**注释：**在一个文档中，不能出现一个以上的 <main> 元素。

<main> 元素不能是以下元素的后代：<article>、<aside>、<footer>、<header> 或 <nav>。

<main>:主要内容

<header>：标签定义文档的页眉（介绍信息）在一个文档中，您可以定义多个标签元素。

**注释：**<header> 标签不能被放在 <footer>、<address> 或者另一个 <header> 元素内部。

<nav>:导航栏

<article>：独立的自包含内容

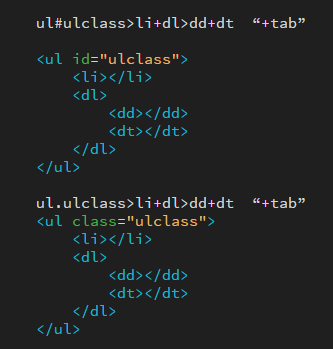
<footer>：页脚通常包含文档的作者、版权信息、使用条款链接、联系信息等等

<dl>+<dt>：描述

例如：姓名：徐涛

## hr水平分割线

Boostrap前端框架



Input：text + Tab



# HTML\_\_Day1

### HTML骨架：！+Tap

1. html: 5

2. !

然后按下tab键盘。

<!DOCTYPE html>: 标签位于文档的最前面，用于向浏览器说明当前文档使用哪种 HTML 或 XHTML 标准规范

### 清空边框

<style>

html, body, div, span, applet, object, iframe,

h1, h2, h3, h4, h5, h6, p, blockquote, pre,

a, abbr, acronym, address, big, cite, code,

del, dfn, em, img, ins, kbd, q, s, samp,

small, strike, strong, sub, sup, tt, var,

b, u, i, center,

dl, dt, dd, ol, ul, li,

fieldset, form, label, legend,

table, caption, tbody, tfoot, thead, tr, th, td,

article, aside, canvas, details, embed,

figure, figcaption, footer, header, hgroup,

menu, nav, output, ruby, section, summary,

time, mark, audio, video {

margin: 0;

padding: 0;

border: 0;

font-size: 100%;

font: inherit;

vertical-align: baseline;

}

/\* HTML5 display-role reset for older browsers \*/

article, aside, details, figcaption, figure,

footer, header, hgroup, menu, nav, section {

display: block;

}

body {

line-height: 1;

}

ol, ul {

list-style: none;

}

blockquote, q {

quotes: none;

}

blockquote:before, blockquote:after,

q:before, q:after {

content: '';

content: none;

}

table {

border-collapse: collapse;

border-spacing: 0;

}

</style>

### 快速写标签：字母+tap键

utf-8是目前最常用的字符集编码方式，常用的字符集编码方式还有gbk和gb2312。

HTML提供了6个等级的标题，即<h1>、<h2>、<h3>、<h4>、<h5>和<h6>

注意： h1 标签因为重要，尽量少用，不要动不动就向你扔了一个h1。 一般h1 都是给logo使用，或者页面中最重要标题信息。

段落标签<p> 文本内容 </p>

Shift+alt+向下键 ：同时输入

<hr />是单标签：在网页中显示默认样式的水平线。

<br />换行标签

“内容+内边距+外边距+边框”组成的，可以通过设定盒子的border、padding和margin

### Div、span标签(重点)

Div、span 是没有语义的 是我们网页布局主要的2个盒子 css+div

div 就是 division 的缩写 分割， 分区的意思 其实有很多div 来组合网页。

span, 跨度，跨距；范围

注意：div一行只能一个，而span一行可以多个



B、i、s、u 只有使用 没有 强调的意思 而strong、em、del、ins语义更强烈

### 图片标签 img

<Img src=”图像” />



注意：使用width或者height时，只要给一个，图片将自动等比例缩放。

### 链接标签(重点) href

<a href="跳转目标" target="目标窗口的弹出方式">文本或图像名称</a>

注意：

Target有两种方式：self为默认值，\_blank为在新窗口中打开。

1.外部链接 需要添加 http:// www.baidu.com

2.内部链接 直接链接内部页面名称即可 比如 < a href="index.html"> 首页 </a >

3.如果当时没有确定链接目标时，通常将链接标签的href属性值定义为“#”(即href="#")，表示该链接暂时为一个空链接。

4.不仅可以创建文本超链接，在网页中各种网页元素，如图像、表格、音频、视频等都可以添加超链接。

text-decoration:none;无下划线

### 锚点定位href和 id（难点）

通过创建锚点链接，用户能够快速定位到目标内容。

创建锚点链接分为两步：

1.使用“a href=”#id名>“链接文本"</a>创建链接文本（被点击的）

<a href="#two">点击名称 < /a>

2.使用相应的id名标注跳转目标的位置。

<h3 id="two">连接名</h3>

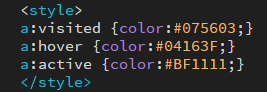
**Css中鼠标点击样式：**

link: 未访问的链接

visited: 访问过之后

hover: 鼠标放到链接上

active: 链接被按下



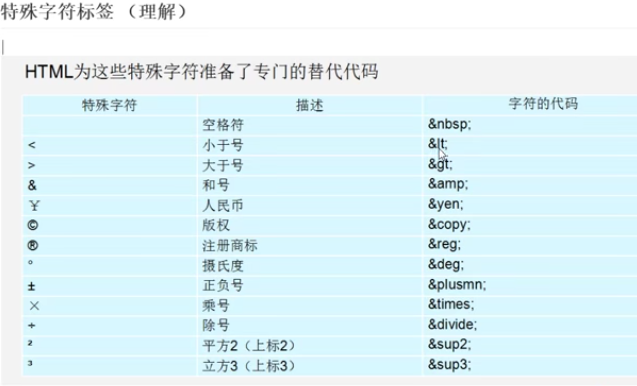
base 标签：<base />

base 可以设置整体链接的打开状态

base 写到 <head> </head> 之间

把所有的连接都默认添加 例如： <base target="\_blank"/> （所有连接都添加弹出窗口显示）

### 特殊符号 （网页显示，不解析）



注意： ； 也要写上去。

### 注释标签(<!-- 注释内容 -->)

只是快捷键 ctrl+/

### 路径

可以分为： 相对路径和绝对路径

## 相对路径

以引用文件之网页所在位置为参考基础，而建立出的目录路径。因此，当保存于不同目录的网页引用同一个文件时，所使用的路径将不相同，故称之为相对路径。

1. 图像文件和HTML文件位于同一文件夹：只需输入图像文件的名称即可，如&lt;img src="logo.gif" /&gt;。

2. 图像文件位于HTML文件的下一级文件夹：输入文件夹名和文件名，之间用“/”隔开，如img src="img/img01/logo.gif。

3. 图像文件位于HTML文件的上一级文件夹：在文件名之前加入“../” ，如果是上两级，则需要使用 “../ ../”，以此类推，如&lt;img src="../logo.gif" /&gt;。

## 绝对路径

绝对路径以Web站点根目录为参考基础的目录路径。之所以称为绝对，意指当所有网页引用同一个文件时，所使用的路径都是一样的

“D:\web\img\logo.gif”，或完整的网络地址，例如“http://www.itcast.cn/images/logo.gif”。

### 列表标签

容器里面装载着文字或图表的一种形式，叫列表。

### 无序列表ul

无序列表的各个列表项之间没有顺序级别之分，是并列的。其基本语法格式如下：

<ul>

<li>列表项1</li>

<li>列表项2</li>

<li>列表项3</li>

......

</ul>

注意：

1. <ul></ul>中只能嵌套<li></li>，直接在<ul></ul>标签中输入其他标签或者文字的做法是不被允许的。

2. <li>与</li>之间相当于一个容器，可以容纳所有标签元素。

3. 无序列表会带有自己样式属性，放下那个样式，一会让CSS来！

### 有序列表ol

<ol>

 <li>列表项1</li>

<li>列表项2</li>

<li>列表项3</li>

......

</ol>

所有特性基本与ul 一致。

### 自定义列表dl

定义列表常用于对术语或名词进行解释和描述，定义列表的列表项前没有任何项目符号。其基本语法如下：

<dl>

 <dt>名词1</dt>

<dd>名词1解释1</dd>

<dd>名词1解释2</dd>

...

<dt>名词2</dt>

<dd>名词2解释1</dd>

<dd>名词2解释2</dd>

...

</dl>

### 表格 table

表格的现在还是较为常用的一种标签，但不是用来布局，常见处理、显示表格式数据。

### 创建表格

在HTML网页中，要想创建表格，就需要使用表格相关的标签。创建表格的基本语法格式如下：

<table>

<tr> <!--行标签 -->

<td>单元格内的文字</td> <!--行标签里的单元格内容 -->

<td>单元格内的文字</td>

...

</tr>

...

</table>

注意：

1. <tr></tr>中只能嵌套<td></td>

2. <td></td>标签，他就像一个容器，可以容纳所有的元素

### 表格属性

写在 <table 属性名 = “属性值” >

<tr>

<td>单元格</td>

<tr>

</table>

表格好看：三参为零（border、cellspacing、cellspacing）

若<tr align = “center” >则单元格内容居中。

### 表头th

<th>单元格内容 </th> <!-- 等于td加粗并居中 -->

### 表格标题caption（在table内第一个）

<caption> </caption>

### 表格的头部内容thead (在table内)

<thead> </thead>

### 表格的身体内容 tbody (在table内)

<tbody> </tbody>

### 并单元格(难点)

跨行合并：rowspan 跨列合并：colspan

合并顺序：先上后下，先左后右

例：<tr>

<td colspan = “2”>内容</td>

<tr/>

合并单元格数，再删除被合并的单元格，不删除的话会被挤到其他位置。

注意：表格中没有列元素，列的个数取决于行的单元格个数。

### 表单标签

在HTML中，一个完整的表单通常由表单控件（也称为表单元素）、提示信息和表单域3个部分构成。

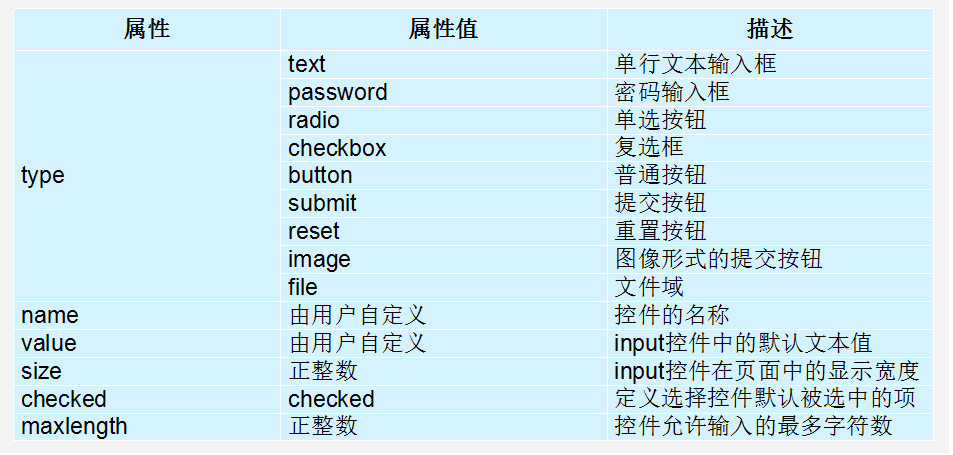
表单控件：包含了具体的表单功能项，如单行文本输入框、密码输入框、复选框、提交按钮、重置按钮等。

提示信息：一个表单中通常还需要包含一些说明性的文字，提示用户进行填写和操作。

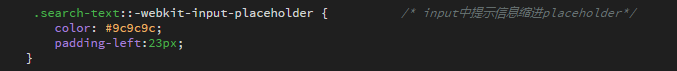
表单域：他相当于一个容器，用来容纳所有的表单控件和提示信息，可以通过他定义处理表单数据所用程序的url地址，以及数据提交到服务器的方法。如果不定义表单域，表单中的数据就无法传送到后台服务器。

### input控件（重点）

在上面的语法中，&lt;input /&gt;标签为单标签，type属性为其最基本的属性，其取值有多种，用于指定不同的控件类型。除了type属性之外，&lt;input /&gt;标签还可以定义很多其他的属性，其常用属性如下表所示。



placeholder="请输入手机号码"

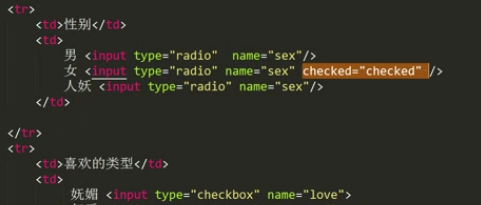


text-indent:10px;控制文字输入位置



type属性说明了是那种表单类型。

radio属性 如果是一组，必须给他一个相同的名字name。



## Button标签

<button type="button">Click Me!</button>

<button> 控件 与 <input type="button"> 相比，提供了更为强大的功能和更丰富的内容

### 表单域 form

在HTML中，form标签被用于定义表单域，即创建一个表单，以实现用户信息的收集和传递，form中的所有内容都会被提交给服务器。创建表单的基本语法格式如下：

<form action="url地址" method="提交方式" name="表单名称">

各种表单控件

</form>

注意：当有表单域后，submit提交按钮、reset重置按钮将起作用。

1. Action

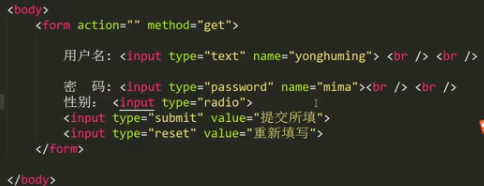
在表单收集到信息后，需要将信息传递给服务器进行处理，action属性用于指定接收并处理表单数据的服务器程序的url地址。

2. method

用于设置表单数据的提交方式，其取值为ge（看见密码）t或post（看不见账户密码）。

3. name

用于指定表单的名称，以区分同一个页面中的多个表单。



### 查文档

经常查阅文档是一个非常好的学习习惯。

W3C : http://www.w3school.com.cn/

MDN: <https://developer.mozilla.org/zh-CN/>

## HTML contenteditable 属性

可编辑的段落。

<p contenteditable="true">这是一段可编辑的段落。请试着编辑该文本。</p>

# CSS\_\_Day2

Color颜色、font-size字体大小

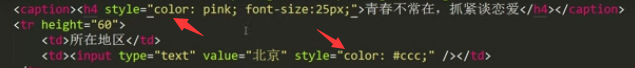
### 行内式（内联样式）

内联样式，又有人称行内样式、行间样式、内嵌样式。

是通过标签的style属性来设置元素的样式，其基本语法格式如下：

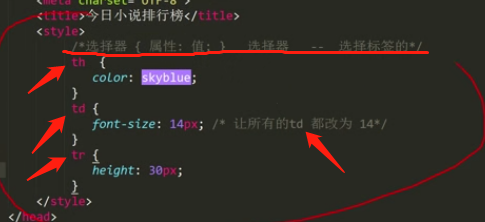
<标签名 style="属性1:属性值1; 属性2:属性值2; 属性3:属性值3;"> 内容</标签名>

内容语法中style是标签的属性，实际上任何HTML标签都拥有style属性，用来设置行内式。其中属性和值的书写规范与CSS样式规则相同，行内式只对其所在的标签及嵌套在其中的子标签起作用。



### 内部样式表 （标签选择器）

内嵌式是将CSS代码集中写在HTML文档的head头部标签中，并且用style标签定义，其基本语法格式如下： html语法中，style标签一般位于head标签中title标签之后，也可以把他放在HTML文档的任何地方。 type="text/CSS" 在html5中可以省略， 写上也比较符合规范， 所以这个地方可以写也可以省略。



### 外部样式表（外链式）推荐使用

链入式是将所有的样式放在一个或多个以.CSS为扩展名的外部样式表文件中，通过link标签将外部样式表文件链接到HTML文档中，其基本语法格式如下：

html 注意： link 是个单标签! 该语法中，link标签需要放在head头部标签中，并且必须指定link标签的三个属性，具体如下：

<head>

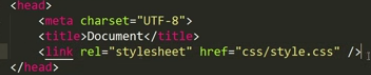
<link href = “css文件路径” rel = ”当前文档与被链接的关系”/>

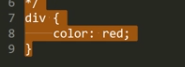
</head>

href：定义所链接外部样式表文件的URL，可以是相对路径，也可以是绝对路径。

type：定义所链接文档的类型，在这里需要指定为“text/CSS”，表示链接的外部文件为CSS样式表。

rel：定义当前文档与被链接文档之间的关系，在这里需要指定为“stylesheet”，表示被链接的文档是一个样式表文件。

html文档内容。

style.css文档内容。

### 三种样式表总结

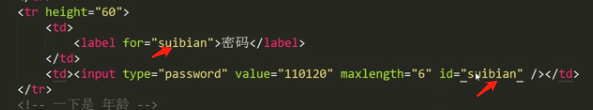


### Lable标签

<label> 标签为 input 元素定义标注（标记）。

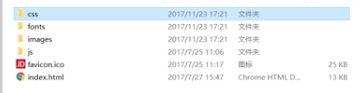
label 元素不会向用户呈现任何特殊效果。不过，它为鼠标用户改进了可用性。如果您在 label 元素内点击文本，就会触发此控件。就是说，当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

<label> 标签的 for 属性应当与相关元素的 id 属性相同。



案例：点击密码，鼠标光标转跳到密码后面填写栏控件

### 网页新建文件



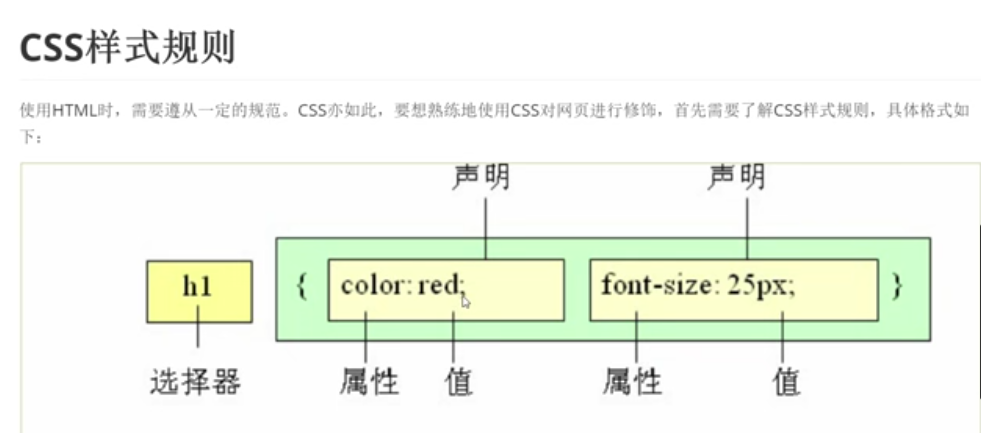
### Css基础选择器

标签选择器能快速为页面中同类型的标签统一样式，同时也是缺点。

格式如下：

标签名{属性1：属性值1；属性2：属性值2}或者

元素名{元素1：元素值1；元素2：元素值2}



标签选择器：可以把某一类标签全部选择出来div span

### 类选择器（使用最多）

上面点声明，下面class调用。

可以选择一个或者多个标签。



.名称：1、长名称或词组可以使用中横线来为选择器命名。

2、不建议使用“\_”下划线来命名css选择器。

# CSS\_Day3

### 多类名选择器（属于类选择器）

Class = “自定义标签1 在自定义标签2”

1、样式显示效果跟HTML元素中的类名先后顺序没有关系,受CSS样式书写的上下顺序有关。

2.、各个类名中间用空格隔开。



类名选择器：<div class=”nav”>这个div的名字



### ID选择器（一般在js中使用）

 id选择器使用“#”进行标识，后面紧跟id名，其基本语法格式如下：

#id名{属性1:属性值1; 属性2:属性值2; 属性3:属性值3; }

该语法中，id名即为HTML元素的id属性值，大多数HTML元素都可以定义id属性，元素的id值是唯一的，只能对应于文档中某一个具体的元素。 用法基本和类选择器相同。 # id选择器和类选择器区别 ：W3C标准规定在同一个页面内，不允许有相同名字的id对象出现，但是允许相同名字的class。（使用次数的区别）

### 通配符选择器（使用较少）

\*代表所有标签的意思

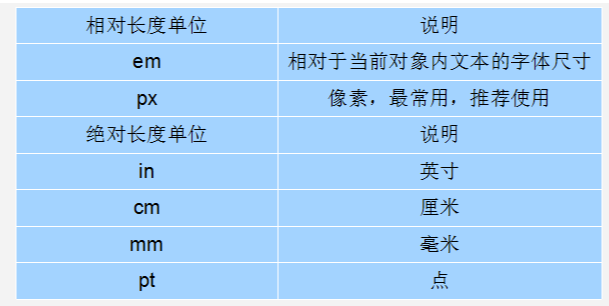
所有选择器中作用范围最广的，能匹配页面中所有的元素。其基本语法格式如下：

\* { 属性1:属性值1; 属性2:属性值2; 属性3:属性值3; } 例如下面的代码，使用通配符选择器定义CSS样式，清除所有HTML标记的默认边距。

### Css字体样式属性

### font-size(字号)

font-size:字号大小 font-size属性用于设置字号，该属性的值可以使用相对长度单位，也可以使用绝对长度单位。其中，相对对长度单位比较常用，推荐使用像素单位px，绝对长度单位使用较少，具体如下：



### font-family（字体）

格式：font-family：“字体”



常用技巧：

1. 现在网页中普遍使用14px+。

2. 尽量使用偶数的数字字号。ie6等老式浏览器支持奇数会有bug。

3. 各种字体之间必须使用英文状态下的逗号隔开。

4. 中文字体需要加英文状态下的引号，英文字体一般不需要加引号。当需要设置英文字体时，英文字体名必须位于中文字体名之前。

5. 如果字体名中包含空格、#、$等符号，则该字体必须加英文状态下的单引号或双引号，例如font-family: "Times New Roman";。

6. 尽量使用系统默认字体，保证在任何用户的浏览器中都能正确显示。

7.我们尽量使用宋体和微软雅黑中文字体，为了照顾不同电脑字体安装问题。

注意：当有很多字体时，体统会先看第一个字体是否支持，若支持则使用第一个字体，若不支持看第二个，若全不支持则使用系统默认字体。

CSS Unicode字体 在 CSS 中设置字体名称，直接写中文是可以的。但是在文件编码（GB2312、UTF-8 等）不匹配时会产生乱码的错误。xp 系统不支持 类似微软雅黑的中文。

方案一： 你可以使用英文来替代。 比如 font-family:"Microsoft Yahei"。

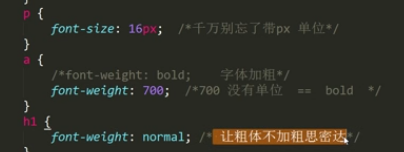
方案二： 在 CSS 直接使用 Unicode 编码来写字体名称可以避免这些错误。

使用 Unicode 写中文字体名称，浏览器是可以正确的解析的。 font-family: "\5FAE\8F6F\96C5\9ED1"，表示设置字体为“微软雅黑”

### font-weight（字体粗细）

font-weight属性用于定义字体的粗细，其可用属性值：normal（正常）、bold（粗体）、bolder(特粗体)、lighter（细体）、100~900（100的整数倍）。

小技巧： css 数字 400 等价于 normal，而 700 等价于 bold。 但是我们更喜欢用数字来表示。



#### 绝对定位

style="position: fixed”

### font-style（字体倾斜i或em）

font-style:字体风格。字体倾斜除了用 i 和 em 标签之外，可以使用CSS 来实现，但是CSS 是没有语义的。

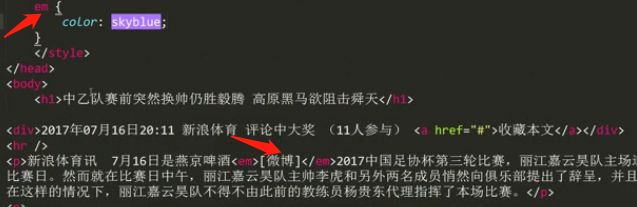
font-style属性用于定义字体风格，如设置斜体、倾斜或正常字体，其可用属性值如下：

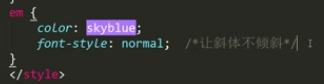
normal：默认值，浏览器会显示标准的字体样式。

italic：浏览器会显示斜体的字体样式。

oblique：浏览器会显示倾斜的字体样式。

小技巧：平时我们很少给文字加斜体，反而喜欢给斜体标签（em，i）改为普通模式。





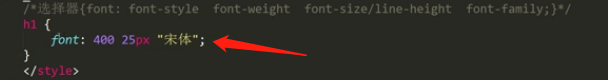
### font:综合设置字体样式 (重点连写)

font属性用于对字体样式进行综合设置，其基本语法格式如下：

css 选择器{font: font-style font-weight font-size/line-height行距） font-family;}

使用font属性时，必须按上面语法格式中的顺序书写，不能更换顺序，各个属性以空格隔开。

注意：其中不需要设置的属性可以省略（取默认值），但必须保留font-size和font-family属性，否则font属性将不起作用。



### CSS外观属性

#### color:文本颜色

#FF（红色）FF（绿色）FF（蓝色）

color属性用于定义文本的颜色，其取值方式有如下3种：

1. 预定义的颜色值，如红色（#FF0000），绿色（#00FF00），蓝色（#00FF00）黑色（#000000）白色（#FFFFFF）等。

缩写：必须两两相同才能缩写。 #F（红色）0（绿色）0（蓝色）--红色。

2.十六进制（最常用），如#FF0000，#FF6600，#29D794等。实际工作中，十六进制是最常用的定义颜色的方式。

3.RGB代码，如红色可以表示为rgb(255,0,0)或rgb(100%,0%,0%)。

需要注意的是，如果使用RGB代码的百分比颜色值，取值为0时也不能省略百分号，必须写为0%。

#### letter-spacing：字间距

letter-spacing：常用的属性值单位有分别为像素px，相对值em。

#### Line-height：行间距

 Line-height属性用于设置行间距，就是行与行之间的距离，即字符的垂直间距，一般称为行高。

line-height常用的属性值单位有三种，分别为像素px，相对值em和百分比%，使用最多的是像素px 一般情况下，行距比字号大7.8像素左右就可以了。

Line-height:100px

#### text-align**：**水平对齐方式

text-align属性用于设置文本内容的水平对齐，相当于html中的align对齐属性。

其可用属性值如下： left：左对齐（默认值） right：右对齐 center：居中对齐 （让盒子里面的内容居中对齐，而不是让盒子）

#### text-indent：首行缩进

text-indent属性用于设置首行文本的缩进，其属性值可为不同单位的数值、em字符宽度的倍数、或相对于浏览器窗口宽度的百分比%，允许使用负值。建议使用em作为设置单位。 1em 就是一个字的宽度 如果是汉字的段落， 1em 就是一个汉字的宽度。

Text-indent : 2em <!--首行缩进两个字-->

#### text-decoration：文本的装饰（取消格式、下划线、贯穿线、上划线、闪烁）

通常我们用于给链接修改装饰效果

none |：取消格式，定义标准的文本。(可以取消倾斜、加粗、删除线等格式)

underline ：下划线 也是我们链接自带的

overline：上划线。

line-through：删除线。（<del></del>）

开发者工具（chrome） 此工具是我们的必备工具，以后代码出了问题，我们首先第一反应就是： “按F12”或者是 “shift+ctrl+i” 打开 开发者工具。

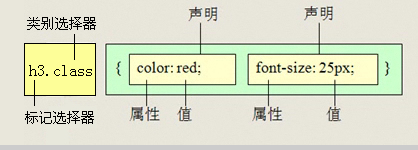
开发者工具（chrome） 此工具是我们的必备工具，以后代码出了问题，我们首先第一反应就是： “按F12”或者是 “shift+ctrl+i” 打开 开发者工具。

### CSS复合选择器

复合选择器是由两个或多个基础选择器，通过不同的方式组合而成的,目的是为了可以选择更准确更精细的目标元素标签。

#### 交集选择器h3.class（用的较少）

交集选择器由两个选择器构成，其中第一个为标签选择器，第二个为class选择器，两个选择器之间不能有空格。



 \*\*记忆技巧：\*\* 交集选择器 是 并且的意思。 即...又...的意思， 比如： p.one 选择的是： 类名为 .one 的 段落p标签。

例如： h3.class{color：red}

#### 并集选择器p,.one, #test(重点)

并集选择器（CSS选择器分组）是各个选择器通过逗号连接而成的，任何形式的选择器（包括标签选择器、class类选择器id选择器等），都可以作为并集选择器的一部分。如果某些选择器定义的样式完全相同，或部分相同，就可以利用并集选择器为它们定义相同的CSS样式。

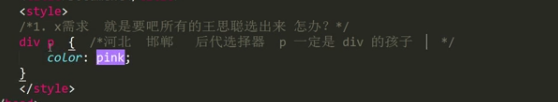
 记忆技巧： 并集选择器和的意思是说，只要逗号隔开的，所有选择器都会执行后面样式。

例如：p,.one, #test {color: #F00;} 表示 .one 和 p 和 #test 这三个选择器都会执行颜色为红色。 通常用于集体声明。 他和他，在一起，在一起，一起的意思。

#### 后代选择器ul li

后代选择器又称为包含选择器，用来选择元素或元素组的后代，其写法就是把外层标签写在前面，内层标签写在后面，中间用空格分隔。当标签发生嵌套时，内层标签就成为外层标签的后代。子孙后代都可以这么选择。 或者说，它能选择任何包含在内的标签。

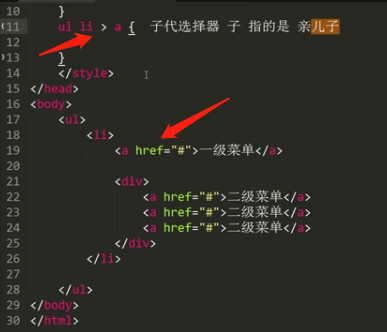
例如：ul li {color:red} ul里面的li的格式变为红色



#### 子元素选择器 >（子代选择器）

子元素选择器只能选择作为某元素子元素的元素。其写法就是把父级标签写在前面，子级标签写在后面，中间跟一个 > 进行连接，注意，符号左右两侧各保留一个空格。 白话： 这里的子 指的是 亲儿子 不包含孙子 重孙子之类。

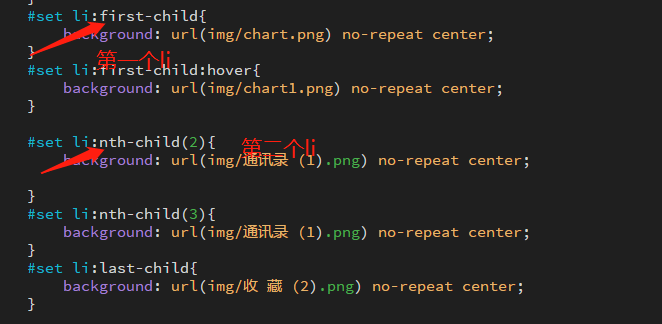
比如： .demo > h3 {color: red;} 说明 h3 一定是demo 亲儿子。 demo 元素包含着h3。



## 伪类选择器

类选择器是一个点，例如：.demo{} ；而我们的伪类用两个点（冒号），例如： ：link{}

### 链接为类选择器

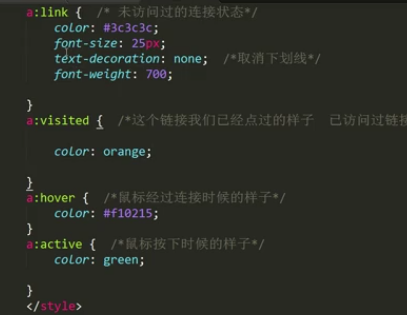


: link未访问的链接

: visited访问过之后的链接 （不能改变背景色）

: hover 鼠标放到链接上

: active 链接被按下 **顺序不能变**



## 显示模式（display）

HTML标签一般分为块标签和行标签。

#### 块级元素(block)

每个块元素通常都独占一行或者多行，可以对其设置宽度、高度、对其等属性。

1. 总是从新行开始
2. 高度，行高、外边距以及内边距都可以控制。
3. 宽度默认是容器的100%
4. 可以容纳内联元素和其他块元素

常见的块元素：<h1>~<6>、<p>、<div>、<ul>、<ol>、<li>等

#### 行内元素(inline)

1. 一行可以有多个元素。
2. 高宽无效
3. 默认宽度是本身内容宽度
4. 行内元素只能容纳文本或其他行内元素(a特殊)

常用行内元素：<a>、<strong>、<b>、<em>、<i>、<del>、<s>、<ins>、<u>、<span>。

#### 行内块元素（inline-block）

在行内元素中有几个特殊的标签——<img />、<input />、<td>，可以对它们设置宽高和对齐属性，有些资料可能会称它们为行内块元素。

行内块元素的特点：

（1）和相邻行内元素（行内块）在一行上,但是之间会有空白缝隙。

（2）默认宽度就是它本身内容的宽度。

（3）高度，行高、外边距以及内边距都可以控制。

#### 标签显示模式转换 （display）

块转行内：display : inline;

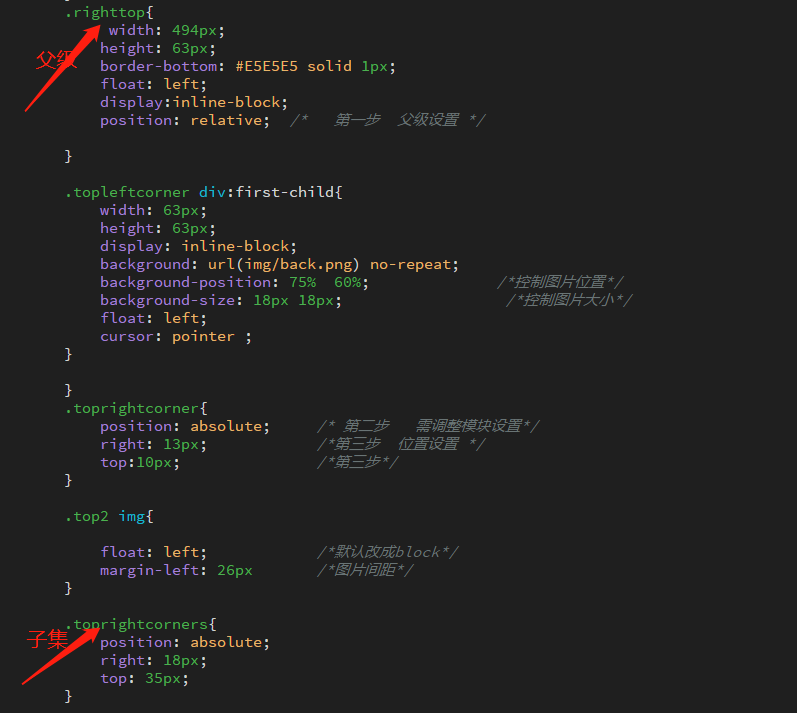
行内转块：display : block;

块、行内元素转换为行内块： display: inline-block;

此阶段，我们只需关心这三个，其他的是我们后面的工作。



## 相对位置（重点）



### 绝对定位（重点）

position: fixed

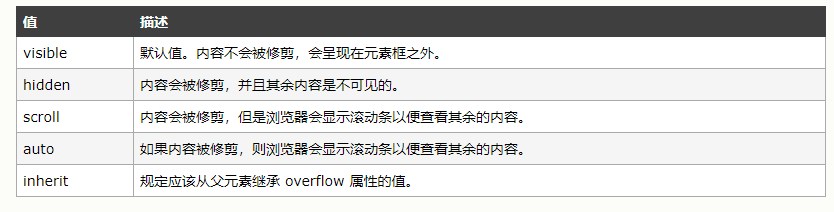
#### 半透明

background: rgba(106,106,106,0.9)

#### 滚动条overflow:auto

#### 鼠标箭头(手)cursor: pointer ;

#### 溢出元素框overflow·



## Overflow：hidden子div全是浮动，外面的框子也能有体积。

#### 背景图片

渐变色background-image: linear-gradient(to right,#ff9000 0,#ff5000 100%);

background-image: linear-gradient(to top, #7A88FF, #7AFFAF);  
等价于：  
 background-image:linear-gradient(0deg, #7A88FF, #7AFFAF);

background-size:cover;

把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。

背景图像的某些部分也许无法显示在背景定位区域中。

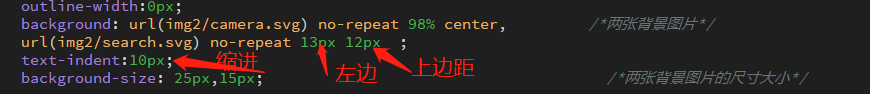
background-size: contain;

把图像图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。

background-size: 100% 100%;

background-size: 50px 50px;

两张背景图运用：



# 注意: inline-block/inline和float

1、inline-block/inline不能使用margin

2、inline-block/inline中间有空白

3、inline-block/inline 不能设置高宽位置

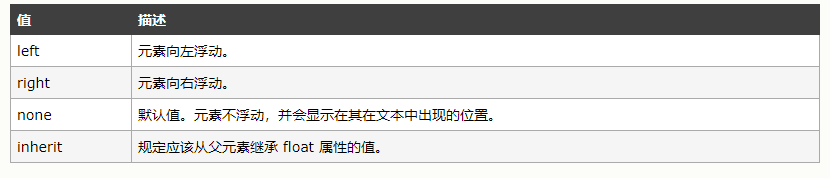
4、添加float自动变成块级框。

5、[float了的元素和内联元素不支持margin：auto](https://www.cnblogs.com/freddyhuang/p/5608698.html) ；text-align: center;

6、margin：百分比%（前面所有的父级一定也要是百分比才有效果）

7、h、p不是内行元素

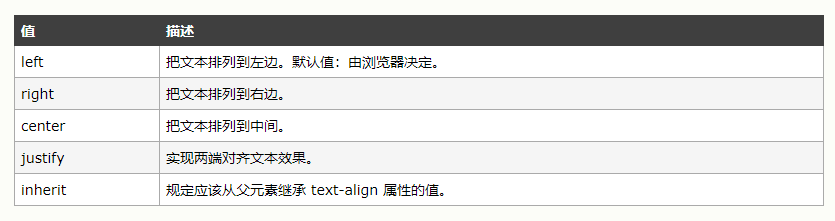
### 浮动float



注意：Flot : right/left/none/inherit 才能margin-right

### 设置行高Line-height

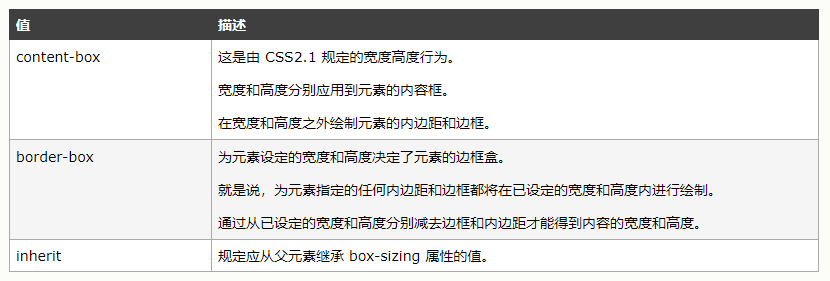
### 设置文本位置text-align



## Display用法



## Box-sizing 内边距和边框设置



## border- radius 边框圆角

边框右上角圆角border-top-right-radius：4px；

## Input外边框

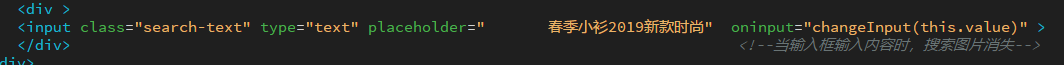


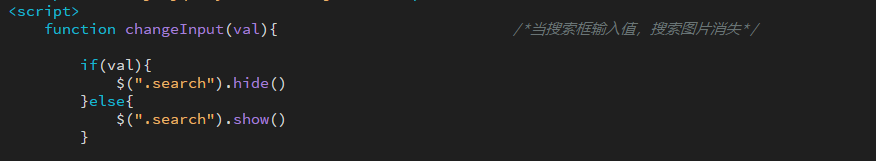
## text-overflow: ellipsis; 多于字数变成省略号

## background : url(img/sou.png) no-repeat ;背景图片不重复

## opacity:0.5透明度

## 当输入框输入内容时，搜索图片消失





## input中输入内容和提示信息缩进text-indent: 2px;

## input标签禁止输入

1. disabled 属性规定应该禁用 input 元素，被禁用的 input 元素，不可编辑，不可复制，不可选择，不能接收焦点,后台也不会接收到传值。设置后文字的颜色会变成灰色。disabled 属性无法与 <input type="hidden"> 一起使用。  
   示例：<input type="text" disabled="disabled" />
2. readonly 属性规定输入字段为只读可复制，但是，用户可以使用Tab键切换到该字段，可选择,可以接收焦点，还可以选中或拷贝其文本。后台会接收到传值. readonly 属性可以防止用户对值进行修改。

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

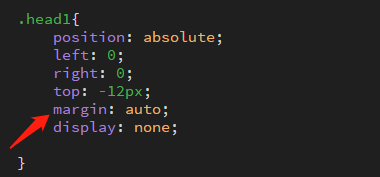
1. readonly unselectable="on" 该属性跟disable类似，input 元素，不可编辑，不可复制，不可选择，不能接收焦点，设置后文字的颜色也会变成灰色，但是后台可以接收到传值。示例：<input type="text"  readonly  unselectable="on" >（还是菜鸟一枚，写的不对的地方欢迎指正！）

## input中提示信息缩进placeholder

### 更改属性



## 绝对定位居中



## A标签不继承父级颜色字体

a{color:inherit; font:inherit;}解决方法

### img标签cursor: pointer（鼠标变手失效）须使用：



### 一个声明中设置所有的列表属性

list-style: none;

## 更改图片透明度opacity

opacity:0.8;

/\* 透明度设置后，可以看到后面的div背景色\*/

## 滤镜filter函数（ Internet Explorer 不支持 filter 属性。）

Ie可以使用：-webkit-filter: opacity(30%); /\* Chrome, Safari, Opera \*/

**注意:** 滤镜通常使用百分比 (如：75%), 当然也可以使用小数来表示 (如：0.75)。

|  |  |  |
| --- | --- | --- |
| **Filter** | **描述** | |
| none | | 默认值，没有效果。 |
| blur(px) | | 给图像设置高斯模糊。"radius"一值设定高斯函数的标准差，或者是屏幕上以多少像素融在一起， 所以值越大越模糊；  如果没有设定值，则默认是0；这个参数可设置css长度值，但不接受百分比值。 |
| brightness(%) | | 给图片应用一种线性乘法，使其看起来更亮或更暗。如果值是0%，图像会全黑。值是100%，则图像无变化。其他的值对应线性乘数效果。值超过100%也是可以的，图像会比原来更亮。如果没有设定值，默认是1。 |
| contrast(%) | | 调整图像的对比度。值是0%的话，图像会全黑。值是100%，图像不变。值可以超过100%，意味着会运用更低的对比。若没有设置值，默认是1。 |
| drop-shadow(h-shadow v-shadow blur spread color) | | 给图像设置一个阴影效果。阴影是合成在图像下面，可以有模糊度的，可以以特定颜色画出的遮罩图的偏移版本。 函数接受<shadow>(在CSS3背景中定义)类型的值，除了"inset"关键字是不允许的。该函数与已有的box-shadow box-shadow属性很相似；不同之处在于，通过滤镜，一些浏览器为了更好的性能会提供硬件加速。<shadow>参数如下：  **<offset-x>** **<offset-y>** (必须)  这是设置阴影偏移量的两个 <length>值. **<offset-x>** 设定水平方向距离. 负值会使阴影出现在元素左边. **<offset-y>**设定垂直距离.负值会使阴影出现在元素上方。查看**<length>**可能的单位. **如果两个值都是0**, 则阴影出现在元素正后面 (如果设置了 <blur-radius> and/or <spread-radius>，会有模糊效果).  **<blur-radius>** (可选)  这是第三个code><length>值. 值越大，越模糊，则阴影会变得更大更淡.不允许负值 若未设定，默认是0 (则阴影的边界很锐利).  **<spread-radius>** (可选)  这是第四个 <length>值. 正值会使阴影扩张和变大，负值会是阴影缩小.若未设定，默认是0 (阴影会与元素一样大小).  注意: Webkit, 以及一些其他浏览器 不支持第四个长度，如果加了也不会渲染。    **<color>** (可选)  查看 <color>该值可能的关键字和标记。若未设定，颜色值基于浏览器。在Gecko (Firefox), Presto (Opera)和Trident (Internet Explorer)中， 会应用color**color**属性的值。另外, 如果颜色值省略，WebKit中阴影是透明的。 |
| grayscale(%) | | 将图像转换为灰度图像。值定义转换的比例。值为100%则完全转为灰度图像，值为0%图像无变化。值在0%到100%之间，则是效果的线性乘子。若未设置，值默认是0； |
| hue-rotate(deg) | | 给图像应用色相旋转。"angle"一值设定图像会被调整的色环角度值。值为0deg，则图像无变化。若值未设置，默认值是0deg。该值虽然没有最大值，超过360deg的值相当于又绕一圈。 |
| invert(%) | | 反转输入图像。值定义转换的比例。100%的价值是完全反转。值为0%则图像无变化。值在0%和100%之间，则是效果的线性乘子。 若值未设置，值默认是0。 |
| opacity(%) | | 转化图像的透明程度。值定义转换的比例。值为0%则是完全透明，值为100%则图像无变化。值在0%和100%之间，则是效果的线性乘子，也相当于图像样本乘以数量。 若值未设置，值默认是1。该函数与已有的opacity属性很相似，不同之处在于通过filter，一些浏览器为了提升性能会提供硬件加速。 |
| saturate(%) | | 转换图像饱和度。值定义转换的比例。值为0%则是完全不饱和，值为100%则图像无变化。其他值，则是效果的线性乘子。超过100%的值是允许的，则有更高的饱和度。 若值未设置，值默认是1。 |
| sepia(%) | | 将图像转换为深褐色。值定义转换的比例。值为100%则完全是深褐色的，值为0%图像无变化。值在0%到100%之间，则是效果的线性乘子。若未设置，值默认是0； |
| url() | | URL函数接受一个XML文件，该文件设置了 一个SVG滤镜，且可以包含一个锚点来指定一个具体的滤镜元素。  例如：  filter: url(svg-url#element-id) |
| initial | | 设置属性为默认值，可参阅： [CSS initial 关键字](http://www.runoob.com/cssref/css-initial.html) |
| inherit | | 从父元素继承该属性，可参阅：[CSS inherit 关键字](http://www.runoob.com/cssref/css-inherit.html) |

## 背景的绘制区域

background-clip: padding-box;



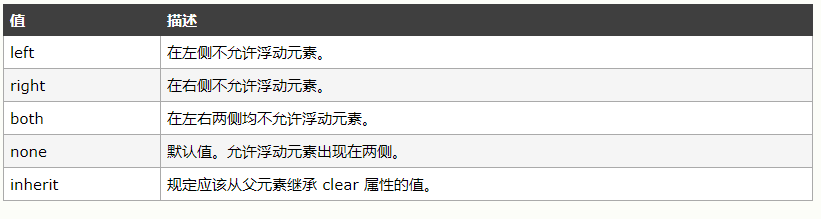
# 引入font文件，自定义class

# 垂直对齐vertical-align 属性



# 不允许浮动clear

clear 属性规定元素的哪一侧不允许其他浮动元素。



# Flex布局

## 一、Flex布局

任何一个容器都可以指定为Flex布局。

.box{

display: flex;

}

行内元素也可以使用Flex布局。

.box{

display: inline-flex;

}

Webkit内核的浏览器，必须加上-webkit前缀。

.box{

display: -webkit-flex; /\* Safari \*/

display: flex;

}

注意，设为Flex布局以后，子元素的float、clear和vertical-align属性将失效。

## 二、基本概念

采用Flex布局的元素，称为Flex容器（flex container），简称”容器”。它的所有子元素自动成为容器成员，称为Flex项目（flex item），简称”项目”。



容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。

项目默认沿主轴排列。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

## 三、容器的属性

以下6个属性设置在容器上。

* flex-direction
* flex-wrap
* flex-flow
* justify-content
* align-items
* align-content

## 3.1 flex-direction属性

flex-direction属性决定主轴的方向（即项目的排列方向）。

.box {

flex-direction: row | row-reverse | column | column-reverse;

}

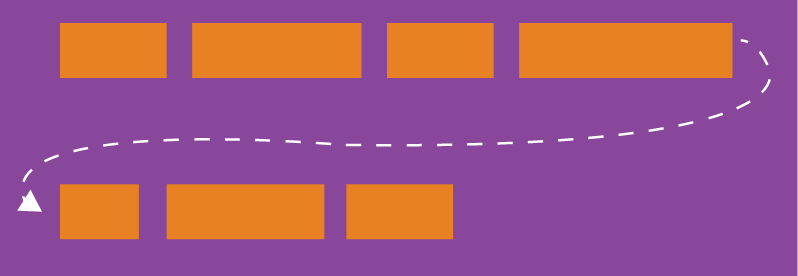


它可能有4个值。

* row（默认值）：主轴为水平方向，起点在左端。
* row-reverse：主轴为水平方向，起点在右端。
* column：主轴为垂直方向，起点在上沿。
* column-reverse：主轴为垂直方向，起点在下沿。

## 3.2 flex-wrap属性

默认情况下，项目都排在一条线（又称”轴线”）上。flex-wrap属性定义，如果一条轴线排不下，如何换行。



.box{

flex-wrap: nowrap | wrap | wrap-reverse;

}

它可能取三个值。

（1）nowrap（默认）：不换行。



（2）wrap：换行，第一行在上方。



（3）wrap-reverse：换行，第一行在下方。



## 3.3 flex-flow

flex-flow属性是flex-direction属性和flex-wrap属性的简写形式，默认值为row nowrap。

.box {

flex-flow: <flex-direction> <flex-wrap>;

}

## 3.4 justify-content属性

justify-content属性定义了项目在主轴上的对齐方式。

.box {

justify-content: flex-start | flex-end | center | space-between | space-around;

}



它可能取5个值，具体对齐方式与轴的方向有关。下面假设主轴为从左到右。

* flex-start（默认值）：左对齐
* flex-end：右对齐
* center： 居中
* space-between：两端对齐，项目之间的间隔都相等。
* space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。

## 3.5 align-items属性

align-items属性定义项目在交叉轴上如何对齐。

.box {

align-items: flex-start | flex-end | center | baseline | stretch;

}



它可能取5个值。具体的对齐方式与交叉轴的方向有关，下面假设交叉轴从上到下。

* flex-start：交叉轴的起点对齐。
* flex-end：交叉轴的终点对齐。
* center：交叉轴的中点对齐。
* baseline: 项目的第一行文字的基线对齐。
* stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。

## 3.6 align-content属性

align-content属性定义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

.box {

align-content: flex-start | flex-end | center | space-between | space-around | stretch;

}



该属性可能取6个值。

* flex-start：与交叉轴的起点对齐。
* flex-end：与交叉轴的终点对齐。
* center：与交叉轴的中点对齐。
* space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。
* space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。
* stretch（默认值）：轴线占满整个交叉轴。

## 四、项目的属性

以下6个属性设置在项目上。

* order
* flex-grow
* flex-shrink
* flex-basis
* flex
* align-self

## 4.1 order属性

order属性定义项目的排列顺序。数值越小，排列越靠前，默认为0。

.item {

order: <integer>;

}



## 4.2 flex-grow属性

flex-grow属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。

.item {

flex-grow: <number>; /\* default 0 \*/

}



如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

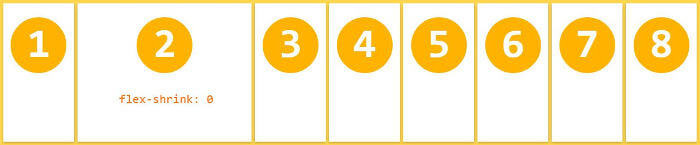
## 4.3 flex-shrink属性

flex-shrink属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。

.item {

flex-shrink: <number>; /\* default 1 \*/

}



如果所有项目的flex-shrink属性都为1，当空间不足时，都将等比例缩小。如果一个项目的flex-shrink属性为0，其他项目都为1，则空间不足时，前者不缩小。

负值对该属性无效。

## 4.4 flex-basis属性

flex-basis属性定义了在分配多余空间之前，项目占据的主轴空间（main size）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即项目的本来大小。

.item {

flex-basis: <length> | auto; /\* default auto \*/

}

它可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。

## 4.5 flex属性

flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

.item {

flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]

}

该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。

建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值。

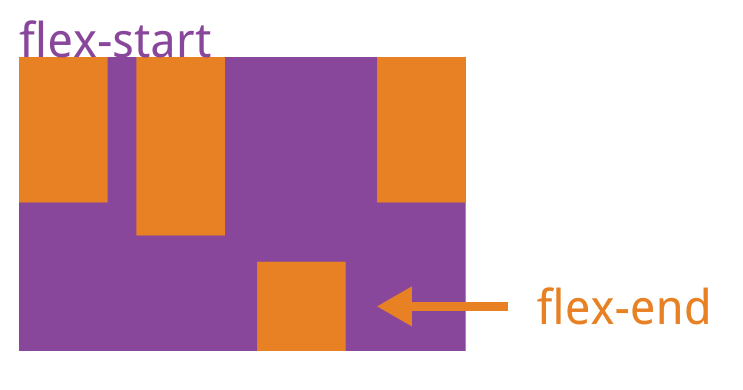
## 4.6 align-self属性

align-self属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch。

.item {

align-self: auto | flex-start | flex-end | center | baseline | stretch;

}



该属性可能取6个值，除了auto，其他都与align-items属性完全一致。

# [css中id和class不能以数字开头](https://www.cnblogs.com/tianmuxi/p/10448613.html)

## 标签select

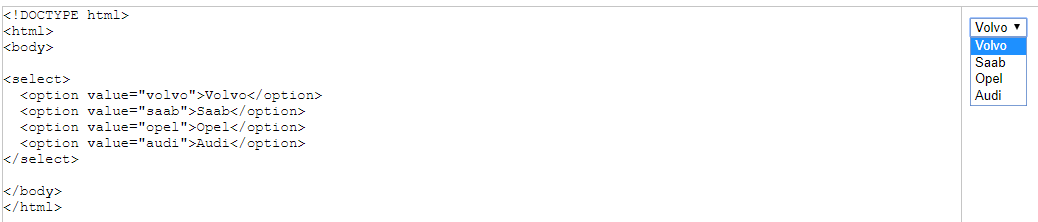
提示和注释：

**提示：**select 元素是一种表单控件，可用于在表单中接受用户输入。

属性

**New:** HTML5 中的新属性。

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [autofocus](http://www.w3school.com.cn/tags/att_select_autofocus.asp) | autofocus | 规定在页面加载后文本区域自动获得焦点。 |
| [disabled](http://www.w3school.com.cn/tags/att_select_disabled.asp) | disabled | 规定禁用该下拉列表。 |
| [form](http://www.w3school.com.cn/tags/att_select_form.asp) | *form\_id* | 规定文本区域所属的一个或多个表单。 |
| [multiple](http://www.w3school.com.cn/tags/att_select_multiple.asp) | multiple | 规定可选择多个选项。 |
| [name](http://www.w3school.com.cn/tags/att_select_name.asp) | *name* | 规定下拉列表的名称。 |
| [required](http://www.w3school.com.cn/tags/att_select_required.asp) | required | 规定文本区域是必填的。 |
| [size](http://www.w3school.com.cn/tags/att_select_size.asp) | *number* | 规定下拉列表中可见选项的数目。 |



例如： 

## visibility 属性规定元素是否可见。

**提示：**即使不可见的元素也会占据页面上的空间。请使用 "display" 属性来创建不占据页面空间的不可见元素。

这个属性指定是否显示一个元素生成的元素框。这意味着元素仍占据其本来的空间，不过可以完全不可见。值 collapse 在表中用于从表布局中删除列或行。

|  |  |
| --- | --- |
| **值** | **描述** |
| 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。 |

## 父级div块不能遮盖子集div块

## 阻止冒泡（点击div外的地方，div隐藏）

        function stopPropagation(e) {  
            var ev = e || window.event;  
            if (ev.stopPropagation) {  
                ev.stopPropagation();  
            }  
            else if (window.event) {  
                window.event.cancelBubble = true;//兼容IE  
            }  
        }  
        $("#btnClick").click(function (e) {  
            $("#pop").show();  
            stopPropagation(e);  
        });  
        $(document).bind('click', function () {  
            $("#pop").hide();  
        });  
        $("#pop").click(function (e) {  
            stopPropagation(e);  
        });

## $(".b12").val($(this).children("span").text()+"元") // 待查阅

Overflow：hidden；div里的div属性是float，使用overflow：hidden显示其体积。

# Css3

## 边框属性

### 边框阴影

box-shadow：10px 10px 5px #888888;

阴影左右位置，阴影上下位置，阴影模糊。

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [border-image](http://www.w3school.com.cn/cssref/pr_border-image.asp) | 设置所有 border-image-\* 属性的简写属性。 | 3 |
| [border-radius](http://www.w3school.com.cn/cssref/pr_border-radius.asp) | 设置所有四个 border-\*-radius 属性的简写属性。 | 3 |
| [box-shadow](http://www.w3school.com.cn/cssref/pr_box-shadow.asp) | 向方框添加一个或多个阴影。 | 3 |

box-shadow: -5px -5px 15px #e3e3e3, 5px 5px 15px #e3e3e3;

/\*四周阴影\*/

### 边框图片

#round

{

border-image:url(/i/border.png) 30 30 30 30 round;

}

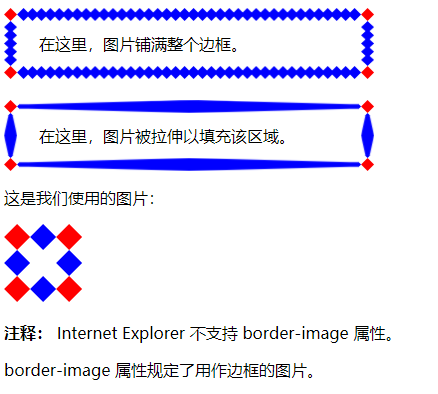
//上右下左。

#stretch

{

border-image:url(/i/border.png) 30 30 30 30 stretch;

}

</style>

## 背景

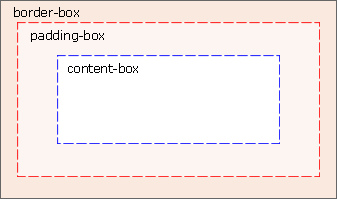
### 控制背景色区域

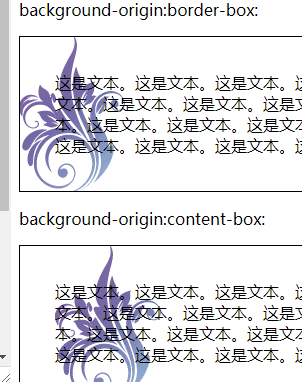
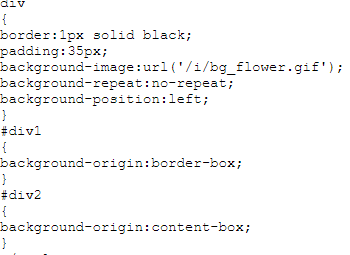
background-origin:content-box;

### background-origin 背景图片的定位区域

background-origin 属性规定背景图片的定位区域。

背景图片可以放置于 content-box、padding-box 或 border-box 区域。



例子：

### 控制背景图片位置

background-position: center center; /\*控制图片位置\*/

### 控制背景图片大小

background-size: 18px 18px; /\*控制图片大小\*/

### CSS3 多重背景图片

CSS3 允许您为元素使用多个背景图像。

background-image:url(/i/bg\_flower.gif),url(/i/bg\_flower\_2.gif);

两个背景图重叠。

## 字体效果

### 文本阴影

规定水平阴影、垂直阴影、模糊距离，以及阴影的颜色；

实例：h1

{

text-shadow: 5px 5px 5px #FF0000;

}



### 自动换行

p {word-wrap:break-word;}

word-wrap 属性允许您允许文本强制文本进行换行 - 即使这意味着会对单词进行拆分；

### 文本可编辑

contenteditable="true"

<div class="textarea" contenteditable="true"></div>

### 新的文本属性

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [hanging-punctuation](http://www.w3school.com.cn/cssref/pr_hanging-punctuation.asp) | 规定标点字符是否位于线框之外。 | 3 |
| [punctuation-trim](http://www.w3school.com.cn/cssref/pr_punctuation-trim.asp) | 规定是否对标点字符进行修剪。 | 3 |
| text-align-last | 设置如何对齐最后一行或紧挨着强制换行符之前的行。 | 3 |
| [text-emphasis](http://www.w3school.com.cn/cssref/pr_text-emphasis.asp) | 向元素的文本应用重点标记以及重点标记的前景色。 | 3 |
| [text-justify](http://www.w3school.com.cn/cssref/pr_text-justify.asp) | 规定当 text-align 设置为 "justify" 时所使用的对齐方法。 | 3 |
| [text-outline](http://www.w3school.com.cn/cssref/pr_text-outline.asp) | 规定文本的轮廓。 | 3 |
| [text-overflow](http://www.w3school.com.cn/cssref/pr_text-overflow.asp) | 规定当文本溢出包含元素时发生的事情。 | 3 |
| [text-shadow](http://www.w3school.com.cn/cssref/pr_text-shadow.asp) | 向文本添加阴影。 | 3 |
| [text-wrap](http://www.w3school.com.cn/cssref/pr_text-wrap.asp) | 规定文本的换行规则。 | 3 |
| [word-break](http://www.w3school.com.cn/cssref/pr_word-break.asp) | 规定非中日韩文本的换行规则。 | 3 |
| [word-wrap](http://www.w3school.com.cn/cssref/pr_word-wrap.asp) | 允许对长的不可分割的单词进行分割并换行到下一行。 | 3 |

## 字体

### @font-face 规则

1. 搜索电脑里的font
2. 将字体文件存放在项目文件夹内，重新命名别用中文
3. 在style内输入下列程序

@font-face{  
    font-family: xutao 666 ;

/\*名称自己取，需要用时需自己调用\*/

src: url(./'Sansation\_Light.ttf'),

url(./'Sansation\_Light.eot'); /\* IE9+ \*/

font-weight:bold;//定义粗体的字体

}/\*定义你自己的字体！！！\*/

## 2D转换

通过 CSS3 转换，我们能够对元素进行移动、缩放、转动、拉长或拉伸。

### Div旋转transform: rotate(30deg)

通过 rotate() 方法，元素顺时针旋转给定的角度。允许负值，元素将逆时针旋转。

<Style>

div{

transform: rotate(30deg);

-ms-transform: rotate(30deg); /\* IE 9 \*/

-webkit-transform: rotate(30deg); /\* Safari and Chrome \*/

-o-transform: rotate(30deg); /\* Opera \*/

-moz-transform: rotate(30deg); /\* Firefox \*/

}

</Style>

### 设置旋转元素的基点位置：

transform: rotate(45deg);

transform-origin:20% 40%;

//div的宽高百分比

### div移动translate() 方法

元素从其当前位置移动，根据给定的 left（x 坐标） 和 top（y 坐标） 位置参数;

实例：

<Style>

div {

//50px代表左右移动，100px代表上下移动

transform: translate(50px,100px);

-ms-transform: translate(50px,100px); /\* IE 9 \*/

-webkit-transform: translate(50px,100px); /\* Safari and Chrome \*/

-o-transform: translate(50px,100px); /\* Opera \*/

-moz-transform: translate(50px,100px); /\* Firefox \*/

}

</Style>

### div放大/缩小scale() 方法

通过 scale() 方法，元素的尺寸会增加或减少，根据给定的宽度（X 轴）和高度（Y 轴）参数；

div

{

//scale(2,4) 把宽度转换为原始尺寸的 2 倍，把高度转换为原始高度的 4 倍。

transform: scale(2,4);

-ms-transform: scale(2,4); /\* IE 9 \*/

-webkit-transform: scale(2,4); /\* Safari 和 Chrome \*/

-o-transform: scale(2,4); /\* Opera \*/

-moz-transform: scale(2,4); /\* Firefox \*/

}

### 旋转（X\Y轴）skew() 方法

通过 skew() 方法，元素翻转给定的角度，根据给定的水平线（X 轴）和垂直线（Y 轴）参数：

div

{

skew(30deg,20deg) 围绕 X 轴把元素翻转 30 度，围绕 Y 轴翻转 20 度。

transform: skew(30deg,20deg);

-ms-transform: skew(30deg,20deg); /\* IE 9 \*/

-webkit-transform: skew(30deg,20deg); /\* Safari and Chrome \*/

-o-transform: skew(30deg,20deg); /\* Opera \*/

-moz-transform: skew(30deg,20deg); /\* Firefox \*/

}



### 2D 转换方法组合matrix() 方法

matrix() 方法把所有 2D 转换方法组合在一起。

matrix() 方法需要六个参数，包含数学函数，允许您：旋转、缩放、移动以及倾斜元素。

实例

如何使用 matrix 方法将 div 元素旋转 30 度：

div

{

transform:matrix(0.866,0.5,-0.5,0.866,0,0);

-ms-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /\* IE 9 \*/

-moz-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /\* Firefox \*/

-webkit-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /\* Safari and Chrome \*/

-o-transform:matrix(0.866,0.5,-0.5,0.866,0,0); /\* Opera \*/

}

### 2D Transform 方法

|  |  |
| --- | --- |
| **函数** | **描述** |
| matrix(*n*,*n*,*n*,*n*,*n*,*n*) | 定义 2D 转换，使用六个值的矩阵。 |
| translate(*x*,*y*) | 定义 2D 转换，沿着 X 和 Y 轴移动元素。 |
| translateX(*n*) | 定义 2D 转换，沿着 X 轴移动元素。 |
| translateY(*n*) | 定义 2D 转换，沿着 Y 轴移动元素。 |
| scale(*x*,*y*) | 定义 2D 缩放转换，改变元素的宽度和高度。 |
| scaleX(*n*) | 定义 2D 缩放转换，改变元素的宽度。 |
| scaleY(*n*) | 定义 2D 缩放转换，改变元素的高度。 |
| rotate(*angle*) | 定义 2D 旋转，在参数中规定角度。 |
| skew(*x-angle*,*y-angle*) | 定义 2D 倾斜转换，沿着 X 和 Y 轴。 |
| skewX(*angle*) | 定义 2D 倾斜转换，沿着 X 轴。 |
| skewY(*angle*) | 定义 2D 倾斜转换，沿着 Y 轴。 |

## 3D转换

### rotateX() 方法

通过 rotateX() 方法，元素围绕其 X 轴以给定的度数进行旋转。

实例

div

{

transform: rotateX(120deg);

-webkit-transform: rotateX(120deg); /\* Safari 和 Chrome \*/

-moz-transform: rotateX(120deg); /\* Firefox \*/

}

### rotateY() 旋转

通过 rotateY() 方法，元素围绕其 Y 轴以给定的度数进行旋转。

实例

div

{

transform: rotateY(130deg);

-webkit-transform: rotateY(130deg); /\* Safari 和 Chrome \*/

-moz-transform: rotateY(130deg); /\* Firefox \*/

}

### 转换属性

下面的表格列出了所有的转换属性：

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [transform](http://www.w3school.com.cn/cssref/pr_transform.asp) | 向元素应用 2D 或 3D 转换。 | 3 |
| [transform-origin](http://www.w3school.com.cn/cssref/pr_transform-origin.asp) | 允许你改变被转换元素的位置。 | 3 |
| [transform-style](http://www.w3school.com.cn/cssref/pr_transform-style.asp) | 规定被嵌套元素如何在 3D 空间中显示。 | 3 |
| [perspective](http://www.w3school.com.cn/cssref/pr_perspective.asp) | 规定 3D 元素的透视效果。 | 3 |
| [perspective-origin](http://www.w3school.com.cn/cssref/pr_perspective-origin.asp) | 规定 3D 元素的底部位置。 | 3 |
| [backface-visibility](http://www.w3school.com.cn/cssref/pr_backface-visibility.asp) | 定义元素在不面对屏幕时是否可见。 | 3 |

## Css过渡

### 属性过渡

应用于宽度、高度和转换属性的过渡效果，时长为 2 秒：

div

{

transition: width 2s, height 2s, transform 2s;//多项属性改变

transition: width 2s;//宽度属性改变

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

-webkit-transition: width 2s; /\* Safari 和 Chrome \*/

-o-transition: width 2s; /\* Opera \*/

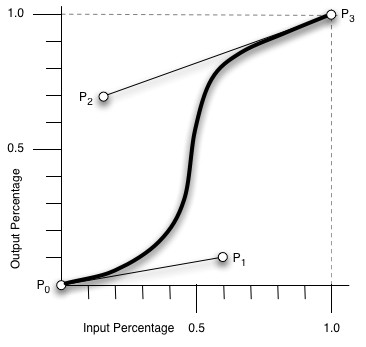
}

### 规定属性名称过渡

transition-property:width;

transition-duration:5s;

|  |  |
| --- | --- |
| **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。 |

[](https://gss0.baidu.com/9vo3dSag_xI4khGko9WTAnF6hhy/zhidao/pic/item/7dd98d1001e93901686997187fec54e736d196a8.jpg)

cubic-bezier即为[贝兹曲线](https://www.baidu.com/s?wd=%E8%B4%9D%E5%85%B9%E6%9B%B2%E7%BA%BF&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)中的绘制方法。图上有四点，P0-3，其中P0、P3是默认的点，对应了[0,0], [1,1]。而剩下的P1、P2两点则是我们通过cubic-bezier()自定义的。cubic-bezier(x1, y1, x2, y2) 为自定义，x1,x2,y1,y2的值范围在[0, 1]。

## 动画

### 动画

<style>

div

{

width:100px;

height:100px;

background:red;

animation:myfirst 5s;

}

@keyframes myfirst

{

from {background:red;}

to {background:yellow;}

}

</style>

</head>

<body>

<div></div>

### 动画（当动画为 25% 及 50% 时改变背景色，然后当动画 100% 完成时再次改变：）

@keyframes myfirst

{

0% {background: red;}

25% {background: yellow;}

50% {background: blue;}

100% {background: green;}

}

动画（改变背景颜色和div位置）

@keyframes myfirst

{

0% {background: red; left:0px; top:0px;}

25% {background: yellow; left:200px; top:0px;}

50% {background: blue; left:200px; top:200px;}

75% {background: green; left:0px; top:200px;}

100% {background: red; left:0px; top:0px;}

}

### CSS3 动画属性

下面的表格列出了 @keyframes 规则和所有动画属性：

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [@keyframes](http://www.w3school.com.cn/cssref/pr_keyframes.asp) | 规定动画。 | 3 |
| [animation](http://www.w3school.com.cn/cssref/pr_animation.asp) | 所有动画属性的简写属性，除了 animation-play-state 属性。 | 3 |
| [animation-name](http://www.w3school.com.cn/cssref/pr_animation-name.asp) | 规定 @keyframes 动画的名称。 | 3 |
| [animation-duration](http://www.w3school.com.cn/cssref/pr_animation-duration.asp) | 规定动画完成一个周期所花费的秒或毫秒。默认是 0。 | 3 |
| [animation-timing-function](http://www.w3school.com.cn/cssref/pr_animation-timing-function.asp) | 规定动画的速度曲线。默认是 "ease"。 | 3 |
| [animation-delay](http://www.w3school.com.cn/cssref/pr_animation-delay.asp) | 规定动画何时开始。默认是 0。 | 3 |
| [animation-iteration-count](http://www.w3school.com.cn/cssref/pr_animation-iteration-count.asp) | 规定动画被播放的次数。默认是 1。 | 3 |
| [animation-direction](http://www.w3school.com.cn/cssref/pr_animation-direction.asp) | 规定动画是否在下一周期逆向地播放。默认是 "normal"。 | 3 |
| [animation-play-state](http://www.w3school.com.cn/cssref/pr_animation-play-state.asp) | 规定动画是否正在运行或暂停。默认是 "running"。 | 3 |
| [animation-fill-mode](http://www.w3school.com.cn/cssref/pr_animation-fill-mode.asp) | 规定对象动画时间之外的状态。 | 3 |

## 多列

### 创建多列

column-count 属性规定元素应该被分隔的列数：

把 div 元素中的文本分隔为三列：

div

{

-moz-column-count:3; /\* Firefox \*/

-webkit-column-count:3; /\* Safari 和 Chrome \*/

column-count:3;

}

### 规定列之间的间隔

column-gap 属性规定列之间的间隔：

规定列之间 40 像素的间隔：

div

{

-moz-column-gap:40px; /\* Firefox \*/

-webkit-column-gap:40px; /\* Safari 和 Chrome \*/

column-gap:40px;

}

### 列规则

column-rule 属性设置列之间的宽度、样式和颜色规则。

规定列之间的宽度、样式和颜色规则：

div

{

-moz-column-rule:3px outset #ff0000; /\* Firefox \*/

-webkit-column-rule:3px outset #ff0000; /\* Safari and Chrome \*/

column-rule:3px outset #ff0000;

}

### 新的多列属性

下面的表格列出了所有的转换属性：

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [column-count](http://www.w3school.com.cn/cssref/pr_column-count.asp) | 规定元素应该被分隔的列数。 | 3 |
| [column-fill](http://www.w3school.com.cn/cssref/pr_column-fill.asp) | 规定如何填充列。 | 3 |
| [column-gap](http://www.w3school.com.cn/cssref/pr_column-gap.asp) | 规定列之间的间隔。 | 3 |
| [column-rule](http://www.w3school.com.cn/cssref/pr_column-rule.asp) | 设置所有 column-rule-\* 属性的简写属性。 | 3 |
| [column-rule-color](http://www.w3school.com.cn/cssref/pr_column-rule-color.asp) | 规定列之间规则的颜色。 | 3 |
| [column-rule-style](http://www.w3school.com.cn/cssref/pr_column-rule-style.asp) | 规定列之间规则的样式。 | 3 |
| [column-rule-width](http://www.w3school.com.cn/cssref/pr_column-rule-width.asp) | 规定列之间规则的宽度。 | 3 |
| [column-span](http://www.w3school.com.cn/cssref/pr_column-span.asp) | 规定元素应该横跨的列数。 | 3 |
| [column-width](http://www.w3school.com.cn/cssref/pr_column-width.asp) | 规定列的宽度。 | 3 |
| [columns](http://www.w3school.com.cn/cssref/pr_columns.asp) | 规定设置 column-width 和 column-count 的简写属性。 | 3 |

## 用户界面

### 用户鼠标调整元素尺寸Resizing

在 CSS3，resize 属性规定是否可由用户调整元素尺寸。

这个 div 元素可由用户调整尺寸（在 Firefox 4+、Chrome 以及 Safari 中）。

CSS 代码如下：

规定 div 元素可由用户调整大小：

div

{

resize:both;

overflow:auto;

}

### 外边框偏移Outline Offset

outline-offset 属性对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓。

轮廓与边框有两点不同：

* 轮廓不占用空间
* 轮廓可能是非矩形

这个 div 在边框之外 15 像素处有一个轮廓。

CSS 代码如下：

规定边框边缘之外 15 像素处的轮廓：

div

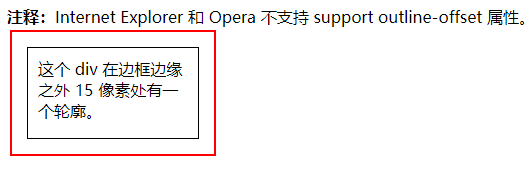
{

border:2px solid black;

outline:2px solid red;

outline-offset:15px;

}



### 新的用户界面属性

下面的表格列出了所有的转换属性：

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [appearance](http://www.w3school.com.cn/cssref/pr_appearance.asp) | 允许您将元素设置为标准用户界面元素的外观 | 3 |
| [box-sizing](http://www.w3school.com.cn/cssref/pr_box-sizing.asp) | 允许您以确切的方式定义适应某个区域的具体内容。 | 3 |
| [icon](http://www.w3school.com.cn/cssref/pr_icon.asp) | 为创作者提供使用图标化等价物来设置元素样式的能力。 | 3 |
| [nav-down](http://www.w3school.com.cn/cssref/pr_nav-down.asp) | 规定在使用 arrow-down 导航键时向何处导航。 | 3 |
| [nav-index](http://www.w3school.com.cn/cssref/pr_nav-index.asp) | 设置元素的 tab 键控制次序。 | 3 |
| [nav-left](http://www.w3school.com.cn/cssref/pr_nav-left.asp) | 规定在使用 arrow-left 导航键时向何处导航。 | 3 |
| [nav-right](http://www.w3school.com.cn/cssref/pr_nav-right.asp) | 规定在使用 arrow-right 导航键时向何处导航。 | 3 |
| [nav-up](http://www.w3school.com.cn/cssref/pr_nav-up.asp) | 规定在使用 arrow-up 导航键时向何处导航。 | 3 |
| [outline-offset](http://www.w3school.com.cn/cssref/pr_outline-offset.asp) | 对轮廓进行偏移，并在超出边框边缘的位置绘制轮廓。 | 3 |
| [resize](http://www.w3school.com.cn/cssref/pr_resize.asp) | 规定是否可由用户对元素的尺寸进行调整。 | 3 |

## Css单位

### 尺寸单位

|  |  |
| --- | --- |
| **单位** | **描述** |
| % | 百分比 |
| in | 英寸 |
| cm | 厘米 |
| mm | 毫米 |
| em | 1em 等于当前的字体尺寸。  2em 等于当前字体尺寸的两倍。  例如，如果某元素以 12pt 显示，那么 2em 是24pt。  在 CSS 中，em 是非常有用的单位，因为它可以自动适应用户所使用的字体。 |
| ex | 一个 ex 是一个字体的 x-height。 (x-height 通常是字体尺寸的一半。) |
| pt | 磅 (1 pt 等于 1/72 英寸) |
| pc | 12 点活字 (1 pc 等于 12 点) |
| px | 像素 (计算机屏幕上的一个点) |

### 颜色单位

|  |  |
| --- | --- |
| **单位** | **描述** |
| (颜色名) | 颜色名称 (比如 red) |
| rgb(x,x,x) | RGB 值 (比如 rgb(255,0,0)) |
| rgb(x%, x%, x%) | RGB 百分比值 (比如 rgb(100%,0%,0%)) |
| #rrggbb | 十六进制数 (比如 #ff0000) |

## Html5 w3c

# Html手机自适应

<meta charset="UTF-8">  
    <meta name="viewport"  
          content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1, user-scalable=0">  
    <meta name="format-detection" content="telephone=no">  
    <meta name="apple-mobile-web-app-capable" content="yes">  
    <meta name="apple-mobile-web-app-status-bar-style" content="black">

原文链接：<http://caibaojian.com/rem-vs-em.html>

## 移动端产生阴影

appearance: none;

-moz-appearance: none;

-webkit-appearance: none;

## 消除移动端点击时的阴影效果

/\* -webkit-tap-highlight-color: rgba(0, 0, 0, 0); \*/

/\* 取消点击时的阴影效果 \*/

## rem 差异 em

上述所有归结如下:[·](http://caibaojian.com/rem-vs-em.html)

* rem 单位翻译为像素值是由 html 元素的字体大小决定的。 此字体大小会被浏览器中字体大小的设置影响，除非显式重写一个具体单位。
* em 单位转为像素值，取决于他们使用的字体大小。 此字体大小受从父元素继承过来的字体大小，除非显式重写与一个具体单位。

## 为什么使用 rem 单位:

Rem 单位提供最伟大的力量并不仅仅是他们提供一致尺寸而不是继承。 相反，它给我们的一个途经去获取用户的偏好来影响网站中每一处使用rem的元素大小，不再是使用固定的 px 单位。

为此，使用 rem 单位的主要目的应该是确保无论用户如何设置自己的浏览器，我们的布局都能调整到合适大小。

一个站点最初设计可以专注于最常见的默认浏览器中字体大小 16px。

## 使用媒体查询（media query）适配布局

/\*

\* 1、流式布局适配方式：

\* 只达到宽度的适配

\* 内容没有自适应改变高度，高度也没有自适应改变

\*

\* 2、rem适配方式：

\* 1、高度和宽度都自适应 内容可自适应 等比例缩放

\* 2、当我在改变rem的基准值html的字体大小的时候，页面上的容器在等比例缩放

\* 3、适配：

\* 假设320px的设备 设置字体的大小是 50px

\* 假设640px的设备 设置字体的大小是 100px

\*/

/\*4、使用媒体查询\*/

rem

@media (min-width: 320px) {

html {

font-size: 50px;

}

}

/\*5、需要适配移动端主流的设备 有很多 这里以iphone6 375px 为例\*/

/\*但这个方法比较麻烦，要一个一个适配 用less可以解决这个问题\*/

@media (min-width: 375px) {

html {

font-size: 58.59375px; /\* 375/640\*100=58.59375px \*/

}

}

@media (min-width: 640px) {

html {

font-size: 100px; /\* 640/320\*50=100px \*/

}

}

</style>

# align或者valign是html属性

# vertical-align:bottom; 只能用在表格里，对div无效。

# url、href、src的区别

## URL

文件地址

## **href**

(Hypertext Reference)指定网络资源的位置，从而在当前元素或者当前文档和由当前属性定义的需要的锚点或资源之间定义一个链接或者关系。

href 目的不是为了引用资源，而是为了建立联系，让当前标签能够链接到目标地址。

## src

在请求 src 资源时会将其指向的资源下载并应用到文档中，比如 JavaScript 脚本，img 图片；

src 用于替换当前内容；

# a标签

## 制作电子邮件链接

<a href="mailto:xxx@yyy">

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [charset](http://www.w3school.com.cn/tags/att_a_charset.asp) | *char\_encoding* | HTML5 中不支持。规定被链接文档的字符集。 |
| [coords](http://www.w3school.com.cn/tags/att_a_coords.asp) | *coordinates* | HTML5 中不支持。规定链接的坐标。 |
| [download](http://www.w3school.com.cn/tags/att_a_download.asp) | *filename* | 规定被下载的超链接目标。 |
| [href](http://www.w3school.com.cn/tags/att_a_href.asp) | *URL* | 规定链接指向的页面的 URL。 |
| [hreflang](http://www.w3school.com.cn/tags/att_a_hreflang.asp) | *language\_code* | 规定被链接文档的语言。 |
| [media](http://www.w3school.com.cn/tags/att_a_media.asp) | *media\_query* | 规定被链接文档是为何种媒介/设备优化的。 |
| [name](http://www.w3school.com.cn/tags/att_a_name.asp) | *section\_name* | HTML5 中不支持。规定锚的名称。 |
| [rel](http://www.w3school.com.cn/tags/att_a_rel.asp) | *text* | 规定当前文档与被链接文档之间的关系。 |
| [rev](http://www.w3school.com.cn/tags/att_a_rev.asp) | *text* | HTML5 中不支持。规定被链接文档与当前文档之间的关系。 |
| [shape](http://www.w3school.com.cn/tags/att_a_shape.asp) | * default * rect * circle * poly | HTML5 中不支持。规定链接的形状。 |
| [target](http://www.w3school.com.cn/tags/att_a_target.asp) | * \_blank * \_parent * \_self * \_top * *framename* | 规定在何处打开链接文档。 |
| [type](http://www.w3school.com.cn/tags/att_a_type.asp) | *MIME type* | 规定被链接文档的的 MIME 类型。 |

# 监听js

node server.js 8000

# 监听html

$ http-server . -c-1