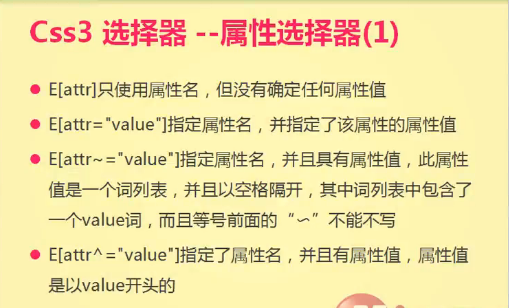
一、属性选择器



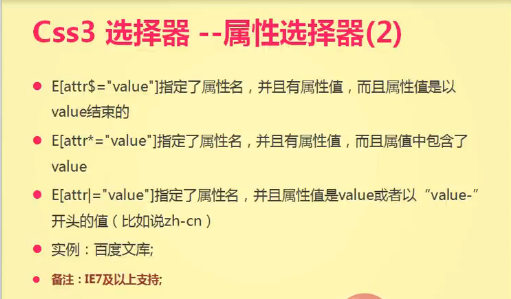
1. E[attr]：这种方法是通过attr属性来选择所有具有该属性的所有E元素；要是同时用方括号跟几个属性，那么就是需要同时具有那几个被写出来的属性才能被选中。



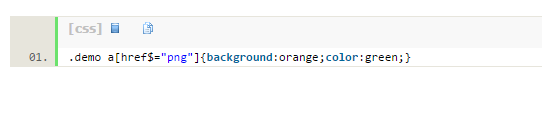
1. E[attr=“value”]：只有属性值等于value的E标签（元素）才能选中；同理，这个方法也可以多个属性并写，进一步缩小选择范围。（IE6不支持）还需要注意，这个属性名和属性值必须匹配，例如class列表有两个词，那么在选择的时候就需要写全两个，中间空格不漏掉，这样才能选择中，否则选择器就不会起作用，这种选择方式属于完全匹配，要是有一点没对上，都不能够选中。
2. E[attr~=“value”]：自定属性名，并且该属性的值是一个词列表（含有多个关键词），只要在这个词列表中有一个属性值和所写value匹配，那么就能够选中该元素，否则不能选中。

上面两个选择器总结起来就是，有~的只要在列表中有一个属性值能够匹配，那么就能选中元素；要是没有~那么就需要完全匹配才能选中。

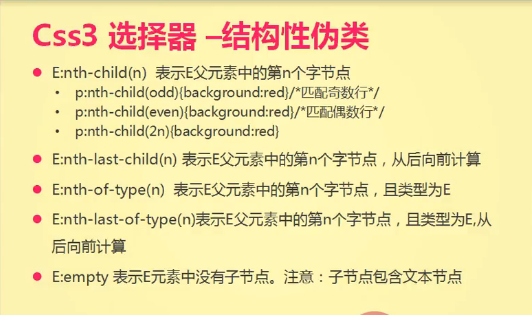
（4）E[attr^=“value”]：就是选择那些属性值以value开头的那些元素。



1. E[attr$=“value”]：这个选择方法和上面一个恰好相反，选择的是那些以value值结尾的元素。这个方法的好处就是给某一类特定的标签或者图片之类的添加样式就会很方便。



1. 结构选择器



1、

1. E：nth-child（n）

表示从前往后找E的父元素中的第n个子节点，并且这个子节点不一定是E类型的标签。其中的n可以像数学中的n一样，从0一直到无穷，因此可以使用。

1. E：nth-last-child（n）

表示E的父元素中的第n个子节点，并且这个子节点不一定须是E类型的标签；但是这个方法是从后往前数（找）第n个元素。

（3）E：nth-of-type（n）

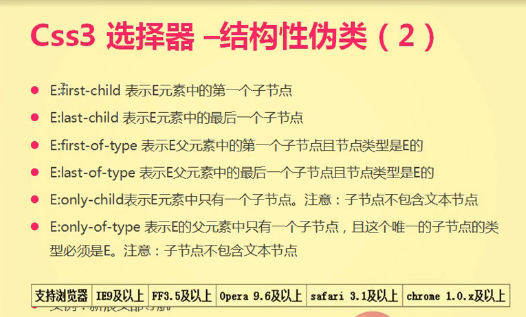
表示找到E元素父级当中类型为E的第n个子节点，并且需要和E的类型一致；

1. body \*：nth-of-type（n）

表示找到body的所有元素中的每种元素的第n个子标签；（注意和上面的区分，上面的还需要找到的和E类型一致，这里没有限定类型）

（5）E：nth-last-of-type（n）

表示找到E元素父级当中类型为E的倒数第n个子节点，同时类型应该和E保持一致。



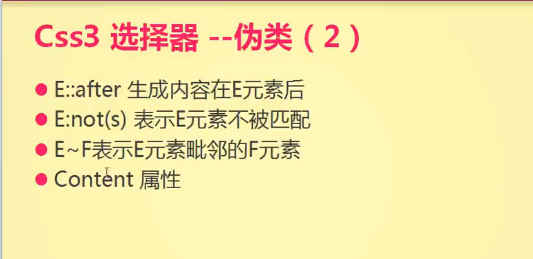
等价写法（意义相同）

1. E：first-child==E：nth-child（1）
2. E：last-child==E：nth-last-child（1）
3. E：first-of-type==E：nth-of-type（1）
4. E：last-of-type==E：nth-last-of-type（1）
5. E：empty——表示E元素中没有子节点（也就是一个空元素），并且空标签还不包括文本节点。
6. 注意，of-type的重点是需要匹配标签的类型，就是选择的元素类型要和指定的元素类型一致；而child则只是关注父节点下的子节点，不关注元素的类型是否一致。
7. 伪类

伪类存在的意义是为了通过选择器找到那些存在于DOM树中的信息以及不能被常规CSS和获取到的信息。



（1）E:target——它的作用是匹配文档(页面)的URI中某个标志符的目标元素；



1. CSS3新增文本属性



1. 在原有的rgb上面新增了一个raga，也就是新增了一个透明度。这样就可以给背景颜色添加透明，但是内容不透明的效果。这样，只要是自己设置颜色的地方，使用rgba就可以设置透明度。（注意：opacity是0-1从完全透明到不透明过渡的）；

在使用rgba和opacity的不同之处就是前者不会影响内容的透明度，但是后者会影响，例如文字，在前者只是颜色变化，但是文字透明度不会变化，后者就会导致文字透明度变化。



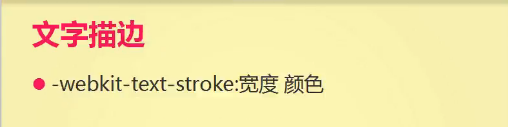
1. 首先横向偏移和纵向偏移都可以去负值，分别对应了向x、y轴的两个方向，在取正值的时候，阴影是向正轴方向的，取负值则是向负轴方向。
2. 关于阴影叠加

就像和box-shadow一样，使用的方法和效果都是一样的，就是按照规则多次书写样式；

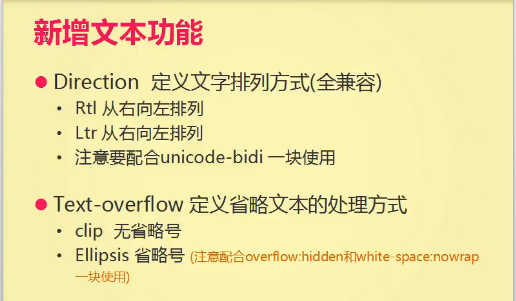


1. 文字描边

只有-webkit的内核的浏览器才支持。



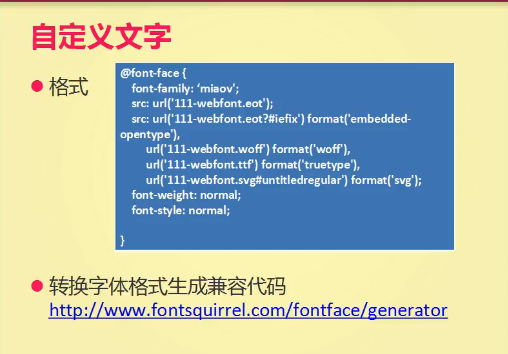
5、文字排版方向



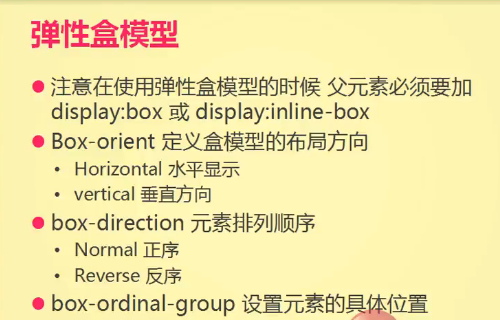
1. direction就是文字的对齐方向，就相当于将文本的父元素进行了浮动。
2. 首先要注意的是，必须是直接将样式添加在包含有文本的元素上，不能跨级添加；其次是必须要注意和overflow：hidden和white-space：nowrap配合使用，否则就没有效果。



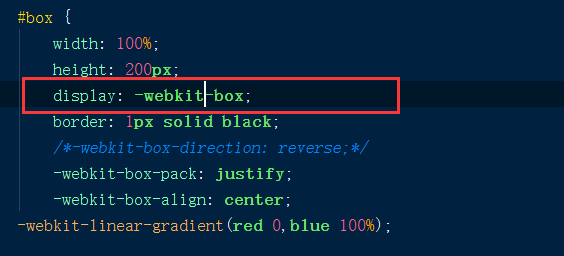
6、自定义字体



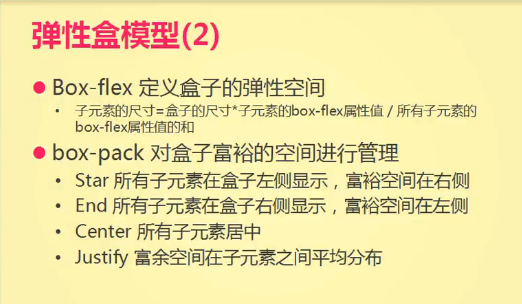
1. 盒模型设计、响应式布局
2. 弹性盒模型



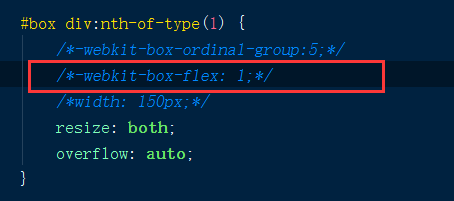
1. 在使用弹性盒模型的时候，需要在box等各种key值前面添加各个浏览器的内核前缀（webkit、Gecko等），这样才能实现相应的效果。

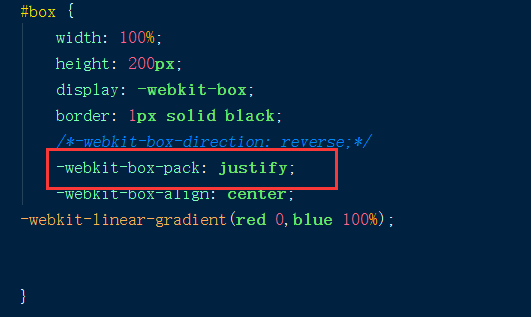


1. 盒模型的布局方向默认的是水平方向。（样式需要添加在父元素上）
2. 元素排列顺序分别是正序和反序，默认是正序排列；（样式需要添加在父元素上）

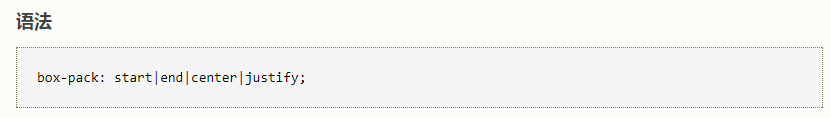


1. 弹性空间管理可以很好地运用于移动端，让各个元素按照一定的比例来进行分配空间尺寸。首先需要注意的是，这里的弹性空间管理是在父元素的基础上，减去某些固定尺寸之后再按照比例分配父元素剩下的空间，也就是在那些元素上添加比例。（样式添加在各个子元素上，哪一个需要哪一个就添加）





1. 富余空间管理添加在父元素上面，这三个属性star和end分别和浮动相类似。

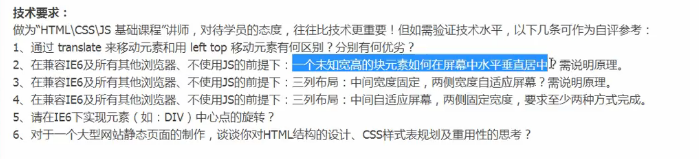


Justify：自适应，就是将剩余空间按照等比例进行分配。

Start/end/center：分别对应了左对齐、右对齐和居中（包括水平和垂直居中）

6、元素对齐方向管理





综上，所有的关于弹性盒布局的属性，除了灵活的去分配每一个子元素的大小（-webkit-box-flex:2;）以外，其他的一些属性都是添加在需要被排版的元素的父级上面的，因此在使用相关属性的时候需要注意样式的书写位置。

**补充：**

1、为什么使用浏览器前缀？

由于不同浏览器的内核不同，因此各个浏览器解释网页、执行代码的方式也会不同，因此，为了解决某些兼容性问题，需要使用浏览器前缀。

1. 何为浏览器内核？有什么作用？

浏览器内核实际上就是“渲染引擎”，渲染引擎决定浏览器渲染网页内容的方式，以及页面的格式信息。

浏览器内核对于浏览器而言，是基础，是依托。如果没有了浏览器内核，那么浏览器是无法独立存在且产生作用的。它的存在，决定了网页的呈现的内容、格式以及效果。所以说，一个好的浏览器，一定是基于有一个稳定、高端、作用明显的浏览器内核的。

1. 盒模型新增属性



1. 盒模型阴影——box-shadow

box-shadow：inset X Y blur color （分别是x、y上的偏移量，blur是模糊半径）

inset：当这个值被添加上的时候，那么阴影就在元素内部；

Spread：是在原有的尺寸上先扩展之后在开始模糊。

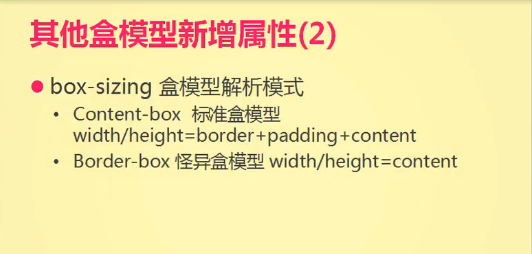
这个方法和text-shadow是一样的，书写叠加的阴影的时候用逗号隔开，按照规则书写就可以了。



2、box-reflect：倒影的结果就是和原来的元素之间呈现镜像关系，这个属性的使用结合渐变是一个很不错的选择。

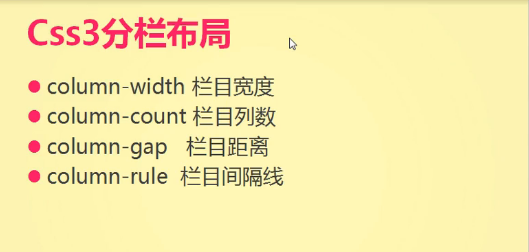
3、resize属性再说会用的时候必须要和overflow：auto搭配使用，否则将不能实现该效果。Resize的值是上面的那三个，依据需求而定，到底是用哪一个值。

4、盒模型



记住，标准盒模型的宽高是指的content的宽高，而在怪异盒模型下盒子的宽高就包含了padding和border的值。

4、响应式布局



（1）分栏布局需要注意，还是需要添加浏览器内核前缀。

（2）在分栏布局的时候，要是父元素宽度一定了，那么将会优先根据分栏的数量来自动确定每一栏的宽度；要是没有确定分几栏，那么就根据宽度来自行调整。

（3）栏目间的分割线就像border一样，写法一样，效果也一样。

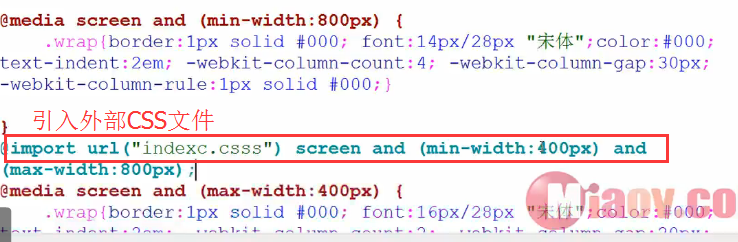
1. 响应式布局

1、不同屏幕宽度

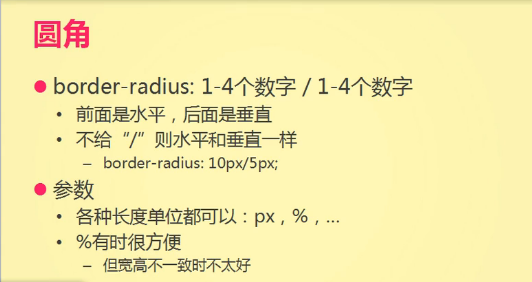


2、横、竖屏切换





1. 新的UI方案
2. 圆角——border-radius



这个属性的值在得到规则图形的时候可以有1-4个，分别对应每一个角；当使用/来分割的时候，/两边的值分别对应的是x轴和y轴的半径。

在画圆形的时候使用百分比就会很方便，这也就是说宽高一致的时候百分比就比较好用。

1. 边框背景——border-image



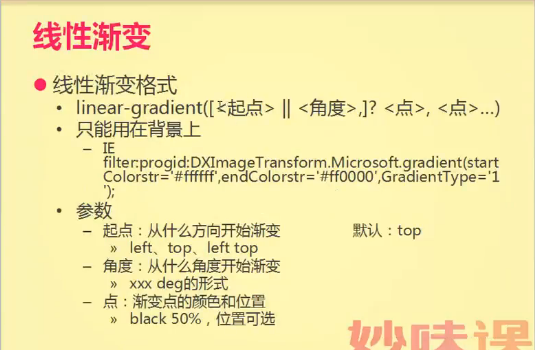
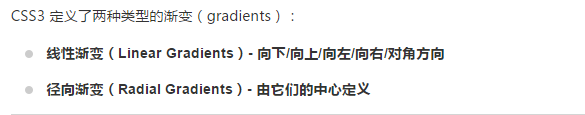
1. 盒子阴影——box-shadow

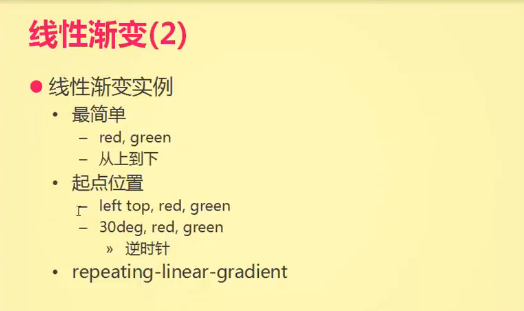
分别是对应了四个参数，前两个是X、Y方向上的阴影方向，当取负数的时候就是在左边和上边有阴影；第三个参数就是模糊半径，也就是阴影的宽度；第四个参数就是阴影的颜色。

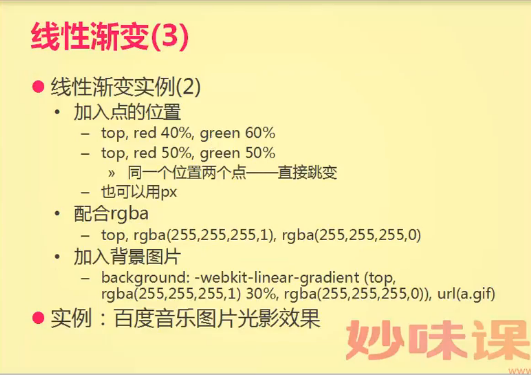
1. border-image

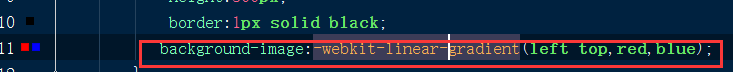
这个属性有四个参数，第一个URL，是图片的存储路径；第二、第三是背景图片的宽高；第四个参数是round/stretch，分别是图片平铺和拉伸。

5、线性渐变——这个属性可以让我们在两个或多个颜色之间实现平滑过渡

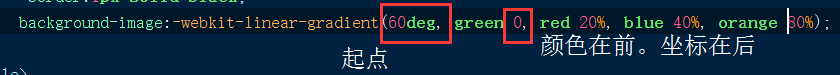




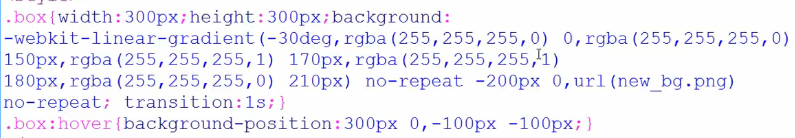




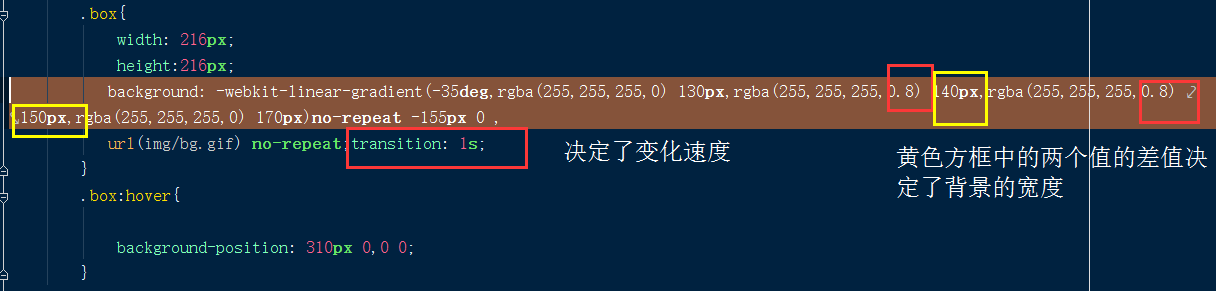
1. 起点是可以控制的，一是可以使用关键字，另一个是使用角度deg；
2. 要是需要更加精确地控制颜色，可以使用百分比或者坐标来控制，到什么地方是什么颜色，注意是颜色在前，坐标在后，这样就比较精确地控制。

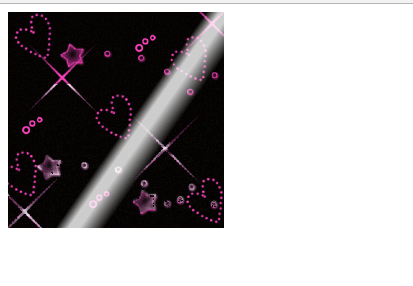


1. 颜色可以是多个叠加，逗号隔开就好；



1. 实例：图片光影效果



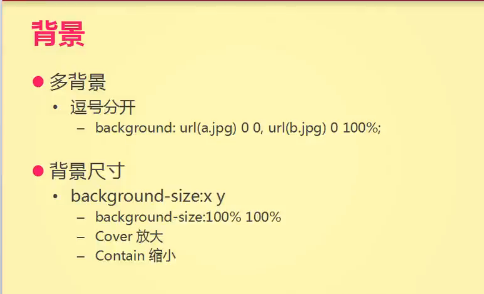


（途中的光斑是可以运动的，从左上角到右下角）

1. 径向渐变



1. 注意的一点是，径向渐变的起点已经没有度数了，只能是关键字或者具体数字以及百分比，因为这个径向渐变本来就是圆的。
2. 所谓起点的设置就是指的是第一种颜色开始渐变的位置。
3. 背景



1. 多背景的显示顺序就是书写的顺序，从前往后显示在元素中。
2. 背景尺寸——background-size：x，y分别对应的是背景的宽高，可以是具体的数字，也可以是百分比。还有两个关键字，cover就是放达到填满整个元素并且有可能超出；而contain则是刚好能够让元素能把背景装下，有可能背景并不能将元素填满。
3. 背景图源点设置background-origin：设置背景开始显示的点。
4. background-position：对背景图片的位置进行控制。要是有多个背景，那么也是按照书写书顺序来设置参数。
5. background-repeat：设置图片背景是否平铺；



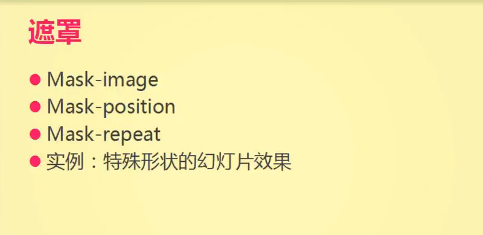


1. background-clip——背景剪裁

背景剪裁的属性值和上面一样有两个，分别是background-clip: padding-box;content-box;效果分别表现为：背景往外延伸到padding，背景只到content区域。

（7）background-origin——显示背景图片的位置，属性值分别是content-box和border-box；

（8）遮罩



1. 2D过渡
2. 过渡——transition

要实现过渡效果，必须要满足两个条件，第一是指定过渡后的样式，第二就是指定过渡的时间长度，要是时间长度不指定，那么将不会有任何的过渡效果，因为其默认时长度是0。

浏览器支持：Internet Explorer 10, Firefox, 和 Opera支持transform 属性.

Chrome 和 Safari 要求前缀 -webkit- 版本.

注意： Internet Explorer 9 要求前缀 -ms- 版本.

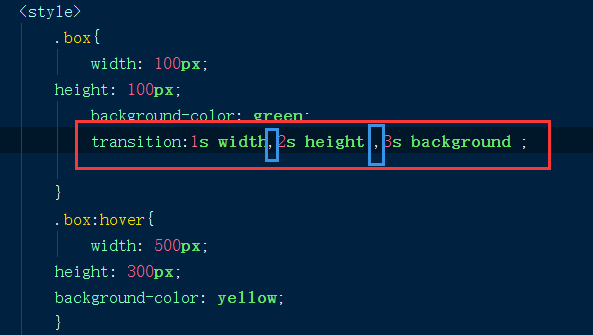


1. 过渡是发生在样式改变的时候，哪一个元素的样式再发生改变，那么哪一个元素的样式中就添加过渡样式、过渡时间transition：\*s。

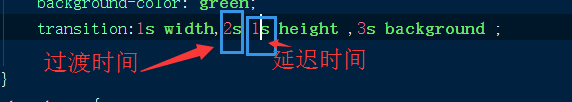
这个过渡的意思，简单理解就是说从当前样式变化到希望改变成的样式这的个过程需要的时间。其中可以分别控制那些属性的过渡时间，例如宽高、背景等。

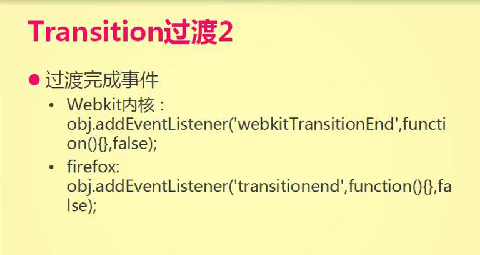
1. transition：time（过渡总时间） attr（需过渡的参数，默认all）
2. 若是只针对某一个参数进行过渡的话，其余没有选择过渡的参数就会在一瞬间完成转换，然后再看到过渡的效果（应该是渲染速度的原因）若是要添加上延迟时间的话，那么在两个时间中第一个事件是过渡时间，相当于一个隐形的速度，第二个参数就是延迟时间。
3. 在多样式过渡的时候，多条样式之间用逗号隔开。
4. 运动形式：transition-timing-function

其中的贝塞尔曲线的功能比较强大，除了贝塞尔曲线之外的几个都是比较常规的属性值。在使用贝塞尔曲线的时候要注意，贝赛尔曲线的四个参数分别是x1，y1，x2，y2，其中的两个x必须按照要求在0-1之间来取值，但是两个y值是可以不受0-1的限制的。



1. 延迟时间transiton-delay：





（7）在执行代码的时候，没执行一次transition都会触发一次transitionEnd事件。因此，在某些情况下会出现一个循环的情况。

2、2D变换



（1）2D变换的时候需要添加浏览器内核前缀。

1. 旋转——rotate（）

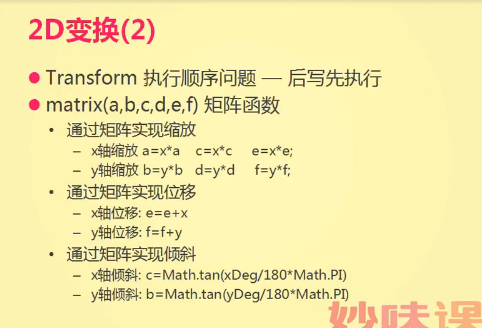
其中rotate中的参数是度数，这个度数可以是负数，这样元素的旋转就是逆时针的旋转，默认的是正数，顺时针旋转。



1. 倾斜（斜切）



这里的skew当做倾斜，后面的X、Y分别是进行倾斜的方向。当然，要是不指明是哪一个方向的斜切，那就必须要写两个参数分别对应的是X、Y两个方向上的斜切角度，只一个参数是不行的，那样只会默认是X方向上的斜切角度。



1. transform-origin：基点（单个或多个关键字、具体坐标数值）

这个属性可以实现钟表效果，当这个属性被使用的时候，那么当前元素就会围绕基点来进行变换，要是加上了其他的属性（如scale），就可以实现一些以前只能用js来实现的效果。

1. 注意transform的执行顺序是先写的后执行。

在2D旋转的这个属性运用中（实现时钟效果模拟），运用到了css一些作图的技巧，比如画圆，实现这样的效果就是边角的弧度调整为50%就能实现。在模拟表盘上的刻度以及指针的时候，为了减少代码的书写，就必须运用到js来实现一些css代码甚至html标签的书写。

指针的模拟是简单的运用了div来实现的，然后这三个div同时都是围绕着最外层表盘这个div来进行旋转的。为了美观，在旋转的基点添加了一个化为圆的div，来充当指针的中心。

接下来在js用到了时间函数date（）；这里的做法选择的是声明一个时间对象，然后这个时间对象分别获取时、分 、秒的具体数值。然后根据时、分、秒的每一个数值对应的的角度进行换算，每一秒钟，指针所在的位置。

3、3D变换





注意一点，就是在**3d的旋转中，随着元素的旋转，坐标轴也会随之旋转**的。

（1）3D变换——依然是在transform上来实现

在使用3D的效果的时候需要首先在外层预先进行3d空间的建立，也就是使用-webkit-transform-style: preserve-3d;来将这个元素的空间转换为3D的空间。还有一个值是flat，也就是平面2D。

1. 在建立3D空间之后，还必须设置景深，也就是一种视觉上的距离。值越小，元素的纵深越大，3D效果越明显。需设置在父元素。

perspective：num+px

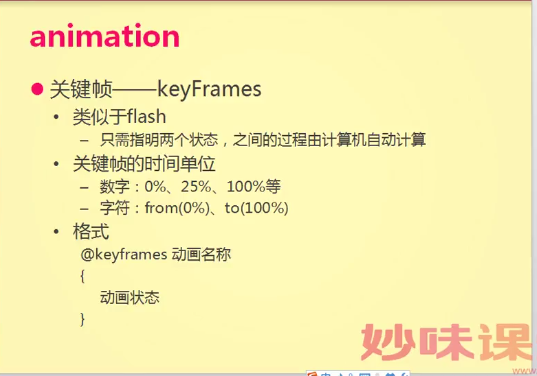
定义元素3D显示的效果，展示出一种“近大远小”的视觉效果，相当于是说元素与观察者之间的距离。这样才能出现3D效果，因此在使用的时候必须要设置景深。同时，还有一个景深基点，就是定义观察者的视觉角度，默认的是从正面（center center）来观察元素，但是可以通过添加参数（关键字、数值）来进行调整。

（3）translateZ（）——z轴方向上的距离移动。例如在做一个立方体的时候，有两个面会重叠在一起，为了实现立方体的效果就必须要对其中一个面就行位置调整，已知z轴是垂直于屏幕向外的，往屏幕外是整数，反之是负数。

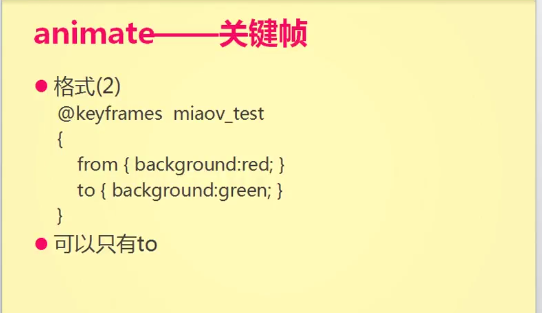
（4）-webkit-transform-origin

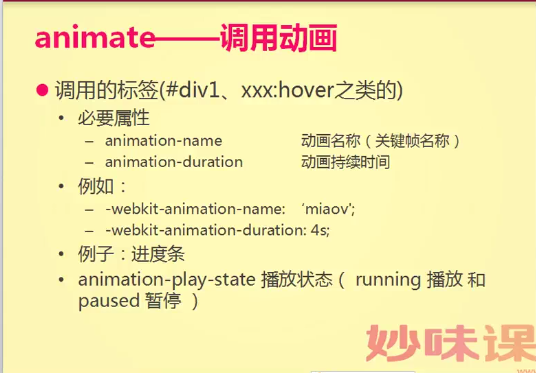
这个属性可以设置元素在旋转的时候的中心，也可以说成是旋转源，也就是旋转中心。在进行3d旋转的时候是可以更改的，只是相对于2d的来说多了一个z坐标轴的定位。

4、动画



1. 关键帧的使用和flash很相似，就是分别定义某一些阶段的动作，在外部看来又有点像一个js中的函数，keyframe就是申明一个关键帧，就像js中function申明一个函数。注意在使用关键帧的时候是需要分别指明每个浏览器内核前缀的，要是不加前缀的话就可能只在不标准浏览器中才能正常执行相关代码。同时还有注意这个关键帧也是需要命名的，这样才方便在animatiion出调用这个关键帧。
2. 注意，在关键帧中义的百分比是对动画时间进行的分配，不要搞混了。在关键帧的代码就是像js一样对将执行的动作写入其中，然后在后面的





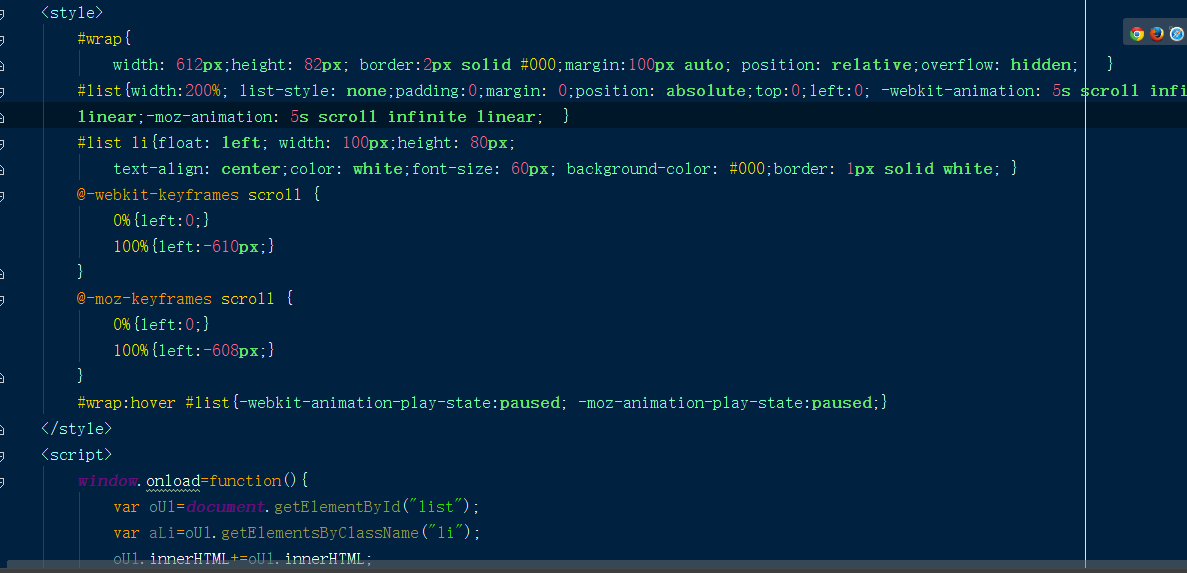
1. 调用动画的时候可以直接用复合形式，包括动画执行时间、延迟时间、执行的形式（速度变化形式）、是否重复登。



1. 对于动画播放状态也是有办法控制的，就是使用animation-play-state，这个属性有两个值，分别是running（默认）和paused；
2. 对于暂停后的动画还有另外两个播放选项，就是从上一次停止的地方开始播放（默认alternate），另一个就是直接回到开头开始播放（normal）



例1——纯CSS无缝滚动



例2——图片轮换

例子中有一个问题就是“下一张”这个按钮会消失，不明白为什么会消失，但是使用代码中的show这个类进行处理，解决了这个问题，初步猜测是由于透明度变化引起的，但是单独加透明度并不能解决这个问题。

**媒体类型**

1. **@media规则**

允许在相同样式表为不同媒体设置不同的样式。

1. 多媒体查询

CSS3多媒体查询继承了CSS2的多媒体类型的所有思想：取代了岔咋后设备的类型，CSS3根据设置自适应显示。媒体查询可以做很多事情：

Viewport的宽度与高度；

设备的宽度与高度

朝向（只能手机横屏，竖屏）

分辨率



