介绍行业：

软件：

开发用的软件有哪些？

dw（）

hb（）

sublime

Webstorm

Notepad++

index.html或default.html首页默认名字

网页扩展名：html,htm

ui： ps firworks ai vi

网络三剑客：



Ps里改变成像素：（px）

Shift连续选择 不连续选择ctrl

1.找到编辑-打开首选项-打开单位与标尺-将文字和标尺改为像素

颜色的三原色 红（red） 绿（green ） 蓝（ blue）

#cccccc #000000 #ffffff

Rgb()

保存图片的格式：

1. png 2. jpg 3.gif 他们三者的区别？

GIF，JPG，png的区别？

GIF格式只有256种颜色，支持透明，动画。

JPG格式颜色丰富，文件小。

png格式是无损压缩，文件大，支持透明，半透明。

title 图片正常显示鼠标经过出现文字说明

alt 图片不能正常显示显示文字

ps切图： 有图psd格式图片层

Jpg,png格式图片切图

1. 选框选中你需要的图片
2. Ctrl+c 复制
3. Ctrl+n 新建图层
4. Ctrl+v 粘贴
5. Ctrl+s 保存 （保存位置）
6. Ctrl+alt+z 连续撤回
7. ctrl+e 合并图层
8. Ctrl+alt+shift+s 保存
9. crtl+d 取消选框
10. I 吸管

浏览器： 谷歌（谷歌调试） 火狐 ie 360 ..





1. Internet Explorer 简称ie，微软公司旗下浏览器
2. google chrome浏览器，google旗下浏览器，追求简洁、快速、安全，个人认为是最好用、速度最快的浏览器。
3. firefox浏览器，简称FF浏览器，mozilla公司旗下浏览器，个人认为第二好用浏览器。
4. safari浏览器，苹果公司旗下浏览器，在苹果系统下是很优秀的浏览器
5. opera浏览器，挪威厂商opera旗下浏览器

1.什么是web标准？

WEB标准，即网站标准。目前通常所说的WEB标准一般指网站建设采用基于XHTML语言的网站设计语言,WEB标准中典型的应用模式是“css+div”（什么是css+div）。实际上，WEB标准并不是某一个标准，而是一系列标准的集合。

网页主要由三部分组成：结构html（Structure）、表现css（Presentation）和行为js（Behavior）。对应的网站标准也分三方面：结构化标准语言，主要包括HTML；表现标准语言主要包括CSS；行为标准主要包括对象模型（如W3C DOM）、ECMAScript等。这些标准大部分由W3C组织（什么是W3C组织）起草和发布，也有一些是其他标准组织制订的标准，比如ECMA（European Computer Manufacturers Association）的ECMAScript标准。

分为三层：

结构：html 表现：css 行为 ：js

Html

Html是什么?

HTML 是用来描述网页的一种语言。 能在浏览器中显示

超文本标记语言 （Hpyer Text Markup Language）

html描述网页的一种语言

css是什么?

层叠样式表，用来控制网页的外观

层叠样式表Cascading Style Sheets

用于控制网页外观

标记语言是：有一套标记标签

标签：<标签名></标签名>

Eg:<html></html>

<img>

标签

<!DOCTYPE html>-----生命当前文档类型为：html

<html>-----页面标签

<head>-----页面头部标签

<meta charset = “UTF-8”>声明当前网页的编码格式。UTF-8----->万国码用UTF-8不会因系统的简、繁体而乱码。

<title>我是标题</title>-----页面标题，用来改变 浏览器标题的文字

<style></style>-----内部样式书写位置

<scirpt></scirpt>-----内部JS书写位置

</head>

<body>-----网页主体内容书写区域

</body>

</html>

标签上写属性固定写法： 属性=“属性值”，

例如：<p style = “color:red;”></p>

h1-h6 标题标签 推荐用h2 特点：默认加粗，h1最大h6最小

p段落标签

br 换行标签 折行（空标签）

img 图片 src 指定路径 =“../图片名称.png”

span 放特殊文字的

常用的特殊符号

空格：HTML的空格用 &nbsp;表示。【所有标签都为英文半角】

小于号：&lt;

大于号：&gt;

&：&amp;

双引号：&quot;

版权：&copy;

注册：&reg;

列表标签：

1. 无序列表 ul li <ul><li></li></ul>

无序列表的属性值：disc实心圆 square正方块 circle空心圆

2.有序列表 ol li

属性：type类型

start从第几个开始

reversed=“reversed”倒序

3.自定义列表 dl dt dd （）

超链接 锚点标签 a 用来跳转页面

1. href用来指定链接的地址。

地址可以是：

1. 网址
2. 邮件地址
3. 本地文件地址
4. 空链接
5. 同一文档内容某一区块，类似于书签
6. Target 属性对应的值决定是在本窗口打开还是新窗口打开

\_blank \_search new打开新窗口

\_parent \_self本窗口(原窗口)

一.标签分为

1.双标签：有开始标签 有闭合

举例:h1-h6 p ul li ol dl dt dd span a

2.单标签：只有一个

举例：br img meta link hr

导航条:ul li

单导航

<ul>

<li>你好</li><li>你好</li><li>你好</li>

</ul>

双导航

<ul>

<li><a href=”#”>拉拉</a><a href=”#”>拉拉</a></li>

<li><a href=”#”>拉拉</a><a href=”#”>拉拉</a></li>

</ul>

图片

src 指定路径

属性

Src = “../image/图片名称.png”

1. 我对这个题的看法 以上就是我对这个题的看法

什么是？ 作用？应用场景？ 优缺点？

背景

前景图 img banner 展示信息 更新速度比较快

背景图 background 用在点缀地方 更新速度比较慢背景

## 背景复合属性background

背景属性包含背景色，背景图，背景重复，背景位置和背景复合属性等。

背景颜色background-color

背景图片background-image:url();

背景重复：background-repeat:

属性值：repeat（平铺）/no-repeat（不平铺）/repeat-x（水平平铺）/repeat-y（垂直平铺）

背景位置：background-position:水平 垂直;

可以取的值为关键字，数值，百分比

水平：left center right（或者数值）

垂直：top center bottom（或者数值）

background复合属性在一个声明中设置所有的背景属性，如背景颜色、背景图像、背景重复及背景位置的设置，属性值的顺序可调换。

如：background: red url(1.jpg) no-repeat right top;

方向（左右）（上下）

**b**  加粗标签 <b></b> bold

**strong**  加粗标签（强调意义）<strong></strong>

*i*  倾斜标签 <i></i>

*em* 倾斜标签（强调意义）<em></em>

字体样式：

**font :改变字体 color 颜色 size：20px 大小**

**Face：“微软雅黑” 字体**

**weight：800**

**（normal 不加粗 400正常 bold加粗 bolder更加粗 ）**

**div：**用来分块

1.有左中右结构一定要分大div

1. 有块级标签配合着行内标签是左右结构不需要加div
2. 独立标签能完成的事不需要加div
3. 多个标签需要div

iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。

引用页面效果

Frameborder是否显示边框 （0 ,1）

Height 高度

[scrolling](http://www.w3school.com.cn/tags/att_iframe_scrolling.asp" \o "HTML <iframe> 标签的 scrolling 属性) 是否显示滚动条 no yes auto

width 代表的是宽

7.iframe的作用以及优缺点？

《1》、iframe是用来在网页中插入第三方页面，早期的页面使用iframe主要是用于导航栏这种很多页面都相同的部分，这样在切换页面的时候避免重复下载。

优点：

（1）、便于修改，模拟分离，像一些信息管理系统会用到。

缺点：

（1）、iframe的创建比一般的DOM元素慢了1-2个数量级

（2）、iframe标签会阻塞页面的的加载，如果页面的onload事件不能及时触发，会让用户觉得网页加载很慢，用户体验不好，在Safari和Chrome中可以通过js动态设置iframe的src属性来避免阻塞。

（3）、iframe对于SEO不友好，替换方案一般就是动态语言的Incude机制和ajax动态填充内容等。

marquee 实现滚动

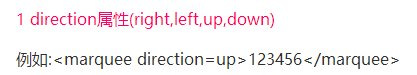
滚动方向 direction up (上) down（下） left （左） right（右）

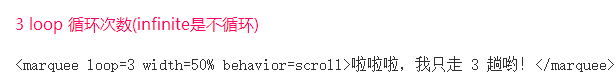
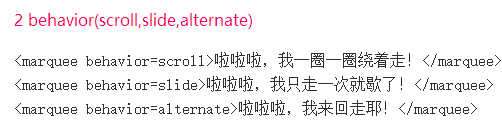
滚动方式 behavior scroll(循环滚动) slide 只滚动一次 alternate来回交替滚动

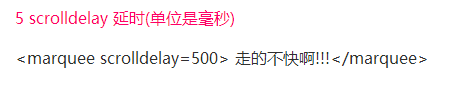
滚动循环 loop 循环次数

滚动背景颜色 bgcolor

滚动延迟scrolldelay（单位毫秒)







表格：<table></table> 表格的最大标签

表格的标签：

caption 表格的标题 双标签

位置放在开始table的下面

行标签 tr <tr></tr>

列标签 td <td></td>

表头单元格 th

加粗且居中<th></th>

<thead></thead> 表格的头部（只能一次）

<tbody></tbody> 隐形标签 表格的中间（n次）

<tfoot></tfoot> 表格的底部 （脚注）（只能一次）表格的属性 ：

宽度（width）

高度（height）

**border-collapse**简述：设置行和单元格的边是否合并  
概述：border-collapse是设置表格的行<tr>或<td>之间是否留有距离的样式，<tr>与<td>之间的距离，在表格里可以通过 cellspacing="0"来实现。换句话说，当border-collapse设置为collapse时，表格自身的cellspacing属性便没有作用了。

**语法：**

**border-collapse**：separate | collapse

**取值：**

**separate：**

边框独立 分开的

**collapse：**

相邻边被合并 折叠的意思

<style type="text/css">

table {

width: 300px;

border-collapse: collapse;

}

table, th, td {

text-align: left;

border: 1px solid #000000;

}

</style>

<table>

<caption>User Info</caption>

<tr>

<th>Name</th>

<th>Email</th>

</tr>

<tr>

<td>John</td>

<td>john@mail.com</td>

</tr>

</table>

cellspacing="0"

bordercolor 边框线颜色 table tr td

bgcolor 表格背景颜色 table tr td

align 水平方向的位置 left左/center中/right右

vertical-align 竖直方向的位置 top上/middle中/bottom下/baseline基线 (同上) 要有确定的高度（height）

单元格的合并

作用于单元格（th,td） 属性是作用于单元格上的 也就是th td;

rowspan 跨行 上下合并 跨几行写数字几 最小的就是2

用法：td rowspan="2"></td> 将多余的对应的单元格删掉

colspan 跨列 左右合并 跨几列写数字几 最小的就是2

用法：<td colspan="2"></td> 将多余的对应的单元格删掉

form表单 input 输入控件

<form action="#"><!--action属性 指的是，表单提交时进入的页面 属性值：url网址--></form>

<input/> 单标签 input也有属性 input 又包括许多type类型

<form>

用户名：<input type="text" />单行文本框

<input type="text" value="请输入..."/>这个显示字体为黑色， 而且要添加

value 显示出来的字体是黑色 我们可以用color改变它的字体颜色

**<input type="text" placeholder="请输入..."/>文字显示为灰色，鼠标选中，默认字体消失**

outline:none;去掉文本框上点击时的蓝边

密码：<input type="password" />密码框

普通按钮<input type="button" value="按钮名称" />

单选<input type="radio" name="值" />

多选<input type="checkbox" />

默认选中选项 属性是 **checked="checked**";(默认选中)

不让你选择这个选项 ，禁止选择此选项， 属性是 **disabled="disabled**"禁止选择，

文件域<input type="file" />

图片域<input type="image" src="../imgs/top1.png" />

提交按钮：<input type="submit" value="提交" />

重置按钮：<input type="reset" value="取消" />

<textarea>文本域，多行文本框</textarea>

属性：cols可見列数 rows可见行数

<select size="1"> <!--size可显示选项数-->

<option>下拉文本框</option>

<option>1</option>

<option>2</option>

</select>

</form>

select{

-webkit-appearance:none;去掉下拉框的的小三角

}

表现Css

将css与html结合

1. .在标签写样式

<p style=”color:#fff”>你好</p>

2.内嵌 （放在head里面写style）

<head>

<style>

p{ color: red; background:yellow; width:100px; height:100px}

</style>

</head>

3．链接

新建css文件 在head里链接<link rel="stylesheet" href="css链接" />

4.导入

@import url ( css/styel.css) 共同的css样式链接

优先级：

1. 链接的方式：

行内样式>内嵌>链接>导入

链接样式和导入样式的区别及优缺点？

区别1：Link是HTML标签，;@import 属于css范畴  
区别2：Link引用css时在页面载入时同时加载  
@import需要页面网页完全载入以后加载，  
区别3: Link是HTML标签，无兼容问题;@import是在css2.1提出的。低版本的浏览器不支持（只有ie5以上的才识别）

区别4：link支持使用JavaScript控制dom去改变样式。而@import不支持。

Link方式的优先权重高于@import

CSS选择器

1. 标签选择器 标签{属性：属性值}

2. 类选择器 类名可以多处使用（同样的标签不同的效果 不同样的标签同样的样式）

步骤：先起名 class=“p-one” **.** 类名{属性：属性值}”

3. id选择器 id名只能使用一次(唯一性)

步骤：先起名id=”id名” # id名{属性：属性值}”

4.群组选择器

#id名，#id名{属性：属性值}

权重值：id 100 class 10 标签 1 全局 0

选择器优先级

全局<标签<类<id<!important

提高优先级是!important

例：div{

background: red!important;

}

div{

background: blue;

}

5.伪类选择器有

:Link 超链接的普通样式

:visited 被点击访问过后的样子

:hover 滑过的样式

:active 鼠标按下的样式

有一点需要特别注意，那就是他们的先后顺序，L-V-H-A

要让他们遵守一个爱恨原则LoVe/HAte,也就是Link--visited--hover--active。

提示：在 CSS 定义中，a:hover 必须被置于 a:link 和 a:visited 之后，才是有效的。

提示：在 CSS 定义中，a:active 必须被置于 a:hover 之后，才是有效的。

：hover 鼠标滑过 .box ul li a：hover{} 把 a 变成块元素 padding值加给a

浮动

浮动的产生：

是当前元素脱离普通流，浮动的框可以左右移动，直至他的外边缘遇到包含框或者另一个浮动框的边缘。

float：left左浮 right右浮 none默认值

clear:left; 清除左浮

clear:right;清除右浮

clear:both;左右全部清除

clear:none; 默认

margin-left 左 righ 右 top上 bottom下 盒子之间的边距

css属性：

\*{padding:0;margin:0}

li{list-style:none} 去掉li的项目符号

a{text-decoration:none;}去除[超链接标签]的下划线

3.border: 1px solid #ccc;

4.Width

5.Height

盒模型：modelbox 简称盒模型

Padding作用：文字与文字间的距离

Padding:10px;一个值 上右下左

Padding:10px 20px;两个值 上下 左右

Padding:10px 20px 30px;三个值 上 左右 下

Padding:10px 20px 40px 50px;四个值 上右下左

Margin作用：标签与标签的距离

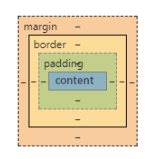
margin:10px;一个值 上右下左

margin:10px 20px;两个值 上下 左右

margin:10px 20px 30px;三个值 上 左右 下

margin:10px 20px 40px 50px;四个值 上右下左

盒模型的构成content+padding+margin+border

盒模型的示意图：

宽度的构成 content+左右的padding +左右的border +左右的margin

高度的构成 content+上下的padding+上下的border +上下的margin

calc(表达式) 计算 计算的值是动态的

1.计算单位 em px rem %

2.计算 + - \* /

3.可以使用混合单位进行计算

4.表达式中 + 和 - 时 前后必须有空格 eg：width:calc(10 + 5)

5.表达式中\* 和/ 前后可以没有空格 但推荐 加上 空格

应用场景： 1.居中定位元素,行内元素给宽高没有作用（转换成块）

详解：

padding margin border border-radius 详解

padding/margin 四个值： 上 右 下 左

padding/margin 三个值 上 左右 下

padding/margin 两个值 上下 左右

padiing/margin 1个值 上 右 下 左

Margin的几个值和padding基本相同

border

border-top: 上边框线

border-left: 左边框线

border-width 边框线的粗细

border-style 边框线的类型

solid 实线 dashed 虚线 dotted 点线 double 双实线

border-color 边框线的颜色

圆角

border-radius：5px；

border-top-left-radius 左上角

border-top-right-radius 右上角

border-bottom-left-radius

border-bottom-right-radius

圆

border-radius ：50% 100% 高度的一半

元素

1.行内元素（内联元素 display:inline）

1.行内元素在一行显示

2.只能容纳文本或者其他行内元素;

3.不可设置宽度高度（margin-top、margin-bottom、padding-top、padding-bottom无效）

常用行内元素有： a span input I b textarea strong select label br time

2.块状元素(display：block)：

1.块元素一般都独占一行,

2.可以容纳行内元素和其他块元素,

3.常用块级元素有： h1-h6 p div dl dt dd ul li ol table form hr

3.行内块元素（display:inline-block） img input (即具有行内元素的特性，又具有块元素的特性)

4.display :none(不显示)

float就是隐性的把内联元素转化为块元素，但是他不占据一行。对外是内联元素的属性。

他有个坏处就是会影响兄弟元素。相当于：display:inline-block;  
 那为什么不直接display:inline-block;因为这个玩艺在ie6下有几个px的bug。

元素的转化

display：block 将行内元素转化为块元素

display：inline 将块元素转化为行内元素

display：lnline-block 既有行内元素 又有块元素

溢出隐藏 overflow:hidden

Overflow:auto 出现滚动条 让多余的东西在滚动条里面

隐藏 display:none

visibility:hidden

相同点都是用来隐藏元素的

visibility:hidden 隐藏掉 占位

display:none 隐藏掉 不占位

做出来鼠标划过出来一个div

Css属性：

width

height

max-width 最大的宽度

该属性值会对元素的宽度设置一个最高限制。因此，元素可以比指定值窄，但不能比其宽。不允许指定负值。

max-height

min-width

min-height

color

backgroud

font-size

font-style:

italic 倾斜

letter-spacing 规定文字间距

word-spacing 字母与字母

font-weight 字体加粗

normal不粗 bold加粗 bolder更粗

400

text-decoration: none underline

display: inline inline-bolock block none

line-height

padding

margin

text-align:center

text-indent 首行缩进

border

list-style

outline:none点击文本框时去掉蓝色框

font-weight

normal bold bolder

400

font-family:"" 改变字体种类

backround-size 改变背景图的大小

cover 沾满 数值

float

overflow:hidden auto;

display: inline inline-bolock block none

:before 在某某之前插入

:before 伪元素在元素之前添加内容。

这个伪元素允许元素内容的最前面插入生成内容。默认地，这个伪元素是行内元素，不过可以使用属性 display 改变这一点。

h1: :before {

content: " 80";

background: red;

color:#FF00FF;

font-size:9px;

}

:after 在某某之后插入

:after 伪元素在元素之后添加内容。

这个伪元素允许元素内容的最后面插入生成内容。默认地，这个伪元素是行内元素，不过可以使用属性 display 改变这一点。

h1:after{

content: "9090";

background: red;

color:#FF00FF;

font-size:9px;

}

必须配合 content

:first-letter用于选取指定选择器的首字母

:first-line 选取指定选择器的首行。

<p>环境与偶空环境与偶空环境与偶空环境与偶空环境与偶空环境与偶

<br>空环境与偶空环境与偶空环境与偶空环境与偶空环境与偶空</p>

p:first-line{

background: #f00;

}

::selection 设置对象被选择时的样式(不可以设置边框属性)

:first-child 匹配父元素的第一个子元素

:last-child 匹配父元素的最后一个子元素

:nth-last-child(1)倒数第几个子元素

:nth-child 匹配父元素的第n个子元素

下面这些也需要讲解

语法：nth-child(3n) {指定样式}匹配所有索引是3的倍数的元素

语法：nth-child(even) {指定样式}匹配所有索引是偶数的元素

语法：nth-child(odd) {指定样式}匹配所有索引是奇数的元素

disabled匹配用户界面上处于禁用状态

enabled 匹配用户界面上处于可用状态

E[att^=”val”] 选择具有att属性且属性值为以val开头的字符串的E元素

E[att$=”val”] 选择具有att属性且属性值为以val结尾的字符串的E元素

E[att\*=“val”] 选择具有att属性且属性值为包含val的字符串的E元素

外边距折叠：

上下外边距折叠取最大值

左右外边距折叠取求和

table布局和div布局的区别，优缺点？

table 布局

缺点

1.显示样式和数据绑定在一起

2.布局的时候灵活度不高

3.一个页面可能会有大量的table元素 代码冗余度高

4.增加带宽（[带宽](https://baike.baidu.com/item/%E5%B8%A6%E5%AE%BD" \t "https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E5%B8%A6%E5%AE%BD/_blank)是指在单位时间（一般指的是1秒钟）内能传输的数据量。网络和高速公路类似，带宽越大，就类似高速公路的车道越多，其通行能力越强。）

5.搜索引擎不喜欢这样的布局

搜索引擎（从使用者的角度看，搜索引擎提供一个包含搜索框的页面，在搜索框输入词语，通过浏览器提交给搜索引擎后，搜索引擎就会返回跟用户输入的内容相关的信息列表。）



优点

1.理解比较简单

2.不同的浏览器看到的效果一般相同

3.显示数据还是很好的

DIV+CSS布局

基本思想: 数据和样式分离 其优势：

1.符合w3c标准，微软等公司均为w3c支持者

2.搜索引擎更加友好

3.样式的调整更加方便，内容和样式的分离，使页面和样式的调整变得更加方便。

现在YAHOO MSN等国际门户网站，网易，新浪等国内门户网站和主流的web2.0网站 均采用DIV+CSS布局，更加印证了DIV+CSS是大势所趋

4.CSS的极大的优势表现在简洁的代码，对于一个大型网站来说，可以节省大量带宽。并且搜索引擎也更喜欢简洁的代码

5.表现和结构分离，在团队开发中更容易分工合作而减少相互关联性

DIV+CSS并不是要我们抛弃table 因为table在显示数据时，特别方便

因此在使用DIV+CSS时，该使用table时，就得使用.

清除浮动的方式：

zoom ie专有的，用来清除浮动。清除margin重叠问题

1.代码示例

|  |
| --- |
| **父级的div定义height**  <style>  .div1{background:#000080;border:1px solid red;/\*解决代码\*/height:200px;}  .div2{background:#800080;height:100px;margin-top:10px  border:1px solid red;}  .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="div2">  div2  </div>  1.父级的div定义height  优点：代码少，容易掌握  缺点：只适合高度固定的布局，要给出精确的高。  如果高度和父级不一样时，会产生问题  建议: 不建议使用，只建议高度固定的布局时使用 |

2.代码示例

|  |
| --- |
| **(受影响的上边)结尾处加空div标签 clear:both**  或者给受影响的div2{clear:both;}  <style>  .div1{background:#000080;border:1px solid red}  .div2{  background:#800080;  border:1px solid red;  height:100px;  margin-top:10px;  }    .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}    /\*清除浮动代码\*/  .clear{clear:both}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="clear"></div>  <div class="div2">  div2  </div>  2.结尾处加空div标签 clear:both  优点：简单代码少，浏览器支持好，不会出现问题  缺点：初学者不懂原理，如果页面浮动布局多，就要增加很多空div  建议：不推荐使用，但这种方式是以前经常使用清除浮动的方法 |

3.代码示例

|  |
| --- |
| **父级div定义伪类：after 和zoom**  <style >  .div1{background:#000080;border:1px solid red;}  .div2{background:#800080;height:100px;margin-top:10pxborder:1px solid red;}  .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}    /\*清除浮动代码\*/  .clear:after{display:block;clear:both;content:"";visibility:hidden;height:0}  .clear{zoom:1}  </style>  <div class="div1 clear">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="div2">  div2  </div>  3.父级div定义伪类：after 和zoom  代码：.clear：after{content:"";display:block;clear:both; visibility:hidden;  height:0}  .clear{zoom:1}  优点：浏览器支持好  缺点：代码多，不好理解  推荐：使用，定义公共类，减少css代码 |

4.代码示例

|  |
| --- |
| **父级定义overflow:hidden**  <style >  .div1{background:#000080;border:1px solid red;/\*解决代码\*/width:98%;overflow:hidden}  .div2{background:#800080;border:1px solid red;height:100px;margin-top:10px;width:98%}    .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="div2">  div2  </div>  4.父级定义overflow:hidden  原理：必须定义width或者zoom：1；同时不能定义height使用  overflow:hidden时浏览器会自动检测浮动区域的高度  优点：代码少，浏览器支持好  缺点：不能配合position使用因为超出的尺寸会被隐藏 |

5.代码示例

|  |
| --- |
| **父级div定义overflow:auto**  <style>  .div1{background:#000080;border:1px solid red;/\*解决代码\*/width:98%;overflow:auto}  .div2{background:#800080;border:1px solid red;height:100px;margin-top:10px;width:98%}    .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="div2">  div2  </div>  5.父级定义overflow:auto  原理：必须定义width或者zoom：1；同时不能定义height使用  overflow:auto时浏览器会自动检测浮动区域的高度  优点：代码少，容易掌握  缺点：内部高度超出父盒子会出现滚动条 |

6.代码示例

|  |
| --- |
| **父级div也一起浮动**  <style>  .div1{background:#000080;border:1px solid red;  /\*解决代码\*/ float:left;  width:98%;margin-bottom:10px; }  .div2{background:#800080;border:1px solid red;height:100px;width:98%;/\*解决代码\*/clear:both}    .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="div2">  div2  </div>  原理：所有代码一起浮动，就变成了一个整体  优点：没有优点  缺点：会产生新的浮动问题。  建议：不推荐使用，只作了解。 |

7.代码示例

|  |
| --- |
| **父级div定义display:table**  <style>  .div1{background:#000080;border:1px solid red;  /\*解决代码\*/ display:table;  width:98%; margin-bottom:10px;}  .div2{background:#800080;border:1px solid red;height:100px;width:98%;}    .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  </div>  <div class="div2">  div2  </div>  原理：将div属性变成表格  优点：没有优点  缺点：会产生新的未知问题  建议：不推荐使用，只作了解 |

8.代码示例

|  |
| --- |
| **结尾处加br标签clear:both**  <style>  .div1{background:#000080;border:1px solid red;margin-bottom:10px;zoom:1}  .div2{background:#800080;border:1px solid red;height:100px}    .left{float:left;width:20%;height:200px;background:#DDD}  .right{float:right;width:30%;height:80px;background:#DDD}    .clearfloat{clear:both}  </style>  <div class="div1">  <div class="left">Left</div>  <div class="right">Right</div>  <br class="clearfloat" />  </div>  <div class="div2">  div2  </div>  原理：父级div定义zoom:1来解决IE浮动问题，结尾处加br标签clear:both  建议：不推荐使用，只作了解 |

单位：px em rem 区别及特点？

浏览器默认下来就是px em rem 详解16像素

px：绝对度量单位，就是像素

em：相对度量单位，但是计算起来实在费劲。 em指的是相对于当前对象内文本的字体尺寸。 如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。换句话说，如果当前div的字体大小是12像素，那么1em就是12像素，如果div字体大小是24像素，1em就是24像素。

rem，也是和字体相关的相对度量单位， 但是，rem是相对于根元素的字体大小进行设置的， 如果html元素中的字体大小设置为24px， 那么针对任意html内的元素设置1rem均表示的是24px，大大节省了我们计算字体大小的时间。

用法：

1. body选择器中声明Font-size=62.5%；

2. 将你的原来的px数值除以10，然后换上em/rem作为单位；

3. 重新计算那些被放大的字体的em/rem数值。避免字体大小的重复声明。

pX特点：

1. IE无法调整那些使用px作为单位的字体大小；

2. 国外的大部分网站能够调整的原因在于其使用了em或rem作为字体单位；

3. Firefox能够调整px和em，rem，但是96%以上的中国网民使用IE浏览器(或内核)。

EM特点

1. em的值并不是固定的；

2. em会继承父级元素的字体大小。

REM特点

1.是相对于根元素的字体大小进行设置的，rem的值比较固定

多余显示省略号：

<p style="width:100px;height:20px;text-overflow:ellipsis; white-space:nowrap;overflow:hidden; ">a b c d e f g h i j k l , msa sd sd sa w df f </p>

Width:宽度

white-space:nowrap; 文本强制显示一行

overflow:hidden;溢出隐藏

text-overflow:ellipsis； 显示省略号

**text-overflow：clip默认值显示 | ellipsis显示省略号**

透明度

opacity 数值0.1-0.9之间 实现的是背景与文字都透明 1是不透明

ie浏览器中设置透明需要加入ie过滤器：filter：alpha(opacity=数值)

背景透明·；background: rgba(r,g,b,a); r 红色g 绿色b 蓝色 （数值0-255） a透明度

cursor:pointer 改变鼠标。为小手状态 move 改变鼠标为移动状态

盒子阴影 Box-shadow

box-shadow:10px水平 10px垂直 10px模糊距离 #000阴影颜色

水平正值右边 负值左边 垂直正值下面 负值上边



文字阴影：

Text-shadow:1px 3px 4px red;

text-shadow: h-shadow v-shadow blur color;

none：无阴影，默认值。

h-shadow:设置对象的阴影水平偏移值。可以为负值

v-shadow:设置对象的阴影垂直偏移值。可以为负值

blur:设置对象的阴影模糊值。不允许负值，可以缺省。

color:设置对象的阴影的颜色。 可以缺省。缺省的时候默认是#333深灰色，不是黑色。

只有外阴影没有内阴影

CSS

.ts1{

text-shadow:10px 20px 1px #666,20px 40px 3px #999;}

### 凸起的效果，有点像轻微的浮雕

CSS

.ts5{

text-shadow:-1px -1px #fff,1px 1px #333;

color:#ccc;

background-color:#888;}

### 、凹下去的效果，和凸出的效果刚好相反。

CSS

.ts6{

font-family:Verdana, Geneva, sans-serif;

color:#666;

text-shadow:-1px -1px #000,-2px -2px #333,1px 1px #fff,2px 2px #eee;}

* 1. **gradient渐变**

所谓渐变是指从一种颜色慢慢过渡到另外一种颜色。

1. 绘制线性渐变

使用的代码如下：

background:linear-gradient(to bottom,red,yellow);

第一个参数可指定的参数值如下：

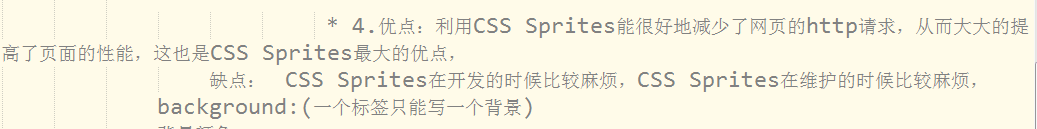
|  |  |
| --- | --- |
| 参数值 | 渐变方向 |
| To bottom | 从上往下 |
| To right | 从左往右 |
| To top | 从下往上 |
| To left | 从右往左 |

第二个参数和第三个参数分别表示渐变的起点色和终点色。

什么是精灵技术？

把页面上的 ICON，栏目背景，图片按钮等有规则的合并一张背景图,然后用background-position来实现背景图片的定位。

精灵技术优缺点



优点：利用css sparites能很好的减少网页的http请求；从而大大的提高了页面的性能，这也是css sparites最大的优点

缺点：css sparites在开发的时候比较麻烦，css sparites在维护的时候比较麻烦，background（一个标签只能写一个背景）

Background: url 是否平铺 位置

background-image：url（img链接）

background-repeat：是否平铺 no-repeat

background-position： 横向（负值）

第一步：在ps里面切图 把小图标整合在一个图层上 （中间的空隙留得较大）

第二歩： 给每个li加上你整合好的图片

第三歩：找到每个li 去写背景位置

阿里巴巴矢量图标库用法：

1. 登录
2. 找到图标管理我的项目
3. 新建项目
4. 回到图标库找到想要的图片 将图标加入购物车
5. 回到购物车将图片放到项目里面
6. 将其下载
7. **定位的四个属性 position**
8. 默认值（静态）：static
9. 绝对定位：absolute
10. 相对定位：relative
11. 固定定位：fixed
12. **定位的意义**
13. 相对定位的概念：设置相对定位的元素，相对于该元素原来位置偏移(相对定位是相当于对于自己定位的)
14. 绝对定位的概念：设置绝对定位的元素，相对于具有相同定位属性的父级元素偏移（绝对定位相对于有相对定位的父元素定位的）
15. 固定定位：相对于浏览器定位
16. 父元素没有相对定位时绝对定位相对于浏览器定位

position:absolute 绝对定位

1.不占位

2.定位原点是body（0，0）

3.找到精确位置需配合属性使用left/right/top/bottom

position: relative相对定位

1.占位

2.定位原点是自身

3.找到精确位置需配合属性使用left/right/top/bottom

position:fixed;固定定位

1.不占位

2.定位原点是body（0，0）

3.找到精确位置需配合属性使用left/right/top/bottom

相对定位和绝对定位配合使用：

1.被定为者是相对定位

2.定位者是个绝对位

3.相对定位和绝对定位配合使用 z-index 提高优先级 一定是给相对定位的元素

1. **定位属性方向值运用规范**

|  |  |
| --- | --- |
| **值** | **描述** |
| Absolute | 设置绝对定位的元素，相对于具有相对定位属性的父级元素偏移  元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| fixed | 生成绝对定位的元素，相对于浏览器窗口进行定位。元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| relative | 相对定位的元素，相对于该元素原来位置偏移因此元素的位置通过 "left", "top", "right" 以及 "bottom" 属性进行规定。 |
| Static | 默认值。没有定位（元素的位置通过top, bottom, left, right）。 |

定位 z-index 提高定位的显示顺序 谁的index值大谁就优先

1. **元素的层叠顺序**
2. z-index属性：设置元素的层叠顺序，属性值为无单位的整数值，值较大的元素会叠加在值较小的元素之上；
3. 元素可拥有负的z-index属性值，默认值是：auto,默认层是：0层
4. 当没有设置z-index属性或者层数相同时，后面的元素显示在上面
5. z-index的属性值，只能为整数，正整数，0，负整数
6. z-index值越大越靠上

## 相对定位及绝对定位的区别

1. 相对定位（position：relative）是以自身为基点，进行定位
2. 绝对定位（position：absolute）是以父元素为基点，进行定位

居中一个定位元素 （三种方式）

top:50%;margin-left:盒子负一半 margin-top:盒子负一半 left：50% （宽高、定位）

.box{

width:300px;

height:300px;

background: #ff0;

position: /\*fixed、absolute\*/

top:50%;

margin-left:-150px;

margin-top:-150px;

left:50%;

}

left：50%；top:50% transform: translate(-50%,-50%);

.box{

width:300px;

height:300px;

background: #ff0;

position: relative; /\*fixed、absolute\*/

left:50%;top:50%;

/\*

transform: 该属性允许我们对元素进行旋转、缩放、移动或倾斜

translate（x,y）定义2D转换(水平垂直值)\*/

transform:translate(-50%,-50%);

}

新增：.div{

width:200px;height:100px;

background:yellow;

/\*rotate(angle) 定义旋转，在参数中规定（deg）角度。

transform:rotate(25deg);

}

top:0;left:0;right:0;bottom:0;margin:auto 需要有宽高

.box{

width:300px;

height:300px;

background: #ff0;

position: /\*fixed、absolute\*/

top:0;

left:0;

right:0;

bottom:0;

margin: auto;

}

Css三角

宽高为0 border-left 像素 实线 transparent

border-right 像素 实线 transparent

上三角 bottom 下三角 top

相反方向 左右三角 上下为transparent(透明)

代码演示：

.nav{

width: 0;

height: 0;

border-left: 5px solid green;

border-right: 5px solid transparent;

border-top: 5px solid transparent;

border-bottom: 5px solid transparent;

/\*border:10px solid transparent;

border-top: 10px solid green;\*/

}

布局

1. 圣杯布局 2.双飞翼 3.三列等高

圣杯布局：

1.将三者都 float:left , 再加上一个position:relative (因为相对定位后面会用到）

2.center部分 width:100%占满

3.此时center占满了，所以要把left拉到最左边，使用margin-left:-100%

4.这时left拉回来了，但会覆盖center内容的左端 要把center内容拉出来所以在外围box加上 padding:0 220px 0 200px

5. center内容拉回来了，但left也跟着过来了，所以要还原 就对left使用相对定位 left:-200px 同理，right也要相对定位还原 right:-220px

6.到这里大概就自适应好了。

如果想box高度保持一致可以给left center right都加上min-height:130px

双飞翼布局：

1.html代码将中间内容放在前面

2.将三者都浮动

3.将中间内容宽度设为100%

4.中间沾满，所以要把左边使用 margin-left:-100%,右边走margin-left:-220px；

5.将中间的增加一个div内层然后margin 右边宽度 左边宽度

三列等高

1.三个盒子没有高度给其浮动

2.第二步：给三个盒子写margin-bottom:-1000px padding-bottom:1000px

3.父盒子写overflow：hidden

4.给盒子内容

兼容

兼容是什么？各个浏览器显示的效果是不一样，各种内核的浏览器都有自己的标准，为了不使属性混淆，所以各家在各自标准前加了一个前缀

为什么出现兼容？因为浏览器内核不一样所以出现兼容

内核 谷歌、苹果 -webkit-

火狐 -moz

ie -trident –ms

欧朋 -o

常用兼容 css js

1.浮动写margin值 在ie6会出现双倍间距

解决办法 display：inline

-margin:值/2

2.图片默认间距 给浮动解决或者在一行设置

3.高度小于9px ie6会超出所写高度

overflow:hidden溢出隐藏解决

1. IE6下图片有空隙产生,怎么解决设置img 为display:block
2. png图片的透明效果ie6不支持 IE6下不支持fixed固定定位
3. 不同浏览器中默认文字大小不同

ie中body默认字大小 14px

FF中body默认字大小 16px

1. 不同浏览器中标签默认的边距和填充不同，解决此浏览器兼容性问题常用

\*{margin:0;padding:0;}

1. ul，ol列表的默认值左缩进是

IE默认为左缩进40px，通过ul、ol的margin属性设置。

FF默认为左缩进40px，通过ul、ol的padding属性设置。

dl无缩进，但其内部的说明元素dd默认缩进40px，而名称元素dt没有缩进。要清除列表样式，一般可以设置：

ul,ol,dd{

list-style-type:none; /\* 清除列表样式符号 \*/

margin-left:0; /\* 清除IE 浏览器左边界，即列表项缩进 \*/

padding-left:0; /\* 清除非IE 浏览器左边，即列表项缩进 \*/

}

8. 浏览器默认样式margin值：

FF默认为8px IE默认为10px

IE默认为10px，通过margin属性设置。

FF默认为8px,通过padding属性设置。

要清除页边距，一般应同时清除这两个属性值：

body{

padding:0; /\* 清除非IE浏览器页边距 \*/

margin:0; /\* 清除IE浏览器页边距 \*/

}

9.段间距

IE默认为10px，通过p的margin-top、margin-bottom属性设置。

FF默认为1.12em，通过p的margin-top、margin-bottom属性设置。

Csshack解决



-backgroud:blue

* “-″减号是IE6专有的hack

\*是ie7的专用

* “\9″ IE6/IE7/IE8/IE9/IE10都生效
* “\0″ IE8/IE9/IE10/ IE11都生效，是IE8/9/10/ IE11的hack
* “\9\0″ 只对IE8/IE9/IE10生效，是IE8/IE9/10的hack

注释的方式<!--[if IE 9 ]> <![endif]-->

Lt 小于 lte小于等于 gt大于 gte大于等于

例子：

<!--[if IE 6]>

<style>

.left{

width:500px;

height:200px;

background:#ff0;

}

</style>

<![endif]-->

css hack方法是：属性级Hack：比如IE6能识别减号“-”和星号“\*”，减号“-”和星号“\*”同时存在时只识别减号，IE7只能识别星号“\*”，但不能识别减号”- ”，而firefox两个都不能认识。

## Transition

## p{width:100px; background: red;height:40px; line-height: 40px;

## /\*过渡属性:（不填指的是所有属性，填写一个只变化一个）

## 花费的时间 速度曲线（匀速等） 从何时开始\*/

## transition: 2s ease-in-out 1s;}

## /\*鼠标滑过的效果变化\*/

## p:hover{width: 300px; font-size: 24px; background: #ff0; text-shadow:20px 20px 1px #000;}

## 

1. **transition实现的动画有什么特点**

CSS3 过渡是元素从一种样式逐渐改变为另一种的效果。

要实现这一点，必须规定两项内容：

（1）规定把效果添加到哪个 CSS 属性上

（2）规定效果的时长

1. **一个及多个过渡属性的设置**

下面的表格列出了所有的转换属性：（下面针对每一个属性都有详细的介绍）

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [transition](http://www.w3school.com.cn/cssref/pr_transition.asp" \o "CSS3 transition 属性) | 简写属性，用于在一个属性中设置四个过渡属性。 | 3 |
| [transition-property](http://www.w3school.com.cn/cssref/pr_transition-property.asp" \o "CSS3 transition-property 属性) | 规定应用过渡的 CSS 属性的名称。默认值all | 3 |
| [transition-duration](http://www.w3school.com.cn/cssref/pr_transition-duration.asp" \o "CSS3 transition-duration 属性) | 定义过渡效果花费的时间。默认是 0。 | 3 |
| [transition-timing-function](http://www.w3school.com.cn/cssref/pr_transition-timing-function.asp" \o "CSS3 transition-timing-function 属性) | 规定过渡效果的时间曲线。默认是 "ease"。 | 3 |
| [transition-delay](http://www.w3school.com.cn/cssref/pr_transition-delay.asp" \o "CSS3 transition-delay 属性) | 规定过渡效果何时开始。默认是 0。 | 3 |

**transition-property 属性规定应用过渡效果的 CSS 属性的名称。**

**提示：**过渡效果通常在用户将鼠标指针浮动到元素上时发生。

**注意：**请始终设置 [transition-duration](http://www.w3school.com.cn/cssref/pr_transition-duration.asp" \o "CSS3 transition-duration 属性) 属性，否则时长为 0，就不会产生过渡效果。

|  |  |
| --- | --- |
| **默认值：** | all |
| **继承性：** | no |
| **版本：** | CSS3 |

**语法:**

transition-property: none|all|*property*;

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 没有属性会获得过渡效果。 |
| all | 所有属性都将获得过渡效果。(长选) |
| *property* | 定义应用过渡效果的 CSS 属性名称列表，列表以逗号分隔。 |

单独的使用这个属性

Div{Transition-property:width}(单独使用这一个属性时在演示的时候不起作用，需要配合以下的所有属性一起使用)。

1. **动画时长**

transition-duration 属性规定完成过渡效果需要花费的时间（以秒或毫秒计）。

|  |  |
| --- | --- |
| **默认值：** | 0 |
| **继承性：** | no |
| **版本：** | CSS3 |

**语法**

transition-duration: *time*;

|  |  |
| --- | --- |
| **值** | **描述** |
| *time* | 规定完成过渡效果需要花费的时间（以秒或毫秒计）。  默认值是 0，意味着不会有效果。 |

单独的使用这个属性

Div{Transition-duration:0.5s}(单独使用这一个属性时在演示的时候不起作用，需要配合其他属性一起使用)。

案例：鼠标放上，盒子宽度变长

<style>

div

{

width:100px;

height:100px;

background:blue;

transition-property: width;

transition-duration: 2s;

-moz-transition-property: width; /\* Firefox 4 \*/

-moz-transition-duration: 2s; /\* Firefox 4 \*/

}

div:hover

{

width:300px;

}

</style>

</head>

<body>

<div></div>

<p>请把鼠标指针移动到蓝色的 div 元素上，就可以看到过渡效果。</p>

<p><b>注释：</b>本例在 Internet Explorer 中无效。</p>

1. **动画的速度曲线**

transition-timing-function 属性规定过渡效果的速度曲线。

该属性允许过渡效果随着时间来改变其速度。

|  |  |
| --- | --- |
| **默认值：** | ease |
| **继承性：** | no |
| **版本：** | CSS3 |

## 语法

transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|cubic-  
bezier(*n*,*n*,*n*,*n*);

|  |  |
| --- | --- |
| **值** | **描述** |
| linear | 规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）。 |
| ease | 规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）。 |
| ease-in | 规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。 |
| ease-out | 规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）。 |
| ease-in-out | 规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）。 |
| cubic-bezier(*n*,*n*,*n*,*n*) | 在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值。（了解即可，不做过多要求） |

linear ：匀速   
ease-in：减速   
ease-out：加速   
ease-in-out：先加速再减速   
cubic-bezier：三次贝塞尔曲线，可以定制

**提示：**请在实例中测试不同的值，这样可以更好地理解它们的工作原理。

案例：规定过渡效果的速度曲线

<style>

div

{

width:100px;

height:100px;

background:blue;

transition:width 2s;

transition-timing-function:ease-in-out;

/\* Firefox 4 \*/

-moz-transition:width 2s;

-moz-transition-timing-function:linear;

}

div:hover

{

width:300px;

}

</style>

</head>

<body>

<div></div>

<p>请把鼠标指针移动到蓝色的 div 元素上，就可以看到过渡效果。</p>

<p><b>注释：</b>本例在 Internet Explorer 中无效。</p>

</body>

</html>

1. **动画的延时**

**定义和用法**

transition-delay 属性规定过渡效果何时开始。

transition-delay 值以秒或毫秒计。

|  |  |
| --- | --- |
| **默认值：** | 0 |
| **继承性：** | no |
| **版本：** | CSS3 |

**语法**

transition-delay: *time*;

|  |  |
| --- | --- |
| **值** | **描述** |
| *time* | 规定在过渡效果开始之前需要等待的时间，以秒或毫秒计。 |

1. **Transition的出发情况：**

触发分为伪类触发 比如 :hover : focus  :checked  :active  js触发，咱们这里只是练习对鼠标滑过时的出发

透明度

Html4+css2 pc网页 html5+css3 手机

Html5比html4 新增了：

1. 新增语义的特性（利于搜索引擎搜到）
   1. 主体结构（布局）

Article （分小块） section（分块） aside（侧边）

nav(导航) time（时间日期）（time是行内元素）

* 1. 非主体结构 (文字)

Header（头部） footer （底部） address （信息 电话号码） hgroup （h1-h6的组合） main 主体

1. 本地存储特性 webstorage

localStorage 长时间存储(没有时间限制的数据存储)

Sessionstorage 短时间存储

1. 音频 audio src 音乐地址 （无宽高）

autoplay 自动播放 loop 循环播放 controls 控制

1. 视频 video src 视频地址（可设置宽高）

Width height controls 控制

autoplay 自动播放 loop 循环播放

1. Canvas 在浏览器对其进行操作了
2. 取消些过时的标记 （font center u s ）
3. 全新的表单控件
   1. input框autofocus 自动获得焦点
   2. search 搜索框
   3. tel 电话号码框

例子：电话：</label><input type="tel" pattern="[0-9]{11}">

pattern="[0-9]{11}"正则表达式[0-9]0-9的数字{11}11位电话号码

* 1. url 网址地址框
  2. email 邮箱地址 有验证功能
  3. number 数字 只能写数字 step=“2”每次加的数值 max=“20” 最大值

min=“0”最小值

* 1. color 颜色
  2. file 上传文件 multiple=“multiple”实现上传多个文件的属性和属性值
  3. date，month 月 , week 周 , time , datetime-local 年月日时分秒 , 时间日期

required="required" 不写东西无法实现提交（文本框搜索框等的属性）

webkit表单输入框placeholder的颜色值改变：  
如果想要默认的颜色显示红色，代码如下：  
input::-webkit-input-placeholder{color:red;}

cmd的简单命令：

1.ipconfig 查看ip地址

2.上箭头回到上一次输入的东西

3.复制过来的东西我们需要用鼠标右键 点击粘贴

1. 复制cmd里面的内容，直接拉， 标记一下
2. Ctrl+c 退出
3. node空格-v 检查node的版本

Nodejs 用法：

npm install 是用来下载东西的

第一步：npm install http-server -g

第二步：找到文件夹 打开cmd 运行 http-server

第三步：http-server -p 6363（端口号）如果端口被占用，可以换端口

弹性盒：

display:flex: (弹性盒 子元素水平排列 写给父盒子)

父盒子弹性盒display: flex;作用是可以让父盒子设置弹性盒而子盒子水平排列

flex:1 （子元素的平均分布 写给子元素）

flex-direction: column; （弹性盒垂直） 必须配合弹性盒

display: flex;默认水平排列，弹性盒垂直排列flex-direction: column;写给父元素

justify-content:center 水平居中

想让子元素水平方向写在父盒子

align-items:center 垂直居中

justify-content: space-between;左中右分

背景图的添加：

<style type="text/css">

\*{

margin: 0;

padding: 0;

}

body,html{

width: 100%;

height: 100%;

}

.box{

width: 100%;

height: 100%;

background: url(../4-26-3/img/ad.jpg) no-repeat;

background-size: cover;

}

</style>

<div class="box">

</div>