



[home](#) [feed](#) | [javascript](#) [php](#) [python](#) [java](#) [mysql](#) [ios](#) [android](#) [node](#)

文 css学习归纳总结（一）

css web前端开发 trigkit4 2014年11月12日发布

选择器的分组

CSS选择器分为

1. 群组选择器 如： `p, body, img, div{}`
2. 兄弟选择器 如： `p + p { color:#f00; }`
3. 属性选择器 如： `p[title] { color:#f00; }`
4. 包含（后代）选择器 如： `body ul li{}`
5. 子元素选择器 如： `div > p{}`
6. ID选择器 如： `#myDiv{}`
7. 类选择器 如： `.class1{}`
8. 伪元素选择器 如： `E:first-line, E:before`
9. 伪类选择器 如： `E:first-child, E:visited, E:focus, E:enabled`
10. 标签选择器 如： `p { font-size:1em; }`

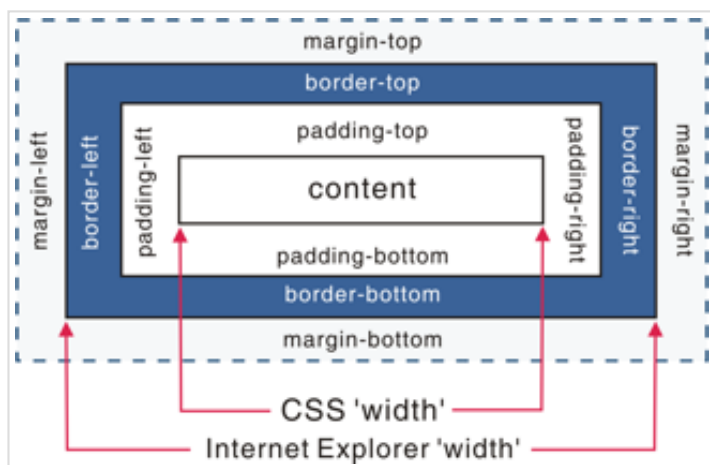
子选择器和后代选择器的区别：

后代选择器的写法就是把外层的标记写在前面，内层的标记写在后面，之间用空格分隔。
子选择器只对直接后代有影响的选择器，而对“孙子后代”以及多层后代不产生作用。

注意：在IE6中，只支持后代选择器，不支持子选择器

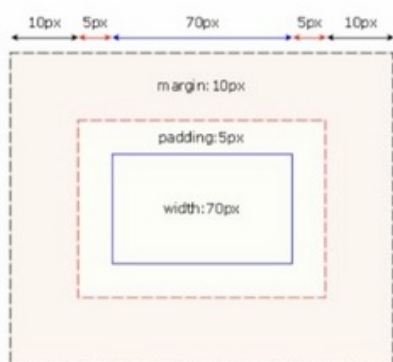
盒子模型

css框模型介绍：



盒子模型的概念

假设框的每个边上有 10 个像素的外边距和 5 个像素的内边距。如果希望这个元素框在页面内只占 100 个像素，就需要将内容的宽度设置为 70 像素，请看下图：

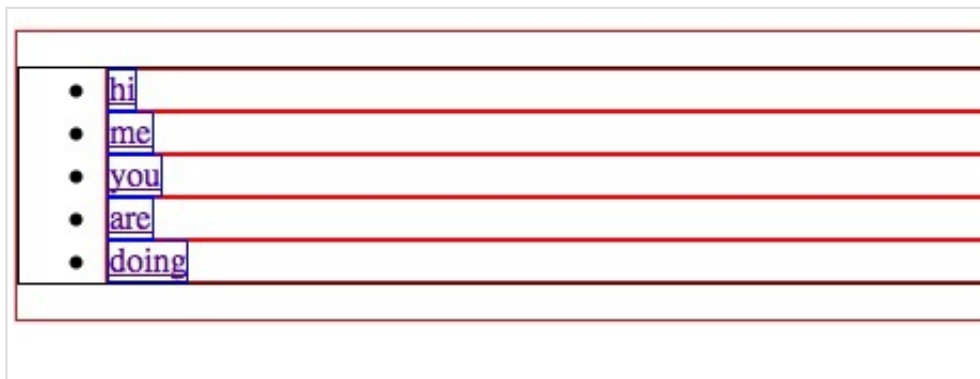


盒子在页面中所占的宽度
是由左边距+左边框+左填充+内容宽度+右填充+右边框+右边距组成
盒子在页面中所占的高度
是由上边距+上边框+上填充+内容高度+下填充+下边框+下边距组成

盒子模型的概念

在 CSS 中，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。

给 ul 下的 li 设置宽度，那么他们将平均掉它们所占容器的宽度。div >ul> li> a 标签所占宽度和高度：



当你为 body 元素指定 div 时，每个 div 默认的宽为浏览器的100%，高度在每个浏览器的解析都是不一样的，比如在火狐下是131px，在 chrome 下则是142px，具体机制不明

css字体速记规则

当用 `css` 定义字体样式的时候，你也许会这样做：

```
font-weight: bold;
font-style: italic;
font-variant: small-caps;
font-size: 1em;
line-height: 1.5em;
font-family: verdana,sans-serif
```

没有必要这么做，因为你可以使用css速记属性：

```
font: bold italic small-caps 1em/1.5em verdana,sans-serif
```

值得注意的是：`css` 速记版本只会在你同时指定 `font-size` 和 `font-family` 属性时才会生效。如果你没有指定 `font-weight`，`font-style`，或者 `font-variant`，那么这些值将会自动默认为 `normal`

避免多余选择器

有时候你会有如下的写法：

```
ul li { ... }
ol li { ... }
table tr td { ... }
```

但实际上这是没必要的写法，因为 `` 会且只会与 `` 或 `` 连用，就像 `<td>` 只能存在于 `<tr>` 和 `<table>` 中一样，这儿真没有把它们重复一次的必要。

!Important

所有被 `!important` 标记的样式，即使它后来被重写，浏览器也只会采用被标记的那条。

```
#content{
  height:960px !important;
  height:900px;
}
```

!important: IE7/IE8/IE9/FireFox 可以识别上面附加 **!important** 的语句，看到附加 **!important** 的语句后，就不会再去执行 第二句，尽管他们也“认识”第二句，但是附加“!important”的语句拥有绝对优先级，只要有它存在，第二句就不允许执行。

IE6会执行 **!important** 的语句后，再去执行第二句 “height:900px”，用后面的样式覆盖了前面的样式，所以在IE6下，最终的height值是900px。

@import and <link>

在决定使用哪种方法去包含你的样式表之前，你应该明白这两种方法被用来干嘛？

链接式：

link就是把外部CSS与网页连接起来，将其加在 <head> 头部标签中，具体形式如下：

```
<link href="styles.css" type="text/css" />
```

导入样式：

通过两个或以上的 **css** 文件共同部分，在 **css** 页面引入 **css** 文件，通过 **@import url (” “)**

```
@import url("global.css");  
@import url(global.css);  
@import "global.css";
```

以上三种方式都有效，

区别：

- 1) **link**属于XHTML标签，而**@import**是CSS提供的；
- 2) 页面被加载的时，**link**会同时被加载，而**@import**引用的CSS会等到页面被加载完再加载；
- 3) **@import**只在IE5以上才能识别，而**link**是XHTML标签，无兼容问题；
- 4) **link**方式的样式的权重 高于**@import**的权重。

行内元素、块元素和空元素

行内元素特点

- 1、和其他元素都在一行上；
- 2、元素的高度、宽度、行高及顶部和底部边距不可设置；
- 3、元素的宽度就是它包含的文字或图片的宽度，不可改变。

行内元素在设置 水平方向的 `padding-left`、`padding-right`、`margin-left`、`margin-right` 都产生边距效果，但竖直方向的 `padding-top`、`padding-bottom`、`margin-top`、`margin-bottom` 却不会产生边距效果。

块级元素特点

- 1、每个块级元素都从新的一行开始，并且其后的元素也另起一行。（真霸道，一个块级元素独占一行）
- 2、元素的高度、宽度、行高以及顶和底边距都可设置。
- 3、元素宽度在不设置的情况下，是它本身父容器的100%（和父元素的宽度一致），除非设定一个宽度。

在html中，``、`<a>`、`<label>`、`<input>`、``、`` 和 `` 就是典型的行内元素（inline）元素。

块级元素是可以设置宽高的，但是它的实际宽高是本身宽高+padding。block元素要单独占一行。内联元素不单独占一行，给他设置宽高是没有用的。

块级元素和行内元素列表：

块级元素列表	
<address>	定义地址
<caption>	定义表格标题
<dd>	定义列表中定义条目
<div>	定义文档中的分区或节
<dl>	定义列表
<dt>	定义列表中的项目
<fieldset>	定义一个框架集
<form>	创建 HTML 表单
<h1>	定义最大的标题
<h2>	定义副标题
<h3>	定义标题
<h4>	定义标题
<h5>	定义标题
<h6>	定义最小的标题
<hr>	创建一条水平线
<legend>	元素为 fieldset 元素定义标题
	标签定义列表项目
<noframes>	为那些不支持框架的浏览器显示文本，于 frameset 元素内部
<noscript>	定义在脚本未执行时的替代内容
	定义有序列表
	定义无序列表
<p>	标签定义段落
<pre>	定义预格式化的文本
<table>	标签定义 HTML 表格
<tbody>	标签表格主体（正文）
<td>	表格中的标准单元格
<tfoot>	定义表格的页脚（脚注或表注）
<th>	定义表头单元格
<thead>	标签定义表格的表头
<tr>	定义表格中的行

行内元素列表	
<a>	标签可定义锚
<abbr>	表示一个缩写形式
<acronym>	定义只取首字母缩写
	字体加粗
<bdo>	可覆盖默认的文本方向
<big>	大号字体加粗
 	换行
<cite>	引用进行定义
<code>	定义计算机代码文本
<dfn>	定义一个定义项目
	定义为强调的内容
<i>	斜体文本效果
	向网页中嵌入一幅图像
<input>	输入框
<kbd>	定义键盘文本
<label>	标签为 input 元素定义标注（标记）
<q>	定义短的引用
<samp>	定义样本文本
<select>	创建单选或多选菜单
<small>	呈现小号字体效果
	组合文档中的行内元素
	语气更强的强调的内容
<sub>	定义下标文本
<sup>	定义上标文本
<textarea>	多行的文本输入控件
<tt>	打字机或者等宽的文本效果
<var>	定义变量

可变元素列表--可变元素为根据上下文语境决定该元素为块元素或者内联元素	
<button>	按钮
	定义文档中已被删除的文本
<iframe>	创建包含另外一个文档的内联框架（即行内框架）
<ins>	标签定义已经被插入文档中的文本
<map>	客户端图像映射（即热区）
<object>	object对象
<script>	客户端脚本

空元素

知名的空元素：
 <hr> <input> <link> <meta>
鲜为人知的是：<area> <base> <col> <command> <embed> <keygen> <param> <source> <track>
<wbr>

长度单位

em

em 为相对长度单位，相对于当前对象内文本的字体尺寸(**font-size**)。比如：Web页面中 **body** 的文字大小在用户浏览器下默认渲染是 **16px**，所以，此时的 **1em = 16px**；

in

英寸 (Inches)。绝对长度单位。

1in = 2.54cm = 25.4 mm = 72pt = 6pc = 96px

pt

点 (Points)。绝对长度单位。

1in = 2.54cm = 25.4 mm = 72pt = 6pc = 96px

small

<small> 标签呈现小号字体效果。

****，****和****，**<i>**标签

`` 标签和 **``** 标签一样，用于强调文本，但它强调的程度更强一些。
`em` 是 斜体强调标签，更强烈强调，表示内容的强调点。相当于html元素中的 **`<i>...</i>`**；
`` 和 **`<i>`** 是视觉要素，分别表示无意义的加粗，无意义的斜体。
em 和 **strong** 是表达要素(phrase elements)。

b标签和i标签的区别

b 元素现在描述为在普通文章中仅从文体上突出的不包含任何额外的重要性的一段文本。
例如：文档概要中的关键字，评论中的产品名。或者代表强调的排版方式。

i 元素现在描述为在普通文章中突出不同意见或语气或其他的一段文本，
例如：一个分类名称，一个技术术语，一个外语中的谚语，一个想法等。或者代表斜体的排版方式。

角度单位

deg

度 (Degrees) 。一个圆共360度

`90deg = 100grad = 0.25turn`

grad

梯度 (Gradians) 。一个圆共400梯度

`90deg = 100grad = 0.25turn`

turn

转、圈 (Turns) 。一个圆共1圈

`90deg = 100grad = 0.25turn`

rad

弧度 (Radians) 。一个圆共 2π 弧度

`90deg = 100grad = 0.25turn`

CSS 多类选择器

一个 `class` 值中可能包含一个词列表，各个词之间用空格分隔。例如，如果希望将一个特定的元素同时标记为重要 (important) 和警告 (warning) ，就可以写作：

```
<p class="important warning">
  This paragraph is a very important warning.
</p>
```

对于ID来说，就不能这样写了

所有浏览器下的CSS透明度

元素透明度时常是个恼人的问题，下面这种方式可以实现所有浏览器下的透明度设置：

```
.transparent {  
  zoom: 1;  
  filter: alpha(opacity=50);  
  opacity: 0.5;  
}
```

Zoom 属性是IE浏览器的专有属性，Firefox等浏览器不支持。它可以设置或检索对象的缩放比例。除此之外，它还有其他一些小作用，比如触发IE的 **hasLayout** 属性，清除浮动、清除 **margin** 的重叠等。

当一个元素的 **hasLayout** 属性值为true时，我们说这个元素有一个布局（layout）

css float属性

定义和用法

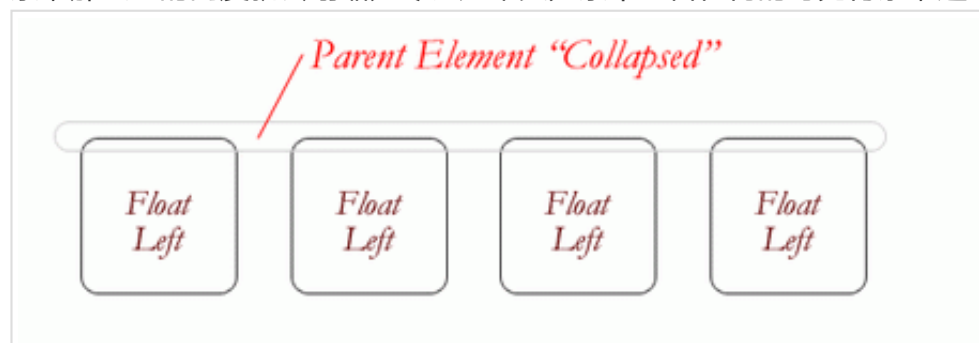
float 属性定义元素在哪个方向浮动。以往这个属性总应用于图像，使文本围绕在图像周围，不过在 **CSS** 中，任何元素都可以浮动。浮动元素会生成一个块级框，而不论它本身是何种元素。

如果浮动非替换元素，则要指定一个明确的宽度；否则，它们会尽可能地窄。

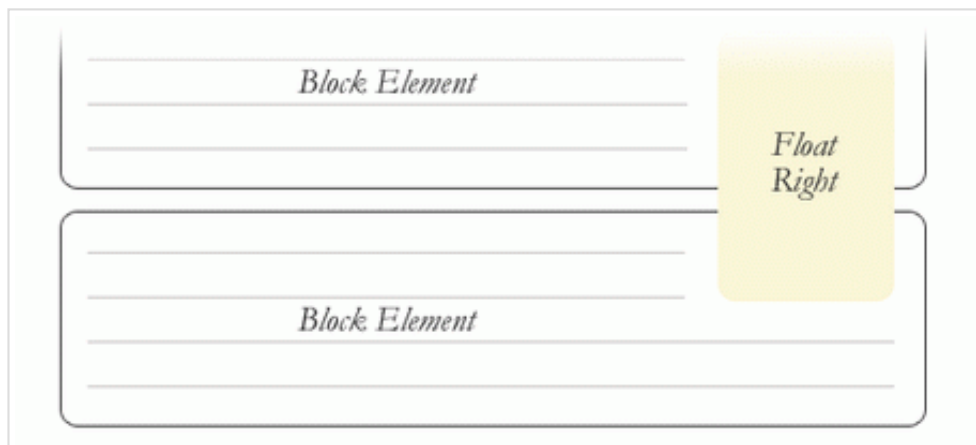
注释：假如在一行之上只有极少的空间可供浮动元素，那么这个元素会跳至下一行，这个过程会持续到某一行拥有足够的空间为止。

伟大的塌陷

使用浮动(float)的一个比较疑惑的事情是他们怎么影响包含他们的父元素的。如果父元素只包含浮动元素，那么它的高度就会塌缩为零。如果父元素不包含任何的可见背景，这个问题会很难被注意到



塌陷的直观对立面更不好，看看下面的情况：



清除浮动

如果你很明确的知道接下来的元素会是什么，可以使用 `clear:both`；来清除浮动。这个方法很不错，它不需要 `hack`，不添加额外的元素也使得它有良好的语义性。

使用css 伪选择符(:after)来清除浮动：

```
.clearfix:after {  
    content: ".";  
    visibility: hidden;  
    display: block;  
    height: 0;  
    clear: both;  
}
```

渐进增强与优雅降级

概念

优雅降级和渐进增强印象中是随着 `css3` 流出来的一个概念。由于低级浏览器不支持 `css3`，但 `css3` 的效果又太优秀不忍放弃，所以在高级浏览中使用 `css3` 而低级浏览器只保证最基本的功能。咋一看两个概念差不多，都是在关注不同浏览器下的不同体验，关键的区别是他们所侧重的内容，以及这种不同造成的工作流程的差异。

渐进增强 `progressive enhancement`：针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

优雅降级 `graceful degradation`：一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

区别

优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

visibility:hidden和display:none

display:none 隐藏对应的元素，在文档布局中不再给它分配空间，它各边的元素会合拢，就当它从来不存在。

visibility:hidden 隐藏对应的元素，但是在文档布局中仍保留原来的空间。

使用CSS **display:none** 属性后，HTML元素（对象）的宽度、高度等各种属性值都将“丢失”；而使用 **visibility:hidden** 属性后，HTML元素（对象）仅仅是在视觉上看不见（完全透明），而它所占据的空间位置仍然存在。

text 文本属性

text-indent

text-indent 属性规定文本块中首行文本的缩进。

注释：允许使用负值。如果使用负值，那么首行会被缩进到左边。使用一个较大的负值进行缩进的话，文字将被隐藏，如 **-9999em**

letter-spacing与word-spacing

letter-spacing 属性与 **word-spacing** 的区别在于，前者是修改字母的间隔，后者是修改单词或单个文字的间隔

text-transform

text-transform 属性处理文本的大小写。这个属性有 4 个值：

none

uppercase: 所有的单词都大写

lowercase: 所有的单词都小写

capitalize: 只对每个单词的首字母大写

text-decoration

text-decoration 有 5 个值：

- none
- underline: 下划线
- overline: 上划线
- line-through: 在文本中间画一个贯穿线
- blink: 让文本闪烁

white-space

- pre-line 合并空白符序列，但保留换行符
- normal
- nowrap 防止元素中的文本换行
- pre 空白符不会被忽略
- pre-wrap 保留空白符序列

博主近来穷的吃土了，如果看完本文的内容后觉得对你有所帮助，想博主更新的勤一点，你不扫一下吗（留言你想学的前端技术，博主有空了就更新啦）



<segmentFault/>



2014年11月12日发布

9 推荐

收藏

你可能感兴趣的文章

- 『译』 前端开发者手册 17 收藏，1.3k 浏览
- 《web前端开发修炼之道》学习笔记（1） 10 收藏，2.6k 浏览
- 从零开始设计技能树: 使用Graphviz建立模型 7 收藏，2.9k 浏览



本文采用 署名-相同方式共享 3.0 中国大陆许可协议，分享、演绎需署名且使用相同方式共享。

讨论区

行级元素的行高是可以设置的吧。line-height的计算不就是计算文本行中每个行级元素的高度。航迹元素的宽高以及上下的margin padding的确是不能设置的，除非 inline-block

代码狗 · 2015年01月13日

请先 [登录](#) 后评论



本文隶属于专栏

[trigkit4](#)

在人生的道路上，我们需要用行动来证明和兑现曾经心动过的梦想！<http://hawx1993.github.io/>



[trigkit4](#)
作者

关注专栏

相关收藏夹

[换一组](#)



[css布局](#)

4 个条目 | 0 人关注



[HTML+CSS](#)

5 个条目 | 0 人关注



[不只是 CSS 的笔记](#)

16 个条目 | 81 人关注

分享扩散：

...

Copyright © 2011-2016 SegmentFault. 当前呈现版本 16.05.04

浙ICP备 15005796号-2 浙公网安备 33010602002000号

移动版 桌面版