一、HTML介绍

1. 简介

* HTML是网页内容的载体。内容就是网页制作者放在页面上想要让用户浏览的信息，可以包含文字、图片、视频等。
* CSS样式是表现。就像网页的外衣。比如，标题字体、颜色变化，或为标题加入背景图片、边框等。所有这些用来改变内容外观的东西称之为表现。
* JavaScript是用来实现网页上的特效效果。如：鼠标滑过弹出下拉菜单。或鼠标滑过表格的背景颜色改变。还有焦点新闻（新闻图片）的轮换。可以这么理解，有动画的，有交互的一般都是用JavaScript来实现的。

2、标签

* 标签由英文尖括号<和>括起来，如<html>就是一个标签。
  + <h></h> 是标题标签，
  + <p></p>是段落标签，
  + <img src="">是图片标签。
* 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/。
* 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。
* 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。

3、语法结构

* <html>

<head>

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

<title>标签的语法</title>

</head>

<body>

<h1>在本教程中，你将学习如何使用 HTML 来创建站点</h1>

<p>当特殊的样式需要应用到个别元素时，就可以使用内联样式。</p>

</body>

</html>

* 语法分析—没有逗号
* http-equiv="Content-Type" 表示描述文档类型
* content="text/HTML; 文档类型mime类型，这里为html,如果JS就是text/javascript，
* charset=utf-8 页面字符集，编码，eg:gb2312,iso-8859-1,utf-8
* meta是html中的元标签，其中包含了对应html的相关信息，客户端浏览器或服务器端的程序会根据这些信息进行处理。
* HTTP-EQUIV类似于HTTP的头部协议，它回应给浏览器一些有用的信息，以帮助正确和精确地显示网页内容。

4、HTML文件基本结构

* <html>

<head>...</head>

<body>...</body>

</html>

* 语法讲解

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。

5、注释

* <!—注释文字-->

快捷键：ctrl+/

二、标签

1、标签用途—语义化

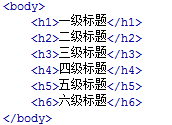
* 语义化：**明白每个标签的用途**（在什么情况下使用此标签合理）比如，网页上的文章的标题就可以用标题标签，网页上的各个栏目的栏目名称也可以使用标题标签。文章中内容的段落就得放在段落标签中，在文章中有想强调的文本，就可以使用 em 标签表示强调等等

2、body标签--段落标签：<p>段落文本</p>

* 每行之间会有一行空白。
* <p style = "margin:0px">我的第三个段落</p>

会向里面缩进px个字符

3、body标签—标题标签：<h1>标题文本</h1>

* 
* h1标签字号最大，h2标签字号相对h1要小，以此类推h6标签的字号最小。

4、body标签--强调标签：可用于文本内

* <em>需要强调的文本</em>

<strong>需要强调的文本</strong>

* 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。

5、自定义标签：<span></span>

* <html>

<head>

<style>

span{

color:blue;

}

</style>

</head>

<body>

<span>…..</span>

</body>

</html>

* <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，相当于自定义CSS。
* HTML 4.0 标准仅支持 16 种颜色名，它们是：aqua、black、blue、fuchsia、gray、green、lime、maroon、navy、olive、purple、red、silver、teal、white、yellow。如果使用其它颜色的话，就应该使用十六进制的颜色值

6、body标签—短文本引用标签：<q>引用文本</q>

* 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。

7、body标签—长文本引用标签：<blockquote>引用文本</blockquote>

* 浏览器对<blockquote>标签的解析是缩进样式。

8、折行标签：<br/>

* <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。
* 只需要加入一个<br/>就行。

9、插入空格：&nbsp;

* 来源：作文网&nbsp;&nbsp;作者：为梦想而飞

10、插入水平横线：<hr/>

* <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。

11、地址标签：<address>联系地址信息</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。

12、代码标签：<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。
* 注：如果是多行代码，可以使用<pre>标签。
* <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。

三、标签第二部分

1、无序标签：<ul>

<li>文本</li>

<li>文本</li>

</ul>

* 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。
* 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。

2、有序标签：<ol>

<li>信息</li>

<li>信息</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。

3、容器标签：<div>…</div>

* 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。
* 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。
* <div id="版块名称">…</div>
* 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。

4、表格标签：

创建表格的四个元素：

table、tbody、tr、th、td

* <table>…</table>：整个表格以<table>标记开始、</table>标记结束。
* <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。）
* <tr>…</tr>：表格的一行，所以有几对tr 表格就有几行。
* <td>…</td>：表格的一个单元格，一行中包含几对<td>...</td>，说明一行中就有几列。
* <th>…</th>：表格的头部的一个单元格，表格表头。
* 表格中列的个数，取决于一行中数据单元格的个数。
* 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

表头，也就是th标签中的文本默认为粗体并且居中显示

5、头部标签—CSS样式

* <style type="text/css">

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

</style>

* 为表格设置样式

6、为表格添加摘要和标题

* 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。

* 标题

用以描述表格内容，标题的显示位置：表格上方

* 语法：

<table>

<table summary="表格简介文本">

<caption>

标题文本

</caption>

<table>

7、超链接标签：<a>

* <a href="目标网址" target=”\_blank” title="鼠标滑过显示的文本">链接显示的文本</a>
* 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）。
* target=”\_blank” ：点击链接打开另外一个新窗口。
* 发送邮件
* 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。
* 
* <a href="mailto:yy@imooc.com?subject=观了不起的盖茨比有感&body=你好，对此评论有些想法">链接的文本</a>

8、图片标签：<img>

* <img src="图片地址" alt="下载失败时的替换文本" title = "提示文本">
* src：标识图像的位置；

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；

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)；

* 图像可以是GIF，PNG，JPEG格式的图像文件。

四、表单标签—交互方式

1、表单标签

* <form method="传送方式" action="服务器文件">……</form>
* <form> ：<form>标签是成对出现的，以<form>开始，以</form>结束。
* 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)。
* method ： 数据传送的方式（get/post）
* 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！）。input模块
* text：文本框
* password：密码框
* raido：单选按钮
* checkbox：复选框
* file：文件选择框
* submit：提交按钮
* <form method="post" action="save.php">

<label for="username">用户名:</label> //label，小模块

<**input** type="text" name="username" id="username" value="" /><br/>

<label for="pass">密码:</label>

<**input** type="password" name="pass" id="pass" value="" />

<**input** type="submit" value="确定" name="submit" />

<**input** type="reset" value="重置" name="reset" />

</form>

2、文本输入框：input

* <form>

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

</form>

* 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。

* name：为文本框命名，以备后台程序ASP 、PHP使用。
* value：为文本输入框设置默认值。(一般起到提示作用)

3、文本域—多行文本输入框

* <textarea rows="行数" cols="列数">文本</textarea>
* <textarea>标签是成对出现的，以<textarea>开始，以</textarea>结束。
* cols ：多行输入域的列数。
* rows ：多行输入域的行数。
* 在<textarea></textarea>标签之间可以输入默认值。
* 语法结构：

<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>

4、单选框AND复选框

* 语法：

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

* type:

当 type="radio" 时，控件为单选框。

当 type="checkbox" 时，控件为复选框。

* value：提交数据到服务器的值（后台程序PHP使用）。
* name：为控件命名，以备后台程序 ASP、PHP 使用。

单选框中name值要相同；复选框中name值不要相同。·

* checked：当设置 checked="checked" 时，该选项被默认选中。

5、下拉表框--selecte

* 语法结构：

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

<label>爱好:</label>

<select>

<option value="看书">看书</option>

<option value="旅游" selected="selected">旅游</option>

<option value="运动">运动</option>

<option value="购物">购物</option>

<select>

</form>

* selected="selected"：

设置selected="selected"属性，则该选项就被默认选中。

* <select multiple="multiple">

<option value=”返回服务器的值”>显示的值</option>

……

<select>

6、提交按钮

* <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

7、重置按钮：reset

* <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

8、lable标签

* 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">

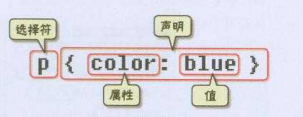
注意：标签的 for 属性中的值应当与相关控件的 id 属性值一定要相同。

一、CSS

1、简介

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。

2、CSS语法结构

* 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;

}

3、注释

CSS : /\*注释内容\*/

HTML : <!—注释内容-->

二、CSS样式

1、内联式--直接写在现有的HTML标签中

* 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。
* 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。

2、嵌入式--写在当前的文件中

* 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。

3、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：
* css样式文件名称以有意义的英文字母命名，如 main.css。
* rel="stylesheet" type="text/css" 是固定写法不可修改
* <link>标签位置一般写在<head>标签之内。

4、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。

就近原则（离被设置元素越近优先级别越高）。

三、CSS中的选择器

1、简介

* 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。

2、标签选择器

* 标签选择器其实就是html代码中的标签。如右侧代码编辑器中的<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。

3、类选择器

* 语法：

.类选器名称{css样式代码;}

* 注意：
* 英文圆点开头
* 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\*/

4、ID选择器

* ID选择器都类似于类选择符，但也有一些重要的区别：
* 为标签设置id="ID名称"，而不是class="类名称"。
* ID选择符的前面是井号（#）号，而不是英文圆点（.）。

5、类选择器和ID选择器区别

* 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。

* 只能应用于一个地方

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）

* <span class="stress bigsize">文本内容</span> 可以
* <span id="stressid bigsize">文本内容</span> 不可以

6、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：

.first>span{border:1px solid red;}

* <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</p>
* 类设置在<p>中，指定段落里的span字段样式。

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。

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;}

9、伪类选择符

* 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;}

只需要加这个就好。

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;}

四、CSS其他属性

1、继承

* 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。
* 边框border:1px solid red;不具备继承属性。

2、特殊性

* 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：

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。

3、层叠

* 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。
* 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）。
* 就近原则。

4、重要性

* 用！import设置最高权值—放在分号之前

p{color:red!important;}

* 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。

五、文字排版

1、字体

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

2、字号颜色

* body{**font-size**:12px;**color**:#666;}
* .stress{font-size:20px;color:red;}

3、粗体

* p span{f**ont-weight**:bold;}

4、斜体

* p a{**font-style**:italic;}

5、下划线

* p {**text-decoration**:underline;}

6、删除线

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

7、缩进

* p{t**ext-indent**:2em;}

8、行间距

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

9、字间距

* 中文字符间距

h1{letter-spacing:50px;}

* 英文单词间距

h1{ word-spacing:50px;}

10、对齐---块状元素

* 居中对齐：h1{ text-align:center;}
* 左对齐：h1{ text-align:left;}
* 右对齐：h1{ text-align:right;}

六、CSS盒模型

1、标签元素元素

* 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、块状元素

* 设置display:block就是将元素显示为块级元素。

如：span{display:block;}

* 块级元素特点：
* 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）
* 元素的高度、宽度、行高以及顶和底边距都可设置。
* 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

* 通过代码display:inline将元素设置为内联元素

如：div{display:inline;}

* 内联元素特点：
* 和其他元素都在一行上；
* 元素的高度、宽度及顶部和底部边距不可设置；
* 元素的宽度就是它包含的文字或图片的宽度，不可改变。
* 当行内元素之间有“回车”、“tab”、“空格”时就会出现间隙。

4、内联块状元素

* 代码display:inline-block就是将元素设置为内联块状元素。

a{display:inline-block;}

* inline-block 元素特点：
* 和其他元素都在一行上；
* 元素的高度、宽度、行高以及顶和底边距都可设置。

5、盒子模型

* <div><ul><ol><p><h><table> 都具备盒子模型的特征：

具有内填充padding，边框border，边界margin

(1)、边框—border

* 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)
* 语法结构：
* div{border:2px solid red;}
* div{

border-width:2px;

border-style:solid;

border-color:red;

}

* 注意：
* border-style（**边框样式**）常见样式有：

dashed（虚线）| dotted（点线）| solid（实线）。

* border-color（**边框颜色**）中的颜色可设置为十六进制颜色，如:

border-color:#888;//前面的井号不要忘掉。

* 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），最常还是用象素（px）。

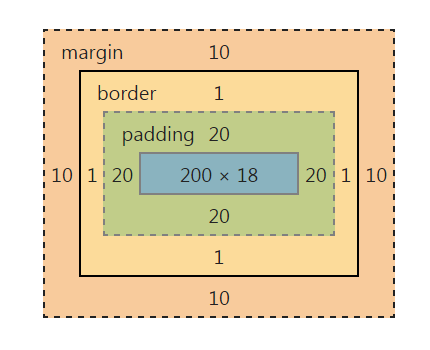
* 单独设置边框的一边：.

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

也可以使用border-left,border-right,border-top

(2)、宽度和高度

* css内定义的宽（width）和高（height），指的是填充以里的内容范围。



* 语法结构：

css代码：

div{

width:200px;

padding:20px;

border:1px solid red;

margin:10px; }

(3)、填充padding

* 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)。
* 语法结构：
* div{

padding-top:20px;

padding-right:10px;

padding-bottom:15px;

padding-left:30px;

}

* div{padding:20px 10px 15px 30px;}
* 上下左右都一样：div{padding:10px;}
* 上下一样，左右一样：div{padding:10px 20px;}

(4)、边界margin

* 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。
* padding和margin的区别，padding在边框里，margin在边框外。
* 语法结构跟padding一样，left，right，top，bottom

七、CSS布局模型

1、简介

布局模型与盒模型一样都是 CSS 最基本、 最核心的概念。 但布局模型是建立在盒模型基础之上，又不同于我们常说的 CSS 布局样式或 CSS 布局模板。如果说布局模型是本，那么 CSS 布局模板就是末了，是外在的表现形式。

* CSS包含3种基本的布局模型，用英文概括为：Flow、Layer 和 Float。

在网页中，元素有三种布局模型：

流动模型（Flow）

浮动模型 (Float)

层模型（Layer）

2、流动模型（Flow）

* 流动（Flow）是默认的网页布局模式。也就是说网页在默认状态下的 HTML 网页元素都是根据流动模型来分布网页内容的。
* 流动模型特点：
* **块状元素**都会在所处的包含元素内**自上而下按顺序垂直延伸分布**，因为在默认状态下，块状元素的宽度都为100%。实际上，块状元素都会以行的形式占据位置。
* **内联元素**都会在所处的包含元素内**从左到右水平分布显示。**（内联元素可不像块状元素这么霸道独占一行）

3、浮动模型（Float）

* 任何元素在默认情况下是不能浮动的，但可以用 CSS 定义为浮动，如 div、p、table、img 等元素都可以被定义为浮动。
* 语法结构：
* 左浮动：

div{

width:200px;

height:200px;

border:2px red solid;

**float:left**;

}

* 右浮动：float:right

4、层模型（Layer）

* 层布局模型就像是图像软件PhotoShop中非常流行的图层编辑功能一样，每个图层能够精确定位操作，CSS定义了一组定位（positioning）属性来支持层布局模型。
* 绝对定位(position: absolute)
* 相对定位(position: relative)
* 固定定位(position: fixed)
* 需要设置position:absolute(表示绝对定位)，这条语句的作用将元素从文档流中拖出来，然后使用left、right、top、bottom属性相对于其最接近的一个具有定位属性的父包含块进行绝对定位。一般是相对于浏览器窗口。
* 语法结构：

div{

……

position：absolute;

left:xx;

right:xx;

top:xx;

bottom:xx;

}

* 需要设置position:relative（表示相对定位），它通过left、right、top、bottom属性确定元素在正常文档流中的偏移位置。相对定位完成的过程是首先按static(float)方式生成一个元素(并且元素像层一样浮动了起来)，然后相对于以前的位置移动，移动的方向和幅度由left、right、top、bottom属性确定，**偏移前的位置保留不动**。会新建一个元素。
* 语法结构：

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

left:100px;

top:50px;

}

* 需要设置position:fixed来定位。它的相对移动的坐标是视图（屏幕内的网页窗口）本身，不会随着浏览器窗口变化，就是固定视图位置。
* 语法结构：

#div1{

width:200px;

height:200px;

border:2px red solid;

position:fixed;

left:100px;

top:50px;

}

* absolute元素 参照 relative元素设置
* 语法结构：

#div1{position=relative}

#div2{position=absolute

top:xx;

right:xx;}

<div id=”div1”>

<div id=”div2”></div>

</div>

* 参照元素必须是父元素，而且是relative属性的。

八、代码简写

1、盒型代码简写

* 通常有下面三种缩写方法:
* 如果top、right、bottom、left的值相同，如下面代码：

margin:10px 10px 10px 10px;

可缩写为：

margin:10px;

* 如果top和bottom值相同、left和 right的值相同，如下面代码：

margin:10px 20px 10px 20px;

可缩写为：

margin:10px 20px;

* 如果left和right的值相同，如下面代码：

margin:10px 20px 30px 20px;

可缩写为：

margin:10px 20px 30px;

* 注意：padding、border的缩写方法和margin是一致的。

2、颜色简写

* 关于颜色的css样式也是可以缩写的，当你设置的颜色是16进制的色彩值时，如果每两位的值相同，可以缩写一半。
* 例子1：

p{color:#000000;}

可以缩写为：

p{color: #000;}

* 例子2：

p{color: #336699;}

可以缩写为：

p{color: #369;}

3、字体缩写

* 网页中的字体css样式代码也有他自己的缩写方式，下面是给网页设置字体的代码：

body{

font-style:italic;

font-variant:small-caps;

font-weight:bold;

font-size:12px;

line-height:1.5em;

font-family:"宋体",sans-serif;

}

这么多行的代码其实可以缩写为一句：

body{

font:italic small-caps bold 12px/1.5em "宋体",sans-serif;

}

* 注意：
* 使用这一简写方式你至少要指定 font-size 和 font-family 属性，其他的属性(如 font-weight、font-style、font-variant、line-height)如未指定将自动使用默认值。
* 在缩写时 font-size 与 line-height 中间要加入“/”斜扛。
* 一般情况下因为对于中文网站，英文还是比较少的，所以下面缩写代码比较常用：

body{

font:12px/1.5em "宋体",sans-serif;

}

* 只是有字号、行间距、中文字体、英文字体设置。

4、颜色值

* 在网页中的颜色设置是非常重要，有字体颜色（color）、背景颜色（background-color）、边框颜色（border）等，设置颜色的方法也有很多种：
* 英文命令颜色：

p{color:red;}

* RGB颜色：

这个与 photoshop 中的 RGB 颜色是一致的，由 R(red)、G(green)、B(blue) 三种颜色的比例来配色。

p{color:rgb(133,45,200);}

每一项的值可以是 0~255 之间的整数，也可以是 0%~100% 的百分数。如：

p{color:rgb(20%,33%,25%);}

* 十六进制颜色：

这种颜色设置方法是现在比较普遍使用的方法，其原理其实也是 RGB 设置，但是其每一项的值由 0-255 变成了十六进制 00-ff。

p{color:#00ffff;}

5、长度值

* 长度单位总结一下，目前比较常用到px（像素）、em、% 百分比，要注意其实这三种单位都是相对单位。
* 像素

像素为什么是相对单位呢？因为像素指的是显示器上的小点（CSS规范中假设“90像素=1英寸”）。实际情况是浏览器会使用显示器的实际像素值有关，在目前大多数的设计者都倾向于使用像素（px）作为单位。

* em

就是本元素给定字体的 font-size 值，如果元素的 font-size 为 14px ，那么 1em = 14px；如果 font-size 为 18px，那么 1em = 18px。如下代码：

p{font-size:12px;text-indent:2em;}

上面代码就是可以实现段落首行缩进 24px（也就是两个字体大小的距离）。

下面注意一个特殊情况：

但当给 font-size 设置单位为 em 时，此时计算的标准以 p 的父元素的 font-size 为基础。如下代码：

html:

<p>以这个<span>例子</span>为例。</p>

css:

p{font-size:14px}

span{font-size:0.8em;}

结果 span 中的字体“例子”字体大小就为 11.2px（14 \* 0.8 = 11.2px）。

* 百分比

p{font-size:12px;line-height:130%}

设置行高（行间距）为字体的130%（12 \* 1.3 = 15.6px）。

九、CSS样式

1、水平居中

* 行内元素
* 如果被设置元素为文本、图片等行内元素时，水平居中是通过给父元素设置 text-align:center 来实现的。
* 语法结构：

.txtCenter{

text-align:center;

}

* 定宽块状元素：块状元素的宽度width为固定值。
* 满足定宽和块状两个条件的元素是可以通过设置“左右margin”值为“auto”来实现居中的。

<style>

div{

border:1px solid red;/\*为了显示居中效果明显为 div 设置了边框\*/

width:200px;/\*定宽\*/

margin:20px auto;/\* margin-left 与 margin-right 设置为 auto \*/

}</style>

* 不定宽块状元素：块状元素的宽度width不固定。
* 不定宽度的块状元素有三种方法居中（这三种方法目前使用的都很多）：

1、加入 table 标签

2、设置 display: inline 方法：与第一种类似，显示类型设为 行内元素，进行不定宽元素的属性设置

3、设置 position:relative 和 left:50%：利用 相对定位 的方式，将元素向左偏移 50% ，即达到居中的目的

* 加入tale标签
* 是利用table标签的长度自适应性---即不定义其长度也不默认父元素body的长度（table其长度根据其内文本长度决定），因此可以看做一个定宽度块元素，然后再利用定宽度块状居中的margin的方法，使其水平居中。
* 第一步：为需要设置的居中的元素外面加入一个 table 标签 ( 包括 <tbody>、<tr>、<td> )。
* 第二步：为这个 table 设置“左右 margin 居中”（这个和定宽块状元素的方法一样）。
* 语法结构：

<div><table><tbody>

<tr><td>

文本

</td></tr>

</tbody></table></div>

<style>

table{

border:1px solid;

margin:0 auto;

}

</style>

* 设置成行内元素，改变块级元素的 display 为 inline 类型（设置为 行内元素 显示），然后使用 text-align:center 来实现居中效果。
* 语法结构：

<div>文本</div>

div{text-align：center；}

div ul{display：inline；}

要把块状元素转化成内联元素。

2、垂直居中—父元素高度确定的单行文本

* **父元素高度确定的单行文本**的竖直居中的方法是通过**设置父元素的 height 和 line-height 高度一致**来实现的。(height: 该元素的高度，line-height: 顾名思义，行高（行间距），指在文本中，行与行之间的 基线间的距离 )。
* **line-height 与 font-size 的计算值之差，在 CSS 中成为“行间距”。分为两半，分别加到一个文本行内容的顶部和底部。**
* 这种文字行高与块高一致带来了一个弊端：当文字内容的长度大于块的宽时，就有内容脱离了块。
* 语法结构：

<div class="container">

hi,imooc!

</div>

css代码：

<style>

.container{

**height:100px;**

**line-height:100px;**

background:#999;

}

</style>

3、垂直居中---父元素高度确定的多行文本以及图片

* 使用插入 table (包括tbody、tr、td)标签，同时设置 vertical-align：middle。
* css 中有一个用于竖直居中的属性 vertical-align，在父元素设置此样式时，会对inline-block类型的子元素都有用。
* html代码：

<body>

<table><tbody><tr><td class="wrap">

<div>

<p>看我是否可以居中。</p>

</div>

</td></tr></tbody></table>

</body>

css代码：

table td{height:500px;background:#ccc}

* 因为 td 标签默认情况下就默认设置了 vertical-align 为 middle，所以我们不需要显式地设置了。

4、隐性改变display：块状元素、内联元素、内联块状元素

position : absolute

float : left 或 float:right

* 简单来说，只要html代码中出现以上两句之一，元素的display显示类型就会自动变为以 display:inline-block（块状元素）的方式显示，当然就可以设置元素的 width 和 height 了，且默认宽度不占满父元素。
* <style>

.container a{

position:absolute;

width:200px;

background:#ccc;

}

</style>