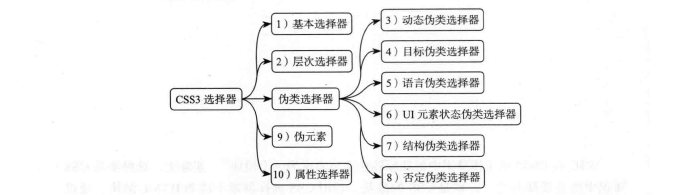
1. CSS3
   1. css3选择器

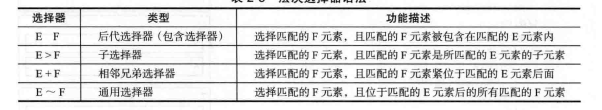
css3选择器有5大类：基本、层次、伪类、伪元素、属性。

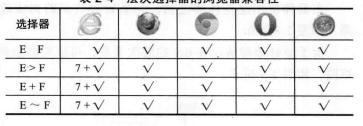


* + 1. 基本

“\*”、“E”、“#id”、“.class”、“class1，class2”5种。

* + 1. 层次





除了ie6都能用。

* + 1. 属性选择器

|  |  |
| --- | --- |
| E[attr] | 只使用属性名，但没有确定任何属性值 |
| E[attr=value] | 指定属性名，并指定了该属性的属性值 |
| E[attr~=value] | 指定属性名，并且具有属性值，此属性值是一个**词列表**，并且以空格隔开，其中词列表中**包含了一个value词**，而且等号前面的〜不能不写 |
| E[attr^=value] | 指定了属性名，并且有属性值，属性值是以value开头的 |
| E[attr$=value] | 指定了属性名，并且有属性值，而且属性值是以value结束的 |
| E[attr\*=value] | 指定了属性名，并且有属性值，而且属值中**包含**了value |
| E[attr|=value] | 指定了属性名，并且属性值**是value**或者**以value-开头**的值（比如说zh-cn）; |

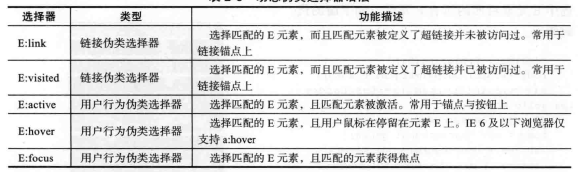
除了ie6都能用。

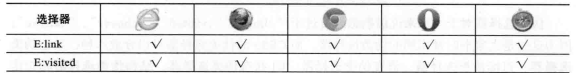
* + 1. 伪元素选择器

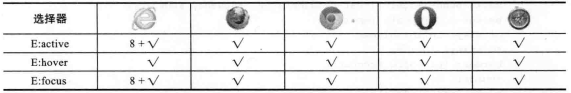
伪元素为DOM文档外的元素，css2中只有1个:，在css3中为2个:。“::before”、“::after”、“::first-letter”、“::first-line”所有浏览器都支持，不过ie只支持单:，其他浏览器两种都支持。

css3新增“::selection”为被鼠标选择的文本部分，如选中通常背景为蓝色，可以使用它来改变，只有现代浏览器支持。

* + 1. 动态伪类选择器







:hover、:visited都支持，其他ie下有比较大的问题。

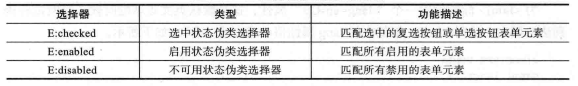
* + 1. 目标伪类选择器

“E:target”：即匹配如a标签等的target目标，现代浏览器支持。

* + 1. 语言伪类选择器

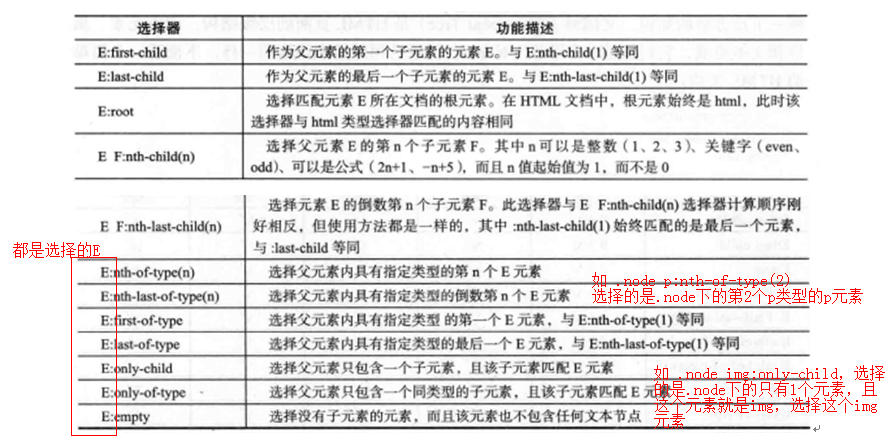
即，元素的lang属性，现代浏览器支持。

* + 1. 元素状态选择器



现代浏览器支持。

* + 1. 结构伪类选择器



现代浏览器支持。

这些结构伪类选择器和类选择器一样，如div.class，那么选择的是有class的这个div，如果是div .class，那么选择的是div下的有class样式这个元素。E:first-child选择的是E；E F:nth-child(n)选择的是F。

* + 1. 否定伪类选择器

“:not()”，如input:not([type=submit])。现代浏览器支持。

selectivizr是一个JavaScript工具，使IE浏览器也可以支持CSS3伪类以及属性选择器

* 1. 边框

在css2的border-width、border-color、border-style基础上，增加每条边的border-color、border-image、border-radius。

* + 1. 每条边border-color

基本都不支持。

* + 1. border-image

ie不支持。

* + 1. border-radius

现代浏览器都支持。

* + 1. box-shadow



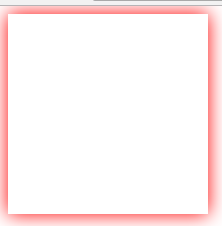
现代浏览器都支持。

投影方式只有1个选项inset，就是将外阴影变成内阴影。X-offset可以是正负值，正值，则阴影在对象的右边，反之在左边。Y-offset为正值，阴影在对象的底部，反之在顶部。阴影模糊半径和阴影扩展半径均为可选参数，两者独立，互不相干。



* 1. 背景

在css2的5个属性background-color、background-image、background-repeat、background-attachment、background-position基础上，又增加4个属性：background-origin、background-clip、background-size、background-break。

在原来css2的5个属性基础上，又增加4个属性。原来的5个属性为：

（1）background-color

（2）background-image

默认值为none，设置背景图片会覆盖背景颜色。

（3）background-repeat

值可以为：repeat-x、repeat-y、repeat、no-repeat，默认值为repeat。

（4）background-attachment

值可以为scroll或fixed，默认值为scroll。表示当页面有scrollbar滚动时，背景图片是否随着滚动。除在ie6中，fixed属性值只支持html和body外，其他浏览器都支持所有元素。

（5）background-position

可以为(x,y)、(left/center/right/top/bottom,～)，默认值为(0,0)。

* + 1. background-origin

需要和background-position属性一起使用，用来改变background-position的计算方式，值可以为padding-box、border-box、content-box。现代浏览器都支持。

* + 1. background-clip

定义背景图片的裁剪区域。现代浏览器都支持。

* + 1. background-size

指定背景图片水平和垂直方向的大小。值有：

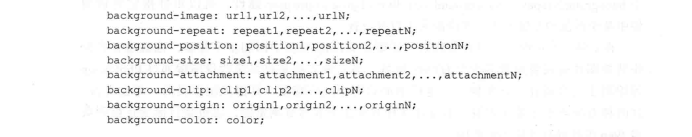


* auto：默认值，保持原始大小。
* int：指定显示多少px。
* percentage：以容器宽高百分比来设置背景图像的宽度和高度，不会按比例，从而导致失真。
* cover：按比例调整背景图片，按最小边扩展，使之铺满整个容器（从而导致较大边丢失）；或按最大边缩小（从而导致较小边丢失）。
* contain：按比例调整背景图片，始终按最小边调整，从而可能出现空白。
  + 1. background-break

仅firfox支持。

* + 1. css3多背景

设置多个背景图片，在background-image中指定多个地址，用逗号隔开。其他属性也用逗号隔开分别控制对应顺序的图片表现。



现代浏览器都支持。

* 1. 文本

文本功能有3大类：字体、颜色、文本。

字体重要的属性有：font-family、font-weight、font-style、font-size，可以组成复合属性：font:font-style font-weight/line-height font-family。

颜色属性为color。

文本属性重要的有：text-align、vertical-align、line-height、text-decoration、direction、word-sapcing、letter-spacing、white-space。

* + 1. text-shadow

css2中便有该属性，但现代浏览器才支持。

一共有4个属性参数：

* color：如果没有设置则为文本自身的颜色。
* x/y-offset：x/y轴偏移，可以为正负值。
* blur-radius：阴影向外模糊半径，为0时则不具有模糊效果（仍然有阴影）。

默认值为none。

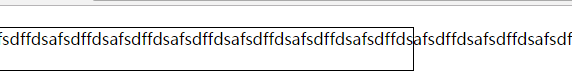
* + 1. text-overflow

属性值可以为：clip，只裁剪，不显示...。ellipsis，裁剪，显示...。所有浏览器都支持。

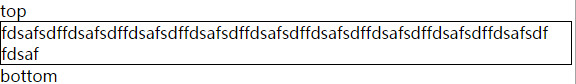
* + 1. 文本换行

文本会自动换行。英文会在空格或标点符号处换行，不会在单词中间换行。中文可以在任何地方换行。

默认情况下，对于block元素，里面内容超过其宽度后，不会撑开，但内容会显示。



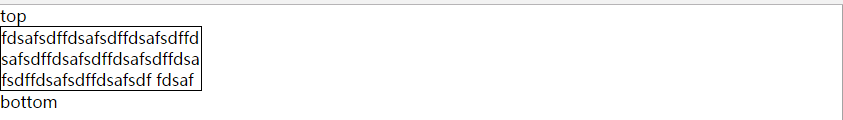
对于inline-block元素，内容会撑开。



对于文本换行，可以使用如下属性控制：

* word-wrap

属性值有：normal、break-word。可以实现长单词和url地址的自动换行。所有浏览器都支持。



* word-break

ie支持比较好。属性值有：normal（默认值）、break-all（可强制截断单词）、keep-all（不强制截断单词）。

* white-space

基本都支持，可以使多个单词都不换行。属性值有：normal（多个空白视为1个空白）、pre（多个空白的话都被保留）、nowrap（多个空白视为1个空白、且无论多个单词都1行显示）、pre-wrap（多个空白都保留，且也不换行）。

综上，word-wrap和word-break可以让单个单词截断换行，white-space可以然多个单词1行显示。因此，white-space:nowrap常和text-overflow：ellipsis。

* 1. 颜色
     1. web页面的安全色

显示器由像素表现色彩。不同型号电脑可能不同位的元色彩来计算色彩，如8位、16位等。不同浏览器有不同的调色板，因此，对于不同颜色，浏览器会选择调色板中最接近的色彩来显示。约定用16进制来组合了216中颜色，可以安全应用于所有web浏览器。

* + 1. 色彩模式
* RGB色彩模式

使用光的3原色红、绿、蓝混合产生。

* CMYK色彩模式

使用颜料的3原色青色、洋红、黄色加上黑色。这4中颜色混合主要用于印刷。

* 索引色彩模式

被限定在256种颜色以内的模式，主要用于web安全色彩和制作透明GIF图片。使用ps制作GIF图片时，必须使用索引色彩模式。

* 灰度模式

即无色彩模式，用于处理黑、白、灰色图片。

* 双色调模式

在黑白图片中加入颜色，RGB、CMYK都不能直接转为双色调模式，必须先转为灰度模式，再转入双色调模式。占用空间较小。

* 位图模式

也是使用黑白共同处理颜色。RGB、CMYK也都不能直接转为位图模式，必须先转为灰度模式，再转入位图模式。

* + 1. opacity

用于呈现半透明效果。值可以为0-1之间的任意浮点数，或为inherit。

* opacity和alpha通道

alpha通道用于对元素设置透明度，针对背景色、边框颜色等。opacity则是对整个元素设置1个透明度。

* + 1. CSS3颜色模式
* RGBA颜色模式

在RGB颜色模式基础上，增加alpha通道透明度参数。现代浏览器都支持。

* HSL颜色模式

即，通过色调（H）、饱和度（S）、亮度（L）3个颜色通道控制颜色。现代浏览器都支持。

* HSLA颜色模式

在HSL颜色模式基础上，增加alpha通道透明度参数。现代浏览器都支持。

颜色模式的选择？

* 1. 盒模型
     1. box-sizing

属性值有：content-box、border-box和inherit。现代浏览器都支持。

* + 1. overflow-x/y

值有visible/hidden/scroll/auto（需要时才会出现滚动条）/no-display（不显示元素，等同于设置了none）/no-content（不显示内容，等同于设置了visibility:hidden）。

所有浏览器都支持。

* + 1. resize

用户可调整元素大小。ie不支持。

* + 1. outline

元素外围的轮廓线。与border相似，可以设置的属性有：outline-color、outline-width、outline-style、outline-offset、inherit。

现代浏览器都支持。

* 1. flex布局
     1. 概念

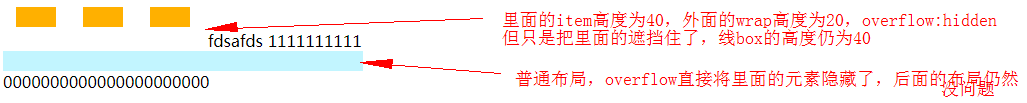
**1.容器/项目**

容器包裹项目，由display定义，可以为flex或inline-flex，分别代表块元素和内联块元素（也就是说可以设置宽度、margin啥的）。有如下要点：

（1）容器内的每1个子元素都是容器的项目。可以查看子元素的属性，没有1个属性要求定义说这是子元素，即，只要包裹在flex中，它就是子元素。

（2）子元素无论是div还是span，都不再有原来的块元素、内联元素性质。准确而言，这些子元素都类似于inline-block。对于width，哪怕是div，如果没有设置宽度，因flex-basis默认值为auto，flex-grow默认值为0，因此它的默认宽度就是0。对于height，如果没有设置，且没有设置align-items，因align-items默认值为stretch，会撑满整个父元素高度，这个与inline-block不同；如果设置了不是stretch值的align-items，则height为0。

（3）flex容器和元素一样，如果没有定义height，将由子元素撑开。如果定义了height，则和其他的不同，其他的样式布局、显示都由overflow决定；而flex只是显示由overflow决定，布局仍然按子元素高度来。如果子元素都定义了height，则如下：



如果有1个没有定义，那么和普通的一样。

（4）子元素float、clear、vertical-align无效。

**2.主轴/侧轴**

与过去的流体布局不同，flex自定义布局方向。每条主轴/侧轴都有方向、起点、终点、长度、长度属性等属性。

**3.容器属性**

可以定义容器元素的主轴方向、换行、对齐，以及侧轴对齐等。

**4.项目属性**

可以定义项目元素的伸缩（大小）属性、对齐、排序。

其他细节可以参考：

http://www.w3cplus.com/css3/flexbox-layout-and-calculation.html

* + 1. 容器属性

**1.display**

决定容器是以块状（flex）还是行内元素（inline-flex）的方式显示。

**2.flex-direction**

定义项目排列方向，由此也定义了主轴和侧轴（没有必要定义为column、column-reverse，使用纵轴属性也一样啊）。值有：

* row：从左到右。
* row-reverse：从右到左。
* column：从上到下。
* column-reverse：从下到上。

**3.flex-wrap**

是否折行/列。默认保持在1行/列内。值有nowrap：不换行/列；wrap：换行/列；wrap-reverse：反方向换行/列。

**4.flex-flow**

flex-direction和flex-wrap属性的简写。

**5.justify-content**

项目main对齐方式。值有：

* flex-start（默认）：main轴开始方向对齐。
* flex-end：main轴结尾方向对齐。
* center：main轴居中对齐。
* space-between：保持在一行，main轴两端对齐。
* space-around：等间距对齐（第1个和最后1个前后也有距离）。

**6.****align-items**

项目侧轴对齐方式。值有：

* flex-start：cross轴开始方向对齐。
* flex-end：cross轴结尾方向对齐。
* center：cross轴居中对齐。
* baseline：文本的基准线对齐。
* stretch（默认）：在cross轴方向填满整个容器（item如果设置该轴方向固定尺寸时不会生效）。

**7.align-content**

多行/列时，所有行/列的对齐方式。值有flex-start、flex-end、center、space-between、space-around、stretch。

* + 1. 项目属性

**1.order**

排序顺序

**2.flex-grow**

尺寸扩展，接收一个无单位的数值作为比例，默认值0。在分配完容器内非项目的尺寸之后的剩余尺寸分按这个比例分配。记住是剩余尺寸。即，可以对item设置width，如果flex也有width，那么就有剩余空间。flex-grow的值就是和剩余空间的比值。比如，剩余空间为100px，如果设置为0.5，就是50px，最大貌似不能超过1。

**3.****flex-shrink**

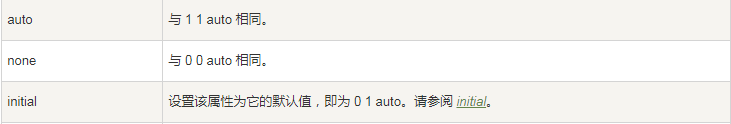
当空间不够时，才缩小item的尺寸，否则并不缩小，默认值1。如果为0，则不缩小。这个值和剩余空间不是比例关系，而是与所有item的这个值相关。如果都是1，则都按同比例缩小，一直到满足容器宽度为止。

**4.flex-basis**

初始尺寸，和width或height属性一样，可以不用，优先级高于width和height。默认值auto，即，如果该项目未指定长度，则长度将根据内容决定。

**5.flex**

是flex-grow、flex-shrink、flex-basis简写方式。flex-shrink和flex-basis可以省略

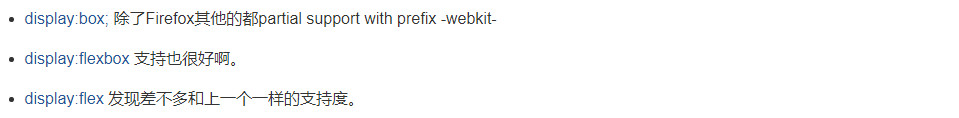


**6.align-self**

定义自己在容器中的对齐方式，可以覆盖容器设置的对齐方式。

* + 1. 兼容性

前后主要有3个版本，上面的是最新版本，之前还有display=box、display=flexbox等写法。



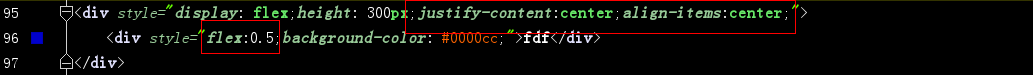
最新的浏览器基本都能使用最新版本，但需要加ms、webkit等前缀。通常使用网上的一些库来兼容所有浏览器和所有版本。没必要兼容box、flexbox写法。如果非要兼容，可以参考如下：

https://github.com/codio/Flex.less/blob/master/flex.less

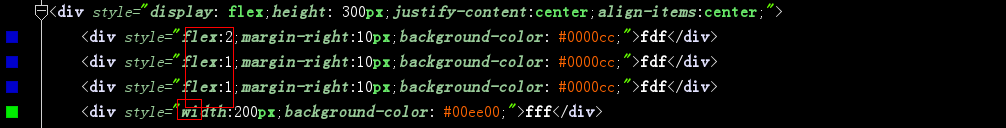
https://zhuanlan.zhihu.com/p/21938048?refer=jscss

https://zhuanlan.zhihu.com/p/21640023

* + 1. 实例

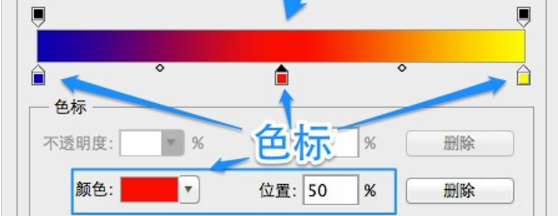








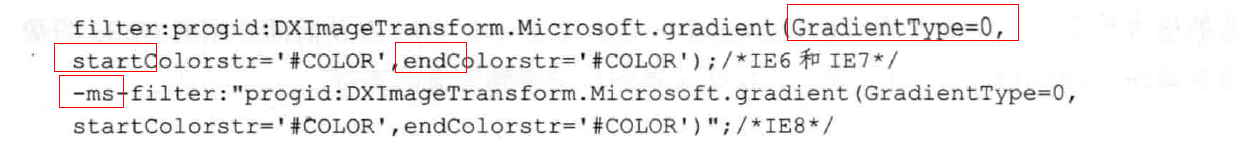
* 1. 多列布局
  2. 渐变
     1. 色标



在创建渐变过程中，可以指定多个中间颜色值，即色标。每个色标包含1种颜色和1个位置。浏览器从1个色标淡出到下一个，从而创建平滑的渐变效果。

* + 1. 兼容性

ie10及现代浏览器都支持，不过都得使用自己的前缀。ie9及以前版本，可以使用滤镜实现。支持最好的属性是background-image。



* + 1. 线性渐变
* webkit

-webkit-linear-gradient([<point>|<angle>]?,<stop>,<stop>[,<stop>]\*)，一共有3个参数：

* point：left/top/right/bottom关键字组合（可以是1个或2个）。
* <angle>：通过角度确定方向，0表示从下到上，90表示从左到右。
* <stop>：色标，需要2个值1组，位置和颜色，位置可以为0-1之间的浮点数，0表示起点，1表示终点。
* firefox

-moz-linear-gradient([<point>|<angle>]?,<stop>,<stop>[,<stop>]\*)，参数同上。

* opera

-o-linear-gradient([<point>|<angle>]?,<stop>,<stop>[,<stop>]\*)，参数同上。

* ie

-ms-linear-gradient([<point>|<angle>]?,<stop>,<stop>[,<stop>]\*)，参数同上。

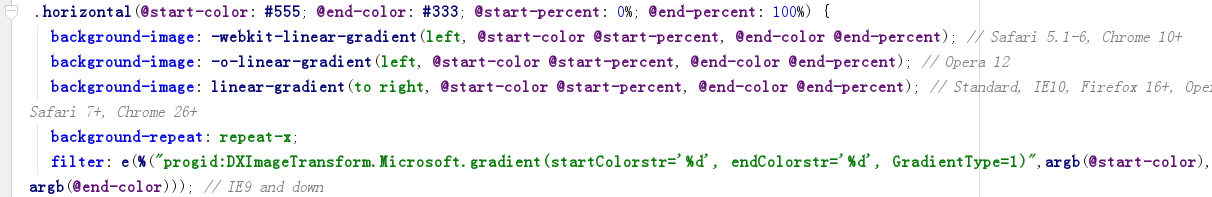
* w3c标准

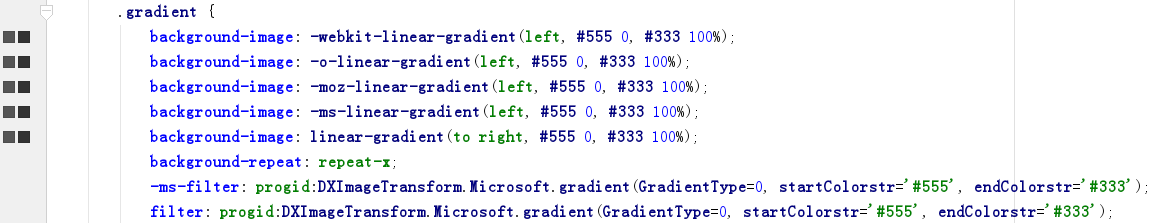
linear-gradient([[<angle>|<side-or-corner>],]?,<color-stop>[,<color-stop>]\*)，主要有3个参数：

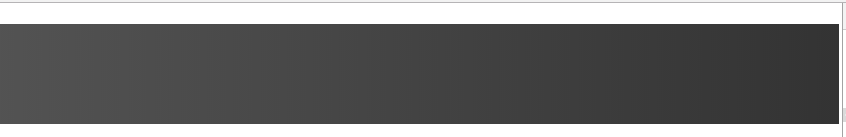
* <angle>：通过角度确定方向，0表示从下到上，90表示从左到右。
* <side-or-corner>：关键字to top/left/right/bottom。
* color-stop：色标。

现代浏览器都支持标准写法。

如，bootstrap中有mixin如下：







ie下：



* + 1. 径向渐变

-webkit-radial-gradient([<position>|<angle>]?,[<shape>|<size>]?,<stop>,<stop>[,<stop>]\*)。

-moz-radial-gradient([<position>|<angle>]?,[<shape>|<size>]?,<stop>,<stop>[,<stop>]\*)。

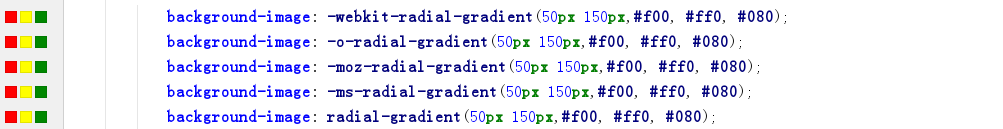
-o-radial-gradient([<position>|<angle>]?,[<shape>|<size>]?,<stop>,<stop>[,<stop>]\*)。

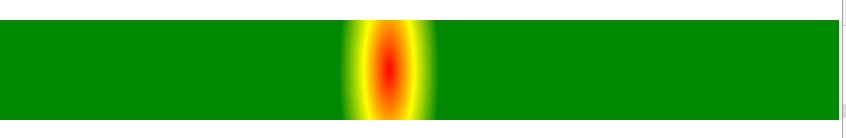
-ms-radial-gradient([<position>|<angle>]?,[<shape>|<size>]?,<stop>,<stop>[,<stop>]\*)。

radial-gradient([<position>|<angle>]?,[<shape>|<size>]?,<stop>,<stop>[,<stop>]\*)。

参数：

* position/angle：同linear。
* sharp：渐变形状，可以为circle圆形渐变或eclipse椭圆渐变。
* size：渐变结束形状，值可以是：
* closest-side：指定径向渐变的半径长度为从圆心到离圆心最近的边。
* closest-corner：指定径向渐变的半径长度为从圆心到离圆心最近的角。
* farthest-side：指定径向渐变的半径长度为从圆心到离圆心最远的边。
* farthest-corner：指定径向渐变的半径长度为从圆心到离圆心最远的角。
* 显示指定length/percentage。
* stop：色标。





* 1. transform
     1. 变形函数及兼容性

2D的变形函数有：

translate移动、scale缩放、rotate旋转、skew扭曲，现代浏览器都支持。

3D变形函数有：

rotateX、rotateY、rotate3d、translateX、translateY、scaleX、scaleY、skewX、skewY、矩阵matrix。

* 1. transition

过渡，可替代js实现位移动画。transition属性是一个简写属性，语法为transition:property duration timing-function delay。ie10及现代浏览器支持。

不是所有的property都支持。

timing-function可以为linear、ease（渐慢）、ease-in（渐快）/out（渐慢）、ease-in-out（先加速后减速）。

触发过渡可以是:hover、:active等伪元素触发，也可以是媒体查询触发，或者使用js触发。

* 1. animation

animation也是1个复合属性，简写形式为animation:name duration timing-function delay iteration-count direction。

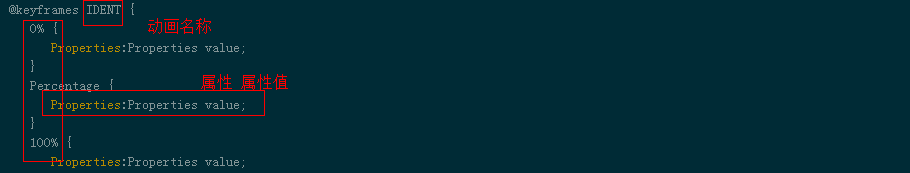
* name：关键帧动画名称，必须和1个@keyframes规则对应。
* duration：动画持续时间。
* timing-function：动画播放方式，和transition的timing-function相似。
* delay：延迟时间。
* iteration-count：循环次数。
* direction：动画播放方向。

ie10及现代浏览器都支持。

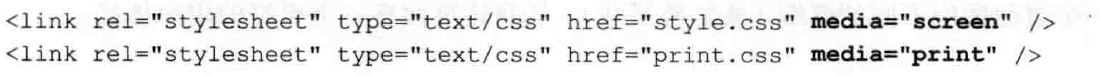
* + 1. 关键帧

animation制作动画分为两步：首先通过关键帧Keyframes声明1个动画，然后用animation调用这个动画。

Keyframes语法：命名由@keyframes开头，后面紧接着动画的名称和{}。{}是一些不同时间段样式规则。1个@keyframes样式规由多个0%到100%之间的百分比构成，然后在每个百分比中定义需要有动画效果的元素和样式，可以使用from、to，和0%、100%意思一样。加上百分符号。



* 1. 媒体类型和response设计
     1. 根据不同媒体类型获取不同css文件
* link方法

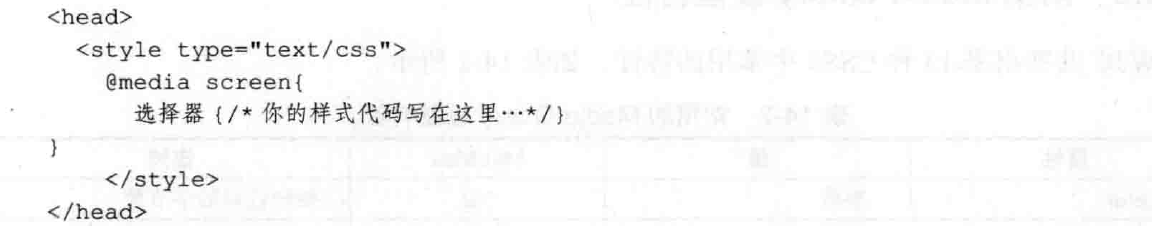


* @import方法



* @media方法

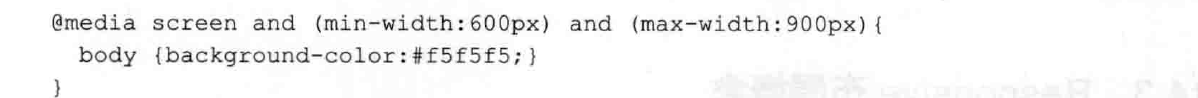




* + 1. 媒体特性（media query）

现代浏览器都支持。





* + 1. meta标签

由于智能手机使用了比屏幕尺寸大很多的虚拟可视区域，为此，需要在meta中定义相关设置viewport，控制网页显示区域。

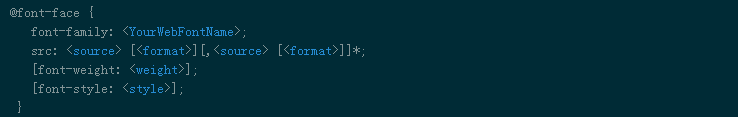


content中可以定义如下属性值：

* width：可视区宽度，可以为数字、device-width。
* height：可视区高度。
* initial-scale：初始显示时缩放级别，1.0为实际尺寸。
* maximum-scale：最大显示级别，如为1.0则表示禁止用户放大到实际尺寸之上。
* user-scale：用户是否可以放大缩小页面，no为禁止。
  1. @font-face

浏览器根据指定命令，将服务器端的字体下载到客户端缓存，并嵌入web页面。所有浏览器都支持。

* + 1. 语法



* YourWebFontName：字体名称，对应font-family，如font-family:"YourWebFontName"。
* source：字体存放路径，可以是相对路径也可以是绝路径。
* format：字体的格式，用来帮助浏览器识别，其值主要有以下几种类型：truetype、opentype、truetype-aat、embedded-opentype、avg等。
* font-weight：字体是否为粗体。
* font-style：字体样式，如斜体。
  + 1. 字体格式
* **.ttf**

Windows和Mac中最常见的字体,支持的浏览器有IE9+、Firefox3.5+、Chrome4+、Safari3+、Opera10+、iOS Mobile Safari4.2+。

* **.otf**

1种原始的字体格式，ie不支持。

* **.woff**

Web字体中最佳格式，支持的浏览器有IE9+、Firefox3.5+、Chrome6+、Safari3.6+、Opera11.1+。

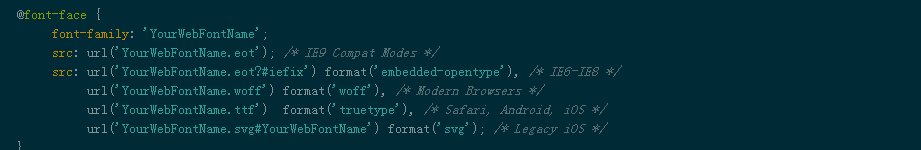
* **.eot**

IE专用字体，IE4+支持。

* **.svg**

ie不支持。

因此，至少需要配置.woff和.eot两种格式字体，甚至还需要.svg等字体达到更多种浏览版本的支持。



* + 1. 创建字体

首先，获取某种字体，要么付费购买，要么下载免费的。

然后，登录第三方工具，将字体转换为.woff和.eot及更多格式字体。

然后，调用字体。

* + 1. 创建图标字体

首先，创建矢量插图，输出为SVG格式。

然后，导入到icoMoon web app，导出字体。

然后，使用。