax技术?

首先你得有web服务器,能够通过浏览器去执行你的html和你的python
注意一点:我们平常写的html,直接在浏览器打开时 使用的是file协议
而ajax是基于HTTP请求的,所以要求你的html能够使用http的协议打开
如果你能做到用http协议去打开你的html并且能够正常显示的话,就代表你的web服务器搭建成功

$.get() 方法:

//发送ajax请求 1.url 2.可选 发送get请求时携带的参数 ,3,可选 回调函数,请求完之后做什么事 4,可选,返回的数据类型 json
$.get(url,{请求的参数},function(data) {},'json')

$.post()

$.post(url,{请求的参数},function(data) {},'json')

$.ajax()

$.ajax({
url:'/cgi-bin/5.py',//当前请求的url地址
type:'get',//当前请求的方式 get post
data:{id:100,username:'zhangsan'},//请求时发送的参数
dataType:'json',//返回的数据类型

var tom = {
name:'tom',
age:18
}

json格式的数据：

{
"name":'tom',
"age":18
}

与json对象不同的是，json数据格式的属性名称需要用双引号引起来，用单引号或者不用引号会导致读取数据错误。

json的另外一个数据格式是数组，和javascript中的数组字面量相同。

['tom',18,'programmer']

---

6 BootStrap

简单、直观、强悍的前端开发框架，让web开发更迅速、简单。 来自Twitter，是目前很受欢迎的前端框架之一。 Bootrstrap是基于HTML、CSS、JavaScript的，让书写代码更容易。 移动优先，响应式布局开发。

bootstrap中文网址：http://www.bootcss.com/

success:function(data){
//ajax请求成功后执行的代码
console.log(data);
},
error:function(){
//ajax执行失败后执行的代码
alert('ajax执行错误');
},
timeout:2000,//设置当前请求的超时时间 毫秒,必须时异步请求才会生效
async:true// 是否异步 true为异步 false 同步
})

ajax异步 同步

//设置ajax的全局配置 async:false 设置当前请求为同步
$.ajaxSetup({
async:false
})
关于ajax中 异步 和 同步
ajax默认就是异步请求,
async (默认: true) 默认设置下，所有请求均为异步请求。
如果需要发送同步请求，请将此选项设置为 false。
同步请求,就发ajax请求发出去后必须等待ajax的结果返回后才能继续往下执行
一般情况下都使用异步操作就可以,除非有特殊情况,必须等ajax的结果回来后才能做处理的,就用同步

注意

1.ajax是无刷新请求服务器,所以我们在浏览器中是感觉不到,也看不到ajax的具体请求和执行情况的.,
因此我们需要借助浏览器的调试工具 (F12打开) 进行查看
2.ajax的请求是基础HTTP协议的,就要求你当前打开这个带有ajax的html时必须使用http协议
3.ajax要求同源策略
http://127.0.0.1:8000/cgi-bin/1.py
即: 协议(http https) 域名或IP 以及端口(80 443 8000 8080 ...)都必须一致
4.关于返回的数据类型 在get() post() ajax() 都可以设置返回的数据类型 'json'
如果要求返回json格式数据,那么就必须返回json,如果不正确,
在get和post方法将拿不到data中返回的数据,在ajax方法中则会进去error方法
5.在python中返回json格式数据,
引入 json模块
json.dumps(数据) 使用json_dumps方法进行json格式的编码转换
6.在使用ajax方法时.会创建一个对象
XMLHttpRequest
那么在ajax的方法中使用的 $(this) 就代表 ajax的对象
$(this) 永远代表一个对象,没有指明对象时 代表的时window对象,
在它有对象时 代表的就是当前的这个对象