# Html+CSS

## 基本标签

现在一般使用xhtml1.0版本，这种版本比较规范。

### 1）<p>标签，段落

添加段落，默认段前段后有空白。

### 2）<hx>标签，标题

添加标题。标题标签共有6个，h1、h2、h3、h4、h5、h6，分别为一级标题、二级标题、三级标题、四级标题、五级标题、六级标题，依据重要性递减。<h1>是最高等级。

### 3）<strong>、<em>，加强语气

<em>表示强调，在浏览器中默认用斜体表示，<strong>表示更强烈的强调，默认用粗体表示。目前国内前端程序员比较喜欢用<strong>。

### 4）<span>，为文字设定单独样式

<span>标签没有语义，作用是设置单独样式。

### 5）<q>、<blockquote>，文本引用

<q>短文本引用，引用的文本不用加双引号，浏览器会自动添加双引号。<q>标签的真正关键点不是它的默认样式双引号，而是它的语义：引用别人的话。

<blockquote>长文本引用，引入大段文字。浏览器对<blockquote>标签的解析式缩进样式。

### 6）<br>，分行显示文本

在需要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。<br />是一个空标签，即没有内容的标签。类似的有<br />、<hr />、<img />。Html忽略回车和空格。

### 7）&nbsp; 空格

### 8）<hr>，添加水平横线

<hr>是一个空标签。<hr />在浏览器中默认样式线条比较粗，颜色为灰色。

### 9）<address>，添加地址信息

<address>标签在浏览器显示样式为斜体

### 10）<code>、<pre>，加入代码

<code>标签可插入一行代码，插入多行代码使用<pre>标签。<pre>标签的作用：预格式化的文本。被包围在pre元素中的文本通常会保留空格和换行符。<pre>标签不只是为显示计算机的源代码时用的，若需要在网页中预显示格式时都可以使用它。

### 11）<ul>、<ol>标签，添加列表

ul-li是没有前后顺序的信息列表。在网页显示中，默认样式为每个li前面自带一个圆点。

<ul>

<li>信息</li>

<li>信息</li>

……

</ul>

<ol>可用于制作有序列表，语法同<ul>。<ul>标签在网页中的默认显示样式为每项<li>前自带一个序号，序号默认从1开始。

### 12）<div>，排版

在网页制作过程中，可以将一些独立的逻辑部分划出来，放在一个<div>标签中，该标签的作用相当于一个容器。为使逻辑更清楚，可以用id属性来为<div>提供唯一的名称，id是唯一的。

### 13）table标签，网页上的表格

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

<tbody></tbody>：表格内容较多时，表格会下载一点显示一点，若加上<tbody>标签，表格要等表格内容全部下载完才会显示。

<tr></tr>：表格的一行，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说明一行有几列。

<th></th>：表格的头部的一个单元格，表格表头。

table表格在没有添加css样式，在浏览器中显示是没有表格线的。

<th>表头，文本默认为粗体并且剧中显示。

用css样式为表格加入边框。eg:

<style type=”text/css”>

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

</style>

上述代码为th，td单元格添加粗细为一个像素的黑色边框。

<caption>标签，为表格添加标题和摘要。

摘要的内容是不会显示出来的，作用是增加表格的可读性（语义化），语法：<table summary=”表格简介文本”>

标题用以描述表格内容，标题显示在表格上方。语法<caption>标题文本</caption>

### 14）<a>标签，链接到另一个页面

<a>标签可以实现超链接。

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

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。

只要为文本加入a标签后，文字的颜色就会自动变为蓝色（被点击过的文本颜色为紫色）

<a>标签默认在当前浏览器窗口打开链接的网页，若要在新的浏览器窗口打开，添加target属性，target=”\_blank”。



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

### 15）<img>标签，为网页插入图片

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

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。

## 表单标签，与用户交互

表单是可以把浏览者输入的数据传送到服务端，这样服务器端程序就可以处理表单传过来的数据。

语法：<form method=”传送方式” action=”服务器文件”>

<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)。

注意：表单的所有控件（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，不然用户输入信息无法提交到服务器。

1. **文本输入框、密码输入框**

语法:<form>

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

</form>

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：为文本框设置默认值，一般起到提示作用。

1. **文本域，支持多行文本输入**

语法：<textarea rows=”行数” cols=”列数”>文本</textarea>

rows和cols可用css样式的width和height代替。col->width。

1. **单选框、复选框，让用户选择**

单选框中的选项只能选择一项，复选框中的选项可以任意选择多项。

语法：<input type=”radio/checkbox” value=”值” name=”名称” checked=”checked” />

type：radio为单选框，checkbox为复选框。

value：提交数据到服务器的值。

name：为控件命名，以备后台php,asp使用。

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

注意：同一组的单选按钮中，name取值一定要一致，这样才能起到单选的作用。

1. **下拉列表，节省空间**

<select>

<option value=””>内容</option>

</select>

selected=”selected”属性，则该选项被默认选中。

在<select>标签中设置multiple=”multiple”属性，可实现多选功能，ctr+单击（mac下command+单击）可以选择多个选项。

1. **提交按钮，提交数据**

<input type=”submit” value=”提交”>

只有当type值设置为submit时，按钮才有提交作用，value为按钮上显示的文字。

1. **重置按钮，重置表单信息**

语法：<input type=”reset” value=”重置”>

1. **<label>标签**

该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能性。当用户单击选中该标签时，浏览器会自动将焦点转到和标签相关的表单控件上。

语法：<label for=”控件id名称”>

注意：标签的for属性值一定要与相关控件的id属性值相同。

eg:

<form>

<label for="male">男</label>

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

<br />

<label for="female">女</label>

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

<label for="email">输入你的邮箱地址</label>

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

</form>

## CSS基本知识

CSS全称为“层叠样式表（Cascading Style Sheets）”，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。

CSS样式由选择符和声明组成，而声明又由属性和值组成。

选择符：又称选择器，指明网页中要应用样式规则的元素。

声明：英文大括号中的部分就是声明，属性和值之间用英文冒号；多条声明之间用英文符“;”隔开。

CSS中用/\*注释语句\*/来注释代码，html用<!---注释语句->

Eg:

p{

font-size:12px;

color:red;

font-weight:bold;

}

## CSS样式基本知识

**1）内联式css样式，直接写在现有的html标签中。**

注意:要写在元素的开始标签中，并且css样式代码要写在style=””双引号中，多条用分号隔开。

<p style="color:red">这里文字是红色。</p>

**2）嵌入式css样式，写在当前文件中**

将css样式代码写在<style type=”text/css”></style>标签之间。一般情况下嵌入式css代码写在<head></head>标签之间。

<style type="text/css">

span{

color:red;

}

</style>

**3）外部式css样式，写在单独的一个文件中**

也称为外联式，该文件以.css为扩展名，在<head>内使用<link>标签将css样式文件链接到HTML文件内。

注意：css样式文件名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。

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

**4）css样式优先级**

三种样式的优先级:内联式>嵌入式>外部式（就近原则），该优先级前提是css样式在相同的权值情况下。

嵌入式大于外部式有一个前提:嵌入式css样式的位置一定在外部式的后面。即<link href=”xx.css”>代码在<style type=”text/css”>...</style>代码的前面。

## CSS选择器

选择器{

样式；

}

选择器指明了{}中的样式的作用对象，也就是样式作用于网页中的哪些元素。

**1）标签选择器**

标签选择器其实就是html代码中的标签，如<html>、<body>、<p>、<img>。

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

**2）类选择器**

类选择器在css样式编码中是最常用到的。其中类是为标签设置的类名。

语法：.类选择器名称{css样式代码;}

注意：英文圆点开头；类选择器名称可以任意起名（但不要起中文）。

.stress{color:red;}

**3）ID选择器**

ID选择器类似于类选择符，但有一些重要的区别：

1. 为标签设置id=“ID名称”，而不是class=“类名称”；
2. ID选择符的前面是井号(#)，而不是英文圆点（.）。

**4）类和ID选择器的区别（谨记id的唯一性）**

相同：可以用于任何元素。

不同：a)ID选择器只能在文档中使用一次（id是唯一的），类选择器可以使用多次；b)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。

.stress{

    color:red;

}

.bigsize{

    font-size:25px;

}

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

**5）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。

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

该代码会是class名为food下的子元素li加入红色实现边框。

**6）包含（后代）选择器**

即加入空格，用于选择指定标签元素下的后辈元素。

.first span{color:red;}

注意：子选择器仅指它的直接后代，而后代选择器是作用于所有子后代元素。

**7）通用选择器**

通用选择器是功能最强大的选择器，使用一个\*号指定，作用是匹配html中的所有标签元素。

**8）伪类选择器**

允许给html不存在的标签（标签的某种状态）设置样式，如给html中一个标签元素的鼠标滑过状态来设置字体颜色：

a:hover{color:red;}

目前，可以兼容所有浏览器的“伪类选择符”就是a标签上使用：hover了。

**9）分组选择符**

分组选择符（，）。

h1,span{color:red;}

相当于下面代码

h1{color:red;}

span{color:red;}

## CSS继承、层叠和特殊性

**1）继承**

css的某些样式具有继承性。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。

p{color:red;}

上述代码，颜色设置不仅应用于P标签，还应用与p标签中的所有子元素文本。

但有一些css样式不具备继承性，如border：1px solid red;该段代码只是给p标签设置了边框为1的像素、红色、实心边框线，而对于子元素标签没起到作用。

**2）特殊性**

p{color:red;}

.first{color:green;}

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

上段代码的两个css样式都匹配到p这个标签上，但是green才是正确颜色。因为浏览器根据权值来判断使用哪种css样式，使用权值高的。

权值规则：

标签的权值为1，类选择法的权值为10，ID选择符的权值最高为100。如下所示：

p{color:red;} **/\*权值为1\*/**

p span{color:green;} **/\*权值为1+1=2\*/**

.warning{color:white;} **/\*权值为10\*/**

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

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

注意：继承也有权值，但很低，有文献指出它只有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。

**4）重要性**

使用!important为某些样式设

置最有最高权值。

注意：!important要写在分号前。

当网页制作者不设置css样式时，浏览器会按照自己的一套样式来显示网页。并且用户也可以在浏览器中设置自己习惯的样式。样式的优先级：浏览器样式<网页制作者样式<用户自己设置的样式，但是!important优先级样式是个例外，权值高于用户自己设置的样式。

## CSS格式化排版

**1）文字排版**

#### a)字体

可以使用css样式为网页中的文字设置字体、字号、颜色等样式属性。

body{font-family:"宋体";}

注意：不要设置不常用的字体，因为如果用户在本地电脑上没有所安装设置的字体，就会显示浏览器默认字体。现在一般网页喜欢使用“微软雅黑”。

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

或

body{font-family:"微软雅黑";}

注意：第一种方法比第二种方法兼容性更好一些。

#### b）字号、颜色

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

#### c）粗体

可以使用css样式来改变文字的样式：粗体、斜体、下划线、删除线。如下代码可以实现粗体：

p span{font-weight:bold;}

#### d）斜体

p a{font-style:italic;}

#### e）下划线

p a{text-decoration:underline;}

#### f）删除线

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

**2）段落排版**

#### a）缩进

p{text-indent:2em;}

2em的意思就是文字的2倍大小。

#### b）行间距（行高）

p{line-height:1.5em;}

#### c）中文字间距、字母间距

使用letter-spacing设置文字间隔或字母间距。

h1{

letter-spacing:50px;

}

如果要设置英文单词之间的间距，采用word-spacing来实现。

h1{

word-spacing:50px;

}

#### d）对齐

使用text-align样式代码设置文本、图片位置。

h1{

text-align:center;//center居中 left 居左 right居右

}

## CSS盒模型

**1）元素分类**

在css中，html中的标签元素大体被分为三种不同类型：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>。

#### a）块级元素

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

a{display:block;}

块级元素特点：

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；

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，是它父容器的100%（和父元素的宽度一致），除非设定一个宽度。

#### b）内联元素

<span>、<a>、<label>、<strong>和<em>是典型的内联元素（行内元素）（inline）。块状元素可以通过display:inline将元素设置为内联元素。

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。

注意：当行内元素之间有”回车”、“tab”、“空格”时就会出现间隙。解决办法：写在一行、之间不要有空格之类的符号或者使用font-size:0

#### c）内联块状元素

内联块状元素就是同时具备内联元素、块状元素的特点，代码display:inline-block就是将元素设置为内联块状元素。<input>、<img>就是内联块状元素。

inline-block元素特点：

和其他元素在同一行；

元素的高度、宽度、行高以及顶和底边距都可设置。

**2）盒模型**

块级标签都具备盒子模型的特征。

1. 边框

盒子模型的边框就是围绕着内容及补白的线，可以设置这条线的粗细、样式、颜色（边框的三个属性）。

div{

border:2px solid red;

}

上面的代码可以拆分为：

div{

border-width:2px;

border-style:solid;

border-color:red;

}

注意：border-style常见有dashed（虚线）、dotted（点线）、solid（实线）。

border-color可以设置为十六进制颜色。

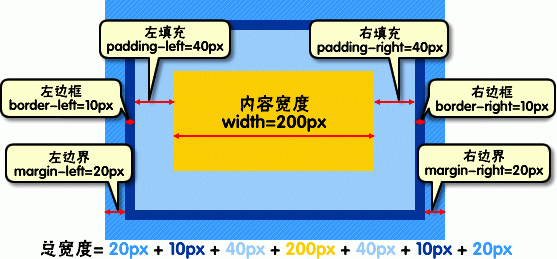
boder-width可以设置为thin|medium|thick（不常用），通常用像素（px）。

css样式允许只为一个方向的边框设置样式：

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

1. 宽度和高度

盒模型的宽度和高度和我们平常所说的物体的宽度和高度理解是不一样的，css内定义的宽（width）和高（height），指的是填充以里的内容范围。因此，一个元素实际宽度=左边界+左边框+左填充+内容宽度+右填充+右边框+右边界。



1. 填充

元素的内容和边框之间可以设置距离，称指为填充。填充分为上、下、左、右（顺时针），可以分开写。

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

如果上、右、下、左填充都为10px，可以写为：

div{padding:10px;}

如果上下填充都一样为10px，可以写为：

div{padding:10px 20px;}

1. 边界

元素与其他元素之间的距离可以使用边界（margin）来设置，与填充类似。边界使用margin标识。

margin与padding区别：padding在边框里，margin在边框外。

## CSS布局模型

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。CSS包括3中布局模型，即流动模型（Flow）、层模型（Layer）和浮动模型（Float）。

**1）流动模型**

流动（Flow）是默认的网页布局模式。流动模型典型特征：第一点，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。第二点，在流动模型下，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。

**2）浮动模型**

设置元素浮动可以让两个块状元素并排显示。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。

div{

width:200px;

height:200px;

border:2px red solid;

float:left;

}

<div id="div1"></div>

<div id="div2"></div>

**3）层模型**

层布局模型就像图像软件photoShop中非常流行的图层编辑功能一样，每个图层都能够精确定位操作。css定义了一组定位属性来支持层布局模型。层模型有三种形式：绝对定位position：absolute；相对定位position：relative；固定定位position：fixed。

1. 绝对定位

设置position:absolute表示绝对定位，作用是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体属性的父包含块进行绝对定位。如果不存在这样的包含块，则相对于body元素，即相对于浏览器窗口。如下代码可以实现div元素相对于浏览器窗口向右移动100px，向下移动50px。

div{

width:200px;

height:200px;

border:2px red solid;

position:absolute;

left:100px;

top:50px;

}

<div id="div1"></div>

1. 相对定位

Position:relative表示相对定位，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起来），然后相对于以前的位置移动，移动幅度有left、right、top、bottom属性确定，偏移前的位置保留不对。以下代码实现相对于以前位置向下移动50px，向右移动100px。

#div1{

width:200px;

height:200px;

border:2px red solid;

position:relative;

left:100px;

top:50px;

}

<div id="div1"></div>

1. 固定定位

fixed表示固定定位，与absolute定位类型类似。但它的相对移动的坐标是视图本身。由于视图本身是固定的，不会随窗口的滚动条滚动而变化，除非你在屏幕中移动窗口屏幕的位置，或者改变浏览器窗口的显示大小，因此固定定位的元素会始终位于浏览器窗口内视图的某个位置，不会受文档流动的印象。这与background-attachment:fixed;属性功能相同。

1. Relative和Absolute组合使用

使用relative帮忙可以实现相对于其他元素进行定位，但必须遵守以下规范：参照定位的元素必须是相对定位元素的前辈元素；参照定位的元素必须加入position:relative;定位元素加入position：absolute；

## 其他

**1）字体缩写**

body{

font-style:italic;

font-variant:small-caps;

font-weight:bold;

font-size:12px;

line-height:1.5em;

font-family:"宋体",sans-serif;

}

可以缩写为：

body{

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

}

注意：使用这一简写方式至少要指定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font-size与line-height中间要加“/”斜杠。

**2）颜色值**

字体颜色color，背景颜色background-color，边框颜色border。设置颜色放法有很多种

a）英文命令颜色

p{color:red;}

b）RGB颜色

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

c）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

**3）长度值**

a）像素

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，目前大多数设计者都倾向于使用像素px作为单位。

b）em

em是本元素给定字体的font-size值。若font-size=18px，则1em=18px。

注意：当给font-size设置单位为em时，此时计算的标准以标签的父元素的font-size为基础。

c）百分比

和em类似，以标签的font-size为标准

## CSS样式设置小技巧

**1）水平居中设置-行内元素**

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

**2）水平居中设置-定宽块状元素**

通过设置左右margin值为auto来实现居中。

**3）水平居中总结-不定宽块状元素方法**

不定宽块状元素有三种方法居中：加入table标签；设置display:inline方法，与第一种方法类似，显示类型为行内元素，进行不定宽元素的属性设置；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。

a）加入table标签

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），因此可以看作一个定宽度块元素，然后再利用定宽度块居中的margin方法，使其水平居中。

b）设置display：inline

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。

与加入table标签相比优势是不用增加无语义标签，但也存在着一些问题，它将块状元素的display类型改为inline，变成了行内元素，所以少了一些功能，比如设定长度值。

c）设置浮动和相对定位

通过给父元素设置float，然后给父元素设置position：relative和left:50%，子元素设置position：relative和left:-50%来实现水平居中。

理解：假想ul层的父层（即下面例子中的div层）中间有条平分线将ul层的父层（div层）平局分成两份，ul层的css代码是将ul层的最左端与ul的父层的（div层）的平分线对齐；而li层的css代码则是将li层的平分线与ul层的最左端（也就是div层的平分线）对齐，从而实现li层的居中。

<body>

<div class="container">

<ul>

<li><a href="#">1</a></li>

<li><a href="#">2</a></li>

<li><a href="#">3</a></li>

</ul>

</div>

</body>

<style>

.container{

float:left;

position:relative;

left:50%

}

.container ul{

list-style:none;

margin:0;

padding:0;

position:relative;

left:-50%;

}

.container li{float:left;display:inline;margin-right:8px;}

</style>

**4）垂直居中**

垂直居中分为两种情况：父元素高度确定的单行文本以及父元素高度确定的多行文本。

1. 父元素高度确定的单行文本

通过设置父元素的height和line-height高度一致来实现。（height：该元素的高度，line-height:行高，指在文本中，行与行之间的基线间的距离。）

line-height和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度时，就有内容脱离了块。

<div class="container">

hi,imooc!

</div>

<style>

.container{

height:100px;

line-height:100px;

background:#999;

}

</style>

1. 父元素高度确定的多行文本

方法一：

使用插入table（包括tbody、tr、td）标签，同时设置vertical-align：middle。在父元素中设置vertical-align会对inline-block类型的子元素都有用。

<body>

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

<div>

<p>看我是否可以居中。</p>

</div>

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

</body>

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

td标签默认情况下就默认设置了vertical-align为middle，所以此处不需要显示设置。

方法二：

该方法兼容性较差，在chrome、firefox及IE8以上的浏览器下可以设置块级元素的display为table-cell（设置为表格单元显示），激活vertical-align属性。

好处是不用添加多余的无意义标签，缺点是兼容性不好，不兼容IE6、7而且这样修改display的block变成了table-cell，破坏了原有的块状元素性质。

**5）隐性改变display类型**

只要html代码中出现position：absolute或者float:left/right二者之一，元素的display显示类型就会自动变为一display:inline-bloack的方式显示。

# 网页布局

网页布局又称版式布局，是网页UI设计师将有限的视觉元素进行有机的排列组合。

网页设计的特点：网页可以自适应宽度；网页的长度理论上可以无限延长。

分栏又叫分列，常见的布局分为一列布局、两列布局、三列布局、混合布局。

1. **一列布局**

通常作为网站的首页，适合放置简单的内容，一列布局经常是固定宽度。

1. **两列布局**
2. **三列布局**
3. 设置百分比，可以利用float实现

.left{

width:200px;

height:500px;

float:left;

background:#ccc;

}

.main{

height:500px;

float:left;

background:#999;

}

.right{

width:300px;

height:500px;

float:right;

background:#ddd;

}

1. 要求左右两列是指定像素宽度，float失效，采用position

.left{

width:200px;

height:500px;

position:absolute;

left:0;

top:0;

background:#ccc;

}

.main{

height:500px;

margin:0 300px 0 200px;

background:#999;

}

.right{

width:300px;

height:500px;

position:absolute;

right:0;

top:0;

background:#ddd;

}

1. **混合布局**
2. **自动居中-列布局**
3. 标准文档

W3C标准：是由万维网联盟指定的一系列标准，包括：

结构化标准语言（HTML和XML）；

表现标准语言（CSS）；

行为标准语言（DOM和ECMAScript）。

倡导结果、样式、行为三者分离。

1. CSS中的定位机制

3种定位机制：标准文档流（Nomal flow），浮动（Floats），绝对定位（Absolute positioning）。

1. 标准文档流

特点：从上到下，从左到右，输出文档内容。由块级元素和行级元素组成。

块级元素：从左到右撑满页面，独占一行，触碰到页面边缘时，会自动换行。

行级元素：可以在同一行显示，不会破坏文档结构。

块级元素和行级元素都是盒子模型。

1. 盒子模型

盒子模型=网页布局的基石。由4部分组成：边框、外边距、内边距、内容。

盒子的3D模型：border第一层，content+padding第二层，background-img第三层，background-color第四层，margin第五层。

1. **横向两列布局**

1）浮动布局及float属性

能够实现横向多列布局，通过设置float属性实现。3个属性值：left左浮动，right右浮动，none不浮动。特点：元素左移或右移，直至触碰到容器为止。设置了浮动的元素，仍旧处于标准文档流中。

设置浮动后的标签会对紧邻该标签之后的元素产生影响。

2）清除浮动的常用方法

对受到浮动影响的元素设置如下属性：

a）clear属性——常用clear:both;

clear:left;或者clear:right;（要求清楚受到了左浮动还是右浮动的影响）

b）同时设置width:100%（或固定宽度）+ overflow:hidden;

空标签也能清除浮动，但是对元素是无意义的，不建议使用。

3）横向两列布局

网页布局最常见的方式之一。

主要应用技能：float属性——使纵向排列的块级元素，横向排列；margin属性——设置两列之间的间距。

父包含块若未设置浮动，子包含块设置了浮动，则父标签的宽度扩展将会受到影响。

当父包含块缩成一条时，用clear:both;清除浮动无效，它一般用于紧邻后面的元素清除浮动。此时应该使用溢出属性overflow:hidden;

Ps:为了保证子元素正常显示，不要设置父包含块的高度，从而使其可以自适应子元素高度。

1. **绝对定位布局**

1）绝对定位布局简介

绝对定位布局是使用position属性实现的网页布局，是css中规定的第三种定位机制，能够实现横向多列布局及较为复杂的定位。比如带有遮罩层效果的提示框、固定层效果、全屏广告等。

Position属性拥有三种定位形式：静态定位、相对定位、绝对定位。

可以设置4个属性值：static（静态定位，默认状态），relative（相对定位），absolute（绝对定位），fixed（固定定位）。

2）相对定位

特点：相对于自身原有位置进行偏移；仍处于标准文档流中；拥有偏移属性和z-index属性。

3）绝对定位

特点：建立了以包含块为基准的定位；完全脱离了标准文档流；拥有偏移属性和z-index属性。

绝对定位与相对定位相比在选择参照基准的时候要复杂的多。

未设置偏移量：无论是否存在已经定位祖先元素，都保持在初始位置；脱离标准文档流。

当一个元素设置绝对定位，没有设置宽度时，则元素宽度根据内容进行调节。

设置偏移量：偏移参照基准：若无已定位祖先元素，以html为偏移参照基准；有已经定位祖先元素，以距其最近的已定位祖先元素为基准进行偏移。

4）使用绝对定位实现横向两列布局

常用于一列宽度固定，另一列宽度自适应。

主要应用技能：relative——父元素相对定位；absolute——自适应宽度元素绝对定位。

注意：固定宽度列的高度>自适应宽度的列！！！（因为设置绝对定位的列脱离文档流，无法撑起整个页面。）

三、网页布局之结构与表现原则