**CSS**

* + **CSS是Cascading Styles Sheets的缩写，中文译名为层叠样式表，是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言**
  + **HTML和CSS的关系就是“内容”和“形式”的关系，由HTML组织网页的结构和内容，而通过CSS来决定页面的表现形式**

**为什么使用CSS**

**由于html的主要功能是描述网页的结构，所以控制网页外观和表现的能力很差，**

**1 无法精确到像素级调整文字大小，行间距等，**

**2 不能对多个网页元素进行统一的样式设置，只能一个一个元素的设置。**

**学习CSS可实现对网页的外观和排版进行更灵活的控制，使网页更美观**

**1.html：用html控制样式**

**代码冗余**

**如果要修改元素的样式，也要一个个的改，修改工作量大**

* + **很多表现用html是根本无法修改的**
  + **最典型的是：**
    - **行距**
    - **以像素级修改字体大小**
    - **为文字设置背景色，边框等**

**2.html：用css控制样式解决代码冗余**

**3.html：用css控制样式解决修改元素的样式（修改字体颜色）**

* + **一条CSS样式规则由选择器(selector)和声明(declarations)组成。**
  + **选择器是为了选中网页中某些元素的**

* + **选择器可以是一个标记名，表示将网页中该标记的所有元素都选中，也就是定义了CSS规则的作用对象。**
  + **选择器也可以是一个自定义的类名，表示将自定义的一类元素全部选中，为了对这一类元素进行标识，必须在这一类的每个元素的标记里添加一个html属性class＝"类名"。**
  + **选择器还可以是一个自定义的id名，表示选中网页中某一个唯一的元素，同样，该元素也必须在标记中添加一个html属性id＝"id名"让CSS来识别。**

* + **而声明则用于定义元素样式。**
    - **声明又可以分为属性和值，属性和值用冒号隔开。（注意CSS的属性和值的写法与html属性的区别）。**
    - **属性和值可以设置多个，从而实现对同一标记声明多种样式风格。**
    - **如果要设置多个属性和值，则每条声明之间要用分号隔开；**
    - **如果属性的某个值不是一个单词，则值要用引号引起来：**

**p {font-family: "sans serif"}；**

* + **如果一个属性有多个值，则每个值之间要用空格隔开：**

**a {margin:6px 4px 3px};**

* + **要为某个属性设置多个候选值，则每个值之间用逗号隔开：**

**p {font-family: "Times New Roman", Times, serif};**

**在HTML中引入CSS的方法**

2017年8月1日

11:40

**在HTML中，引入CSS的方法有**

**行内式**

**内嵌式**

**导入式**

**链接式**

**行内式**

**html标记都有一个通用的style属性，行内式就是在该属性内添加CSS属性和值。**

**这种方式由于CSS属性就在标记内，其作用对象就是标记内的元素。所以不需要指定CSS的选择器，只需要书写CSS属性和值**

**注意：可以将CSS样式规则写在注释符内，不会被忽略**

嵌入式

**嵌入式将页面中各种元素的CSS样式设置集中写在<style>和</style>之间，<style>标记是专用于引入嵌入式CSS的一个html标记，它只能放置在文档头部**

**链接式和导入式**

**目的：都是将一个独立的CSS文件引入到HTML文件**

**区别：**

1.语法

* **链接式是在网页文档头部通过link标记引入外部CSS文件，格式如下：**

**<link href="style1.css" rel="stylesheet" type="text/css" />**

* **而使用导入式，则需要使用如下语句：**

**<style type="text/css">**

**@import url("style.css");**

**</style>**

2.效果

**链接式：会在装载页面主体部分之前装载CSS文件，这样显示出来的网页从一开始就是带有样式效果的**

**导入式：要在整个页面装载完之后再装载CSS文件，如果页面文件比较大，则开始装载时会显示无样式的页面**

3.**import把CSS文件的内容复制到HTML文件中，link直接向CSS文件读取所定义的CSS样式**

**推荐：**

**制作网站则主要应采用链接式方法引入CSS**

**CSS选择器分类**

2017年8月1日

13:17

* + **标记选择器**
  + **伪类选择器**
  + **类选择器**
  + **id选择器**

**标记选择器**

**标记是元素固有的属性，CSS标记选择器用来声明哪种标记采用哪种CSS样式，因此，每一种html标记的名称都可以作为相应的标记选择器的名称。**

**<style type="text/css">**

**p{ /\* 标记选择器 \*/**

**color:blue;**

**font-size:18px;**

**}**

**</style>**

**伪类选择器**

**所谓伪类就是指标记的状态**

**伪类选择器的标记和伪类之间用“:”隔开**

**伪类选择器可以看成是一种特殊的标记选择器，它用来选中在某种状态下的标记**

**类选择器**

**类选择器以半角“.”开头，且类名称的第一个字母不能为数字**

**id选择器**

* + **id选择器的使用方法与class选择器基本相同，不同之处在于一个id选择器只能应用于html文档中的一个元素，因此其针对性更强，而class选择器可以应用于多个元素。**
  + **id选择器以半角“#”开头，且id名称的第一个字母不能为数字**

**选择器的组合**

2017年8月1日

16:04

* + **每个选择器都有它的作用范围，前面介绍了三种基本的选择器，它们的作用范围都是一个单独的集合，**
  + **如标记选择器的作用范围是具有该标记的所有元素的集合，类选择器的作用范围是自定义的某一类元素的集合，有时我们希望对几种选择器的作用范围取交集、并集、子集后对选中的元素再定义样式，这时就要用到复合选择器了，它是通过对几种基本选择器的组合，实现更强、更方便的选择功能。**

**复合选择器就是两个或多个基本选择器，通过不同方式组合而成的选择器。主要有:**

* + **交集选择器**
  + **并集选择器**
  + **后代选择器**

交集选择器

**交集选择器是由两个选择器直接连接构成，其结果是选中两者各自作用范围的交集**

* **其中第一个必须是标记选择器，第二个必须是类选择器或id选择器。**
* **这两个选择器之间不能有空格。**
* **交集选择器将选中同时满足前后二者定义的元素，也就是前者定义的标记类型，并且指定了后者的类别或id的元素**

* **Firefox还支持属性选择器**

**如：input[type=‘submit’]**

**并集选择器**

**对多个选择器进行集体声明，多个选择器之间用“,”隔开**

**h2.special, .special, #one{ /\* 集体声明 \*/**

**text-decoration:underline;**

}

**通配选择符 \***

**如果要对网页中所有的元素进行集体声明可使用通配选择符 \***

**\*{ /\* 全局声明 \*/**

**color:purple; /\* 文字颜色 \*/**

**font-size:15px; /\* 字体大小 \*/**

**}**

**后代选择器**

**写法：把外层的标记写在前面，内层的标记写在后面，之间用空格隔开。**

**和其他所有CSS选择器一样，后代选择器定义的具有继承性的样式同样也能被其子元素继承**

css样式设计原则

* + **定义标记选择器最省事，它不需在元素的html标记里添加class或id属性，因此初学者最喜欢定义标记选择器或由标记选择器组成的包含选择器。但有些标记（如a标记）在网页文档的各部分都出现得很多，它们在各部分的样式风格往往不一样。**
  + **例如导航条内的a标记就要求和文档其他地方的a标记样式不同，我们当然可以将导航条内的各个a标记定义为一个类，但这就要将导航条内的各个a标记都添加一个class属性，实际上，可以将导航条内a标记的父标记（如ul）添加一个id属性（#menu），然后用包含选择器（#menu a）就可以选中导航条内的各个a标记了。**
  + **对于几个不同的选择器，如果它们有一些共同的样式声明，就可以先用并集选择器对它们先集体声明，然后再单独定义某些选择器的特殊样式。**

子选择器

* + **子选择器用于选中元素的直接后代（即儿子），它的定义符号是大于号（>）**

* + **相邻（adjacent-sibling）选择器**

**定义符号是加号（+），相邻选择器将选中紧跟在它后面的一个兄弟元素（这两个元素具有共同的父元素）**

属性选择器

* + **匹配属性**

**a[name] {color:purple; }**

**img[border] {border-color:gray;}**

**[special] {color:red;}**

* + **匹配属性和值**

**input[type="submit"] {background:purple; }**

* + **匹配单个属性值**

**h2[class~="two"] { color: red;}**

**<h2 class="one two three">文字是红色</h2>**

**新增加的伪类选择器**

* + **:focus**

**用于定义元素获得焦点时的样式**

* + **:first-child**

**:first-child伪类选择器用于匹配它的父元素的第一个子元素，也就是说这个元素是它父元素的第一个儿子，而不管它的父元素是哪个。**

**p:first-child{ font-weight: bold; }**

**<body>**

**<p>这一段文字是粗体</p> <!--第1行，被选中-->**

**<h2>下面哪些文字是粗体的呢</h2>**

**<p>这一段文字不是粗体</p>……**

* + **first-letter**

**:first-letter用于选中元素内容的首字符。例如：**

* + **p:first-letter{ font-size: 2em; float: left;}**

**它可以选中段落p中的第一个字母或中文字符。**

* + **:first-line**

**:first-line用于选中元素中的首行文本。例如：**

**p:first-line{ font-weight: bold; letter-spacing: 0.3em;}**

* + **:before和:after**

**p:before, p:after{content: "--"; color:red;}**

**<p>看这一段文字的左右</p>**

**<p>这一段文字左右</p></body>**

**用CSS修饰文本**

2017年8月1日

13:19

**控制文本样式的属性主要有font-属性类和text-属性类，**

**修改颜色的color属性和行高line-height属性。**

**text-indent表示首行缩进，在每段开头空两格通常是用text-indent: 2em;来实现，text-decoration: none;表示去掉下划线，line-height: 150%;表示调整为1.5 倍行间距。**

**text-align：文本对齐属性**

**left (居左，缺省值)**

**right (居右)**

**center (居中)**

**justify (两端对齐)**

**.p2 {text-align:right}**

**注意：行内元素无法应用对齐属性**

**文本颜色属性：color**

**用颜色属性(color)可以改变文本的字体颜色。**

**.p1{color:gray}**

**text-decoration：文本修饰属性**

**none (无，缺省值)**

**underline (下划线)**

**overline (上划线)**

**line-through (当中划线)**

**.p2 {text-decoration: underline overline;}**

**letter-spacing：字间距属性**

**设定字符之间的距离。**

**normal (缺省值)**

**length (长度，可以用绝对单位(cm, mm, in, pt, pc)或者相对单位 (em, ex, px))**

**.p1 {letter-spacing: 3mm}**

**text-indent：文本缩进属性**

**首行缩进。其值有以下设定方法：**

**length (长度，可以用绝对单位(cm, mm, in, pt, pc)或者相对单位 (em, ex, px))**

**percentage (百分比，相当于父对象宽度的百分比)**

**.p1 {text-indent: 8mm}**

**word-spacing：词间距属性**

**设定字符之间的距离。**

**normal (缺省值)**

**length (长度，可以用绝对单位(cm, mm, in, pt, pc)或者相对单位 (em, ex, px))**

**.p1 {word-spacing: 3mm}**

**line-height：行高**

**这个属性设定每行之间的距离。其值有以下设定方法：**

**normal (缺省值)**

**length (长度，可以用绝对单位(cm, mm, in, pt,pc)或者相对单位 (em, ex, px))**

**percentage (百分比，相当于父对象高度的百分比)**

**.p1 {line-height:1cm}**

**CSS的特性**

2017年8月1日

13:20

**CSS具有两个特性：层叠性和继承性。**

**层叠性**

**指当有多个选择器都作用于同一元素时，即多个选择器的作用范围发生了重叠，CSS怎样处理**

**CSS的处理原则：**

* **如果多个选择器定义的规则不发生冲突，则元素将应用所有选择器定义的样式**

* **如果多个选择器定义的规则发生了冲突，则CSS按选择器的优先级让元素应用优先级高的选择器定义的样式。**

**优先级：**

**行内样式 > ID样式 > 类别样式 > 标记样式。**

**总的原则是：越特殊的样式，优先级越高。**

**!important**

**强制改变选择器的优先级，则优先级为:**

**!important >行内样式 > ID样式 > 类别样式 > 标记样式**

**另外，如果在同一个选择器中定义了两条相冲突的规则，则以后一条为准。**

**继承性**

**如果子元素定义的样式没有和父元素定义的样式发生冲突，那么子元素将继承父元素的样式风格，并可以在父元素样式的基础上再加以修改，自己定义新的样式，而子元素的样式风格不会影响父元素**

* + **注意**

**不是所有的CSS属性都具有继承性，一般是CSS的文本属性具有继承性，而其他属性（如背景属性、盒子属性等）则不具有继承性。**

**具有继承性的属性：color, font-类，text-indent, text-align, text-decoration, line-height, letter-spacing, border-collapse等**

**无继承性的属性：**

**text-decoration:none,**

**所有背景属性，**

**所有盒子属性,**

**布局属性等**

**应用CSS修饰文本和超链接**

2017年8月1日

13:22

**动态超链接**

动态超链接是通过CSS伪类选择器实现的，因为伪类可以描述超链接在不同状态下的样式

|  |  |
| --- | --- |
| **伪 类** | **作 用** |
| **a:link** | **超链接的普通样式风格，即正常浏览状态时的样式** |
| **a:visited** | **被单击过的超链接的样式风格** |
| **a:hover** | **鼠标指针悬停在超链接上时的样式风格** |
| **a:active** | **当前激活（在鼠标单击与释放之间发生）的样式风格** |

**注意**

* + **链接伪类选择器的书写应遵循LVHA的顺序，即CSS代码中四个选择器出现的顺序应为**

**a:link→ a:visited→ a:hover→ a:active，若违反这种顺序鼠标停留和激活样式就不起作用了。**

* + **各种a的伪类选择器将继承a标记选择器定义的样式。**
  + **a:link选择器只能选中具有href属性的a标记，而a选择器能选中所有a标记，包括用a标记作为锚点**

**CSS属性的值和单位**

2017年8月1日

13:22

**值是对属性的具体描述，而单位是值的基础。**

**没有单位，浏览器将不知道一个边框是10厘米还是10象素。CSS中较复杂的值和单位有颜色取值和长度单位。**

**Tip：HTML的属性的值一般不要写单位，这是因为html属性的取值可用的单位很少，要么是象素，要么是百分比**

**颜色**

**命名颜色**

**p{color: red; }**

**RGB颜色**

**td{ color: rgb(139,31,185); }**

**td{ color: rgb(12%,201,50%); }**

**其值可以取0~255之间的整数，也可以是0%~100%的百分数，**

**不过Firefox浏览器不支持百分数值。**

**16进制颜色**

**参数取值范围为：00－FF（对应十进制仍为0－255），如果每个参数各自在两位上的数值相同，那么该值也可缩写成“#RGB”的方式。例如：#ffcc33可以缩写为#fc3。**

**长度单位**

**绝对单位**

**英寸（in）、厘米（cm）、毫米（mm）、磅（pt）和pica（pc）。**

**由于同一个长度在不同的显示器或者相同显示器不同的分辨率中显示并不相同，不会按显示器的比例去显示。所以绝对单位很少用。**

**相对单位**

**元素的字体高度（em）**

**em就是元素原来给定的字体font-size的值，如果元素原来给定的font-size值是14px，那么1em就是14px。**

**字母x的高度（ex）**

**ex是以字体中小写x字母为基准的单位，不同的字体有不同的x高度，因此即使font-size相同而字体不同的话，1ex的高度也会不同。**

**象素（px）**

**象素指显示器按分辨率分割得到的小点，因为显示器由于分辨率或大小不同，象素点的大小是不同的，所以象素也是相对单位**

**目前大多数设计者都倾向于使用象素作为单位。**

**百分比**

**td{font-size:12px; line-height: 160%; }**

**/\*设定段落的行高为字体高度的160% \*/**

**hr{ width: 80% }**

**/\* 线段长度是相对于浏览器窗口的80% \*/**

**CSS的滤镜属性简介**

2017年8月1日

13:23

**滤镜(filter)属性是CSS的一个扩展部分，它能够渲染对象元素，创建出艺术效果**

**CSS滤镜的标识符是“filter”，语法：**

**filter:filtername(parameters);**

**.alpha{ /\*类选择器\*/**

**filter:alpha(opacity=50);**

**}**

**滤镜属性可分为无参滤镜和有参滤镜**

**无参滤镜**

**Gray：使对象产生灰度图效果，仅对图像有作用。**

**Invert：使对象产生“底片”效果。**

**Xray：使对象产生“X光片”效果。**

**FlipH：使对象产生水平翻转效果。**

**FlipV：使对象产生垂直翻转效果。**

**Light：使对象产生一种模拟光源的投射效果**

**有参滤镜**

**Alpha：设置对象的透明度。**

**Blur：使对象产生模糊效果。**

**Dropshadow：设置对象的阴影效果。**

**Chroma：将对象中指定的颜色设置为透明色。**

**Glow：在对象的边缘产生类似发光的效果。**

**BlendTrans：设置对象的淡入淡出效果。**

**RevealTrans：设置对象之间的切换效果**

* + **使网页变黑白**

**由于gray滤镜只能使图像变黑白，要使网页整体变黑白需对html标记使用如下滤镜代码：**

**html { filter:progid:DXImageTransform.Microsoft.BasicImage(grayscale=1); }**

* + **用Alpha属性制作渐变的hr分隔条**

**hr {filter: Alpha(opacity=10%, FinishOpacity=100%, Style=1, StartX=0, StartY=0, FinishX=500, FinishY=8);**

**color: #FF0033; }**

* + **设置网页之间的切换效果**

**<meta http-equiv="Page-Enter" content="blendTrans(duration=10)">**

**<meta http-equiv="Page-Exit" content="blendTrans(duration=10)">**

**<meta http-equiv="Page-Enter" content="RevealTrans(duration=10,transition=14)">**

**CSS的盒子模型**

2017年8月1日

13:23

* + **盒子模型是CSS的基石之一，它指定元素如何显示以及（在某种程度上）如何相互交互**
  + **页面上的每个元素都被浏览器看成是一个矩形的盒子，这个盒子由元素的内容、填充、边框和边界组成。**
  + **网页就是由许多个盒子通过不同的排列方式（上下排列，并列排列，嵌套排列）堆积而成。**
  + **每个HTML元素都可以看作是一个装了东西的盒子**
  + **盒子里面的内容到盒子的边框之间的距离即填充(padding),盒子本身有边框(border),而盒子边框外和其它盒子之间，还有边界(magin),如图所示**
  + **默认情况下盒子的边框是无，背景色是透明，所以我们在默认情况下看不到盒子**

**元素盒子大小的计算**

**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**

Tip

* + **当将文档声明DOCTYPE删除后，IE 6对网页的解释会进入quirk（怪异）模式，此时盒子的宽度等于左边界＋宽度＋右边界**
  + **因此当使用了盒子属性后切忌删除DOCTYPE**

**padding属性**

* + **填充padding属性，也称为盒子的内边距。就是盒子边框到内容之间的距离，和表格的填充属性（cellpadding）比较相似。（如果填充属性为0，则盒子的边框会紧挨着内容，这样通常不美观。）**
  + **当对盒子设置了背景颜色或背景图像后，那么背景会覆盖padding和内容组成的范围，并且默认情况下背景图像是以padding的左上角为基准点在盒子中平铺的**

**边框border属性**

* + **盒子模型的margin和padding属性比较简单，只能设置宽度值，最多分别对上、右、下、左设置宽度值。而边框border则可以设置宽度、颜色和样式。**
  + **分别是border-width（宽度）、border-color（颜色）和border-style（样式）其中border-style属性可以将边框设置为实线（solid）、虚线（dashed）、点划线（dotted）、双线（double）等效果**
  + **边框border属性有个有趣的特点，即两条交汇的边框之间是一个斜角，我们可以通过为边框设置不同的颜色，再利用这个斜角，制作出像三角形一样的效果**

**盒子模型的特性**

* + **边界值margin可为负，填充padding不可为负**
  + **边框border默认值为0，即不显示**
  + **行内元素，如a，定义上下边界不影响行高**

**对盒子模型的思考**

* + **边框是实的，我们可以看到实实在在的边框，而填充和边界都是虚的，我们只能看到他们对元素的影响**
  + **盒子模型中只能设置两类颜色，即边框颜色和背景颜色**
  + **盒子模型可设置三类距离，即边界距离margin，填充距离padding和边框值border**

**属性的简写形式**

* + **方法是按照规定的顺序，给出2个、3个或者4个属性值，它们的含义将有所区别，具体含义如下：**
  + **如果给出2个属性值，前者表示上下边框的属性，后者表示左右边框的属性；**
  + **如果给出3个属性值，前者表示上边框的属性，中间的数值表示左右边框的属性，后者表示下边框的属性；**
  + **如果给出4个属性值，依次表示上、右、下、左边框的属性，即顺时针排序。**

**盒子的定位**

2017年8月1日

13:23

**CSS 有三种基本的定位机制**

标准流

浮动

定位

**tip：除非设置浮动属性或定位属性，否则所有盒子都是在标准流中定位。顾名思义，标准流中元素盒子的位置由元素在HTML中的位置决定**

标准流

* + **HTML元素在标准状况下的定位方式**
  + **行内元素在同一行内横向排列**
  + **块级元素占满整个一行，在页面中竖向排列**
  + **元素不会移动到其它地方去，对于嵌套的元素盒子也是嵌套的关系**

* + **行内元素的盒子永远只能在浏览器中得到一行高度的空间（行高由line-height属性决定，如果没设置该属性，则是内容的默认高度），如果给它设置上下border，margin，padding等值，导致其盒子的高度超过行高，那么它的盒子上下部分将和其他元素的盒子重叠。**

**因此，不推荐对行内元素直接设置盒子属性，一般先设置行内元素以块级元素显示，再设置它的盒子属性**

**当增加行内元素的边界和填充时，行内元素a占据浏览器的高度并没有增加，下面这个div块仍然在原来的位置，导致行内元素盒子的上下部分重叠，而左右部分不会受影响**

**display属性**

* + **通过display属性可控制元素是以行内元素显示还是以块级元素显示，或不显示**
  + **Display属性值：**

**block**

* + **总是在新行上开始**
  + **高度，行高以及顶和底边距都可控制**
  + **宽度缺省是它的容器的100%，除非用width设定一个宽度  
    <div>, <p>, <h1>, <form>, <ul> 和 <li>是块元素的例子。**

**Inline**

* + **和其他元素都在一行上**
  + **高，行高及顶和底边距不可改变；**
  + **宽度就是它的文字或图片的宽度，不可改变。  
    <span>, <a>, <label>, <input>, <img>, <strong> 和<em>是inline元素的例子**

**None**

* + **当某个元素被设置成了隐藏元素之后，浏览器会完全忽略掉这个元素，该元素将不会被显示，也不会占据文档中的位置。像title元素默认就是此类型。**
  + **比较visibility: hidden;**
  + **制作下拉菜单、tab面板等有时就需要用display: none把菜单或面板隐藏起来**

**list-item**

**在html中只有li元素默认是此类型，将元素设置为列表项元素后将按列表元素显示，再通过设置列表选项可使元素的左边出现小黑点**

**需要使用display属性切换的情况**

**让一个inline元素从新行开始；**

**让块元素和其他元素保持在一行上；**

**控制inline元素的宽度;**

**控制inline元素的高度（对导航条特别有用）；**

**无须设定宽度即可为一个块元素设定与文字同宽的背景色**

**背景的控制**

2017年8月1日

13:24

**backgroud**

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **可用值** |
| **background** | **设置背景的所有控制选项，是其他所有背景属性的缩写** | **其他背景属性的值的集合** |
| **background-color** | **设置背景颜色** | **rgb颜色**  **命名颜色**  **16进制颜色** |
| **background-image** | **设置背景图片** | **url** |
| **background-repeat** | **设置背景图片的平铺方式** | **repeat（完全平铺）**  **repeat-x（横向平铺）**  **repeat-y（纵向平铺）**  **no-repeat（不平铺）** |
| **background-attachment** | **设置背景图片固定还是随内容滚动** | **scroll（默认值）**  **fixed** |
| **background-position** | **设置背景图片显示的起始位置** | **（[top] [left]**  **[center] [center]**  **[bottom] [right]）**  **（[x%] [y%]**  **[x-pos] [y-pos]）** |

* + **background属性是所有背景属性的缩写形式，就像font属性一样，其缩写顺序为：**

**background: background-color || background-image || background-repeat || background-attachment || background-position**

**如body {background:#EFF4FF url(images/body\_bg.jpg) repeat-x fixed;}**

**盒子的浮动-float**

2017年8月1日

13:26

* + **在标准流中，块级元素的盒子都是上下排列，行内元素的盒子都是左右排列，限制太大**

**为了灵活布局==浮动和定位方式进行盒子的排列**

* + **在标准流中，一个块级元素在水平方向会自动伸展，在它的父元素中占满整个一行；**

**而在竖直方向和其他元素依次排列，不能并排，使用“浮动”方式可以改变排列方式。**

* + **float属性：**

**none：默认值，也就是标准流通常的情况**

**Left：元素向其父元素的左侧靠紧**

**right：元素向其父元素的右侧靠紧**

**设置float为left或right同时盒子的宽度不再伸展，而是收缩，**

**在没设置宽度时，会根据盒子里面的内容来确定宽度。**

**浮动的清除**

* + **clear是清除浮动属性**
    - **left：表示该盒子的左边不允许有浮动的对象**
    - **right：表示该盒子的右边不允许有浮动的对象**
    - **Both：表示两边都不允许有浮动对象，因此该盒子将会在浏览器中另起一行显示**
    - **none（默认值）**
  + **清除浮动是清除其它盒子浮动对该元素的影响，而设置浮动是让元素自身浮动，两者并不矛盾，因此可同时设置元素清除浮动和浮动**

横竖转换

|  |  |
| --- | --- |
| #nav{ |  |
|  | font-size: 14px;} |
|  | #nav a { |
|  | color: #FF0000; |
|  | background-color: #99CCFF; |
|  | text-align: center; |
|  | text-decoration: none; |
|  | display: block; |
|  | padding:6px 10px 4px; |
|  | margin:0 2px; /\*设置了左右边界，使两个a元素间有4px的水平间距\*/ |
|  | border: 1px solid #3399FF; |
|  | float:left; /\*使a元素浮动，实现水平排列\*/ |
|  | } |
|  | #nav a:hover { |
|  | color: #FFFFFF; |
|  | background-color: #993300; } |
|  | </style> |
|  | </head> |
|  | <body> |
|  | <div id="nav"> |
|  | <a href="[#](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\4-7-21%E8%8F%9C%E5%8D%95%E7%AB%96%E6%A8%AA%E8%BD%AC%E6%8D%A2.html)">首 页</a> |
|  | <a href="[#](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\4-7-21%E8%8F%9C%E5%8D%95%E7%AB%96%E6%A8%AA%E8%BD%AC%E6%8D%A2.html)">中心简介</a> |
|  | <a href="[#](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\4-7-21%E8%8F%9C%E5%8D%95%E7%AB%96%E6%A8%AA%E8%BD%AC%E6%8D%A2.html)">政策法规</a> |
|  | <a href="[#](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\4-7-21%E8%8F%9C%E5%8D%95%E7%AB%96%E6%A8%AA%E8%BD%AC%E6%8D%A2.html)">常用下载</a> |
|  | <a href="[#](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\4-7-21%E8%8F%9C%E5%8D%95%E7%AB%96%E6%A8%AA%E8%BD%AC%E6%8D%A2.html)">为您服务</a> |
|  | <a href="[#](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\4-7-21%E8%8F%9C%E5%8D%95%E7%AB%96%E6%A8%AA%E8%BD%AC%E6%8D%A2.html)">技术支持和服务</a> |
|  | </div> |
|  | </body>    1-3-1:电子科技大学首页 |

**定位position**

2017年8月1日

13:26

**position 属性值的含义**：

**static**

表示不使用定位属性定位，也就是盒子按照标准流或浮动方式布局。默认值

**relative**

元素框偏移某个距离。元素仍保持其未定位前的形状，它原本所占的空间仍保留。

**absolute**

元素框从文档流完全删除，并相对于其包含块定位。包含块可能是文档中的另一个元素或者是初始包含块。元素原先在正常文档流中所占的空间会关闭，就好像元素原来不存在一样。元素定位后生成一个块级框，而不论原来它在正常流中生成何种类型的框。

**fixed**

元素框的表现类似于将 position 设置为 absolute，不过其包含块是视窗本身（扫描二维码，广告，IE6不支持）。

**偏移属性**

**偏移属性是指 top、left、bottom、right四个属性，它们的取值可以为像素等绝对单位，也可以是百分比**

#mydiv {

position:absolute;

left: 50%；

top: 30px;

}

**偏移属性仅对设置了定位属性的元素有效**

固定定位：**fixed**

被固定的元素不会随着滚动条的拖动而改变位置。在视野中，固定定位的元素的位置是不会改变的

**相对定位**

* + **使用相对定位的盒子的位置定位依据常以标准流的排版方式为基础，然后使盒子相对于它在原来的标准位置偏移指定的距离。相对定位的盒子仍在标准流中，它后面的盒子仍以标准流方式对待它。**
  + **如果对一个元素定义相对定位属性（position: relative;），那么它将保持在原来的位置上不动。如果再对它通过top、left等属性值设置垂直或水平偏移量，那么它将“相对于”它原来的位置发生移动。**
  + **相对定位的特点：**
    - **元素原来占据的位置仍然会保留，也就是说相对定位的元素未脱离标准流；**
    - **因为是使用了定位属性的元素，所以会和其他元素发生重叠。**
  + **作用**
    - **一是让元素相对于它原来的位置发生位移，同时不释放它原来占据的位置；**
    - **二是让元素的子元素以它为定位基准进行定位，同时它的位置保持不变，这时相对定位的元素成为包含框，一般是为了帮助里面的元素进行绝对定位。**

案例：相对定位-图像阴影.Html(**原理是在img元素外套一个外围容器，将外围容器的背景设置为灰色，作为img元素的阴影，同时不设置填充边界等值使外围容器和图片一样大，这时图像就正好把外围容器的背景完全覆盖。再设置图像相对于原来的位置往左上方偏移几个像素，这样图像的右下方就露出了阴影**)

|  |  |
| --- | --- |
|  | img { |
|  | padding: 6px; |
|  | border: 1px solid #465B68; |
|  | background-color: #fff; |
|  | position: relative; |
|  | left: -5px; |
|  | top: -5px; |
|  | } |
|  | div.shadow { |
|  | background-color: #CCCCCC; |
|  | float:left; |
|  | } |
|  | </style> |
|  | </head> |
|  | <body> |
|  | <div class="shadow"> |
|  | <img src="[images/works.jpg](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\images\works.jpg)" width="150" height="140" /> |
|  | </div> |
|  | </body> |

案例：相对定位-居中.html(

* + **设置wrap的定位是相对于它原来的位置，而它原来默认的位置是在浏览器窗口中左对齐，然后将其左边框移动到浏览器的正中央，这是通过left:50%实现的，这样就找到了浏览器的中线，再使用负边界法将盒子的一半宽度从中线位置拉回到左边，这样就实现了水平居中。**
  + **想一想：如果把#container选择器中（left:50%; margin-left:-380px;）改为（right:50%; margin-right: -380px;），还能实现居中吗？**

)

|  |  |
| --- | --- |
| #container { |  |
|  | position:relative; |
|  | width:760px; |
|  | left:50%; |
|  | margin-left:-380px; |
|  | } |
|  | </style> |
|  | </head> |
|  | <body> |
|  | <div id="container"><img src="[images/big.jpg](file:///G:\old\UESTC\html\51CTO%E4%B8%8B%E8%BD%BD-%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E3%80%8A%E5%9F%BA%E4%BA%8EWeb%E6%A0%87%E5%87%86%E7%9A%84%E7%BD%91%E9%A1%B5%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%88%B6%E4%BD%9C%E3%80%8B%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E6%BA%90%E4%BB%A3%E7%A0%81%E5%92%8C%E7%B4%A0%E6%9D%90\%E7%AC%AC%E5%9B%9B%E7%AB%A0\images\big.jpg)" width="760" height="570" /></div> |
|  | </body> |

绝对定位

* + **绝对定位，盒子的位置以他的包含框为基准进行定位。绝对定位的盒子从标准流中脱离。这意味着它们对其他元素的盒子的定位没有影响，其他的盒子就好像这个盒子不存在一样。**
  + **注意：绝对定位是以他的包含框的边框内侧为基准进行定位，因此改变包含框的填充值不会对绝对定位元素的位置造成影响。**
  + **绝对定位的偏移值是指从它的包含框边框内侧到元素的外边界之间的距离，如果修改元素的margin值会影响元素内容的显示位置。如果不设置绝对定位元素的偏移值，那么它位于未设置绝对定位属性时在标准流中的初始位置**

**注意：**

* + **设置为绝对定位（position:absolute;）的元素，并非总是以浏览器窗口为基准进行定位。实际上，绝对定位元素是以它的包含框为基准进行定位的，所谓包含框是指距离它最近的设置了定位属性的父级元素的盒子。如果它所有的父级元素都没设置定位属性，那么包含框就是浏览器窗口**
  + **我们对em元素的父级元素p设置定位属性，使p元素成为em元素的包含框，这时，em元素就不再以浏览器窗口为基准进行定位了，而是以它的包含框p元素的盒子为基准进行定位，**

|  |  |  |
| --- | --- | --- |
|  | 相对定位  relative | 绝对定位  absolute |
| 定位基准 | 以它自己原来的位置为基准 | 以距离它最近的设置了定位属性的父级元素为定位基准，若它所有的父元素都没设置定位属性，则以浏览器窗口为定位基准 |
| 原来的位置 | 还占用着原来的位置，未脱离标准流 | 占用其原来的位置，已经脱离标准流，其它元不素就当它不存在一样 |

**z-index**

* + **z-index属性用于调整定位时重叠块之间的上下位置。与它的名称一样，想象页面为x-y轴，那么垂直于页面的方向就为z轴，z-index值大的盒子位于值小的盒子的上方，可以通过设置z-index值改变盒子之间的重叠次序。默认的z-index值为0，当两个盒子的z-index值一样时，则保持原来的高低覆盖关系**
  + **z-index属性和偏移属性一样，只对设置了定位属性（position 属性值为 relative 或 absolute 或 fixed）的元素有效。**