#### 一些基础

Win快捷键

**常用的快捷键、**ctrl+c 复制

ctrl+v 粘贴

ctrl+x 剪切（就是移动文件，在原来的文件夹ctrl+x一个文件，然后在新文件夹中ctrl+v粘贴）

ctrl+tab 切换（具体切换什么，要看是什么软件）

alt+F4 关闭程序

F2 重命名

F5 刷新，比如看网页的时候，想刷新网页，按f5

ctrl+z 撤销，就是这一步干错了，就ctrl+z撤销

**windows+E**  打开资源管理器

windows+D 显示桌面

ctrl+空格 **切换中英文，严禁用shift键切换，所有同学现在调整搜狗拼音输入法：**

常见的联网文件传输

C:\Users\Danny\AppData\Local\Microsoft\Windows\Temporary Internet Files

这个文件夹叫做临时文件夹**（文件夹知道存在就可以了，不用自己试着去找）**

**服务器和客户端的传输**

服务器上存放着网页的相关文件，包括html文件、css文件、js文件、图片等。当我们打开浏览器，输入网址，我们的计算机就会对这些文件发出HTTP请求。

服务器收到请求之后，会把这些文件通过HTTP协议，传输到我们的计算机中（保存到了刚才那个临时文件夹中）。这些文件，将在我们计算机本地的浏览器中，进行渲染、呈递

|  |
| --- |
| 1. http://www.icdjs.com/1.html |

就是服务器上面的1.html文件

|  |
| --- |
| 1. http://www.icdjs.com/ |

看似没有精确到一个文件，但是有一个规定，就是**index.html是默认的首页文件**。

index就是英语“目录”的意思。

|  |
| --- |
| 1. http://www.icdjs.com/aaa/b.html |

服务器上面有一个aaa文件夹，这个文件夹里面有一个b.html文件。

本质

**网页是真实物理的文件。并且一个网页是很多的物理文件组成的：html文件、图片文件、js文件、css文件。这些文件要通过特殊软件才能上传到服务器上。然后就能让用户看了。用户通过浏览器，访问网址，服务器上面的文件就会通过http请求悄悄地传输到用户的电脑中的临时文件夹中，在用户的电脑中执行、渲染、呈递**

Html 超文本标记语言

只能增加文本的语义 、结构

可以压缩网页的尺寸的

Sublime使用

P+ tab 自动补全

Ctrl+shift+d复制向下

Ctrl+shift+k 删除当前

Ctrl+shift+上 向上移动当前行

Li\*10 tab 产生10行

点击鼠标中键 多行编辑

Ctrl +左键 选中多行进行编辑 end 到行尾部 home 移动到行头

Ctrl + 滚动 放大缩小

分屏 查看

#### Html部分

Html文档类型

<!DOCTYPE html>

|  |  |
| --- | --- |
| HTML4.01 | **Strict** 严格的，体现在一些标签不能使用，比如u  **Transitional** 普通的  **Frameset** 带有框架的页面 |
| XHTML1.0  严格体现在小写标签、闭合、引号 | **Strict** 严格的，体现在一些标签不能使用，比如u  **Transitional** 普通的**（我们学习的版本）**  **Frameset** 带有框架的页面 |

**Html：xt 用写u一类的标签当做css勾来做**

**Html：5 h5**

Html <head>

head> 元素

<head> 元素是所有头部元素的容器。<head> 内的元素可包含脚本，指示浏览器在何处可以找到样式表，提供元信息，等等。

以下标签都可以添加到 head 部分：<title>、<base>、<link>、<meta>、<script> 以及 <style>

<title> 标签定义文档的标题。

title 元素在所有 HTML/XHTML 文档中都是必需的。

title 元素能够：

* 定义浏览器**工具栏中的标题**
* 提供页面被添加到**收藏夹时**显示的标题
* 显示在搜索**引擎结果中的页面标题**

<base> 元素

<base> 标签为页面上的所有链接规定默认地址或默认目标（target）：

<head>

<base href="http://www.w3school.com.cn/images/" />

<base target="\_blank" />

</head>

<link> 元素

<link> 标签定义文档与外部资源之间的关系。

<link> 标签最常用于连接样式表：

<head>

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

</head>

Media 属性

所有浏览器都支持值为 "screen"、"print" 以及 "all" 的 media 属性。

如果您打印本页面，或者在打印预览模式中打开它，此页面会通过 media="print" 样式表进行格式化。"print" 样式表会把本页面格式化为白底黑字。

rel属性

rel 属性的 "stylesheet 所有浏览器支持

type属性

type 属性规定被链接文档的 MIME 类型。

该属性最常见的 MIME 类型是 "text/css"，该类型描述样式表

Href 属性

超链接的 URL。可能的值：

* 绝对 URL - 指向另一个站点（比如 href="http://www.example.com/theme.css"）
* 相对 URL - 指向站点内的某个文件（href="/themes/theme.css"）

<style> 元素

<style> 标签用于为 HTML 文档定义样式信息。

您可以在 style 元素内规定 HTML 元素在浏览器中呈现的样式：

<head>

<style type="text/css">

body {background-color:yellow}

p {color:blue}

</style>

</head>

 <meta> 元素

元数据可用于浏览器（如何显示内容或重新加载页面），搜索引擎（关键词），或其他 web 服务。

针对搜索引擎的关键词

一些搜索引擎会利用 meta 元素的 name 和 content 属性来索引您的页面。

描述：

<meta name="**description**" content="Free Web tutorials on HTML, CSS, XML" />

关键词：

<meta name="**keywords**" content="HTML, CSS, XML" />

name 和 content 属性的作用是描述页面的内容。

5s延迟后重定向

<head>

<meta **http-equiv="Refresh" content="5;url=http://www.w3school.com.cn**" />

</head>

字符集

第一种：UTF-8

|  |
| --- |
| 1. <meta http-equiv="Content-Type" content="text/html;charset=**UTF-8**"> |

第二种：gb2312

|  |
| --- |
| 1. <meta http-equiv="Content-Type" content="text/html;charset=**gb2312**"> |

也可以写成gbk

|  |
| --- |
| 1. <meta http-equiv="Content-Type" content="text/html;charset=**gbk**"> |

**保存类型要与声明的东西相同**

**utf-8 内容多，文件大**

**gb2312 内容小，文件小加载快**

<meta http-equiv="charset" content="iso-8859-1">

<meta http-equiv="expires" content="31 Dec 2008">

###### <body>

标题 <h1> - <h5>

主标题 一级标签

段落 <p>

HTML标签是分等级的，HTML将所有的标签分为两种：容器级、文本级。

顾名思义，**容器级的标签，里面可以放置任何东西；文本级的标签里面，只能放置文字、图片、表单元素**。

p标签是一个文本级标签。从学习p的第一天开始，就要死死记住：p里面只能放文字、图片、表单元素。其他的一律不能放。’

正确的：

|  |
| --- |
| 1. <h3>个人经历</h3> 2. <p> 3. Angelababy从小怎么怎么样 4. </p> |

错误的：

|  |
| --- |
| 1. <p> 2. **<h3>个人经历</h3>** 3. Angelababy从小怎么怎么样 4. </p> |

里面的话会截断p ，然后后面的就轮空了

连接 <a href=[www.baidu.com](http://www.baidu.com)”> 百度一下</a>

**注释：**在 href 属性中指定链接的地址。

图像

<img src="w3school.jpg" width="104" height="142" />

HTML 标签对大小写不敏感：<P> 等同于 <p>。许多网站都使用大写的 HTML 标签。但是推荐小写

<p> 浏览器会自动地在段落的前后添加空行。（<p> 是块级元素）

**HTML 折行**

如果您希望在不产生一个新段落的情况下了\* 进行换行（新行），请使用 <br /> 标签：

当显示页面时，浏览器会移除**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。**

属性

href a标签的href中互联网资源要加上 http:// 也可以加载本地资源

align 剧中什么的

bgcolor 背景色

background 背景图 文件夹下面路径直接上就行了 在body里面

属性是 key value类型 value一般放在双引号里面，如果自带双引号必须放在单引号厘

标题

H1 –H5

浏览器会自动在标题前后添加空行

Html会自动在块级元素前后添加一个空行，比如段落和标题元素前后

搜索引擎使用标题为您的网页的结构和内容编制索引

标题就最好就是标题的属性

水平线

<hr/>

换行：

|  |
| --- |
| 1. <br /> |

不另起一个段落，进行换行。

|  |
| --- |
| 1. <p> 2. 哈<br />哈哈 3. </p> |

网页中99.9999%需要换行的时候，是因为另起了一个段落，所以要用p，不要用<br />

br是英语break打断的意思。

段落

<

<pre>

这是

预格式文本。

它保留了 空格

和换行。

</pre>

html文本格式化（可以认为是内容样式）

**物理样式 丢弃了**

~~<~~**~~b~~**~~>bond加粗</~~**~~b~~**~~>  
<~~**~~i~~**~~>italic 斜体</~~**~~i~~**~~>  
<~~**~~small~~**~~>小一号的</~~**~~small~~**~~>  
<~~**~~big~~**~~>大一号的字</~~**~~big~~**~~>  
下标<~~**~~sub~~**~~>下标subscript</~~**~~sub~~**~~>  
上标<~~**~~sup~~**~~>supscript上标</~~**~~sup~~**~~>~~

~~<ins>下划线，插入~~

~~<del> 文字中画线 删除~~

~~<tt> 标签呈现类似打字机或者等宽的文本效果。~~

语意标签

|  |  |
| --- | --- |
| [<em>](http://www.w3school.com.cn/tags/tag_em.asp) | 把文本定义为强调的内容。 |
| [<strong>](http://www.w3school.com.cn/tags/tag_strong.asp) | 把文本定义为语气更强的强调的内容。 |
| [<dfn>](http://www.w3school.com.cn/tags/tag_dfn.asp) | 定义一个定义项目。 |
| [<code>](http://www.w3school.com.cn/tags/tag_code.asp) | 定义计算机代码文本。 |
| [<samp>](http://www.w3school.com.cn/tags/tag_samp.asp) | 定义样本文本。 |
| [<kbd>](http://www.w3school.com.cn/tags/tag_kbd.asp) | 定义键盘文本。它表示文本是从键盘上键入的。它经常用在与计算机相关的文档或手册中。 |
| [<var>](http://www.w3school.com.cn/tags/tag_var.asp) | 定义变量。您可以将此标签与 <pre> 及 <code> 标签配合使用。 |
| [<cite>](http://www.w3school.com.cn/tags/tag_cite.asp) | 定义引用。可使用该标签对参考文献的引用进行定义，比如书籍或杂志的标题。 |

**地址**

<address>

Written by <a href="mailto:webmaster@example.com">Donald Duck</a>.<br>

Visit us at:<br>

Example.com<br>

Box 564, Disneyland<br>

USA

</address>

定义缩写

缩写，鼠标移动到上面显示全称abbreviation， title显示全部的内容

<**abbr title="etcetera"**> etc.</**abbr**>

**定义首字母缩写。**首字母缩略词

<acronym title="World Wide Web">WWW</acronym>

Bdo 定义文字的方向 rtl从右到左，ltr从左到右

<bdo dir="rtl">Here is some text</bdo>

<blockquote>块引用

**缩进换行加外边距**

**提示：**请使用 [q 元素](http://www.w3school.com.cn/tags/tag_q.asp)来标记短的引用。

**注释：**如需把页面作为 strict XHTML 进行验证，那么 <blockquote> 元素必须包含块级元素，比如这样：

<blockquote>

<p>here is a long quotation here is a long quotation</p>

</blockquote>

短引用

<q>Here is a short quotation here is a short quotation</q>

浏览器经常在引用的内容周围添加引号。

<q> 与 <blockquote> 的区别

<q> 标签在本质上与 [<blockquote>](http://www.w3school.com.cn/tags/tag_blockquote.asp) 是一样的。不同之处在于它们的显示和应用。<q> 标签用于简短的行内引用。如果需要从周围内容分离出来比较长的部分（通常显示为缩进的块），请使用 <blockquote> 标签。

Html样式

**所有的格式化代码均可移出 HTML 文档，然后移入一个独立的样式表**

**样式的使用方法： style定义样式或者rel=“stylesheet”**

**type=”text/css” 在外链和内部的时候指出规则是什么，内嵌不用**

**外部连接样式 表**

<head>

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

</head>

**内部样式表**

<head>

<style type="text/css">

body {background-color: red}

p {margin-left: 20px}

</style>

</head>

Body 和 后面的p都是样式名

**内嵌样式**

<p style="color: red; margin-left: 20px">

This is a paragraph

</p>

Div

是一个**块级元素**。**换行**是 <div> 固有的唯一格式表现。可以通过 <div> 的 class 或 id 应用额外的样式。

不必为每一个 <div> 都加上类或 id，虽然这样做也有一定的好处。

**可以对同一个 <div> 元素应用 class 或 id 属性，但是更常见的情况是只应用其中一种**。这两者的主要差异是，c**lass 用于元素组**（类似的元素，或者可以理解为某一类元素），而 **id 用于标识单独的唯一的元素。**

<span>

标签被用来组合文档中的**行内元素。没有特殊效应但可以定义样式**

可以对同一个 <span> 元素应用 class 或 id 属性，但是更常见的情况是只应用其中一种。

<font>

规定文本的字体、字体尺寸、字体颜色。

<font size="3" color="red">This is some text!</font>

**建议用样式代替**

Html链接

超文本传输

**文字链接**

<a href="http://www.microsoft.com/">本文本</a>

**图像连接**

<a href="/example/html/lastpage.html">

<img border="0" src="/i/eg\_buttonnext.gif" />

</a>

这样的链接会在图像前加一个空格的，因为有换行，图像属于文本，则前后都会出现短线，正确的做法是 <a><img src=”></a>

A标签属性介绍

**href** 给出网址hypertext reference 超文本引用；

万维网得加上<http://www.w3school.com.cn/> 本网站地址直接鞋/index.html

**target** 给出新标签还是自己页面 \_blank 新页面 \_top 跳出原来的框架？？？？？？

Download 要下载的文件名

<a href="http://www.w3school.com.cn/" target="\_blank">Visit W3School!</a>

<a href="/images/myw3schoolimage.jpg" download="w3logo">

锚

**也可以用id代替name**

首先，我们在 HTML 文档中对锚进行命名（创建一个书签）：

<a name="tips">基本的注意事项 - 有用的提示</a>

然后，我们在同一个文档中创建指向该锚的链接：

<a href="#tips">有用的提示</a>

您也可以在其他页面中创建指向该锚的链接：

<a href="http://www.w3school.com.cn/html/html\_links.asp#tips">有用的提示</a>

在上面的代码中，我们将 # 符号和锚名称添加到 URL 的末端，就可以直接链接到 tips 这个命名锚了

**命名锚经常用于在大型文档开始位置上创建目录。**

邮件 ？？？？

<p>

这是邮件链接：

<a href="mailto:someone@microsoft.com?subject=Hello%20again">发送邮件</a>

</p>

**提示和注释**

**提示：**如果不使用 href 属性，则不可以使用如下属性：download, hreflang, media, rel, target 以及 type 属性。

**提示：**被链接页面通常显示在当前浏览器窗口中，除**非您规定了另一个目标（target 属性）**。

**提示：**请使用 CSS 来设置链接的样式。

**A是一个文本级的标签**

A能放在p里面，但p不能放在里面，可以把a直接当作文本处理

Html图像

<img src=””>

**图片来自本地文件**

<img src="/i/eg\_mouse.jpg" width="128" height="128" />

<**img src=".imag/girl.jpg"**/>

本地的网站，路径是从该html的所在目录开始的， ../上一级的

当图片在文件夹里面的时候：

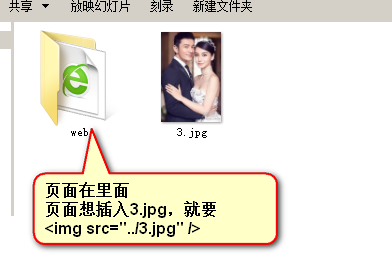


|  |
| --- |
| 1. <img src="**images/**baby.jpg" alt="巴黎结婚照" /> |

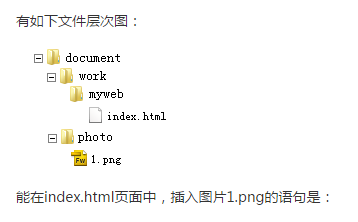
如果在很深的文件夹中，也不怕，可以一直罗列下去：

|  |
| --- |
| 1. <img src="**images/jiehunzhao/**baby.jpg" alt="巴黎结婚照" /> |

如果情况变得复杂，我们的图片在浅一层的文件夹中：



面试题



标准答案：

|  |
| --- |
| 1. <img src="../../photo/1.png" /> |

**来自万维网**

<img src="http://www.w3school.com.cn/i/w3school\_logo\_white.gif" />

**动态图和静态图插入一样的**

浏览器将图像显示在文档中图像标签出现的地方。如果你将图像标签置于两个段落之间，那么浏览器会首先显示第一个段落，然后显示图片，最后显示第二段。

文件无法加载的时候，alt 的值显示出来

<img src="boat.gif" alt="Big Boat">

**图像的排列位置**

<p>图像 <img src="/i/eg\_cute.gif" align="bottom"> 在文本</p>

图像底部和文本框的底部对齐，

<p>图像 <img src ="/i/eg\_cute.gif" align="middle"> 在文本</p>

图像中部和文本框的底部对齐，

<p>图像 <img src ="/i/eg\_cute.gif" align="top"> 在文本</p>

图像的顶部和文本狂对其

<p>请注意，bottom 对齐方式是默认的对齐方式。</p>

<p>

<img src ="/i/eg\_cute.gif" align ="left">

带有图像的一个段落。图像的 align 属性设置为 "left"。图像将浮动到文本的左侧。

</p>

<p>

<img src ="/i/eg\_cute.gif" align ="right">

带有图像的一个段落。图像的 align 属性设置为 "right"。图像将浮动到文本的右侧。

</p>

**图片的大小**

通过改变 img 标签的 "height" 和 "width" 属性的值，您可以放大或缩小图像。

背景图片

<**body background=".idea/imag/girl.jpg"**>

**Gif 和jpg均可以作为html的背景，**

如果图像小于页面，图像就会重复

**图像映射（图片内部有各种可以点的超链接）**

<img src="planets.jpg" border="0" usemap="#planetmap" alt="Planets" />

<map name="planetmap" id="planetmap">

<area shape="circle" coords="180,139,14" href ="venus.html" alt="Venus" />

<area shape="circle" coords="129,161,10" href ="mercur.html" alt="Mercury" />

<area shape="rect" coords="0,0,110,260" href ="sun.html" alt="Sun" />

</map>

**注意在 area中可以定义 target ="\_blank"**

注释：</b>img 元素中的 "usemap" 属性引用 map 元素中的 "id" 或 "name" 属性（根据浏览器），所以我们同时向 map 元素添加了 "id" 和 "name" 属性**。**

nohref 属性规定该区域没有相关的链接。 Href 指出链接位置

|  |
| --- |
| **coords 描述 坐标** |
| *x1,y1,x2,y2* | 如果 shape 属性设置为 "rect"，则该值规定矩形左上角和右下角的坐标。 |
| *x,y,radius* | 如果 shape 属性设置为 "circ"，则该值规定圆心的坐标和半径。 |
| *x1,y1,x2,y2,..,xn,yn* | 如果 shape 属性设置为 "poly"，则该值规定多边形各边的坐标。如果第一个坐标和最后一个坐标不一致，那么为了关闭多边形，浏览器必须添加最后一对坐标 |

Html表格

表格 table

表格的行 tr （table row）

表格数据td （table date）

文本、图片、列表、段落、表单、水平线、表格

表头 th table head

空单元格里面写 &nbsp;

**标题**<caption>我的标题</caption> caption 在tr上 在table中

**跨两列** 不写第三个td就行了

<tr>

<th>姓名</th>

<th **colspan="2"**>电话</th>

</tr>

**跨两行 被夸的部分对应的td不写**

<tr>

<th rowspan="2">电话</th>

<td>555 77 854</td>

</tr>

<tr>

<td>555 77 855</td>

</tr>

Table的属性 <table>

Border 边框粗细 0 无 越大月粗

Cellpadding 内容距离边框的距离

Cellspacing 各个单元格之间的间距

Bgcolor 背景色

Background 背景图片

Frame 一些方向上的边框显示

Td属性

Align 内容的位置

Bgcolor

Background

细分的table内的标签，三个必须一起出现

Thead <thead> 内部必须拥有 <tr> 标签！

Tbody

Tfoot

这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上

内部以char 的字幕对其时，没有支持的浏览器

<thead align="char" char="M" charoff="2">

属性

[valign](http://www.w3school.com.cn/tags/att_tbody_valign.asp) 垂直方向上的对齐方式 top middle bottom baseline

align 水平内容对其方式 right left center justify整理版面 char

**colgroup** 可以跨列定义样式用

<table width="100%" border="1">

<colgroup span="2" align="left"></colgroup>

<colgroup align="right" style="color:#0000FF;"></colgroup>

<tr

Span colgrpoup 跨的元素

**Width**

|  |  |
| --- | --- |
| *pixels* | 设置以像素计的宽度值（例子：width="50"）。 |
| *percent* | 设置以包围元素的百分比计的宽度值（例子：width="50%"）。 |

假如表格的宽度是 100 像素，第一个列组是 20 像素，第二个列组是 50%，那么剩余的可用像素是 30 像素。如果可用像素是 30px，那么可以设置部分一为 "1\*"，部分二为 "2\*"

**Col** 定义每一列的样式

<table width="100%" border="1">

<col align="left" />

<col align="left" />

<col align="right" />

<tr>

Html列表

**无序列表 前面的点的样式 type 定**

<ul type="circle"> difined 设置的前面的点或者什么的

<li>咖啡</li>

<li>茶</li>

<li>牛奶</li>

</ul>

**li是一个容器级标签，li里面什么都能放**

**有序列表** order list 用的少，多数是用ul在前面加上标签即可

<ol start="50">

<li>咖啡</li>

<li>牛奶</li>

<li>茶</li>

</ol>

默认是数字序列的

type="A" 或者 type=”a” 序列变成 字母的

可以嵌套定义

**自定义列表**

dl表示definition list 定义列表

dt表示definition title 定义标题

dd表示definition description 定义表述词儿

dt、dd都是容器级标签，想放什么都可以。所以，现在就应该更加清晰的知道：

**用什么标签，不是根据样子来决定，而是语义**

**<dl>**

<**dt**>计算机</dt>

**<dd**>用来计算的仪器 ... ...</dd>

<dt>显示器</dt>

<dd>以视觉方式显示信息的装置 ... ...</dd>

</dl>

Html的块

<div>

可以通过div 和 span 将html元素组合起来

<div>用于文档布局 和css一起使用，加上id或者class属性

**div 元素的 "align" 属性不被支持。**

div在浏览器中，**默认是不会增加任何的效果改变的**，但是语义变了，div中的所有元素是一个小区域。**div标签是一个容器级标签，**里面什么都能放，甚至可以放div自己

<span>

内联元素，用于**文本的容器**，与css一起使用时，可用于**部分文本设置样式属性**

**span里面只能放置文字、图片、表单元素。 span里面不能放p、h、ul、dl、ol、div。**

Html的布局 ？？

没看懂div如何变化的

<**html**>  
<**head**>  
 <**style type="text/css"**>  
 **div#container**{**width**:500**px**}  
 **div#header** {**background-color**:**#99bbbb**;}  
 **div#menu** {**background-color**:**#ffff99**;**height**:200**px**;**width**:150**px**;**float**:**left**;}  
 **div#content** {**background-color**:**#EEEEEE**;**height**:200**px**;**width**:350**px**;**float**:**left**;}  
 **div#footer** {**background-color**:**#99bbbb**;**clear**:**both**;**text-align**:**center**;}  
 **h1** {**margin-bottom**:0;}  
 **h2** {**margin-bottom**:0;**font-size**:18**px**;}  
 **ul** {**margin**:0;}  
 **li** {**list-style**:**none**;}  
 </**style**>  
</**head**>  
<**body**>  
<**div id="container"**>  
 <**div id="header"**>  
 <**h1**>Main Title of Web Page</**h1**>  
 </**div**>  
 <**div id="menu"**>  
 <**h2**>Menu</**h2**>  
 <**ul**>  
 <**li**>HTML</**li**>  
 <**li**>CSS</**li**>  
 <**li**>JavaScript</**li**>  
 </**ul**>  
 </**div**>  
 <**div id="content"**>Content goes here</**div**>  
 <**div id="footer"**>Copyright W3School.com.cn</**div**>  
</**div**>  
</**body**>  
</**html**>

html表单

action 提交的地址 method 提交方式

<form name="input" action="html\_form\_action.asp" method="get">

**Enctype encoding type**

enctype="text/plain"

enctype 属性规定在发送到服务器之前应该如何对表单数据进行编码。

默认地，表单数据会编码为 "application/x-www-form-urlencoded"。就是说，在发送到服务器之前，所有字符都会进行编码（空格转换为 "+" 加号，特殊符号转换为 ASCII HEX 值）

|  |  |
| --- | --- |
| **值** | **描述** |
| application/x-www-form-urlencoded | 在发送**前编码所有字符（默认）** |
| multipart/form-data | 不对字符编码。  在使用**包含文件上传控件的表单时**，必须使用该值。 |
| text/plain | 空格转换为 "+" 加号，但不对特殊字符编码。 |

**Novalidate**

novalidate="novalidate" 提交表单的时候不进行认证

输入

Type

属性值

|  |  |
| --- | --- |
| **值** | **描述** |
| button | 定义可点击按钮（多数情况下，用于通过 JavaScript 启动脚本）。 |
| checkbox | 定义复选框。 |
| file | **定义输入字段和 "浏览"按钮，供文件上传**。 |
| hidden | 定义隐藏的输入字段。 |
| image | 定义图像形式的提交按钮。 |
| password | 定义密码字段。该字段中的字符被掩码。 |
| radio | 定义单选按钮。 |
| reset | 定义重置按钮。重置按钮会清除表单中的所有数据。 |
| submit | 定义提交按钮。提交按钮会把表单数据发送到服务器。 |
| text | 定义单行的输入字段，用户可在其中输入文本。默认宽度为 20 个字符。 |

文本域

Name:

<input type=”text” name=”name” value="yourmail" size="20"/> 可以添加值类似于hint

Password:

<input type=”password” name=”password”/>

文本域默认缺省宽度 20 字符

单选按钮

input radio 相同的name 不同的 value

<input type="radio" name="sex" value="male" /> Male

<input type="radio" name="sex" value="female" checked="checked"/> Female

下拉框

**select name option value**

<select name="cars">

<option value="volvo" **selected**=”selected”>Volvo</option>

<option value="saab">Saab</option>

<option value="fiat">Fiat</option>

<option value="audi">Audi</option>

</select>

**Optgroup**

<select>

<**optgroup** label="Swedish Cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

</optgroup>

<optgroup label="German Cars">

<option value="mercedes">Mercedes</option>

<option value="audi">Audi</option>

</optgroup>

</select>

复选框

input **checkbox** **不同的name**

<input type="checkbox" name="bike" />

I have a bike

<br />

<input type="checkbox" name="car" />

I have a car

<lebel>

**在单选和复选框中必须定义id ，然后在 lable中for属性指向该id ，并且必须定义value**

<**input type="radio" name="sex" id="famale" value="famale"**/><**label for="famale"**>女</**label**>

<**input type="radio" name="sex" ID="male" value="male"** /><**label for="male"**>男</**label**>

任何表单都有label

提交

**input submit value**

<input type="submit" value="Submit" />

文本域

Textarea rows cols

**<textarea rows="10" cols="30" name=“”” resize=”none”></textarea>里面的文字是默认的文字**

**Resize=none 防止用户拖动**

按钮

<**input type="button" value="什么东西"**>

表单周围的框

**<fieldset>**

本例演示如何在数据周围绘制一个带标题的框。

**<fieldset>**

**<legend>**健康信息</legend>

身高：<input type="text" />

体重：<input type="text" />

</fieldset>

**电子邮件**

<form action="MAILTO:someone@w3school.com.cn" method="post" **enctype**="text/plain">

上传文件

**file name id accept （上传文件的属性）**

<input type="file" name="pic" id="pic" accept="image/gif, image/jpeg" />

如果不限制图像的格式，可以写为：accept="image/\*"。 记得用逗号隔开文件内容

autocomplete

自动完成（补全，记录历史）功能

自动补全 on 在第二次输入的时候会有下拉选项的

form 和 <input> 类型：text, search, url, telephone, email, password, datepickers, range 以及 color。 属性中可以使用这个

<**form action="/example/html5/demo\_form.asp" method="get" autocomplete="on"**>  
 First name:<**input type="text" name="fname"** /><**br** />  
 Last name: <**input type="text" name="lname"** /><**br** />  
 E-mail: <**input type="email" name="email" autocomplete="off"** /><**br** />  
 <**input type="submit"** />  
</**form**>

autofocus

自动获取焦点

文本输入字段被设置为当页面加载时获得焦点

<input type="text" name="fname" autofocus="autofocus" />

Disabled属性

被禁用的 input 元素既不可用，也不可点击。可以设置 disabled 属性，直到满足某些其他的条件为止（比如选择了一个复选框等等）。然后，就需要通过 JavaScript 来删除 disabled 值，将 input 元素的值切换为可用。

disabled 属性无法与 <input type="hidden"> 一起使用。

<input type="text" name="lname" disabled="disabled" />

Form属性 form="form1"

指向自己应该属于的表单id，可以多个

<form action="/example/html5/demo\_form.asp" method="get" id="form1">

First name: <input type="text" name="fname" /><br />

<input type="submit" value="提交" />

</form>

<p>下面的 "Last name" 字段位于 form 元素之外，但仍然是表单的一部分。</p>

Last name: <input type="text" name="lname" form="form1" />

Formaction

该属性适用于 type="submit" 以及 type="image"。的 input标签，目的是在标签内部提交到不同的网址， 造成权限什么的

Formenctype

formenctype="multipart/form-data"

formmethod

formmethod 属性覆盖 form 元素的 [method](http://www.w3school.com.cn/tags/att_form_method.asp) 属性。

可以通过以下方式发送 form-data ：

* 以 URL 变量 (使用 method="get") 的形式来发送
* 以 HTTP post (使用 method="post") 的形式来发送

该属性与 type="submit" 以及 type="image" 配合使用。

formnovalidate

formnovalidate 属性覆盖 form 元素的 [novalidate](http://www.w3school.com.cn/tags/att_form_novalidate.asp) 属性。

如果使用该属性，则提交表单时按钮不会执行验证过程。

该属性适用于 <form> 以及以下类型的 <input>：text, search, url, telephone, email, password, date pickers, range 以及 color。

Formtaget

formtarget 属性覆盖 form 元素的 [target](http://www.w3school.com.cn/tags/att_form_target.asp) 属性。

该属性与 type="submit" 以及 type="image" 配合使用。

**注释：**HTML5 不支持框架和框架集。现在，parent, top 和 *framename* 值大多用于 iframe。

Type= imae 可以做submmit使用，

并且默认地，image 输入类型会发生点击图像按钮时的 X 和 Y 坐标，可以定义width和heigth

<form action="/example/html5/demo\_form.asp" method="get">

First name: <input type="text" name="fname" /><br />

Last name: <input type="text" name="lname" /><br />

<input type="image" src="/i/eg\_submit.jpg" alt="Submit" width="128" height="128"/>

</form>

<p><b>注释：</b>默认地，image 输入类型会发生点击图像按钮时的 X 和 Y 坐标。</p>

List datalist

**list=“datalist的id“ datalist标签（和select有点象） 属性lable（位于列表右边显示是什么的书属性）**

网页：

<input type="url" list="url\_list" name="link" />

<**datalist** id="url\_list">

<option **label="W3School"** value="http://www.w3school.com.cn" />

<option label="Google" value="http://www.google.com" />

<option label="Microsoft" value="http://www.microsoft.com" />

</datalist>

Max和min

max 属性规定输入字段所允许的最大值。

**提示：**max 属性与 [min](http://www.w3school.com.cn/tags/att_input_min.asp) 属性配合使用，可创建合法值范围。

**注释：**max 和 min 属性适用于以下 <input> 类型：number, range, date, datetime, datetime-local, month, time 以及 week

可选属性值为 number date

Maxlenth

maxlength 属性规定输入字段的最大长度，以字符个数计。

maxlength 属性与 <input type="text"> 或 <input type="password"> 配合使用。

Panttern

定义和用法

pattern 属性规定用于验证输入字段的模式。

模式指的是[正则表达式](http://www.w3school.com.cn/js/js_obj_regexp.asp)。您可以在我们的 JavaScript 教程中阅读到这方面的内容。

**注释：**pattern 属性适用于以下 <input> 类型：text, search, url, telephone, email 以及 password 。

**提示：**请使用标准的 "title" 属性来描述模式。

[placeholder](http://www.w3school.com.cn/tags/att_input_placeholder.asp)

hint

**readonly**

只读字段是不能修改的。不过，用户仍然可以使用 tab 键切换到该字段，还可以选中或拷贝其文本。

readonly 属性可以防止用户对值进行修改，直到满足某些条件为止（比如选中了一个复选框）。然后，需要使用 JavaScript 消除 readonly 值，将输入字段切换到可编辑状态。

readonly 属性可与 <input type="text"> 或 <input type="password"> 配合使用。

Size

对于 <input type="text"> 和 <input type="password">，size 属性定义的是可见的字符数。而对于其他类型，size 属性定义的是以像素为单位的输入字段宽度。

由于 size 属性是一个可视化的设计属性，我们推荐您使用 CSS 来代替它。

CSS 语法：<input style="width:100px" /

Step

step 属性规定输入字段的合法数字间隔（假如 step="3"，则合法数字应该是 -3、0、3、6，以此类推）。

**提示：**step 属性可以与 max 以及 min 属性配合使用，以创建合法值的范围。

**注释：**step、max 以及 min 属性适用于以下 <input> 类型：number, range, date, datetime, datetime-local, month, time 以及 week。

会出现上下的箭头

**Required**

required="required"

required 属性规定必需在提交之前填写输入字段。没填就不能提交

如果使用该属性，则字段是必填（或必选）的。

**注释：**required 属性适用于以下 <input> 类型：text, search, url, telephone, email, password, date pickers, number, checkbox, radio 以及 file。

|  |  |  |
| --- | --- | --- |
| [name](http://www.w3school.com.cn/tags/att_input_name.asp) | field\_name | 定义 input 元素的名称。 |
| [pattern](http://www.w3school.com.cn/tags/att_input_pattern.asp) | *regexp\_pattern* | 规定输入字段的值的模式或格式。  例如 pattern="[0-9]" 表示输入值必须是 0 与 9 之间的数字。 |
| [placeholder](http://www.w3school.com.cn/tags/att_input_placeholder.asp) | *text* | 规定帮助用户填写输入字段的提示。 Hint |
| [readonly](http://www.w3school.com.cn/tags/att_input_readonly.asp) | readonly | 规定输入字段为只读。 |
| [required](http://www.w3school.com.cn/tags/att_input_required.asp) | required | 指示输入字段的值是必需的。 |
| [size](http://www.w3school.com.cn/tags/att_input_size.asp) | number\_of\_char | 定义输入字段的宽度。 |
| [src](http://www.w3school.com.cn/tags/att_input_src.asp) | URL | 定义以提交按钮形式显示的图像的 URL。 |
| [step](http://www.w3school.com.cn/tags/att_input_step.asp) | *number* | 规定输入字的的合法数字间隔。 |
| [type](http://www.w3school.com.cn/tags/att_input_type.asp) | * button * checkbox * file * hidden * image * password * radio * reset * submit * text | 规定 input 元素的类型。 |
| [value](http://www.w3school.com.cn/tags/att_input_value.asp) | *value* | 规定 input 元素的值。 在text中是直接填写的的，不是hint |

Html框架

不能将 <body></body> 标签与 <frameset></frameset> 标签同时使用！不过，假如你添加包含一段文本的 <noframes> 标签，就必须将这段文字嵌套于 <body></body> 标签内。（在下面的第一个实例中，可以查看它是如何实现的。

**Frameset 框架结合标签**

里面有rows分成几行 cols 分成几列

cols="25%,50%,25%" cols="120,\* “ 定义宽度的方法

**frame 框架标签**

src=”“ 指出里面内联的网址 name 属性定义一个name属性， 在其他的跳转 target选项

noresize="noresize" 不允许手动调节大小

**frameset中的noframe标签 浏览器无法显示frame的时候显示，必须加上<body>标签**

<frameset cols="25%,50%,25%">

<frame src="/example/html/frame\_a.html">

<frame src="/example/html/frame\_b.html">

<frame src="/example/html/frame\_c.html">

<noframes>

<body>您的浏览器无法处理框架！</body>

</noframes>

</frameset>

**分成三列得到框架**

<frameset cols="25%,50%,25%">

<frame src="/example/html/frame\_a.html">

<frame src="/example/html/frame\_b.html">

<frame src="/example/html/frame\_c.html">

</frameset>

**混合框架**

<frameset rows="50%,50%">

<frame src="/example/html/frame\_a.html">

<frameset cols="25%,75%">

<frame src="/example/html/frame\_b.html">

<frame src="/example/html/frame\_c.html">

</frameset>

</frameset>

html内联框架

**iframe 标签 在任意位置开辟单独的空间来显示网页**

<iframe src="demo\_iframe.htm" width="200" height="200"></iframe>

**height 和 width** 属性用于规定 iframe 的高度和宽度。

属性值的默认单位是像素，但也可以用百分比来设定（比如 "80%"）。

删除边框

frameborder="0"

name属性作为外链显示的target

<iframe src="demo\_iframe.htm" name="iframe\_a"></iframe>

<p><a href="http://www.w3school.com.cn" target="iframe\_a">W3School.com.cn</a></p>

Html背景

背景颜色（Bgcolor）

<body bgcolor="#000000">

背景（Background）

<body background="clouds.gif">

<body background="http://www.w3school.com.cn/clouds.gif">

**Bacground 会导致尺寸不够的时候红蝶显示**

**提示：**如果你打算使用背景图片，你需要紧记一下几点：

* 背景图像是否增加了页面的加载时间。小贴士：图像文件不应超过 10k。
* 背景图像是否与页面中的其他图象搭配良好。
* 背景图像是否与页面中的文字颜色搭配良好。
* 图像在页面中平铺后，看上去还可以吗？
* 对文字的注意力被背景图像喧宾夺主了吗

<script> 元素

<script> 标签用于定义客户端脚本，比如 JavaScript。

html脚本

**script**

<script type="text/javascript">

document.write("Hello World!")

</script>

<script> 标签用于定义客户端脚本，比如 JavaScript。

script 元素既可以包**含脚本语句**，也可以通过 **src 属性指向外部脚本文件。**

必需的 **type 属性规定脚本的 MIME 类型。**

JavaScript 的常见应用时**图像操作、表单验证以及动态内容更新。**

假如此元素内部的**代码没有位于某个函数中，那么这些代码会在页面被加载时被立即执行。<frameset> 标签之后的脚本会被忽略。**

**HTML 4 和 XHTML 在处理脚本中的内容方面有所不同：**

在 HTML 4 中，内容类型声明为 CDATA，就是说不会对实体进行解析。

在 XHTML 中，内容类型声明为 (#PCDATA)，也就是说会对实体进行解析。

这意味着，在 XHTML 中，应该编码所有特殊的字符，或者把所有内容嵌套在 CDATA 部分中。

为了确保在 XHTML 文档中脚本正确进行解析，

**noscript**

<noscript>Sorry, your browser does not support JavaScript!</noscript>

**async="async" 外部脚本异步进行**

<script type="text/javascript" **src**="demo\_async.js" **async="async"**></script>

**Charset**

<script type="text/javascript" src="myscripts.js" charset="UTF-8"></script>

* UTF-8 - Unicode 字符编码
* ISO-8859-1 - 拉丁字母表的字符编码

**Defer**  是否加载结束后才加载

Html实体

|  |  |  |  |
| --- | --- | --- | --- |
| **显示结果** | **描述** | **实体名称** | **实体编号** |
|  | 空格 | &nbsp; | &#160; |
| < | 小于号 | &lt; | &#60; |
| > | 大于号 | &gt; | &#62; |
| & | 和号 | &amp; | &#38; |
| " | 引号 | &quot; | &#34; |
| ' | 撇号 | &apos; (IE不支持) | &#39; |
| ￠ | 分 | &cent; | &#162; |
| £ | 镑 | &pound; | &#163; |
| ¥ | 日圆 | &yen; | &#165; |
| € | 欧元 | &euro; | &#8364; |
| § | 小节 | &sect; | &#167; |
| © | 版权 | &copy; | &#169; |
| ® | 注册商标 | &reg; | &#174; |
| ™ | 商标 | &trade; | &#8482; |
| × | 乘号 | &times; | &#215; |
| ÷ | 除号 | &divide; | &#247; |

Html统一资源定位器 URL 网址

http 超文本传输协议 以 http:// 开头的普通网页。不加密。

https 安全超文本传输协议 安全网页。加密所有信息交换。

ftp 文件传输协议 用于将文件下载或上传至网站。

file 您计算机上的文件。

**scheme://host.domain:port/path/filename**

scheme - 定义因特网服务的类型。最常见的类型是 http

host - 定义域主机（http 的默认主机是 www）

domain - 定义因特网域名，比如 w3school.com.cn

:port - 定义主机上的端口号（http 的默认端口号是 80）

path - 定义服务器上的路径（如果省略，则文档必须位于网站的根目录中）。

filename - 定义文档/资源的名

**URL 编码**

URL 只能使用 ASCII 字符集来通过因特网进行发送。

由于 URL 常常会包含 ASCII 集合之外的字符，URL 必须转换为有效的 ASCII 格式。

URL 编码使用 "%" 其后跟随两位的十六进制数来替换非 ASCII 字符。

URL 不能包含空格。URL 编码通常使用 + 来替换空格

**Html Web Serve**

<http://www.w3school.com.cn/hosting/index.asp>

网站主机教程

Html 多媒体

多媒体值得是 音效 音乐 视频 动画

**视频格式**

.avi window常用，非win不好用

.wmv . windows常用，非win不能用

.mpg .mpeg 跨平台支持所有浏览器

.mov ios常用，win要装插件

.ram .rm **低宽带**优选，质量差

.swf .flv flash 插件

**.mp4**

**声音格式**

.mid .midi 小巧的数字方法的，广泛支持

Rm ram 低宽带

.wav 除了谷歌都支持,**最受欢迎，**无压缩的音乐和演讲用

.mp3 .mpga 通那个用，**压缩**录制音乐格式

Html对象

**<object>作用是支持html助手（插件或者辅助应用程序,连入）,**

**Quicktime播放 wave，播放mp4**

<object width="420" height="360"

classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B"

codebase="http://www.apple.com/qtactivex/qtplugin.cab">

<param name="src" **value="bird.wav"** />

<param name="controller" value="true" />

</object>

**Flash播放 .swf**

<object width="400" height="40"

classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"

codebase="http://fpdownload.macromedia.com/

pub/shockwave/cabs/flash/swflash.cab#version=8,0,0,0">

<param name="src" **value="bookmark.swf"**>

<embed **src="bookmark.swf"** width="400" height="40"></embed>

</object>

**Window Media Player 播放WMV 影片**

<object width="100%" height="100%"

type="video/x-ms-asf" url=**"3d.wmv"** data=**"3d.wmv"**

classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6">

<param name="url" value="3d.wmv">

<param name="filename" value="3d.wmv">

<param name="autostart" value="1">

<param name="uiMode" value="full" />

<param name="autosize" value="1">

<param name="playcount" value="1">

<embed type="application/x-mplayer2" src="3d.wmv" width="100%"

height="100%" autostart="true" showcontrols="true"

pluginspage="http://www.microsoft.com/Windows/MediaPlayer/"></embed>

</object>

HRML 音频

**<!DOCTYPE html> 一定开头声明要使用**

使用 <embed> 元素，定义外部（非 HTML）内容的容器。

<**embed** height="100" width="100" **src="song.mp3"** />

<Object > 元素

<object height="100" width="100" **data="song.mp3**"></object>

使用<audio>元素 ,只支持3种列表 MP3 ogg wav

<audio controls="controls">

<source src="song.mp3" type="audio/mp3" />

<source src="song.ogg" type="audio/ogg" />

<source src="song.wave" type="audio/wav" />

</audio>

<audio> 标签的属性

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [autoplay](http://www.w3school.com.cn/tags/att_audio_autoplay.asp) | autoplay | 如果出现该属性，则音频在就绪后马上播放。 |
| [controls](http://www.w3school.com.cn/tags/att_audio_controls.asp) | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| [loop](http://www.w3school.com.cn/tags/att_audio_loop.asp) | loop | 如果出现该属性，则每当音频结束时重新开始播放。 |
| [preload](http://www.w3school.com.cn/tags/att_audio_preload.asp) | preload | 如果出现该属性，则音频在页面加载时进行加载，并预备播放。  如果使用 "autoplay"，则忽略该属性。 |
| [src](http://www.w3school.com.cn/tags/att_audio_src.asp) | *url* | 要播放的音频的 URL。 |

最好的html解决方法

audio embed

MP3 ogg wav

<audio controls="controls" height="100" width="100">

<source src="song.mp3" type="audio/mp3" />

<source src="song.ogg" type="audio/ogg" />

<embed height="100" width="100" src="song.mp3" />

</audio>

2.雅虎媒体播放器

直接链接就能播放但是要保证script标签内的内容存在

<a href="song.mp3">Play Sound</a>

下面是要保证的标签

<script type="text/javascript" src="http://mediaplayer.yahoo.com/js">

</script>

3.使用超链接（播放器会自动用插件播放）

<a href="song.mp3">Play the sound</a>

Html视频

**<!DOCTYPE html>**

**<embed> 标签**

<embed src="movie.swf" height="200" width="200"/>

**<object>标签**

<object width="320" height="240" data="/i/bookmark.swf"></object>

**<video>标签** Vadio支持ogg，mp4,webm 三种格式

<video width="320" height="240" controls="controls" autoplay="autoplay">

<source src="/i/movie.ogg" type="video/ogg" />

<source src="/i/movie.mp4" type="video/mp4" />

<source src="/i/movie.webm" type="video/webm" />

</video>

Vadio支持ogg，mp4,webm 三种格式

Ogg = 带有 Theora 视频编码和 Vorbis 音频编码的 Ogg 文件

MPEG4 = 带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件

WebM = 带有 VP8 视频编码和 Vorbis 音频编码的 WebM 文件

<video> 与 </video> 之间插入的内容非<source>是供不支持 video 元素的浏览器显示的

Vadio的属性

|  |  |  |  |
| --- | --- | --- | --- |
| [autoplay](http://www.w3school.com.cn/tags/att_video_autoplay.asp) | autoplay | 如果出现该属性，则视频在就绪后马上播放。 | |
| [controls](http://www.w3school.com.cn/tags/att_video_controls.asp) | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 | |
| [height](http://www.w3school.com.cn/tags/att_video_height.asp) | *pixels* | 设置视频播放器的高度。 | |
| [loop](http://www.w3school.com.cn/tags/att_video_loop.asp) | loop | 如果出现该属性，则当媒介文件完成播放后再次开始播放。 | |
| [preload](http://www.w3school.com.cn/tags/att_video_preload.asp) | preload | 如果出现该属性，则视频在页面加载时进行加载，并预备播放。  如果使用 "autoplay"，则忽略该属性。 | |
| [src](http://www.w3school.com.cn/tags/att_video_src.asp) | *url* | 要播放的视频的 URL。 | |
| [width](http://www.w3school.com.cn/tags/att_video_width.asp) | *pixels* | 设置视频播放器的宽度。 | |
| **方法** | | **属性** | **事件** |
| play() | | currentSrc | play |
| pause() | | currentTime | pause |
| load() | | videoWidth | progress |
| canPlayType | | videoHeight | error |
|  | | duration | timeupdate |
|  | | ended | ended |
|  | | error | abort |
|  | | paused | empty |
|  | | muted | emptied |
|  | | seeking | waiting |
|  | | volume | loadedmetadata |
|  | | height |  |
|  | | width |  |

**注释：**在所有属性中，只有 videoWidth 和 videoHeight 属性是立即可用的。在视频的元数据已加载后，其他属性才可用。

最好的html解决方法

1.组合框架

<**video** width="320" height="240" controls="controls">

<source src="movie.mp4" type="video/mp4" />

<source src="movie.ogg" type="video/ogg" />

<**source** **src**="movie.webm" type="video/webm" />

<**object** **data**="movie.mp4" width="320" height="240">

<**embed** **src**="movie.swf" width="320" height="240" />

</object>

</video

2.优酷解决方案

<embed src="http://player.youku.com/player.php/sid/XMzI2NTc4NTMy/v.swf"

width="480" height="400"

type="application/x-shockwave-flash">

</embed>

Xhtml

**Xhtml 可扩展超文本标签语言**

语法：

属性名必须小写（所有的东西都是小写的）

属性值必须加引号

属性不能简写

**用id 属性代替那么属性**

**DTD定义了强制使用的html元素**

**HTML 4.01 针对下列元素定义 name 属性：a, applet, frame, iframe, img, 和map。**

**这些东西没有name属性**

**重要的兼容性提示：**

你应该在 "/" 符号前添加一个额外的空格，以使你的 XHTML 与当今的浏览器相兼容。

**强制使用的xhtml 元素**

强制使用的 XHTML 元素

所有 XHTML 文档必须进行文件类型声明（DOCTYPE declaration）。在 XHTML 文档中必须存在html、head、body元素，而 title 元素必须位于在 head 元素中。

下面是一个最小化的 XHTML 文件模板：

<!DOCTYPE Doctype goes here>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Title goes here</title>

</head>

<body>

</body>

</html>

xHtml文档声明类型

**Strict**

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

在此情况下使用：需**要干净的标记，避免表现上的混乱。请与层叠样式表配合使用。**

**Transitional**

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

在此情况下使用：**当需要利用 HTML 在表现上的特性时，并且当需要为那些不支持层叠样式表的浏览器编写 XHTML 时。**

**Frameset**

<!DOCTYPE html

PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

在此的情况下使用：**需要使用HTML框架将浏览器窗口分割为两部分或更多框架时**。

升级到xhtml

1.添加文档类型声明

## CSS

**解决内容与表现分离的问题**

选择器相关

各种样式定义的优先级

1. 浏览器缺省设置
2. 外部样式表
3. 内部样式表（位于 <head> 标签内部）
4. 内联样式（在 HTML 元素内部）

Css基础语法

一些规范

4个空格作为一个缩进层级

空格

1.选择器 与 { 之间必须包含空格

2.属性名 与之后的 : 之间不允许包含空格， : 与 属性值 之间必须包含空格。

3.列表型属性值 书写在单行时，, 后必须跟一个空格

选择器

1.当集体声明多个 selector 时，每个选择器声明必须独占一行。

2. >、+、~ 选择器的两边各保留一个空格

1.选择器的嵌套层级应不大于 3 级，位置靠后的限定条件应尽可能精确。

2. 在可以使用缩写的情况下，尽量使用属性缩写。

3. 使用 border / margin / padding 等缩写时，应注意隐含值对实际数值的影响，确实需要设置多个方向的值时才使用缩写

4.文本内容必须用双引号包围。

5.当数值为 0 - 1 之间的小数时，省略整数部分的 0。

6.长度为 0 时须省略单位。 (也只有长度单位可省)

7.url() 函数中的路径不加引号。

8.RGB颜色值必须使用十六进制记号形式 #rrggbb。不允许使用 rgb()

9.颜色值可以缩写时，必须使用缩写形式。

10. 颜色值不允许使用命名色值。

11. 颜色值中的英文字符采用小写。如不用小写也需要保证同一项目内保持大小写一致。

12. font-weight 属性必须使用数值方式描述。

13. 层级(z-index)必须清晰明确，页面弹窗、气泡为最高级（最高级为999），不同弹窗气泡之间可在三位数之间调整

Css检测地址

http://jigsaw.w3.org/css-validator/

**选择器+声明**

p { color: #f00; }

**值为若干个单词的时候要加引号**

p {font-family: "sans serif";}

**多重声明（每一行结尾都加上;）**

p {

text-align: center;

color: black;

font-family: arial;

}

**空格和大小写**

如果涉及到与 HTML 文档一起工作的话，class 和 id 名称对大小写是敏感的

**选择器的分组**

你可以对选择器进行分组，这样，被分组的选择器就可以分享相同的声明。用**逗号**将需要分组的选择器分开。在下面的例子中，我们对所有的标题元素进行了分组。所有的标题元素都是绿色的。

h1,h2,h3,h4,h5,h6 {

color: green;

}

选择器

后代选择器

<li><strong>我是斜体字。这是因为 strong 元素位于 li 元素内。</strong></li>

li strong {

font-style: italic;

font-weight: normal;

}

具体应用

后代选择器的功能极其强大。有了它，可以使 HTML 中不可能实现的任务成为可能。

假设有一个文档，其中有一个边栏，还有一个主区。边栏的背景为蓝色，主区的背景为白色，这两个区都包含链接列表。不能把所有链接都设置为蓝色，因为这样一来边栏中的蓝色链接都无法看到。

解决方法是使用后代选择器。在这种情况下，可以为包含边栏的 div 指定值为 sidebar 的 class 属性，并把主区的 class 属性值设置为 maincontent。然后编写以下样式：

div.sidebar {background:blue;}

div.maincontent {background:white;}

div.sidebar a:link {color:white;}

div.maincontent a:link {color:blue;}

有关**后代选择器**有一个易被忽视的方面，**即两个元素之间的层次间隔可以是无限的。**

例如，如果写作 ul em，这个语法就会选择从 ul 元素继承的所有 em 元素，而不论 em 的嵌套层次多深。

因此，ul em 将会选择以下标记中的所有 em 元素：

<ul>

<li>List item 1

<ol>

<li>List item 1-1</li>

<li>List item 1-2</li>

<li>List item 1-3

<ol>

<li>List item 1-3-1</li>

<li>List item <em>1-3-2</em></li>

<li>List item 1-3-3</li>

</ol>

</li>

<li>List item 1-4</li>

</ol>

</li>

<li>List item 2</li>

<li>List item 3</li>

</ul>

**>子元素选择器（Child selectors）**

**只能选择作为某元素子元素的元素，直接子元素**

如果您希望选择只作为 h1 元素子元素的 strong 元素，可以这样写：

**h1 > strong** {color:red;}

这个规则会把第一个 h1 下面的两个 strong 元素变为红色，但是第二个 h1 中的 strong 不受影响：

<h1>This is <strong>very</strong> <strong>very</strong> important.</h1>

<h1>This is <em>really <strong>very</strong></em> important.</h1>

+相邻兄弟

如果需要选择紧接在另一个元素后的元素，而且二**者有相同的父元素，**可以使用相邻兄弟选择器（Adjacent sibling selector）。

例如，如果要增加紧接在 h1 元素后出现的段落的上边距，可以这样写：

h1 + p {margin-top:50px;}

这个选择器读作：“选择紧接在 h1 元素后出现的段落，h1 和 p 元素拥有共同的父元素”。

相邻兄弟选择器能作用于后面的标签

Id选择器 #id

**id 选择器即使不被用来创建派生选择器，它也可以独立发挥作用：**

#sidebar {

border: 1px dotted #000;

padding: 10px;

}

**有 id=”sidebar”的标签**

**列表**

#mostImportant {color:red; background:yellow;}

<ul id="mostImportant">

<li>item 1</li>

<li>item 2</li>

<li>item 3</li>

一个选择器，多种用法

**即使被标注为 sidebar 的元素只能在文档中出现一次，这个 id 选择器作为派生选择器也可以被使用很多次：**

#sidebar p {

font-style: italic;

text-align: right;

margin-top: 0.5em;

}

#sidebar h2 {

font-size: 1em;

font-weight: normal;

font-style: italic;

margin: 0;

line-height: 1.5;

text-align: right;

}

Css 类选择器 .class

**注意：**类名的第一个字符不能使用数字！它无法在 Mozilla 或 Firefox 中起作用。

**元素也可以基于它们的类而被选择：**

td.fancy {

color: #f60;

background: #666;

}

在上面的例子中，类名为 fancy 的表格单元将是带有灰色背景的橙色。

<td class="fancy">

你可以将类 fancy 分配给任何一个表格元素任意多的次数。那些以 fancy 标注的单元格都会是带有灰色背景的橙色。那些没有被分配名为 fancy 的类的单元格不会受这条规则的影响。还有一点值得注意，class 为 fancy 的段落也不会是带有灰色背景的橙色，当然，任何其他被标注为 fancy 的元素也不会受这条规则的影响。这都是由于我们书写这条规则的方式，这个效果被限制于被标注为 fancy 的表格单元（即使用 td 元素来选择 fancy 类）。

多类选择器

在 HTML 中，一个 class 值中可能包含一个词列表，**各个词之间用空格分隔**。

<p class="important warning">

This paragraph is a very important warning.

</p>

这**两个词的顺序无关紧要**，写成 warning important 也可以。

我们假设 **class 为 important 的所有元素都是粗体，而 class 为 warning 的所有元素为斜体，class 中同时包含 important 和 warning 的所有元素还有一个银色的背景** 。就可以写作：

.important {font-weight:bold;}

.warning {font-style:italic;}

**.important.warning** {background:silver;}

这里不能有空格。

也就是说：

1） 不要去试图用一个类名，把某个标签的所有样式写完。这个标签要多携带几个类，共同造成这个标签的样式。

2） 每一个类要尽可能小，有“公共”的概念，能够让更多的标签使用。

到底用id还是用class？

答案：尽可能的用class，除非极特殊的情况可以用id。

原因：id是js用的。也就是说，js要通过id属性得到标签，所以我们css层面尽量不用id，要不然js就很别扭。另一层面，我们会认为一个有id的元素，有动态效果。

Css属性选择器 []属性放在中括号中

可以为拥有指定属性的 HTML 元素设置样式，而不仅限于 class 和 id 属性。

注释：只有在规定了 !DOCTYPE 时，IE7 和 IE8 才支持属性选择器。在 IE6 及更低的版本中，不支持属性选择、

[title]

{ //属性选择器

color:red;

}

[title=W3School]

{ //属性和值选择器

border:5px solid blue;

}

[title~=hello] { color:red; }

多个属性的时候i

例如，为了将同时有 href 和 title 属性的 HTML 超链接的文本设置为红色，可以这样写：

a[href][title] {color:red;}

**例子 5：为 XML 文档使用属性选择器**

属性选择器在 XML 文档中相当有用，因为 XML 语言主张要针对元素和属性的用途指定元素名和属性名。

假设我们为描述太阳系行星设计了一个 XML 文档。如果我们想选择有 moons 属性的所有 planet 元素，使之显示为红色，以便能更关注有 moons 的行星，就可以这样写：

planet[moons] {color:red;}

根据具体的属性值

a[href="http://www.w3school.com.cn/about\_us.asp"] {color: red;}

**上面的都是严格的相同才行**

~根据部分属性值选择

如果需要根据属性值中的词列表的某个词进行选择，则需要使用波浪号（~）。

假设您想选择 class 属性中包含 important 的元素，可以用下面这个选择器做到这一点：

p[class~="important"] {color: red;}

子串匹配属性选择器

|  |  |
| --- | --- |
| **类型** | **描述** |
| [abc^="def"] | 选择 abc 属性值以 "def" 开头的所有元素 |
| [abc$="def"] | 选择 abc 属性值以 "def" 结尾的所有元素 |
| [abc\*="def"] | 选择 abc 属性值中包含子串 "def" 的所有元素 |

a[href\*="w3school.com.cn"] {color: red;}

特定属性选择类型

\*[lang|="en"] {color: red;}

**lang**属性值以en开头的

属性选择器参考手册

|  |  |
| --- | --- |
| **选择器** | **描述** |
| [[*attribute*]](http://www.w3school.com.cn/cssref/selector_attribute.asp) | 用于选取带有指定属性的元素。 |
| [[*attribute*=*value*]](http://www.w3school.com.cn/cssref/selector_attribute_value.asp) | 用于选取带有指定属性和值的元素。 |
| [[*attribute*~=*value*]](http://www.w3school.com.cn/cssref/selector_attribute_value_contain.asp) | 用于选取属性值中包含指定词汇的元素。 |
| [[*attribute*|=*value*]](http://www.w3school.com.cn/cssref/selector_attribute_value_start.asp) | 用于选取带有以指定值开头的属性值的元素，该值必须是整个单词。 |
| [[*attribute*^=*value*]](http://www.w3school.com.cn/cssref/selector_attr_begin.asp) | 匹配属性值以指定值开头的每个元素。 |
| [[*attribute*$=*value*]](http://www.w3school.com.cn/cssref/selector_attr_end.asp) | 匹配属性值以指定值结尾的每个元素。 |
| [[*attribute*\*=*value*]](http://www.w3school.com.cn/cssref/selector_attr_contain.asp) | 匹配属性值中包含指定值的每个元素。 |

交集选择器 ie7开始支持

**同一个标签里标签名和类名都都有，两个必须紧挨着，不能有空格，有空格就变成后代选择器了，且多数情况下是不对的 .既是也是 。一般以标签开头类名结束**

**并集选择器**

选择器之间用逗号隔开

序选择器

**first-child 自己是第一个子标签**

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

last-chilid

**lang**

**q元素的lang属性为no的，标签，定义引号 quote(引号)的样式**

q:lang(no)

{

quotes: "~" "~"

}

**p:first-line 找到首行**

{ color:#ff0000;

font-variant:small-caps;

}

**First-letter 首字母的样式**

p:first-letter

{

color:#ff0000;

font-size:xx-large;

}

：before 在该标签前

**在标签前插入图片**

h1:before {

content:url(logo.gif);

}

：after 在该标签后

h1:after

{ content:url(logo.gif); }

选择器总结

说IE6层面兼容的： 标签选择器、id选择器、类选择器、后代、交集选择器、并集选择器、通配符。

|  |
| --- |
| 1. p 2. #box 3. .spec 4. div p 5. div.spec ie7开始兼容 6. div,p 7. \* |

IE7能够兼容的：儿子选择器、下一个兄弟选择器

|  |
| --- |
| 1. div>p 2. h3+p |

IE8能够兼容的：

|  |
| --- |
| 1. ul li:first-child 2. ul li:last-child |

属性选择器

After before

Css样式

继承性

哪些属性能继承？

color、 text-开头的、line-开头的、font-开头的。

这些关于文字样式的，都能够继承； 所有关于盒子的、定位的、布局的属性都不能继承。

层叠性

**层叠性：就是css处理冲突的能力。 所有的权重计算，没有任何兼容问题！**

**CSS像艺术家一样优雅，像工程师一样严谨。**

**总结**

**选中的时候**

**!important; >行内样式>id样式>类样式>标签样式**

不过如果你想无视麻烦的优先级问题，想把一个属性始终应用在元素上。  
可以在样式后面加个 !important;

**都选中的情况下**

**统计权重 id：class：leble 前面的一定大于后面的**

**不进位 一个class>255leble**

**权重相同的谁在后面看谁的**

**没选中的情况下，谁接近听谁的**

**都选中的情况!important 提升该属性的权重**

div{

color:red **!important**;

}

p{

color:blue;

}

由于div是通过继承性来影响文字颜色的，所以!important无法提升它的权重，权重依然是0。

干不过p标签，因为p标签是实实在在选中了，所以字是蓝色的（以p为准）。

背景

背景色

设置为灰色从元素中的文本向外少有延伸，只需增加一些内边距：

p {**background-color**: gray; padding: 20px;}

background-color 不能继承，其默认值是 transparent2。transparent 有“透明”之意，透明的时候会显示父元素的背景

颜色的表示方法

1. 单词
2. Rgb（x,x,x） 红绿蓝发光合成的颜色0-255个颜色

绿：background-color: rgb(0,255,0); 黑：background-color: rgb(0,0,0)

白: background-color: rgb(255,255,255);

1. 16进制 把rgb的十进制转换成了16进制
2. ff

**十六进制可以简化为3位，所有#aabbcc的形式，能够简化为#abc;**

常见的颜色代码：

#000黑 #fff白 #f00 红 #333 灰 #222深灰 #ccc浅灰

背景图像

Background—image

如果需要设置一个背景图像，**必须为这个属性设置一个 URL 值：**

body {background-image: url(/i/eg\_bg\_04.gif);}

**背景重复**

如果需要在页面上对背景图像进行平铺，可以使用 **background-repeat** 属性。

属性值 repeat 导致图像在水平垂直方向上都平铺，就像以往背景图像的通常做法一样。**repeat-x 和 repeat-y 分别导致图像只在水平或垂直方向上重复**，**no-repeat** **则不允许图像在任何方向上平铺**。默认地，背景图像将从一个元素的左上角开始。请看下面的例子：

body

{

**background-image: url(/i/eg\_bg\_03.gif);**

**background-repeat: repeat-y;**

}



背景定位

**background-position**:center

先，可以使用一些**关键字：top、bottom、left、right 和 center**。通常，这些关键字会**成对出现，另一个默认为ceter**，不过也不总是这样。

**暗黑破坏神的背景图**

**通栏banner**

背景溢出就直接溢出了没什么滚动条

百分数

**换句话说，百分数值同时应用于元素和图像。**也就是说，图像中描述为 50% 50% 的点（中心点）与元素中描述为 50% 50% 的点（中心点）对齐。

如果图像位于 0% 0%，其左上角将放在元素内边距区的左上角。如果图像位置是 100% 100%，会使图像的右下角放在右边距的右下角。

如果你想把一个图像放在水平方向 2/3、垂直方向 1/3 处，可以这样

background-position:66% 33%;

如果只提供一个百分数值，所提供的这个值将用作水平值，垂直值将假设为 50%。这一点与关键字类似。

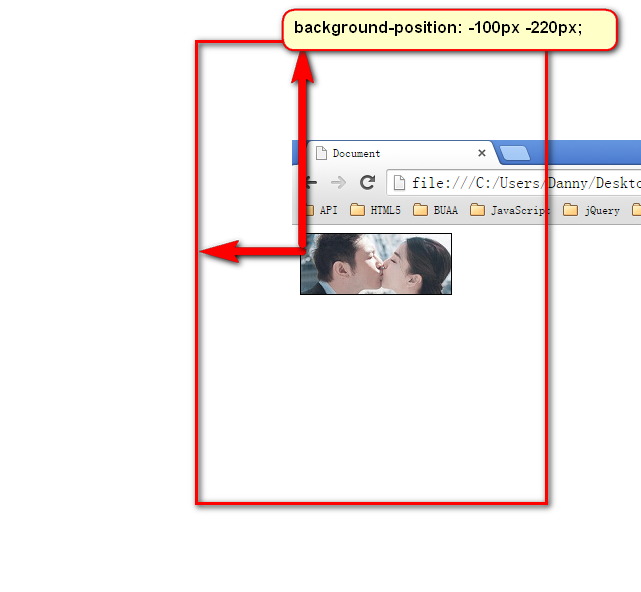
background-position 的默认值是 0% 0%，在功能上相当于 top left。这就解释了背景图像为什么总是从元素内边距区的左上角开始平铺，除非您设置了不同的位置值。

**长度值 精灵用**

长度值解释的是元素内边**距区左上角的偏移。偏移点是图像的左上角。**

比如，如果设置值为 50px 100px，图像的左上角将在元素内边距区左上角向右 50 像素、向下 100 像素的位置上：

原理：

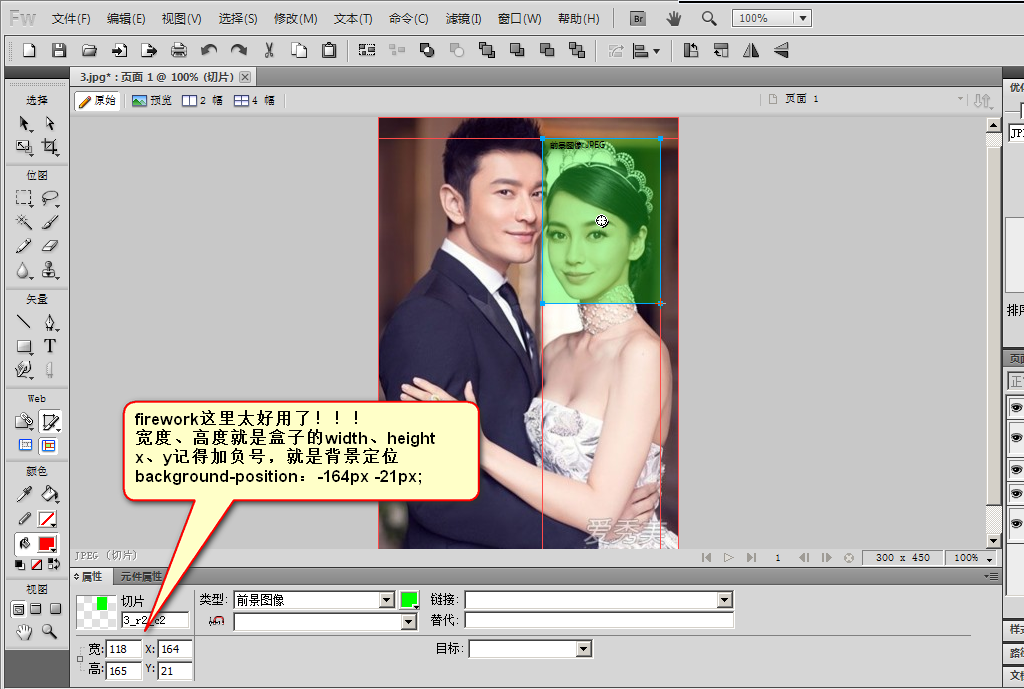


“css精灵”，英语css sprite，所以也叫做“css雪碧”技术。是一种CSS图像合并技术，该方法是将小图标和背景图像合并到一张图片上，然后利用css的背景定位来显示需要显示的图片部分。

css精灵有什么优点，就是减少了http请求。比如4张小图片，原本需要4个http请求。但是用了css精灵，小图片变为了一张图，http请求只有1个了。

淘宝网的精灵图：

 用fireworks精确控制精灵：



**背景关联**

**Background-attachment** 附属物

Fixed 固定在屏幕上，不随滚动滚动 scroll 固定在文档上，随滚动而滚动

**background属性和border一样，是一个综合属性：**

background:red url(1.jpg) no-repeat 100px 100px fixed;

预告一下，CSS3课程中，将学习更多background属性：

background-origin、background-clip、background-size（在CSS2.1背景图片是不能调整尺寸，IE9开始兼容）、

多背景。

**可以任意省略部分**

background: blue url(images/wuyifan.jpg) no-repeat 100px 100px;

Css文本

Text-indent 首行缩进

p {text-indent: 5em;}

**提示：**如果想把一个行内元素的第一行“缩进”，可以用左内边距或外边距创造这种效果。

em 一个汉子的宽度

**使用负值**

如果对一个段落设置了负值，那么首行的某些文本可能会超出浏览器窗口的左边界。为了避免出现这种显示问题，建议针对负缩进再设置一个外边距或一些内边距

p {text-indent: -5em; padding-left: 5em;}

**使用百分比**

百分数要相对于缩进元素父元素的宽度

**继承**

text-indent 属性可以继承，请考虑如下标记：

水平对齐

**Text-align**

eft 把文本排列到左边。默认值：由浏览器决定。

right 把文本排列到右边。

center 把文本排列到中间。

justify 实现两端对齐文本效果。自动设置字与字间隔

inherit 规定应该从父元素继承 text-align 属性的值。

字间隔 word-spacing

如果提供一个正长度值，那么字之间的间隔就会增加。为 word-spacing 设置一个负值，会把它拉近：

字母间隔letter-spacing

默认关键字是 normal（这与 letter-spacing:0 相同）。输入的长度值会使字母之间的间隔增加或减少指定的量：

字符转换

**Text—transform**

默认值 none 对文本不做任何改动，将使用源文档中的原有大小写。顾名思义，uppercase 和 lowercase 将文本转换为全大写和全小写字符。最后，capitalize 只对每个单词的首字母大写。

文本装饰decoration

**underline** 会对元素加下划线，就像 HTML 中的 U 元素一样。

overline 的作用恰好相反，会在文本的顶端画一个上划线。

**line-through** 则在文本中间画一个贯穿线，等价于 HTML 中的 S 和 strike 元素。

blink 会让文本闪烁，类似于 Netscape 支持的颇招非议的 blink 标记。

**none** 值会关闭原本应用到一个元素上的所有装饰，应用：去除超链接的下划线

不过要注意的是，如果两个不同的装饰都与同一元素匹配，胜出规则的值会完全取代另一个值。请考虑以下的规则：

处理空白符 white-space 属性

p {white-space: normal;}

**normal**它会把所有空白符合并为一个空格

**white-space** 属性的值为 pre，浏览器将会注意额外的空格，甚至回车。在这个方面，而且仅在这个方面，任何元素都可以相当于一个 pre 元素。

**注意：**经测试，IE 7 以及更早版本的浏览器不支持该值，因此请使用非 IE 的浏览器来查看上面的实例。

**nowrap**  它会防止元素中的文本换行，除非使用了一个 br 元素。在 CSS 中使用 nowrap 非常类似于 HTML 4 中用 <td nowrap> 将一个表单元格设置为不能换行，不过 white-space 值可以应用到任何元素

**pre-wrap**： 那么该元素中的文本会保留空白符序列，但是文本行会正常地换行

**pre-line**： 会像正常文本中一样合并空白符序列，但保留换行符。

**Pre-wap** 和 pre-line

下面的表格总结了 white-space 属性的行为：

|  |  |  |  |
| --- | --- | --- | --- |
| **值** | **空白符** | **换行符** | **自动换行** |
| pre-line | 合并 | 保留 | 允许 |
| normal | 合并 | 忽略 | 允许 |
| nowrap | 合并 | 忽略 | 不允许 |
| pre | 保留 | 保留 | 不允许 |
| pre-wrap | 保留 | 保留 | 允许 |

文字方向

Direction ltr rtl

Font字体属性

**● 使用font属性，能够将字号、行高、字体，能够一起设置。**

font: 14px/24px “宋体”;

等价于三行语句：

font-size:14px;

line-height:24px;

font-family:"宋体";

**指定字体系列 font-family**

p {font-family: Times, TimesNR, 'New Century Schoolbook',

Georgia, 'New York', serif;} 逗号隔开

**微软雅黑、宋体、黑体。 如果页面中，需要其他的字体，那么需要切图。**

**英语：Arial 、 Times New Roman**

● **我们要将英语字体，放在最前面，这样所有的中文，就不能匹配英语字体，就自动的变为后面的中文字体：后面逗号隔开无数个备选字体**

1 font-family: "Times New Roman","微软雅黑","宋体";

● 所有的中文字体，都有**英语别名，**我们也要知道：

微软雅黑的英语别名：

1 font-family: "**Microsoft YaHei**";

宋体的英语别名：

1 font-family: "**SimSun**";

font属性能够将font-size、line-height、font-family合三为一：

font:12px/30px "Times New Roman","Microsoft YaHei","SimSun";

● 行高可以用**百分比，表示字号的百分之多少**。一般来说，都是大于100%的，因为行高一定要大于字号。

font:12px/200% “宋体” 等价于 font:12px/24px “宋体”;

**使用引号**

只有当字体名中有一个或多个空格（比如 New York），或者如果字体名包括 # 或 $ 之类的符号，才需要在 font-family 声明中加引号，多为单引号

**字体风格 font-style 斜体的使用**

Normal 正常字体

Italic 斜体显示

Oblique 文本倾斜显示

字体变形 font-variant

p {font-variant:small-caps;}

**字体加粗 font-weight**

100-900 数值或者文字

100最细 数字 400 等价于 normal，而 700 等价于 bold，900最粗

Bolder 比继承的字体粗一些，Lighter 比继承的细一些

**字体大小 font-size**

如果您没有规定字体大小，普通文本（比如段落）的默认大小是 16 像素 (16px=1em)。

**使用像素来设置字体大小**  p {font-size:14px;}

**W3C 推荐使用 em 尺寸单位设置大小**

em 的值会相对于父元素的字体大小改变。

**结合使用百分比和 EM**

在所有浏览器中均有效的方案是为 body 元素（父元素）以百分比设置默认的 font-size 值：

body {font-size:100%;}

h1 {font-size:3.75em;}

h2 {font-size:2.5em;}

p {font-size:0.875em;}

Line-height:设置行间距

**Padding 作用在行上，行高和文字字号不同的行高、字号，一般都是偶数，为了居中**

**百分比设置**

默认为110%-120% 115%

小点的设置为90% 大点的设置为200%

**像素设置**

默认约**20px**

小点的设置为1px。大点的设置为

**数值设置**

默认>1

小于1的就会重合

**综合**

**font:italic bold 12px/30px arial,sans-serif;**

**设置文字在垂直方向上的位置：**

**单行居中：行高设置为height**

**多行居中：padding 设置为1/2 的剩余高度**

css链接 中间用： 应用

**改变链接颜色**

a:link {color:#FF0000;} /\* 未被访问的链接 \*/

a:visited {color:#00FF00;} /\* 已被访问的链接 \*/

a:hover {color:#FF00FF;} /\* 鼠标指针移动到链接上 \*/

a:active {color:#0000FF;} /\* 正在被点击的链接 \*/

hover ai’huai

**文本修饰**

a:link {text-decoration:none;}

a:visited {text-decoration:none;}

a:hover {text-decoration:underline;}

a:active {text-decoration:underline;}

**背景色**

background-color 属性规定链接的背景色：

**方框的链接**

display:block;

font-weight:bold;

font-size:14px;

font-family:Verdana, Arial, Helvetica, sans-serif;

color:#FFFFFF;

background-color:#98bf21;

width:120px;

text-align:center;

padding:4px;

text-decoration:none

css列表

**改变（无序）列表的标志**

ul {**list-style-type** : square}

square 方块

disc 实心点

circle 空心圈

none 没有

**有序列表的标志**

ol.decimal {list-style-type: **decimal**} 十进制的数

ol.lroman {list-style-type: lower-roman} 小号罗马字母

ol.uroman {list-style-type: upper-roman} 大写罗马字母

ol.lalpha {list-style-type: lower-alpha} abcde

ol.ualpha {list-style-type: upper-alpha} ABCDE

cjk-ideographic 一二三四五

**将标志改为图像**

ul li {**list-style-image** : url(xxx.gif)}

**列表的放置位置 list-style-position**

|  |  |
| --- | --- |
| inside | 列表项目标记放置在文本以内，且环绕文本根据标记对齐。 |
| outside | 默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。 |
| inherit | 规定应该从父元素继承 list-style-position 属性的值 |

**可以直接写list-style 然后写出各个默认值**

list-style: square inside url('/i/eg\_arrow.gif')

css表格

**表格边框**

table

{ border-collapse:collapse; } //记得加上html5的开头

table,th, td

{ border: 1px solid black; }、

**table、th 以及 td 元素都有独立的边框。需要单独定义**

**宽度和高度**

**Width:100%**

**Height：50px；**

**表格文本对齐**

**Text-align 水平的对齐方式 left right center**

**Vertical-align top bottom center**

**表格内边距**

**Td 和 th 设置padding**

**padding:15px;**

**背景和字体**

**th{**

**background-color:green;**

**color:white;}**

**border-spacing**

规定相邻单元的边框之间的距离。使用 px、cm 等单位。不允许使用负值。

如果定义一个 length 参数，那么定义的是水平和垂直间距。

如果定义两个 length 参数，那么第一个设置水平间距，而第二个设置垂直间距。

border-collapse:separate;

border-spacing:10px 50px;

**empty-cells （仅分离的可用）**

border-collapse:separate;

empty-cells:hide;

hide 隐藏空的td

**caption-side**

caption

{ caption-side:bottom; }

Bottom top inherit（继承父类） 位置

**Table-layout**

Fixed 宽高是固定的

Automatic 宽度随着内容变化

外部轮廓outline

Outline：color style width

**Outline-color**

**Outline-style**

|  |  |
| --- | --- |
| none | 默认。定义无轮廓。 |
| dotted | 定义点状的轮廓。 |
| dashed | 定义虚线轮廓。 |
| solid | 定义实线轮廓。 |
| double | 定义双线轮廓。双线的宽度等同于 outline-width 的值。 |
| groove | 定义 3D 凹槽轮廓。此效果取决于 outline-color 值。 |
| ridge | 定义 3D 凸槽轮廓。此效果取决于 outline-color 值。 |
| inset | 定义 3D 凹边轮廓。此效果取决于 outline-color 值。 |
| outset | 定义 3D 凸边轮廓。此效果取决于 outline-color 值。 |
| inherit | 规定应该从父元素继承轮廓样式的设置。 |

**Outline-width**

thin规定细轮廓。

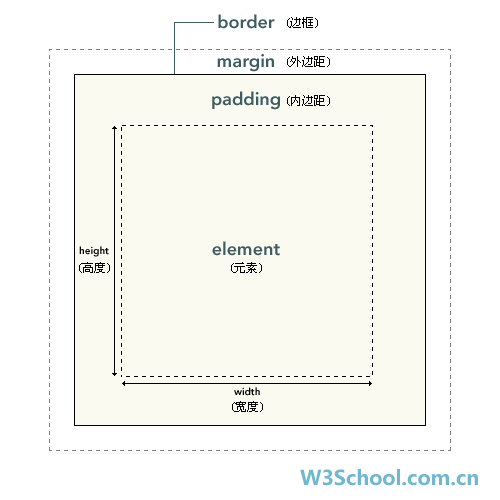
medium默认。规定中等的轮廓。

thick规定粗的轮廓。

*length*允许您规定轮廓粗细的值。

inherit规定应该从父元素继承轮廓宽度的设置。

Css盒子模型



width 和 height 指的是内容区域的宽度和高度。增加内边距、边框和外边距不会影响内容区域的尺寸，但是会增加元素框的总尺寸。 开发的时候要用**专业的清除默认样式的样式报表，去除影响**

**要保持盒子占有空间的大小，width和padding的增减是是相反的**

Padding 属性

Padding区域有background的

**可以直接padding：** 上右下左的顺序来

1个对应4个 2个对应上下、左右 3个 对应上右下左=右

也可以padding-top： Padding-right: Padding-bottom Padding-left:

如果只用小属性写的时候，没有写哪个，哪个方向上padding就是0

**要懂得，用小属性层叠大属性：**

|  |
| --- |
| padding: 20px;  padding-left: 30px; |

这个顺序不能改变,否则前面的小属性会被后面的大属性层叠，覆盖

最终结果 left:30px top right buttom 20px

一些属性默认带有padding;

Padding对盒子大小的硬性

内边距 会影响盒子大小

行内元素 尽量不用 上下的padding和margin

**继承的宽度 padding不会挤开** 。子盒子内不给宽度的时候默认父亲的，但是，加了左右padding不会撑开父亲，总宽度仍然是这些

同时高度必须自己定义不然无法显示宽度

**Border 属性**

width style color

1px solid black

色如果不写，默认是黑色。另外两个属性不写，要命了，显示不出来边框。

如果公司里面的设计师，处女座的，追**求极高的页面还原度，那么不能使用css来制作边框。**

就要用到图片，**就要切图了。所以，比较稳定的就几个：solid、dashed 虚线、dotted 点**

**border-style 也可以分开定义 border-top-style border-right-style border-bottom-s30ty**

|  |  |
| --- | --- |
| n1one | 定义无边框。 |
| hidden | 与 "none" 相同。不过应用于表时除外，对于表，hidden 用于解决边框冲突。 |
| dotted | 定义点状边框。在大多数浏览器中呈现为实线。 |
| dashed | 定义虚线。在大多数浏览器中呈现为实线。 |
| solid | 定义实线。 |
| double | 定义双线。双线的宽度等于 border-width 的值。 |
| groove | 定义 3D 凹槽边框。其效果取决于 border-color 的值。 |
| ridge | 定义 3D 垄状边框。其效果取决于 border-color 的值。有阴影的样子的东西 |
| inset | 定义 3D inset 边框。其效果取决于 border-color 的值。 |
| outset | 定义 3D outset 边框。其效果取决于 border-color 的值。 |
| inherit | 规定应该从父元素继承边框样式。 |

**Border-width: border-top-width**

可以指定长度值，比如 2px 或 0.1em；

或者使用 3 个关键字 thin 、medium（默认值） 和 thick。

**因此，如果您希望边框出现，就必须声明一个边框样式。**

**Border-color: border-top-color**

border是一个大综合属性，

|  |
| --- |
| 1. border:1px solid red; |

就是把4个边框，都设置为1px宽度、线型实线、red颜色。

border属性能够被拆开，有两大种拆开的方式：

1） 按3要素:border-width、border-style、border-color

2） 按方向：border-top、border-right、border-bottom、border-left

**按3要素拆开：**

|  |
| --- |
| 1. **border-width**:10px; → 边框宽度 2. **border-style**:solid; → 线型 3. **border-color**:red; → 颜色。 |

等价于：

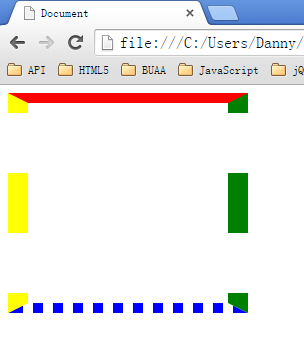
|  |
| --- |
| 1. border:10px solid red; |

现在心里要明白，**原来一个border是由三个小属性综合而成：**

**border-width border-style border-color。**

如果某一个小要素后面是空格隔开的多个值，那么就是**上右下左**的顺序：

|  |
| --- |
| 1. border-width:10px 20px; 2. border-style:**solid dashed dotted**; 3. border-color:**red green blue yellow**; |



**按方向来拆**

|  |
| --- |
| 1. border-top:10px solid red; 2. border-right:10px solid red; 3. border-bottom:10px solid red; 4. border-left:10px solid red; |

等价于

|  |
| --- |
| 1. border:10px solid red; |

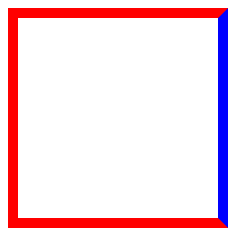
**按方向还能再拆一层，就是把每个方向的，每个要素拆开，一共12条语句：**

|  |
| --- |
| 1. border-top-width:10px; 2. border-top-style:solid; 3. border-top-color:red; 4. border-right-width:10px; 5. border-right-style:solid; 6. border-right-color:red; 7. border-bottom-width:10px; 8. border-bottom-style:solid; 9. border-bottom-color:red; 10. border-left-width:10px; 11. border-left-style:solid; 12. border-left-color:red; |

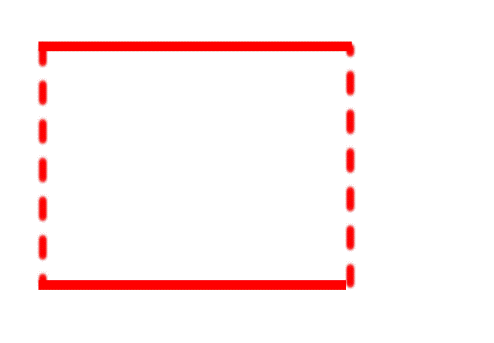
等价于

|  |
| --- |
| 1. border:10px solid red; |

工作中到底用什么？很简答：什么简单用什么？

 写法：

|  |
| --- |
| 1. border:10px solid red; 2. border-right-color:blue; |

写法：

|  |
| --- |
| 1. border:10px solid red; 2. border-style:solid dashed; |

border可以没有，

|  |
| --- |
| 1. border:none; |

某一条边没有：

|  |
| --- |
| 1. border-left: none; |

也可以调整左边边框的宽度为0：

|  |
| --- |
| 1. border-left-width: 0; |

**黑科技：可控三角形**

div{

width: 0px;

height: 0px;

border: 20px solid white;

border-top-color: green;

border-bottom: none;

}

四、标准文档流

宏观的讲，我们的web页面和photoshop等设计软件有本质的区别：web页面的制作，是个“流”，必须从上而下，像“织毛衣”。而设计软件，想往哪里画个东西，都能画。

我们要看看标准流有哪些微观现象：

1. **空白折叠现象：** 两个**img之间的空隙**是因为换行被转换为空格造成的

图片和表单都是纯文本，遵循**多个空格合成一个，并且换行也是空格的设计**

比如，如果我们想让img标签之间没有空隙，必须紧密连接：

|  |
| --- |
| 1. <img src="images/0.jpg" /><img src="images/1.jpg" /><img src="images/2.jpg" /> |

2） **高矮不齐，底边对齐**：



3） **自动换行，一行写不满，换行写。**

4.1 块级元素和行内元素

学习的初期，你就要知道，标准文档流等级森严。标签分为两种等级：

1） 块级元素

● 霸占一行，不能与其他任何元素并列

● 能接受宽、高

● 如果不设置宽度，那么宽度将默认变为父亲的100%。

2） 行内元素

● 与其他行内元素并排

● 不能设置宽、高。默认的宽度，就是文字的宽度。可以设置行高

在HTML中，我们已经将标签分过类，当时分为了：文本级、容器级。

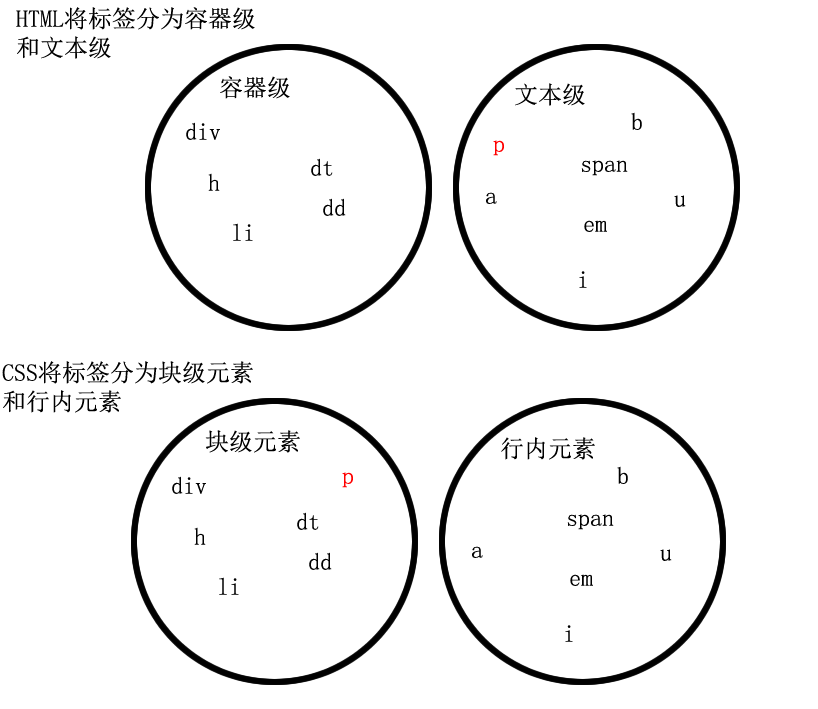
文本级：p、span、a、b、i、u、em、img

容器级：div、h系列、li、dt、dd

CSS的分类和上面的很像，就p不一样：

所有的文本级标签，都是行内元素，除了p，p是个文本级，但是是个块级元素。

所有的容器级标签都是块级元素。



4.2 块级元素和行内元素的相互转换

块级元素可以设置为行内元素

行内元素可以设置为块级元素

|  |
| --- |
| 1. div{ 2. **display: inline;** 3. background-color: pink; 4. width: 500px; 5. height: 500px; 6. } |

display是“显示模式”的意思，用来改变元素的行内、块级性质

inline就是“行内”。

一旦，给一个标签设置

|  |
| --- |
| 1. **display: inline;** |

那么，这个标签将立即变为行内元素。此时它和一个span无异：

● 此时这个div不能设置宽度、高度；

● 此时这个div可以和别人并排了

同样的道理，

|  |
| --- |
| 1. span{ 2. **display: block;** 3. width: 200px; 4. height: 200px; 5. background-color: pink; 6. } |

“block”是“块”的意思

让标签变为块级元素。此时这个标签，和一个div无异：

● 此时这个span能够设置宽度、高度

● 此时这个span必须霸占一行了，别人无法和他并排

● 如果不设置宽度，将撑满父亲

标准流里面限制非常多，标签的性质恶心。比如，我们现在就要并排、并且就要设置宽高。

所以，移民！脱离标准流！

css中一共有三种手段，使一个元素脱离标准文档流：

1） 浮动

2） 绝对定位

3） 固定定位

浮动

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

Float属性

Float：浮动的时候要一起浮动

**脱标、贴边、字围、收缩。**

脱标：

可以设置宽高，脱离了标准文档流，

下面的标准流会成为第一个元素，渲染到了第一个元素位置，部分被float的盖住

不需要displlay ，可以并排并且可以设宽高

贴边

浮动的元素互相贴靠，不会忘往里钻，盒子内部的元素也遵循这个原则，

|  |  |  |
| --- | --- | --- |
|  |  |  |

字围：

可以挡住盒子挡不住盒子里面的字

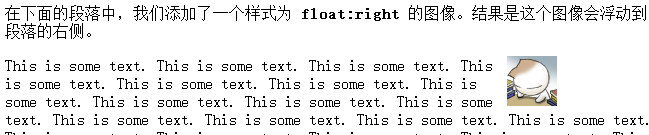
**第一个例子**

img

{ float:right;

margin-right:100px;

}



**第二个例子**

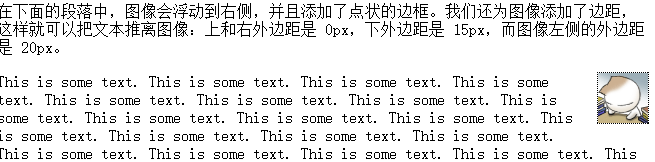
img {

**float:right;**

**border:1px dotted black;**

**margin:0px 0px 15px 20px;**

**}**



**带标题的图像浮于右侧**

div{

float:right;

width:120px;

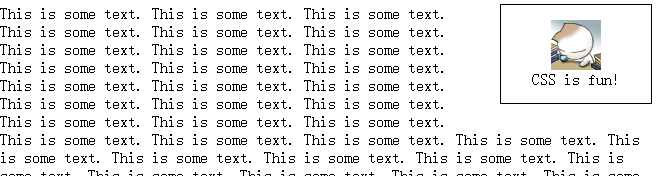
margin:0 0 15px 20px;

padding:15px;

border:1px solid black;

text-align:center;

}



**首字母浮于左侧**

span{

float:left;

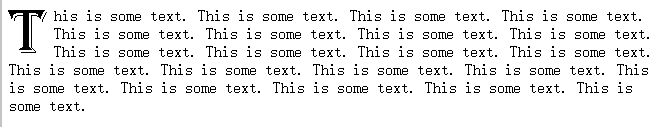
width:0.7em;

font-size:400%;

font-family:algerian,courier;

line-height:80%;

}



**创建水平菜单（导航列表）**

<style type="text/css">

ui{

**float:left;**

width:100%;

padding:0;

margin:0;

**list-style-type:none;**

}

a{

float:left;

width:7em;

text-decoration:none;

color:white;

background-color:purple;

padding:0.2em 0.6em;

border-right:1px solid white;

}

a:hover {background-color:#ff3300}

**li {display:inline} //uxia 段内元素**

</style>

控制浮动的办法让浮动可以分开(给父标签高)

**1.给浮动的元素的祖先元素加高度。**

如果一个元素要浮动，那么它的祖先元素一定要有高度。**高度的盒子，才能关住浮动。**

高度height很少出现。因为能被内容撑高！

**2.Clear**

浮动元素哪侧没有其他的元素

|  |  |  |
| --- | --- | --- |
| **值** | **描述** | |
| left | 在左侧不允许浮动元素。 | |
| right | 在右侧不允许浮动元素。 | |
| both | 在左右两侧均不允许浮动元素。 | |
| none | 默认值。允许浮动元素出现在两侧。 | |
| inherit | 规定应该从父元素继承 clear 属性的值。 | |
| **clear:both;** | | |

**clear就是清除，both指的是左浮动、右浮动都要清除。意思就是：清除别人对我的影响。写在后面的上**

**这种方法有一个非常大的、致命的问题，margin失效了。**

**3.隔墙法 clear:both;**

**外墙法：在两个div间新建一个新空的div，携带两个属性，但是没有高度**

**clear:both;**

**height：10px； margin依然不可用，但是自己高度当组margin**

如果我们现在想让第一个div，自动的根据自己的儿子，撑出高度，我们就要想一些“小伎俩”，“奇淫技

.

**4.内枪法： 在div内部新建一个空div 外部的div被内部的儿子撑出高了**

**5.overflow:hidden; 在父标签加上属性**

一个父亲不能被自己浮动的儿子，撑出高度。但是，只要给父亲加上overflow:hidden; 那么，父亲就能被儿子撑出高了。这是一个偏方。

6：伪元素 盒子的开头和结尾加上：：before ：：after

.clearfix:after {

content:””;

Visibility:hidden;

Display:block;

Height:0;

Clear:both;

}

.clearfix{

Zoom:1;

}

清除浮动： 真正的叫法 闭合浮动

4. 双伪元素

.clearfix:before,.clearfix:after{  
 display: table;  
 content: "";  
}  
.clearfix:after {  
 clear: both;  
}  
.clearfix {  
 zoom: 1;  
}

3.6 浏览器兼容问题

**上述知识点遇见的浏览器兼容问题**

第一，IE6，不支持小于12px的盒子，任何小于12px的盒子，在IE6中看都大

解决办法很简单，就是将盒子的字号，设置小（小于盒子的高），比如0px。 装饰用

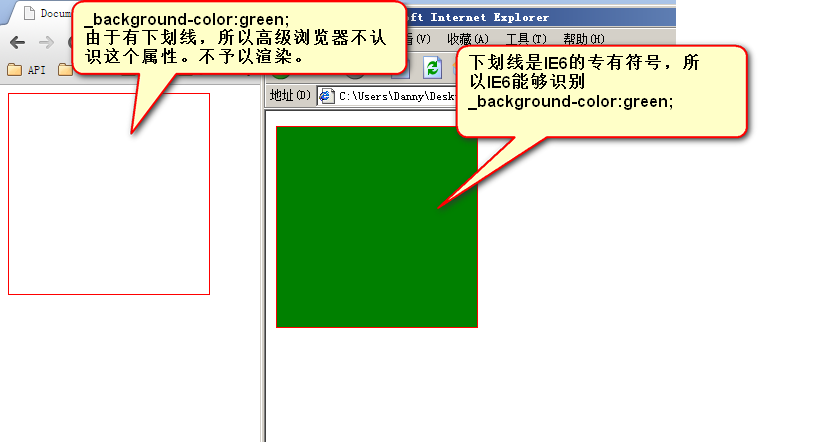
|  |
| --- |
| 1. height: 4px; 2. \_font-size: 0px; |

我们现在介绍一下浏览器hack。hack就是“黑客”，就是使用浏览器提供的后门，针对某一种浏览器做兼容。

**IE6留了一个后门，就是只要给css属性之前，加上下划线，这个属性就是IE6认识的专有属性。**

比如：

|  |
| --- |
| 1. \_background-color: green; |



解决微型盒子，正确写法：

|  |
| --- |
| 1. height: 10px; 2. **\_font-size:0;** |

第二，IE6不支持用overflow:hidden;来清除浮动的

解决办法，以毒攻毒。追加一条

|  |
| --- |
| 1. \_zoom:1; |

完整写法：

|  |
| --- |
| 1. overflow: hidden; 2. **\_zoom:1;** |

实际上，\_zoom:1;能够触发浏览器hasLayout机制。这个机制，不要深究了，因为就IE6有。我们只需要让IE6好用，具体的实现机制，有兴趣的同学，自行百度。

强调一点， overflow:hidden;的本意，就是溢出盒子的border的东西隐藏，这个功能是IE6兼容的。不兼容的是overflow:hidden;清除浮动的时候。

我们刚才学习了两个IE6的兼容问题，这两个IE6的兼容问题，都是通过多写一条hack来解决的。

这个我们称为伴生属性。

|  |
| --- |
| 1. height:6px; 2. **\_font-size:0;** |

|  |
| --- |
| 1. overflow:hidden; 2. **\_zoom:1;** |

3. 伪元素

.clearfix:after {

content:””;

Visibility:hidden;

Display:block;

Height:0;

Clear:both;

}

.clearfix{

Zoom:1;

}

清除浮动： 真正的叫法 闭合浮动

4. 双伪元素

.clearfix:before,.clearfix:after{  
 display: table;  
 content: "";  
}  
.clearfix:after {  
 clear: both;  
}  
.clearfix {  
 zoom: 1;  
}

Css定位 position

**Position**

**Relative 相对定位**

h2{

position:relative;

**left:-20px**  //相对哪个边移动多少 }

**不脱标，形影分离，所有的功能都在自己原来的位置**

↘：

|  |
| --- |
| 1. position: relative; 2. top: 10px; 3. left: 40px; |

↙：

|  |
| --- |
| 1. position: relative; 2. **right: 100px; → 往左边移动** 3. top: 100px; |

**作用**

相对定位有坑，所以一般不用于做“压盖”效果。页面中，效果极小。就两个作用：

1） **微调**元素

2） 做绝对定位的参考，子绝父相（讲绝对定位的时候说）

Absolute 绝对定位

绝对定位之后，标签就不区分所谓的行内元素、块级元素了，不需要display:block;就可以设置宽、高了：

{ //绝对坐标，相对于页面的左方和上方

position:absolute;

left:100px;

top:150px

}

**参考点**

绝对定位的参考点，如果用**top**描述，那么定位参考点就是**页面的左上角**，而不是浏览器的左上角：

如果用**bottom描**述，那么就是**浏览器首屏窗口尺寸**，对应的页面的左下角：

一个绝对定位的元素，如果**父辈元素中出现了也定位了的元素**，那么将**以父辈这个元素**，为参考点。

● 要听最近的已经定位的祖先元素的，不一定是父亲，可能是爷爷：

|  |
| --- |
| 1. <div class="box1"> → 相对定位 2. <div class="box2"> → 没有定位 3. <p></p> → 绝对定位，将以box1为参考，因为box2没有定位，box1就是最近的父辈元素 4. </div> 5. </div> |

|  |
| --- |
| 1. <div class="box1"> → 相对定位 2. <div class="box2"> → 相对定位 3. <p></p> → 绝对定位，将以box2为参考，因为box2是自己最近的父辈元素 4. </div> 5. </div> |

● 不一定是相对定位，任何定位，都可以作为参考点

|  |
| --- |
| 1. <div> → 绝对定位 2. <p></p> → 绝对定位，将以div作为参考点。因为父亲定位了。 3. </div> |

子绝父绝、子绝父相、子绝父固，都是可以给儿子定位的。但是，工程上子绝、父绝，没有一个盒子在标准流里面了，所以页面就不稳固，没有任何实战用途。工程上，“子绝父相”有意义，父亲没有脱标，儿子脱标在父亲的范围里面移动。

|  |
| --- |
| 1. <div class=”box1”> → 绝对定位 2. <div class=”box2”> → 相对定位 3. <div class=”box3”> → 没有定位 4. <p></p> → 绝对定位，以box2为参考定位。 5. </div> 6. </div> 7. </div> |

● 绝对定位的儿子，无视参考的那个盒子的padding。

下图中，绿色部分是div的padding，蓝色部分是div的内容区域。那么此时，div相对定位，p绝对定位。

p将无视父亲的padding，在border内侧为参考点，进行定位：

**绝对定位的盒子居中**

非常简单，当做公式记忆下来。就是left:50%; margin-left:负的宽度的一半。

Fixed 固定定位

**脱标的，**就是相对浏览器窗口定位。页面如何滚动，这个盒子显示的位置不变。

**fixed**{ //相对于窗口位置固定

position:fixed;

left:5px;

top:5px;

}

z-index

**数值大的压着数值小的**

**只有定位了的元素才能有z-index ，绝对/相对/固定定位，而浮动的东西不能用**

是否在同一个垂直维度上，决定是否遮挡，如果不在一个维度上就会有前后显示，决定那个在最前面，有透明属性的时候可以当背景用，可以为正负int值

Z-index 仅能在定位元素上奏效（例如 position:absolute;）！

**● 从父现象：父亲怂了，儿子再牛逼也没用。**

**滚动显示**

Background-color:#00FFFF;

width:150px;

height:150px;

**overflow 溢出**

元素中的内容超出了给定的宽度和高度属性，overflow 属性可以确定是否显示滚动条等行scroll 滚动条显示内容

visible 可见（默认的，会超出和尺寸显示）

hidden（隐藏多余的部分）

auto 系统自己判定，没有超出的时候就是visible

**clip 剪裁**

属性剪裁**绝对定位元素。**唯一合法的形状值是：rect (*top*, *right*, *bottom*, *left*)，相对于左和上的方向画出显示窗口，**rect 矫正**

img

{ position:absolute;

clip:rect(10px 500px 200px 10px) }

**vertical-align**

图像和文字的上部还是下部对齐

|  |  |
| --- | --- |
| baseline | 默认。元素放置在父元素的基线上。 |
| sub | 垂直对齐文本的下标。 下边界和文字的下标对齐 |
| super | 垂直对齐文本的上标 下边界和文字的上标对齐 |
| top | 把元素的顶端与行中最高元素的顶端对齐 上边界对齐 |
| text-top | 把元素的顶端与父元素字体的顶端对齐 上边界对齐 |
| middle | 把此元素放置在父元素的中部。 中间对齐 |
| bottom | 把元素的顶端与行中最低的元素的顶端对齐。 |
| text-bottom | 把元素的底端与父元素字体的底端对齐。 |
| length | 将下表对齐设置为0；向上移动为正，向下移动为负 |
| % | 使用 "line-height" 属性的百分比值来排列此元素。 和上面的相似相对于文本的宽度上下调 |
| inherit | 规定应该从父元素继承 vertical-align 属性的值。 |

相对定位

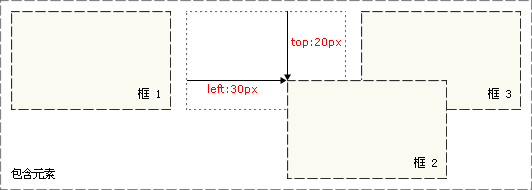
#box\_relative {

position: relative;

left: 30px;

top: 20px;

}



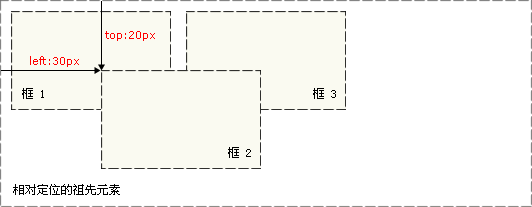
注意，在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其它框。

绝对定位

绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

普通流中其它元素的布局就像绝对定位的元素不存在一样：

如下图所示：



绝对定位的元素的位置相对于**最近的已定位祖先元素**，如果元素没有已定位的祖先元素，那么它的位置相对于**最初的包含块**。

**提示：**因为绝对定位的框与文档流无关，所以它们可以覆盖页面上的其它元素。可以通过设置 [z-index 属性](http://www.w3school.com.cn/cssref/pr_pos_z-index.asp)来控制这些框的堆放次序。

浮动定位

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

Float属性

Float：rigth 浮动在文本外，文本遇到了会自动换行

脱标、贴边、字围、收缩。

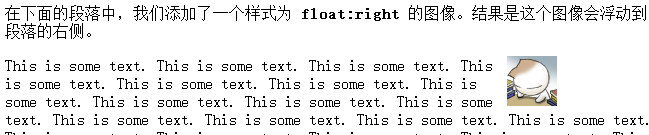
**第一个例子**

img

{ float:right;

margin-right:100px;

}



**第二个例子**

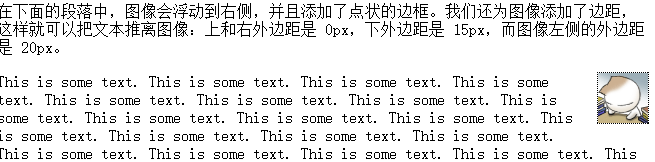
img {

**float:right;**

**border:1px dotted black;**

**margin:0px 0px 15px 20px;**

**}**



**带标题的图像浮于右侧**

div{

float:right;

width:120px;

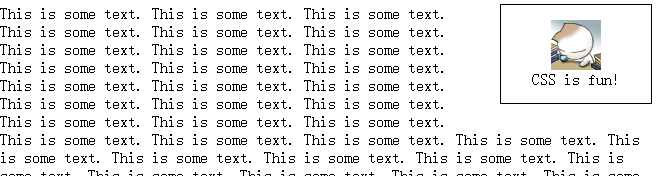
margin:0 0 15px 20px;

padding:15px;

border:1px solid black;

text-align:center;

}



**首字母浮于左侧**

span{

float:left;

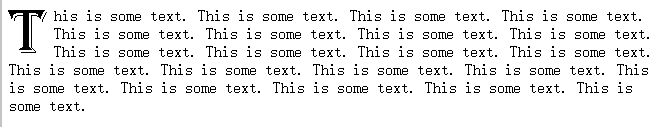
width:0.7em;

font-size:400%;

font-family:algerian,courier;

line-height:80%;

}



**创建水平菜单（导航列表）**

<style type="text/css">

ui{

**float:left;**

width:100%;

padding:0;

margin:0;

**list-style-type:none;**

}

a{

float:left;

width:7em;

text-decoration:none;

color:white;

background-color:purple;

padding:0.2em 0.6em;

border-right:1px solid white;

}

a:hover {background-color:#ff3300}

**li {display:inline} //uxia 段内元素**

</style>

控制浮动的办法让浮动可以分开(给父标签高)

**1.给浮动的元素的祖先元素加高度。**

如果一个元素要浮动，那么它的祖先元素一定要有高度。**高度的盒子，才能关住浮动。**

高度height很少出现。因为能被内容撑高！

**2.Clear**

浮动元素哪侧没有其他的元素

|  |  |  |
| --- | --- | --- |
| **值** | **描述** | |
| left | 在左侧不允许浮动元素。 | |
| right | 在右侧不允许浮动元素。 | |
| both | 在左右两侧均不允许浮动元素。 | |
| none | 默认值。允许浮动元素出现在两侧。 | |
| inherit | 规定应该从父元素继承 clear 属性的值。 | |
| **clear:both;** | | |

**clear就是清除，both指的是左浮动、右浮动都要清除。意思就是：清除别人对我的影响。**

**这种方法有一个非常大的、致命的问题，margin失效了。**

**3.隔墙法**

**外墙法：在两个div间新建一个新空的div，隔开两个**

如果我们现在想让第一个div，自动的根据自己的儿子，撑出高度，我们就要想一些“小伎俩”，“奇淫技巧

**4.内枪法： 在div内部新建一个空div 外部的div被内部的儿子撑出高了**

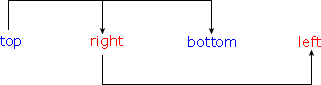
**5.overflow:hidden; 在父标签加上属性**

一个父亲不能被自己浮动的儿子，撑出高度。但是，只要给父亲加上overflow:hidden; 那么，父亲就能被儿子撑出高了。这是一个偏方。

Margin

接受任何长度单位，像素px、 英寸 in、毫米mm、em、百分数（相对于父元素）

**值复制**



换句话说，如果为外边距指定了 3 个值，则第 4 个值（即左外边距）会从第 2 个值（右外边距）复制得到。如果给定了两个值，第 4 个值会从第 2 个值复制得到，第 3 个值（下外边距）会从第 1 个值（上外边距）复制得到。最后一个情况，如果只给定一个值，那么其他 3 个外边距都由这个值（上外边距）复制得到

**提示：**Netscape 和 IE 对 body 标签定义的默认边距（margin）值是 8px。而 Opera 不是这样。相反地，Opera 将内部填充（padding）的默认值定义为 8px，因此如果希望对整个网站的边缘部分进行调整，并将之正确显示于 Opera 中，那么必须对 body 的 padding 进行自定义。

外边距合并（塌陷现象）

**标准文档流中**，**竖直方向的margin不叠加，以较大的为**准，两个相邻的边距会合并留下较大的边距

**浮动之后两个盒子之间是margin叠加，**不塌陷

水平方向没有叠加

**盒子居中**

Margin-left：auto

Margin-right:auto

简写

Margin:0 auto;

1)使用margin:0 auto; 的盒子，**必须有width**，有明确的width

2） 只有**标准流的盒子**，才能使用**margin:0 auto**; 居中。

也就是说，当一个盒子浮动了、绝对定位了、固定定位了，都不能使用margin:0 auto;

3） **margin:0 auto;是在居中盒子，不是居中文本**。 P是块级元素

**文本的居中**

text-align:center;

text-align:right; 文本居右

默认居中的

**要善于使用父亲的padding 而不是儿子的margin**

如果**父亲没有border，那么儿子的margin实际上踹的是“流”**，踹的是这“行”。所以，父亲整体也掉下来了

这个p有一个margin-top踹父亲，试图将自己下移

|  |
| --- |
| 1. <div> 2. <p></p> 3. </div> |

结果：

|  |  |
| --- | --- |
|  |  |

**margin这个属性，本质上描述的是兄弟和兄弟之间的距离； 最好不要用这个marign表达父子之间的距离。**

所以，我们一定要善于使用父亲的padding，而不是儿子的margin。

**IE BUG**

**连续浮动会出现双倍浮动**

解决方案：

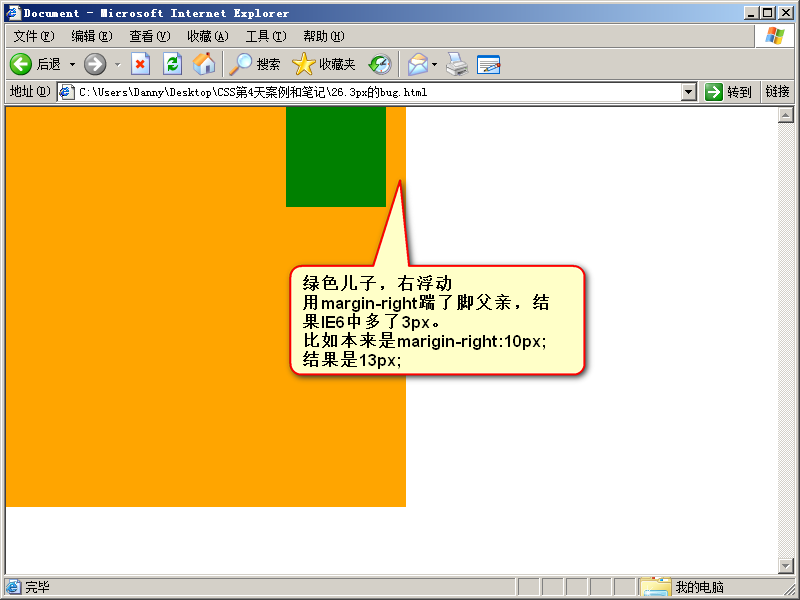
1）使**浮动的方向和margin的方向，相反。**

所以，你就会发现，我们特别喜欢，浮动的方向和margin的方向相反。并且，前端开发工程师，把这个当做习惯了。

|  |
| --- |
| 1. float: left; 2. margin-right: 40px; |

2) 也可以单独手动第一个元素一般的边界

IE6的3px bug



解决办法：

不用管，因为根本就不允许用儿子踹父亲。所以，如果你出现了3px bug，说明你的代码不标准。

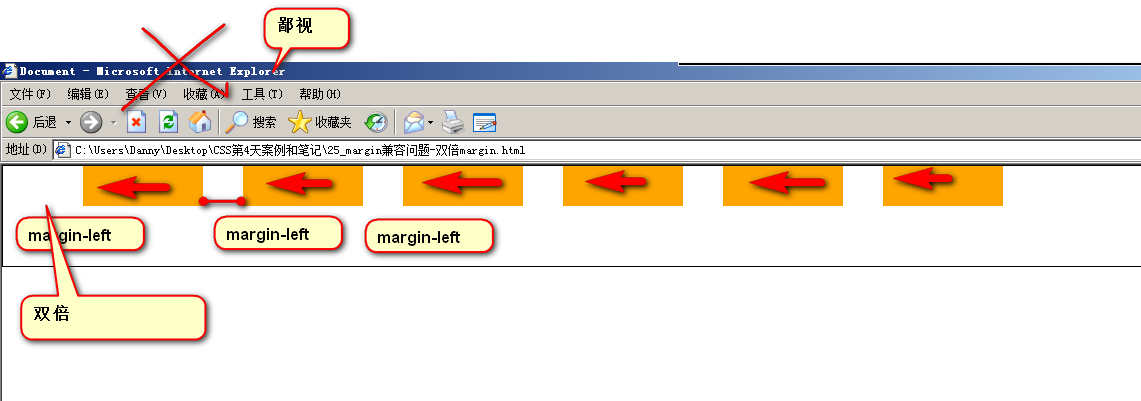
**IE6，千万不要跟他死坑、较劲，它不配。 格调要高，老师给你讲，就是为了增加面试的成功率。不是为了让你成为IE6的专家。**

关于margin的IE6兼容问题

IE6双倍margin bug

**当出现连续浮动的元素，携带和浮动方向相同的margin时，队首的元素，会双倍marign。**

|  |
| --- |
| 1. <ul> 2. <li></li> 3. <li></li> 4. <li></li> 5. </ul> |



解决方案：

1）使浮动的方向和margin的方向，相反。

所以，你就会发现，我们特别喜欢，浮动的方向和margin的方向相反。并且，前端开发工程师，把这个当做习惯了。

|  |
| --- |
| 1. float: left; 2. margin-right: 40px; |

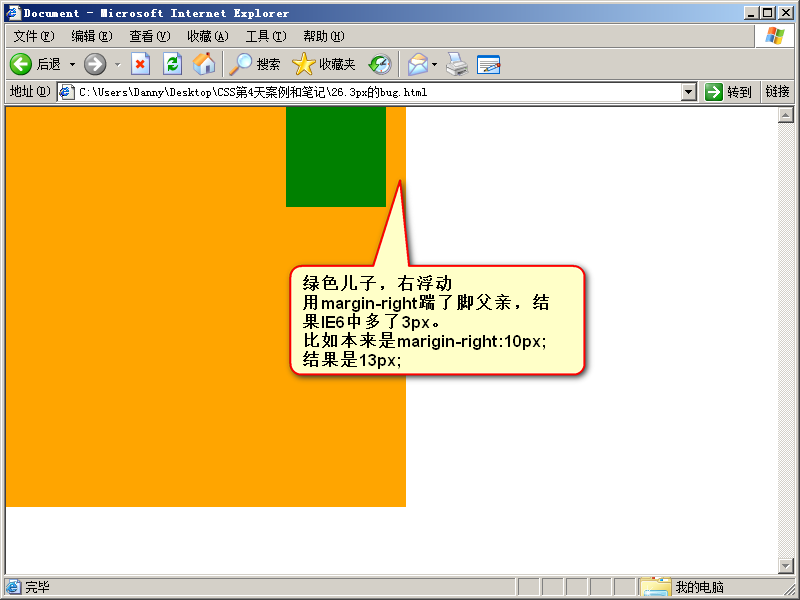
2）使用hack（没必要，别惯着这个IE6）

单独给队首的元素，写一个一半的margin

|  |
| --- |
| 1. <li class="no1"></li> |

|  |
| --- |
| 1. ul li.no1{ 2. \_margin-left:20px; 3. } |

IE6的3px bug



解决办法：

不用管，因为根本就不允许用儿子踹父亲。所以，如果你出现了3px bug，说明你的代码不标准。

**IE6，千万不要跟他死坑、较劲，它不配。 格调要高，老师给你讲，就是为了增加面试的成功率。不是为了让你成为IE6的专家。**

**创建无表格的首页 float创建**

**浮动不会**

css伪类

**同一个标签，根据用户的某种状态不同，有不同的样式。这就叫做“伪类”。**

**用法**

selector : pseudo-class {property: value}

标签名 : 伪类 {name:value}

**超链接**

a:link {color: #FF0000} 链接

a:visited {color: #00FF00} 访问过

a:hover {color: #FF00FF} 悬停，在上上面

a:active {color: #0000FF} 激活

最标准的，就是把link、visited、hover都要写。但是前端开发工程师在大量的实践中，发现不写link、visited浏览器也挺兼容。所以这些“老油条”们，就把a标签简化了：

a:link、a:visited都是可以省略的，简写在a标签里面。也就是说，a标签涵盖了link、visited的状态。

|  |
| --- |
| .nav ul li a{  **display: block;**  width: 120px;  height: 50px;  **text-decoration: none;**  background-color: purple;  color:white;  }  .nav ul li a:hover{  background-color: orange;  } |

**focus使用鼠标焦点选中了**

input:focus {background-color:yellow;}

**first-child 自己是第一个子标签**

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

**lang**

**q元素的lang属性为no的，标签，定义引号 quote(引号)的样式**

q:lang(no)

{

quotes: "~" "~"

}

伪元素

**p:first-line 找到首行**

{ color:#ff0000;

font-variant:small-caps;

}

**First-letter 首字母的样式**

p:first-letter

{

color:#ff0000;

font-size:xx-large;

}

**：before 在该标签前**

**在标签前插入图片**

h1:before {

content:url(logo.gif);

}

**：after 在该标签后**

h1:after

{ content:url(logo.gif); }

对齐

**块级元素的对齐 !DOCTYPE 水平对齐**

**.center{**

Margin-left:auto;

margin-right:auto;

width:70%;

background-color:#b0e0e6;

**}**

**绝对定位**

Display属性

**块级元素和行内元素**

1） 块级元素

● 霸占一行，不能与其他任何元素并列

● 能接受宽、高

● 如果不设置宽度，那么宽度将默认变为父亲的100%。

2） 行内元素

● 与其他行内元素并排

● 不能设置宽、高。默认的宽度，就是文字的宽度。

|  |  |
| --- | --- |
| none | 此元素不会被显示。 |
| block | 此元素将显示为块级元素，此元素前后会带有换行符。链接显示为块元素可使整个链接区域可点击（不仅仅是文本），同时也允许我们规定宽度。 |
| inline | 默认。此元素会被显示为内联元素，元素前后没有换行符。 |
| inline-block | 行内块元素。（CSS2.1 新增的值） |
| list-item | 此元素会作为列表显示。 |
| run-in | 此元素会根据上下文作为块级元素或内联元素显示。 |
| table | 此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。 |
| inline-table | 此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。 |
| table-row-group | 此元素会作为一个或多个行的分组来显示（类似 <tbody>）。 |
| table-header-group | 此元素会作为一个或多个行的分组来显示（类似 <thead>）。 |
| table-footer-group | 此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。 |
| table-row | 此元素会作为一个表格行显示（类似 <tr>）。 |
| table-column-group | 此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。 |
| table-column | 此元素会作为一个单元格列显示（类似 <col>） |
| table-cell | 此元素会作为一个表格单元格显示（类似 <td> 和 <th>） |
| table-caption | 此元素会作为一个表格标题显示（类似 <caption>） |
| inherit | 规定应该从父元素继承 display 属性的值 |

标准流里面限制非常多，标签的性质恶心。比如，我们现在就要并排、并且就要设置宽高移民！脱离标准流！

css中一共有三种手段，使一个元素脱离标准文档流：

1） 浮动

2） 绝对定位

3） 固定定位

Css定位

**Position**

**Relative**

h2{

position:relative;

**left:-20px**  //元素的原始左侧位置减去 20 像素

}

**Absolute** { //绝对坐标，相对于页面的左方和上方

position:absolute;

left:100px;

top:150px

}

**fixed**{ //相对于窗口位置固定

position:fixed;

left:5px;

top:5px;

}

**滚动显示**

background-color:#00FFFF;

width:150px;

height:150px;

**overflow 溢出**

元素中的内容超出了给定的宽度和高度属性，overflow 属性可以确定是否显示滚动条等行scroll 滚动条显示内容

visible 可见（默认的，会超出和尺寸显示）

hidden（隐藏多余的部分）

auto 系统自己判定，没有超出的时候就是visible

**clip 剪裁**

属性剪裁**绝对定位元素。**唯一合法的形状值是：rect (*top*, *right*, *bottom*, *left*)，相对于左和上的方向画出显示窗口，**rect 矫正**

img

{ position:absolute;

clip:rect(10px 500px 200px 10px) }

**vertical-align**

图像和文字的上部还是下部对齐

|  |  |
| --- | --- |
| baseline | 默认。元素放置在父元素的基线上。 |
| sub | 垂直对齐文本的下标。 下边界和文字的下标对齐 |
| super | 垂直对齐文本的上标 下边界和文字的上标对齐 |
| top | 把元素的顶端与行中最高元素的顶端对齐 上边界对齐 |
| text-top | 把元素的顶端与父元素字体的顶端对齐 上边界对齐 |
| middle | 把此元素放置在父元素的中部。 中间对齐 |
| bottom | 把元素的顶端与行中最低的元素的顶端对齐。 |
| text-bottom | 把元素的底端与父元素字体的底端对齐。 |
| length | 将下表对齐设置为0；向上移动为正，向下移动为负 |
| % | 使用 "line-height" 属性的百分比值来排列此元素。 和上面的相似相对于文本的宽度上下调 |
| inherit | 规定应该从父元素继承 vertical-align 属性的值。 |

**z-index**

是否在同一个垂直维度上，决定是否遮挡，如果不在一个维度上就会有前后显示，决定那个在最前面，有透明属性的时候可以当背景用，可以为正负int值

Z-index 仅能在定位元素上奏效（例如 position:absolute;）！

相对定位

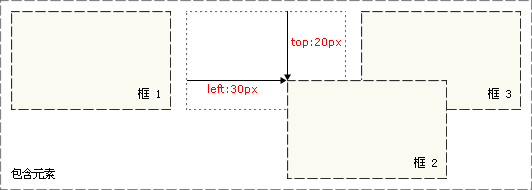
#box\_relative {

position: relative;

left: 30px;

top: 20px;

}



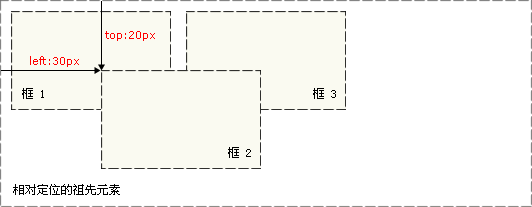
注意，在使用相对定位时，无论是否进行移动，元素仍然占据原来的空间。因此，移动元素会导致它覆盖其它框。

绝对定位

绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

普通流中其它元素的布局就像绝对定位的元素不存在一样：

如下图所示：

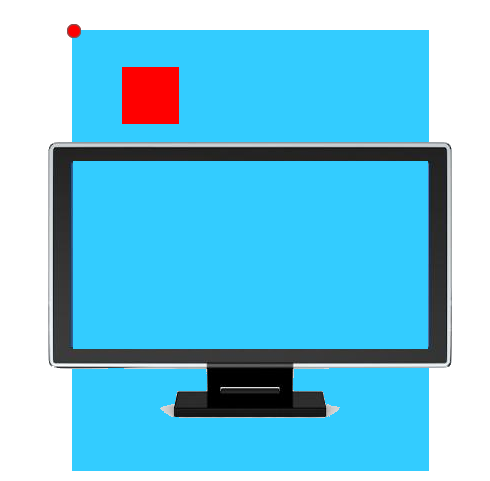


绝对定位的元素的位置相对于**最近的已定位祖先元素**，如果元素没有已定位的祖先元素，那么它的位置相对于**最初的包含块**。

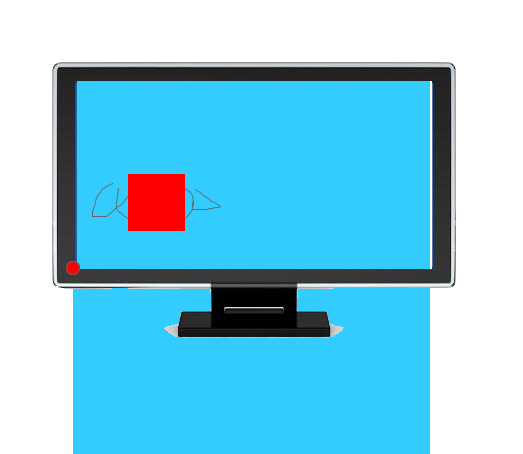
**提示：**因为绝对定位的框与文档流无关，所以它们可以覆盖页面上的其它元素。可以通过设置 [z-index 属性](http://www.w3school.com.cn/cssref/pr_pos_z-index.asp)来控制这些框的堆放次序。

参考点

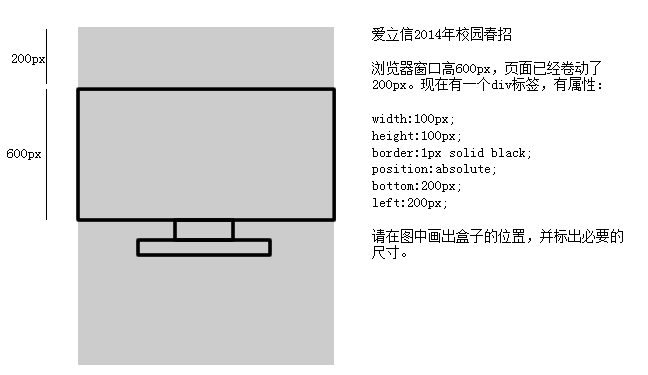
**绝对定位的参考点，如果用top描述，那么定位参考点就是页面的左上角，而不是浏览器的左上角：**



**如果用bottom描述，那么就是浏览器首屏窗口尺寸，对应的页面的左下角**：

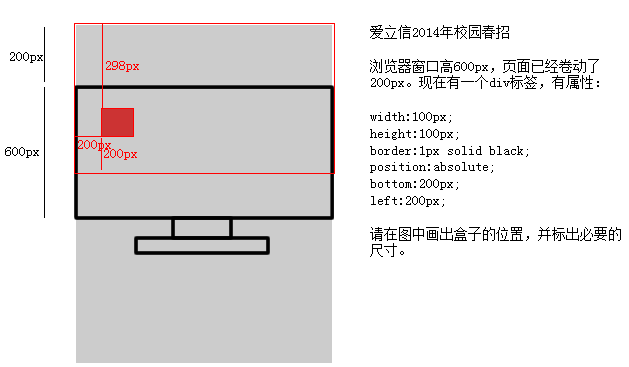


面试题：



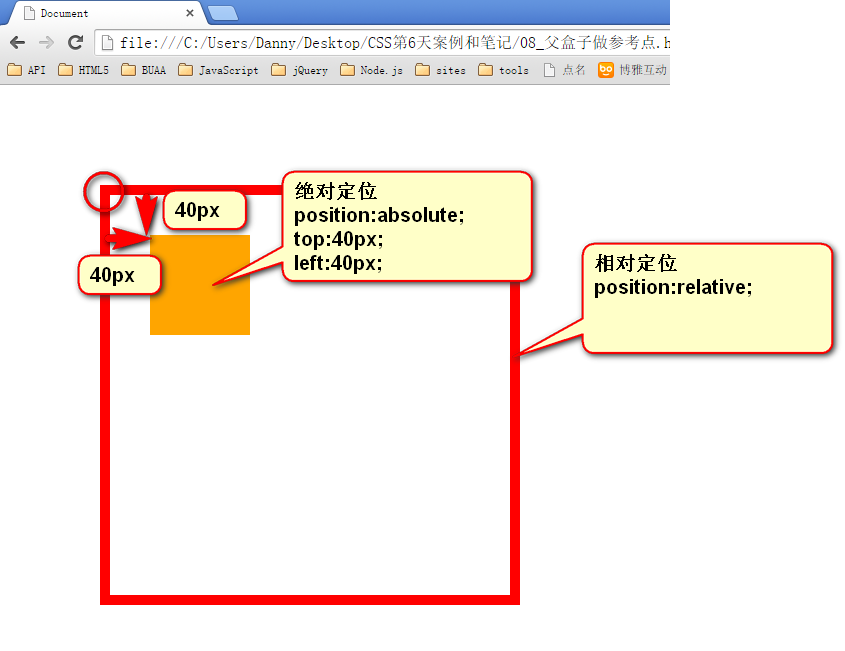
答案：

用bottom的定位的时候，参考的是浏览器首屏大小对应的页面左下角。



3.3 以盒子为参考点

一个绝对定位的元素，如果父辈元素中出现了也定位了的元素，那么将以父辈这个元素，为参考点。



● 要听最近的已经定位的祖先元素的，不一定是父亲，可能是爷爷：

|  |
| --- |
| 1. <div class="box1"> → 相对定位 2. <div class="box2"> → 没有定位 3. <p></p> → 绝对定位，将以box1为参考，因为box2没有定位，box1就是最近的父辈元素 4. </div> 5. </div> |

|  |
| --- |
| 1. <div class="box1"> → 相对定位 2. <div class="box2"> → 相对定位 3. <p></p> → 绝对定位，将以box2为参考，因为box2是自己最近的父辈元素 4. </div> 5. </div> |

● 不一定是相对定位，任何定位，都可以作为参考点

|  |
| --- |
| 1. <div> → 绝对定位 2. <p></p> → 绝对定位，将以div作为参考点。因为父亲定位了。 3. </div> |

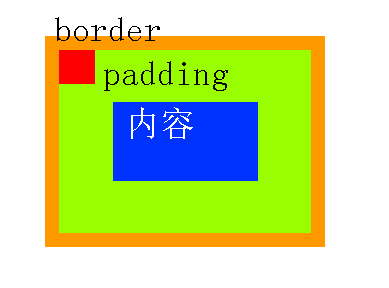
**子绝父绝、子绝父相、子绝父固**，都是可以给儿子定位的。但是，工程上子绝、父绝，没有一个盒子在标准流里面了，所以页面就不稳固，没有任何实战用途。工程上，“子绝父相”有意义，父亲没有脱标，儿子脱标在父亲的范围里面移动。

|  |
| --- |
| 1. <div class=”box1”> → 绝对定位 2. <div class=”box2”> → 相对定位 3. <div class=”box3”> → 没有定位 4. <p></p> → 绝对定位，以box2为参考定位。 5. </div> 6. </div> 7. </div> |

● 绝对定位的儿子，无视参考的那个盒子的padding。

下图中，绿色部分是div的padding，蓝色部分是div的内容区域。那么此时，div相对定位，p绝对定位。

p将无视父**亲的padding，在border内侧为参考点，**进行定位：



3.4 绝对定位的盒子居中

绝对定位之后，所有标准流的规则，都不适用了。所以margin:0 auto;失效。

|  |  |
| --- | --- |
|  |  |

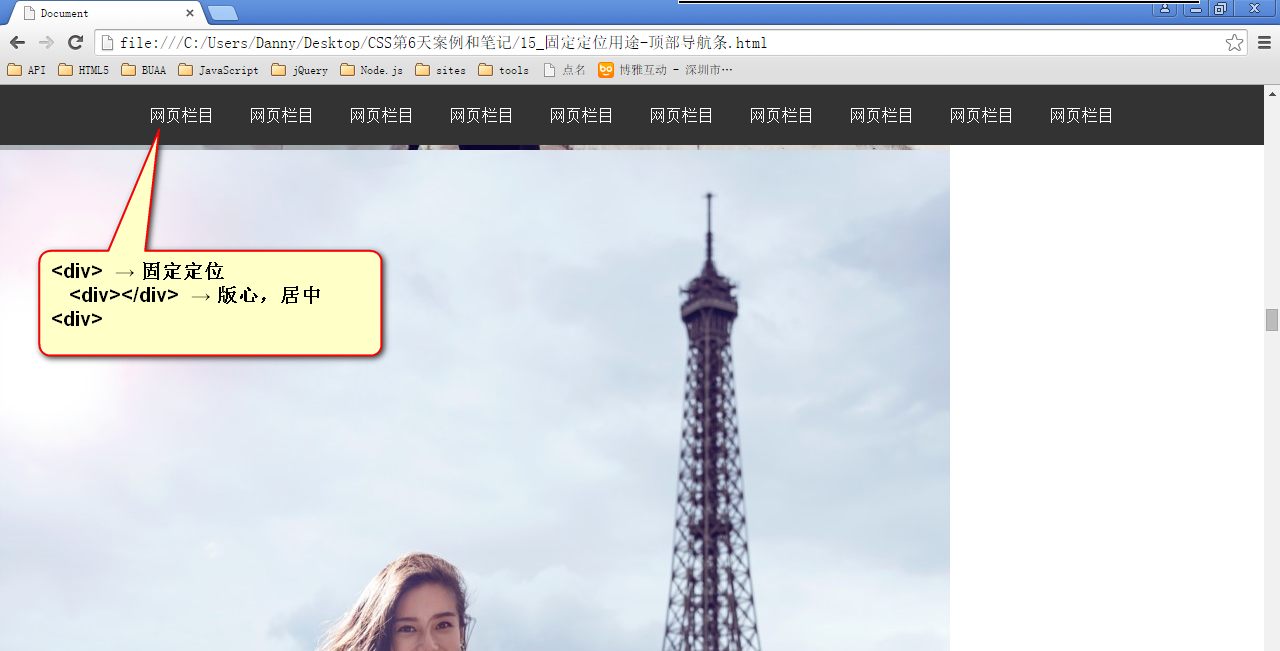
|  |
| --- |
| 1. width: 600px; 2. height: 60px;   **position: absolute;**   1. **left: 50%;** 2. top: 0; 3. **margin-left: -300px; → 宽度的一半** |

非常简单，当做公式记忆下来。**就是left:50%; margin-left:负的宽度的一半。**

固定定位

固定定位，就是相对浏览器窗口定位。页面如何滚动，这个盒子显示的位置不变。

固定定位脱标！



**IE6不兼容。**

z-index

● z-index值表示谁压着谁。数值大的压盖住数值小的。

● 只有定位了的元素，才能有z-index值。也就是说，不管相对定位、绝对定位、固定定位，都可以使用z-index值。而浮动的东西不能用。

● z-index值没有单位，就是一个正整数。默认的z-index值是0。

● 如果大家都没有z-index值，或者z-index值一样，那么谁写在HTML后面，谁在上面能压住别人。定位了的元素，永远能够压住没有定位的元素。

**● 从父现象：父亲怂了，儿子再牛逼也没用。**

**没有单位：**

|  |
| --- |
| 1. z-index: 988; |

**在做通栏的时候回出现好多错误**

浮动定位

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

Float属性

Float：rigth 浮动在文本外，文本遇到了会自动换行

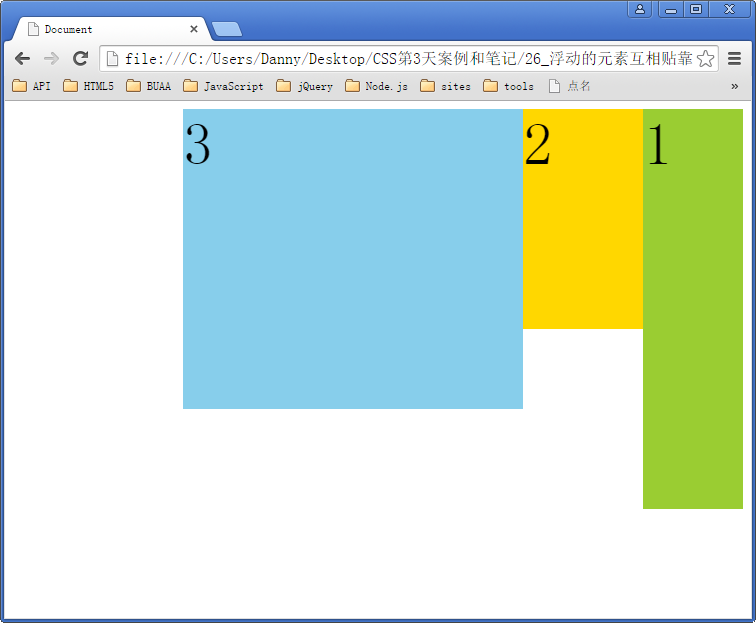
1. **旦一个元素浮动了，那么，将能够并排了，并且能够设置宽高了。无论它原来是个div还是个span。**
2. **5.2 浮动的元素互相贴靠**

**如果有足够空间，那么就会靠着2哥。如果没有足够的空间，那么会靠着1号大哥。**

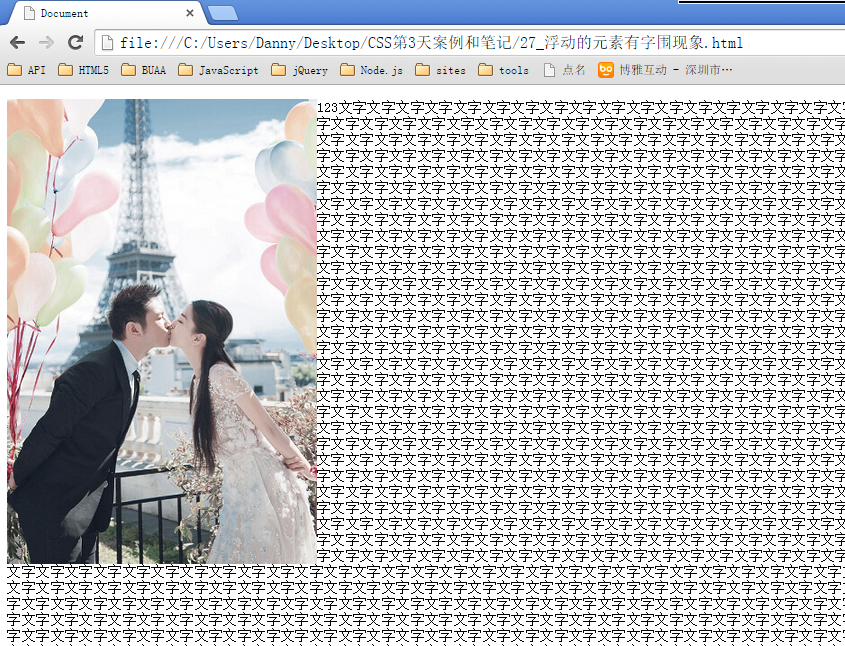
**如果没有足够的空间靠着1号大哥，自己去贴左墙。**

|  |  |  |
| --- | --- | --- |
|  |  |  |

右浮动： float:right;



1. 字围效果



div挡住了p，但是p中的文字不会被挡住，形成“字围”效果。

关于浮动我们要强调一点，浮动这个东西，我们在初期一定要遵循一个原则：

**永远不是一个东西单独浮动，浮动都是一起浮动，要浮动，大家都浮动。**

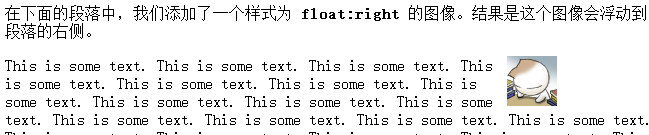
**第一个例子**

img

{ float:right;

margin-right:100px;

}



**第二个例子**

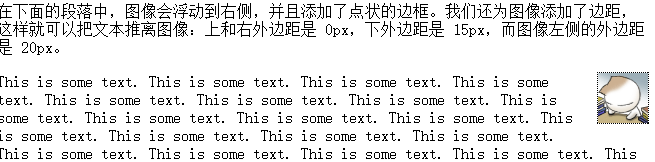
img {

**float:right;**

**border:1px dotted black;**

**margin:0px 0px 15px 20px;**

**}**



**带标题的图像浮于右侧**

div{

float:right;

width:120px;

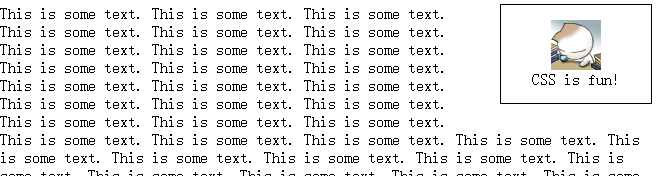
margin:0 0 15px 20px;

padding:15px;

border:1px solid black;

text-align:center;

}



**首字母浮于左侧**

span{

float:left;

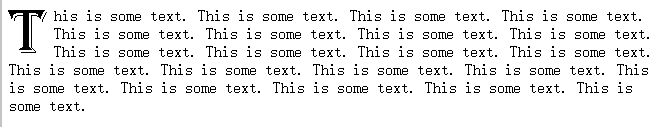
width:0.7em;

font-size:400%;

font-family:algerian,courier;

line-height:80%;

}



**创建水平菜单（导航列表）**

<style type="text/css">

ui{

**float:left;**

width:100%;

padding:0;

margin:0;

**list-style-type:none;**

}

a{

float:left;

width:7em;

text-decoration:none;

color:white;

background-color:purple;

padding:0.2em 0.6em;

border-right:1px solid white;

}

a:hover {background-color:#ff3300}

**li {display:inline} //uxia 段内元素**

</style>

**Clear**

浮动元素哪侧没有其他的元素

|  |  |
| --- | --- |
| **值** | **描述** |
| left | 在左侧不允许浮动元素。 |
| right | 在右侧不允许浮动元素。 |
| both | 在左右两侧均不允许浮动元素。 |
| none | 默认值。允许浮动元素出现在两侧。 |
| inherit | 规定应该从父元素继承 clear 属性的值。 |

**创建无表格的首页 float创建**

<head>

<style type="text/css">

div.container

{

width:100%;

margin:0px;

border:1px solid gray;

line-height:150%;

}

div.header,div.footer

{

padding:0.5em;

color:white;

background-color:gray;

clear:left;

}

h1.header

{

padding:0;

margin:0;

}

div.left

{

float:left;

width:160px;

margin:0;

padding:1em;

}

div.content

{

margin-left:190px;

border-left:1px solid gray;

padding:1em;

}

</style>

</head>

<body>

<div class="container">

<div class="header"><h1 class="header">W3School.com.cn</h1></div>

<div class="left"><p>"Never increase, beyond what is necessary, the number of entities required to explain anything." William of Ockham (1285-1349)</p></div>

<div class="content">

<h2>Free Web Building Tutorials</h2>

<p>At W3School.com.cn you will find all the Web-building tutorials you need,

from basic HTML and XHTML to advanced XML, XSL, Multimedia and WAP.</p>

<p>W3School.com.cn - The Largest Web Developers Site On The Net!</p></div>

<div class="footer">Copyright 2008 by YingKe Investment.</div>

</div>

</body>

Css分类

**显示为内联元素**

Display:none

**显示为块级元素**

**隐藏元素**

**Visibility：**

|  |  |
| --- | --- |
| visible | 默认值。元素是可见的。 |
| hidden | 元素是不可见的。 |
| collapse | 当在表格元素中使用时，此值可删除一行或一列，但是它不会影响表格的布局。被行或列占据的空间会留给其他内容使用。如果此值被用在其他的元素上，会呈现为 "hidden"。 |
| inherit | 规定应该从父元素继承 visibility 属性的值。 |

**光标属性**

**<span style="cursor:move">**

|  |  |
| --- | --- |
| *url* | 需使用的自定义光标的 URL。  注释：请在此列表的末端始终定义一种普通的光标，以防没有由 URL 定义的可用光标。 |
| default | 默认光标（通常是一个箭头） |
| auto | 默认。浏览器设置的光标。 |
| crosshair | 光标呈现为十字线。 |
| pointer | 光标呈现为指示链接的指针（一只手） |
| move | 此光标指示某对象可被移动。 |
| e-resize | 此光标指示矩形框的边缘可被向右（东）移动。 |
| ne-resize | 此光标指示矩形框的边缘可被向上及向右移动（北/东）。 |
| nw-resize | 此光标指示矩形框的边缘可被向上及向左移动（北/西）。 |
| n-resize | 此光标指示矩形框的边缘可被向上（北）移动。 |
| se-resize | 此光标指示矩形框的边缘可被向下及向右移动（南/东）。 |
| sw-resize | 此光标指示矩形框的边缘可被向下及向左移动（南/西）。 |
| s-resize | 此光标指示矩形框的边缘可被向下移动（南）。 |
| w-resize | 此光标指示矩形框的边缘可被向左移动（西）。 |
| text | 此光标指示文本。 |
| wait | 此光标指示程序正忙（通常是一只表或沙漏）。 |
| help | 此光标指示可用的帮助（通常是一个问号或一个气球） |

FW使用

新建时 默认72 的分辨率

Ctrl|+alt+r 标尺的快捷键 ，在标尺上拉出参考线，按住shift显示像素，用切图工具也可以实现

测上下的padding的时候，要先测出行高，做切片得到padding，

测左右padding的时候，文字一般直接测出左边的padding就行了，右侧可能换行

这时height为内容高，加上padding为盒子高

项目创建

规划好页面

项目文件夹

1.首页 index.html

2.样式 css文件夹 css文件 相同样式 （全局样式 公共样式）

Base.css（基本样式） global.css (全局样式)

3.图片 images 文件

4.特效 js文件夹 js 文件

Css 初始化

**@charset** "UTF-8";  
/\*css 初始化 \*/  
html, body, ul, li, ol, dl, dd, dt, p, h1, h2, h3, h4, h5, h6, form, fieldset, legend, img { margin:0; padding:0; }  
fieldset, img,input,button { border:none; padding:0;margin:0;outline-style:none; }  
ul, ol { list-style:none; }  
input { padding-top:0; padding-bottom:0; font-family: "SimSun","宋体";}  
select, input { vertical-align:middle; }  
select, input, textarea { font-size:12px; margin:0; }  
textarea { resize:none; } /\*防止拖动\*/  
img {border:0; vertical-align:middle; } /\* 去掉图片低测默认的3像素空白缝隙\*/  
table { border-collapse:collapse; }  
body {  
 font:12px/150% Arial,Verdana,"\5b8b\4f53";  
 color:#666;  
 background:#fff  
}  
.clearfix:before,.clearfix:after{  
 content:"";  
 display:table;  
}  
.clearfix:after{clear:both;}  
.clearfix{  
 \*zoom:1;/\*IE/7/6\*/  
}  
a{color:#666; text-decoration:none; }  
a:hover{color:#C81623;}  
h1,h2,h3,h4,h5,h6{text-decoration:none;font-weight:normal;font-size:100%;}  
s,i,em{font-style:normal;text-decoration:none;}  
.col-red{color: #C81623**!important**;}  
  
/\*公共类\*/  
.w { /\*版心 提取 \*/  
 width: 1210px;margin:0 auto;  
}  
.fl {  
 float:left  
}  
.fr {  
 float:right  
}  
.al {  
 text-align:left  
}  
.ac {  
 text-align:center  
}  
.ar {  
 text-align:right  
}  
.hide {  
 display:none  
}

Shortcut和结尾公共则在bass.css

Favicon

<link rel="shortcut icon" href="[favicon.ico](view-source:file:///C:/Users/andy/Desktop/%E4%BA%AC%E4%B8%9C%E7%BD%91%E7%AB%99/favicon.ico)" />

常见的标签一些性质

S del 删除线

I em 倾斜

U ins 下划线

字体加粗 font-weight: 700;

让字体不加粗: font-weight:normal;

字体倾斜: font-style:italic; 不用

字体不倾斜： font-style:normal;

不下划线 不删除线: text-decoration: none;

定位： position：static; 静态定位 约等于标准流

浮动的不浮动: float:none; none left right

定位的不定位: position: static; absolute relative fixed

标签 1 类 10 id 100 行内 1000

网页稳定：

Width 和height 最稳定

其次 padding

最后才考虑margin

正常流 normal flow

浮动 定位 脱标 out of flow

浮动目的：

可以让多个块级 元素 放到一行上。

Float: left right none;

搜索栏



搜索栏和下面的文字是两个不同的块

**搜索栏由input和button组成 ，button和img属于行内块元素。和其他元素之间总有间隔，因此链接在一起的方法是都浮动**

**Ctrl+l 定位行号**

淘宝多盒子并排显示边框和选中

**多个盒子并排的时候**：边界宽， margin-left：-1px；由于浮动的作用才会盖住

Div和其他标签都可以用hover了

Div：hover{

Border:1px solid red;

Position:relative; //防止遮挡

}

当并排的都有定位的时候

Div：hover{

z-index=1；

}

绝对定位的遮盖方法

当需要绝对定位来遮盖的时候，可以将这个绝对定位放在一个空的相对定位里，然后位于前面也采用相对定位，后来的在上面就可以制作定位了

定位同事出现left right

**定位的时候同时有left、right、top、buttom的时候，执行left 和top**

背景半透明

Css3

Background：rgba（0,0,0,0.5）； 红绿蓝 透明度

Opacity：0.5 盒子透明度 ，里面的文字什么的也就透明了

行内元素没有上下的padding margin 但是有左右的