# 什么是浏览器内核

“渲染引擎、浏览器内核”,负责对网页语法的解释并渲染（显示）网页。

## 浏览器内核有哪些？

Trident内核：主要代表为[IE浏览器](https://www.baidu.com/s?wd=IE%E6%B5%8F%E8%A7%88%E5%99%A8&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvPy7-nvm1mWw-P1DLPWb40ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHb4njfYnWDYrjcdPW6znHn3Ps" \t "https://zhidao.baidu.com/question/_blank)  
 Gecko内核：主要代表为Firefox  
 Presto内核：主要代表为Opera  
 Webkit内核：产要代表为Chrome和Safari

不同的浏览器内核对网页编写语法的解释也有不同，因此同一网页在不同的内核的浏览器里的渲染（显示）效果也可能不同。

## CSS3 前缀

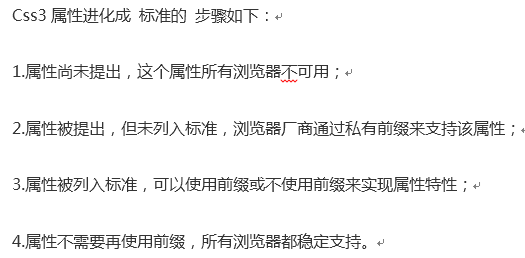
不同内核的浏览器，部分CSS3属性需要添加不同的前缀，也将其称之为浏览器的私有前缀，添加上私有前缀之后的CSS3属性可以说是对应浏览器的私有属性。

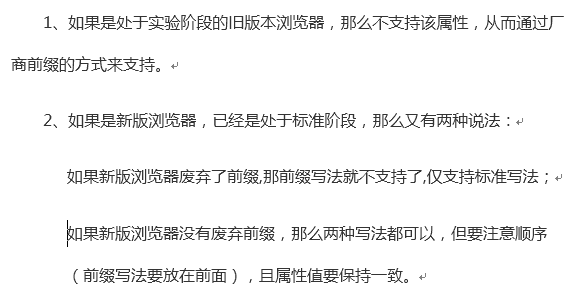


如果是手机等移动端一般采用的是 IOS 或安卓系统，那么基本上它的引擎是 webkit，直接参考-webkit-即可。

## 如何判断是否要加前缀

CSS3 的新属性border-radius、opacity等，这几个属性我们在前面的使用中，并没有添加所谓的浏览器厂商前缀。那是因为，这些属性已经在主流浏览器或版本成为了标准。





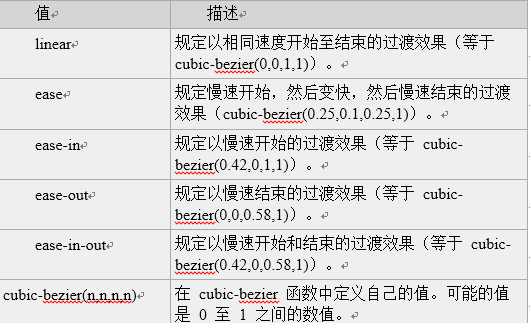
# 过渡transition：

## 1.语法：





## 2.补充：

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

## 3.兼容性：

因为transition最早是有由webkit内核浏览器提出来的，mozilla和opera都是最近版本才支持这个属性，而我们的大众型浏览器IE全家都是不支持，Firefox,Safari,Chrome,Opera都还不支持W3C的标准写法，所以在应用transition时我们有必要加上各自的前缀，最好在放上我们W3C的标准写法，这样标准的会覆盖前面的写法，只要浏览器支持我们的transition属性，那么这种效果就会自动加上去，如：

-moz-transition: all 5s ease 1s;  
-webkit-transition: all 1s ease 1s;  
-o-transition: all 1s ease 1s;  
transition: all 1s ease 1s;

## 4.过渡案例：手风琴菜单，气泡

# 2D转换transform：

通过 CSS3的transform转换，能够对元素进行移动、缩放、转动。

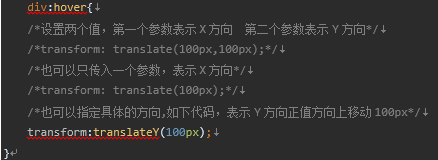
## 2D移动：translate()

作用：把元素从原来的位置移动，参照元素左上角原点。

语法：

（1）transform:translate(x,y) 或transform:translate(x)

（2）transform:translateX(x) 或 transform:translateY(y)



## 2D缩放：scale()

作用：让元素根据中心原点对对象进行缩放，默认的值1。

（1）0.01到0.99之间的任何值，使一个元素按倍数缩小；

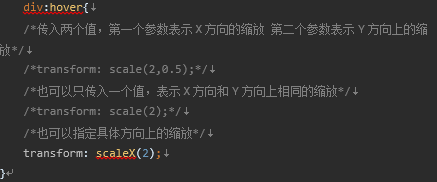
（2）任何大于或等于1.01的值，让元素按倍数放大。

（3）缩放是参照元素中心点。

语法：

（1）transform: scale(x,y) 或transform:scale(x)

（2）transform:scaleX(x) 或 transform:scaleY(y)



## 2D旋转：rotate()

作用：通过指定的角度参数，对元素根据对象原点指定一个2D旋转。

（1）如果这个值为正值，元素相对原点中心顺时针旋转；

（2）如果这个值为负值，元素相对原点中心逆时针旋转。

语法：tranform:rotate(角度值)



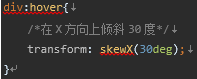
## 2D翻转：skew()

能够让元素倾斜显示。skew()函数不会旋转，而只会改变元素的形状

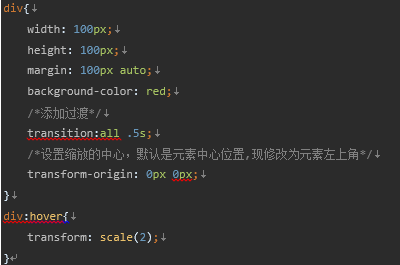
语法：

（1）transform: skew(x,y) 或transform:skew(x)

（2）transform:skewX(x) 或 transform:skewY(y)



## transform-origin:改变被转换元素的位置



# 3D转换transform：

## 3D移动：translate3d()

作用：使元素在这三个纬度中移动。

语法：

（1）translate3d(x,y,z)

（2）translateX(x),translateY(y), translateZ(z)

## 3D缩放：scale3d()

作用：使元素在这三个纬度中缩放。

语法：

（1）scale3d(x,y,z)

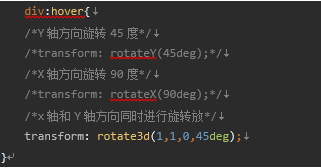
（2）scaleX(x),scaleY(y),scaleZ(z)

## 3D旋转:rotate3d()

作用：使元素在这三个纬度中旋转。

语法：

（1）rotate3d(x,y,z,angle)：指定需要进行旋转的坐标轴

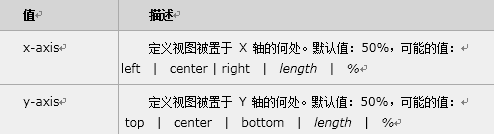
* 1. rotateX(angle)，rotateY(angle) ，rotateZ(angle)

## perspective(length) 透视/景深效果

为一个元素设置三维透视的距离。仅作用于元素的后代，而不是其元素本身。当perspective:none/0;时，相当于没有设perspective(length)。比如你要建立一个小立方体，长宽高都是200px。如果你的perspective < 200px ，那就相当于站在盒子里面看的结果，如果perspective 非常大那就是站在非常远的地方看（立方体已经成了小正方形了），意味着perspective 属性指定了观察者与z=0平面的距离，使具有三维位置变换的元素产生透视效果

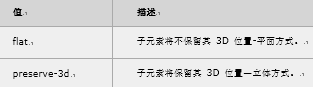
## perspective-origin3D 元素的基点位置





## transform-style

使被转换的子元素保留其 3D 转换(需要设置在父元素中)



# 动画：

动画是CSS3中具有颠覆性的特征之一，可通过设置多个节点来精确控制一个或一组动画，常用来实现复杂的动画效果.

## 1. 步骤：

（1）通过@keyframes指定动画序列；

（2）通过百分比将动画序列分割成多个节点；

（3）在各节点中分别定义各属性

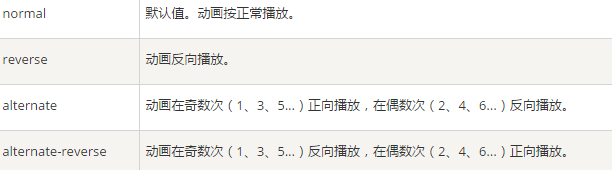
（4）通过animation将动画应用于相应元素；

## 2. animation用于设置动画属性：

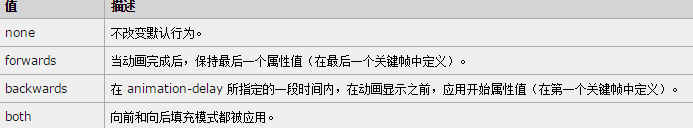
简写：



animation-direction：



animation-fill-mode：



动画案例：汽车/钟表/宇宙

## 3. 补充

animation-play-state:paused; 暂停动画