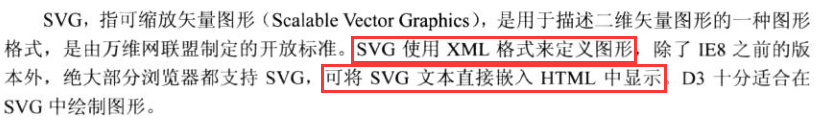
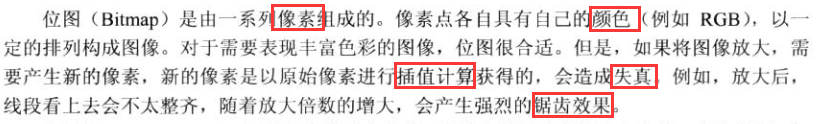
# SVG



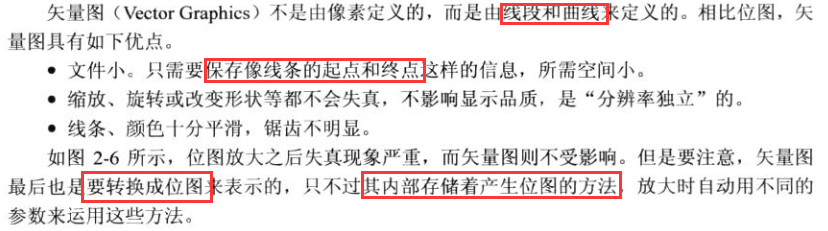
# 位图

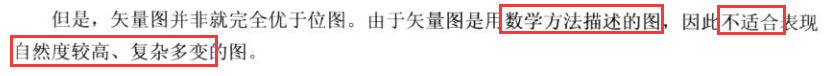
位图由像素组成，像素有颜色。放大位图会失真，产生锯齿效果



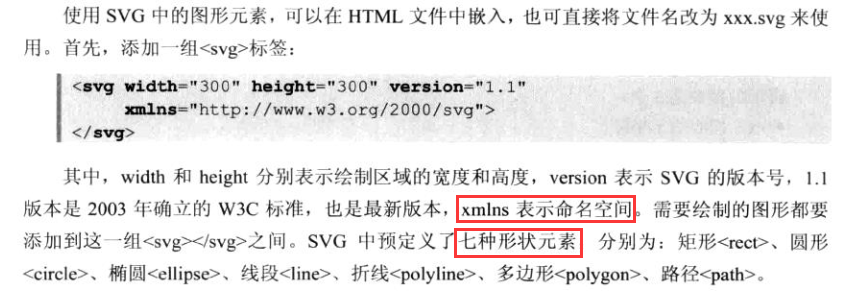
# 矢量图

矢量图由线段和曲线定义，显示时需要转换成位图。不适合表现复杂的图

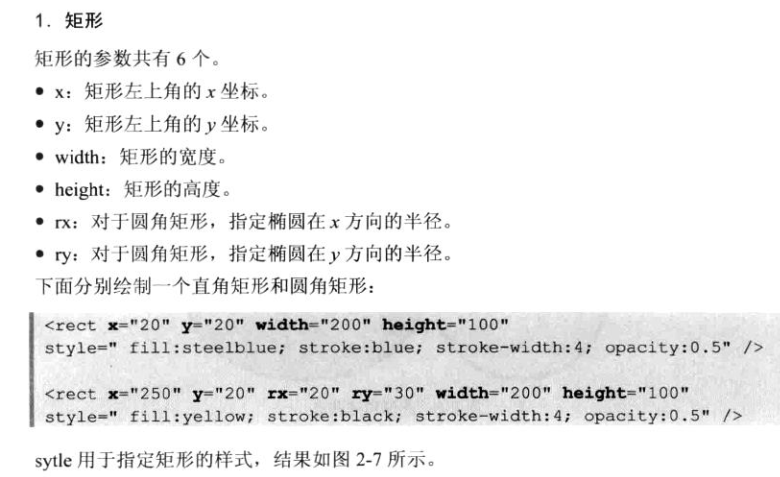


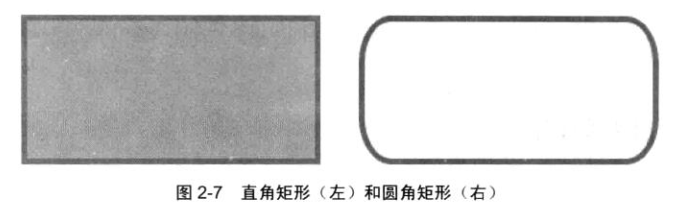


# 图形元素

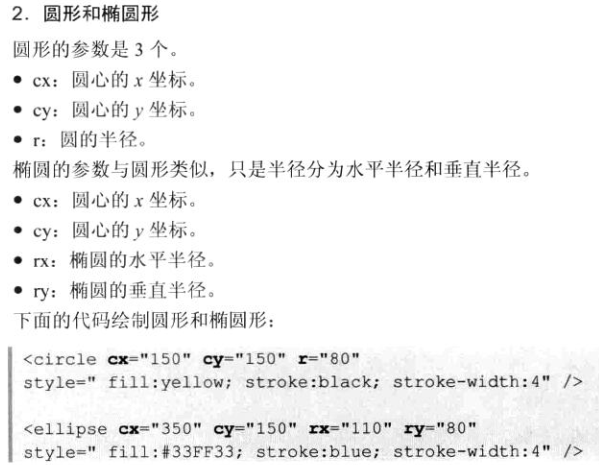


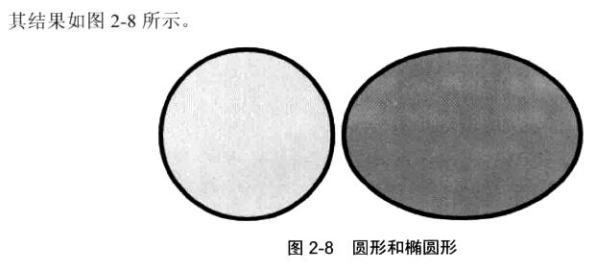
## 矩形



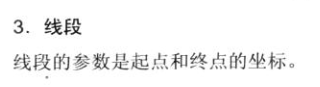


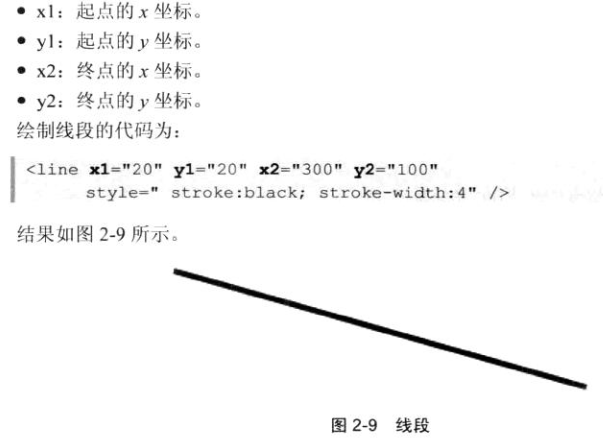
## 圆形和椭圆形



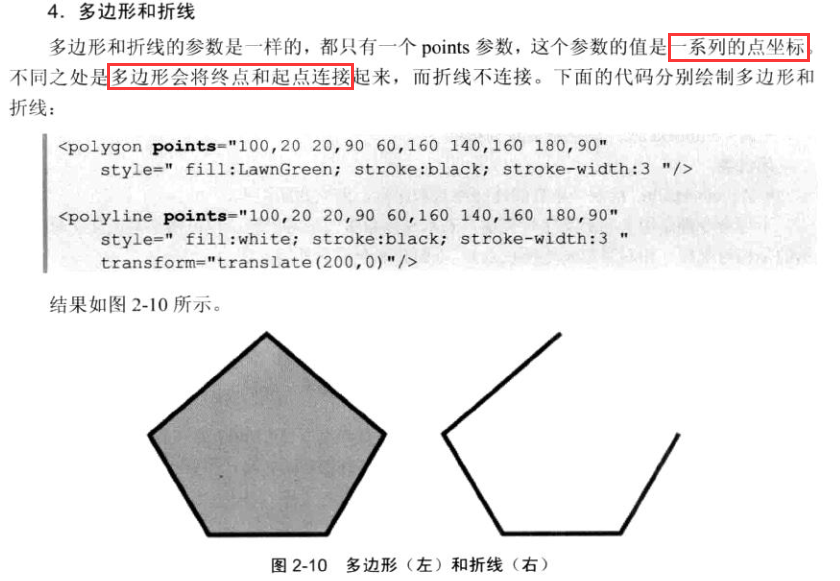


## 线段

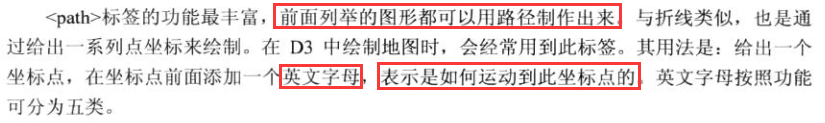




## 多变形和折线



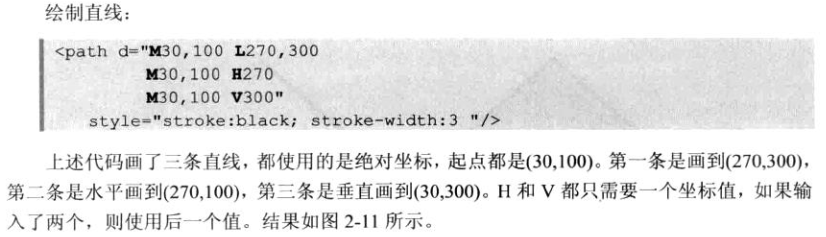
## 路径



大绝对，小相对

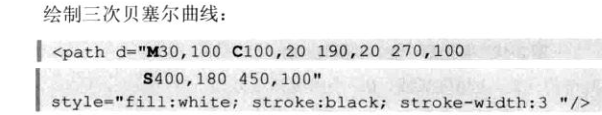


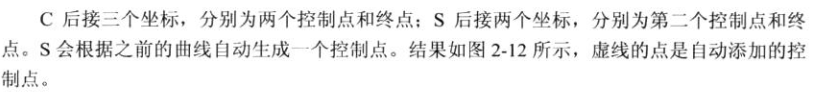
### 直线

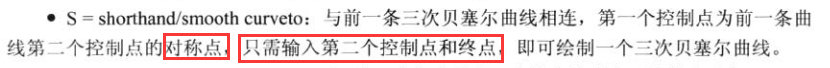


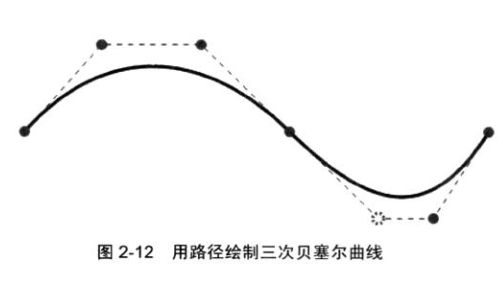


### 三次贝塞尔曲线

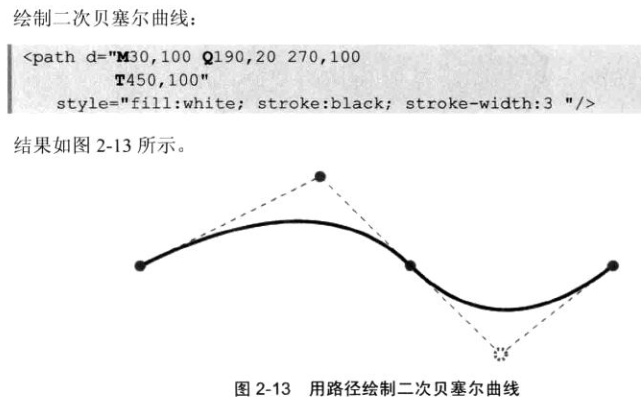






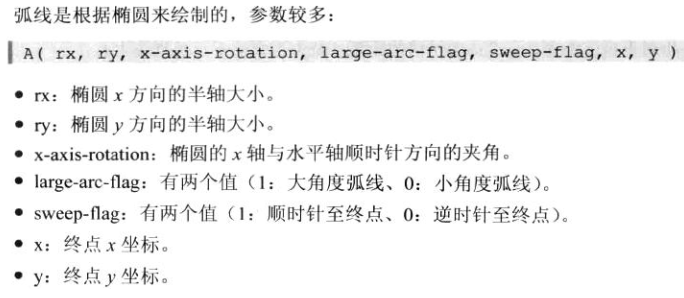


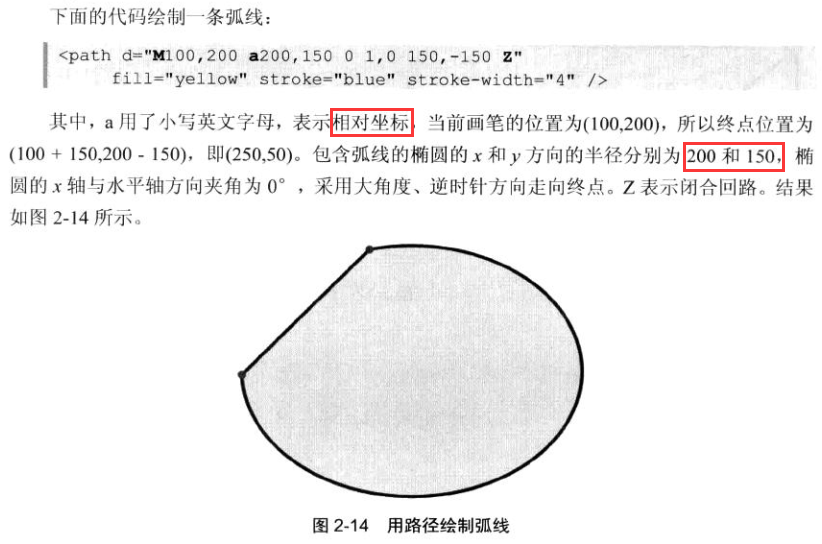
### 二次贝塞尔曲线



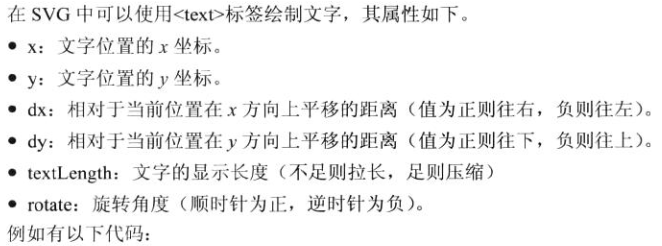


## 弧线



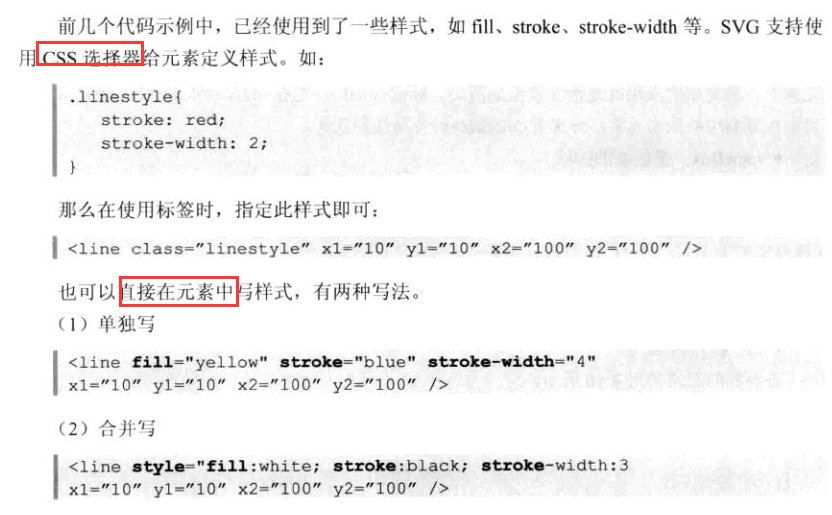


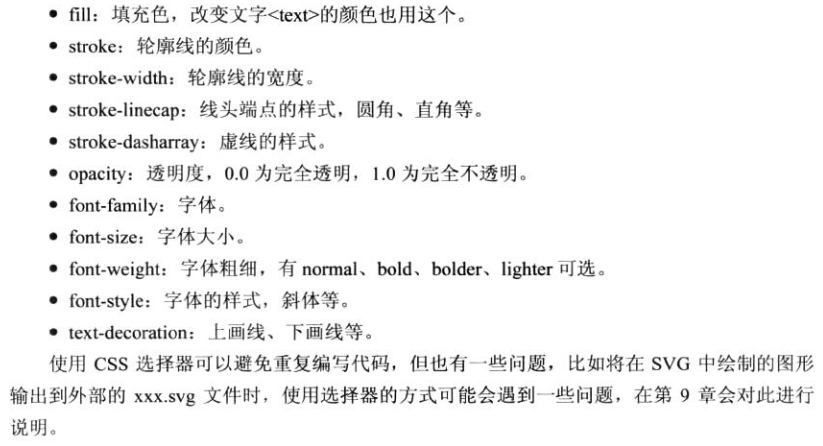
## 文字





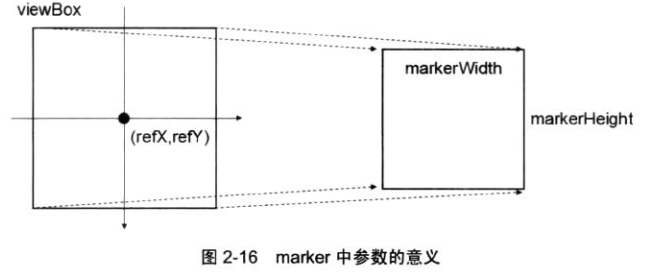
## 样式

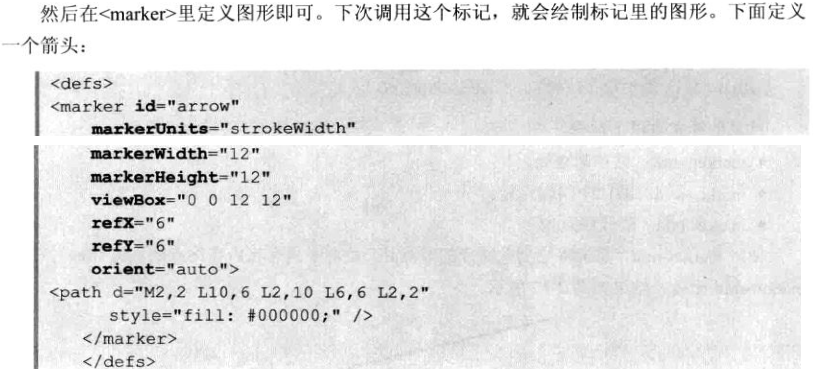


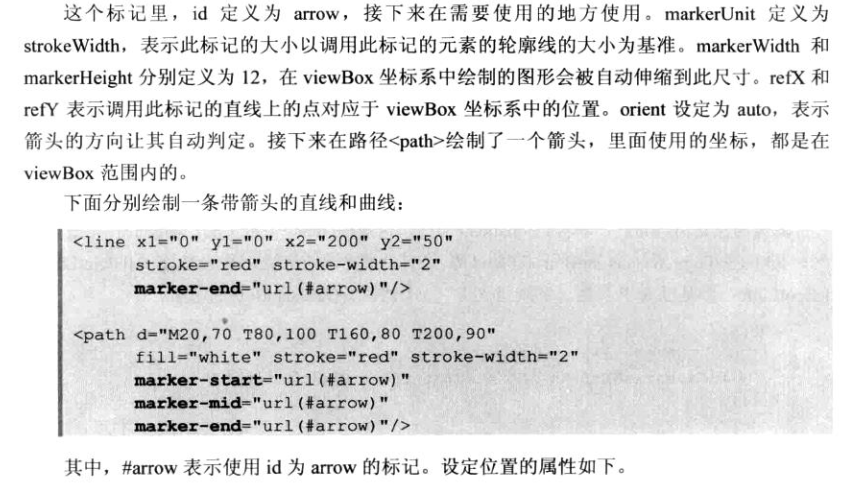


## 标记











# 其他

## 2@贝塞尔曲线扫盲 - 随便写写

- 前端乱炖

http://www.html-js.com/article/1628

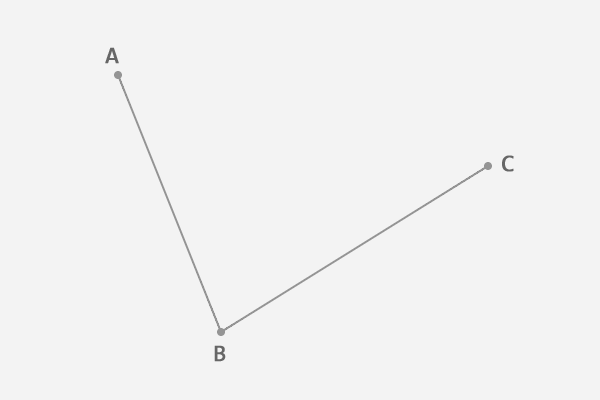
相信很多同学都知道“[贝塞尔曲线](http://en.wikipedia.org/wiki/B%C3%A9zier_curve)”这个词，我们在很多地方都能经常看到。但是，可能并不是每位同学都清楚地知道，到底什么是“贝塞尔曲线”，又是什么特点让它有这么高的知名度。

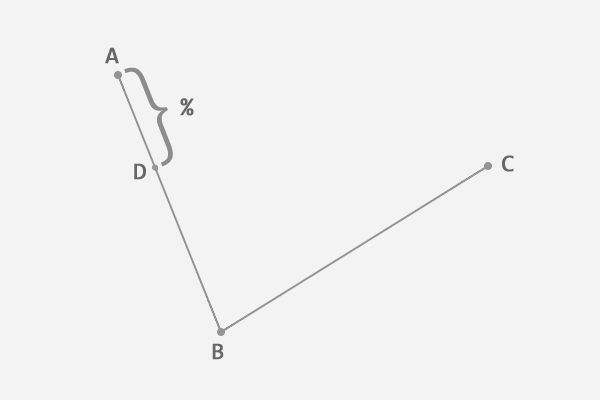
贝塞尔曲线的数学基础是早在 1912 年就广为人知的[伯恩斯坦多项式](http://en.wikipedia.org/wiki/Bernstein_polynomial)。但直到 1959 年，当时就职于雪铁龙的法国数学家 [Paul de Casteljau](http://en.wikipedia.org/wiki/Paul_de_Casteljau) 才开始对它进行图形化应用的尝试，并提出了一种数值稳定的 [de Casteljau 算法](http://en.wikipedia.org/wiki/De_Casteljau's_algorithm)。然而贝塞尔曲线的得名，却是由于 1962 年另一位就职于雷诺的法国工程师 [Pierre Bézier](http://en.wikipedia.org/wiki/Pierre_B%C3%A9zier) 的广泛宣传。他使用这种只需要很少的控制点就能够生成复杂平滑曲线的方法，来辅助汽车车体的工业设计。

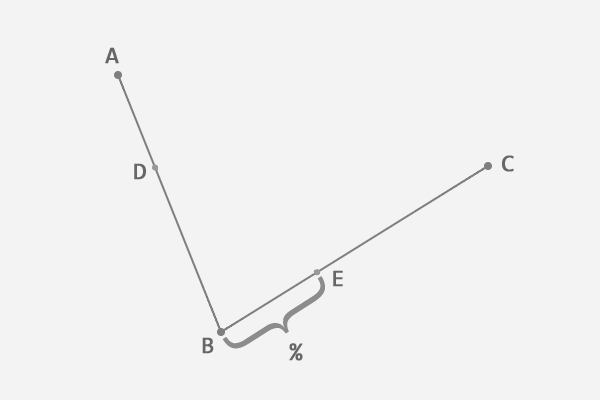
正是因为控制简便却具有极强的描述能力，贝塞尔曲线在工业设计领域迅速得到了广泛的应用。不仅如此，在计算机图形学领域，尤其是矢量图形学，贝塞尔曲线也占有重要的地位。今天我们最常见的一些矢量绘图软件，如 Flash、Illustrator、CorelDraw 等，无一例外都提供了绘制贝塞尔曲线的功能。甚至像 Photoshop 这样的位图编辑软件，也把贝塞尔曲线作为仅有的矢量绘制工具（钢笔工具）包含其中。

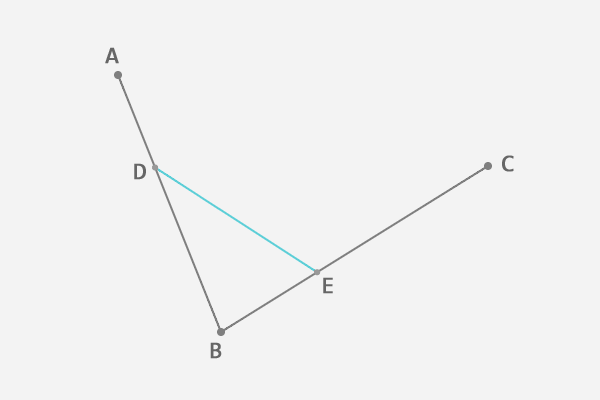
贝塞尔曲线在 web 开发领域同样占有一席之地。CSS3 新增了 [transition-timing-function](http://www.w3.org/TR/css3-transitions/#transition-timing-function-property) 属性，它的取值就可以设置为一个三次贝塞尔曲线方程。在此之前，也有不少 JavaScript 动画库使用贝塞尔曲线来实现美观逼真的缓动效果。

下面我们就通过例子来了解一下如何用 de Casteljau 算法绘制一条贝塞尔曲线。

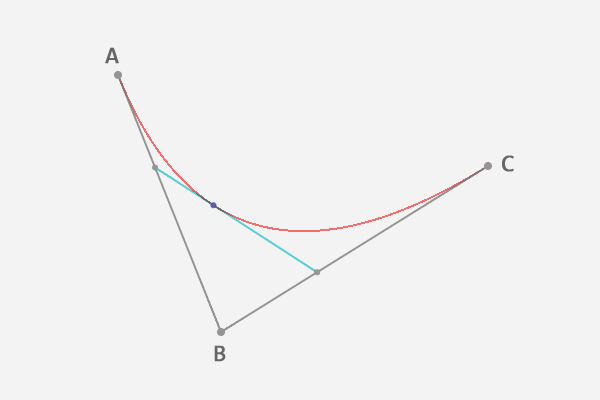
在平面内任选 3 个不共线的点，依次用线段连接。

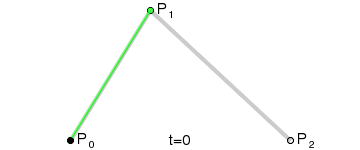
在第一条线段上任选一个点 D。计算该点到线段起点的距离 AD，与该线段总长 AB 的比例。

根据上一步得到的比例，从第二条线段上找出对应的点 E，使得 AD:AB = BE:BC。

连接这两点 DE。

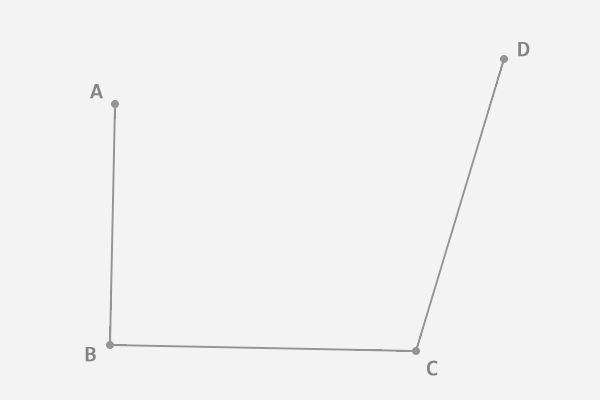
从新的线段 DE 上再次找出相同比例的点 F，使得 DF:DE = AD:AB = BE:BC。

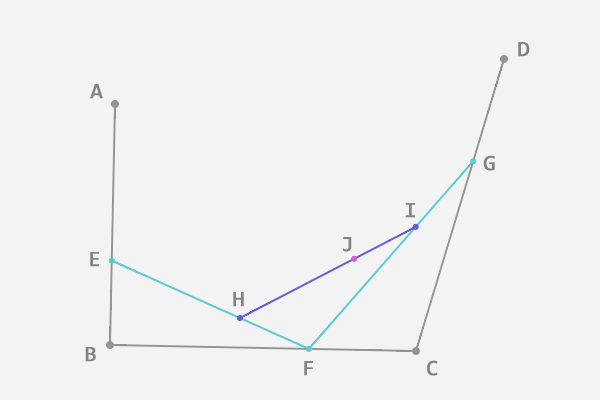
到这里，我们就确定了贝塞尔曲线上的一个点 F。接下来，请稍微回想一下中学所学的极限知识，让选取的点 D 在第一条线段上从起点 A 移动到终点 B，找出所有的贝塞尔曲线上的点 F。所有的点找出来之后，我们也得到了这条贝塞尔曲线。

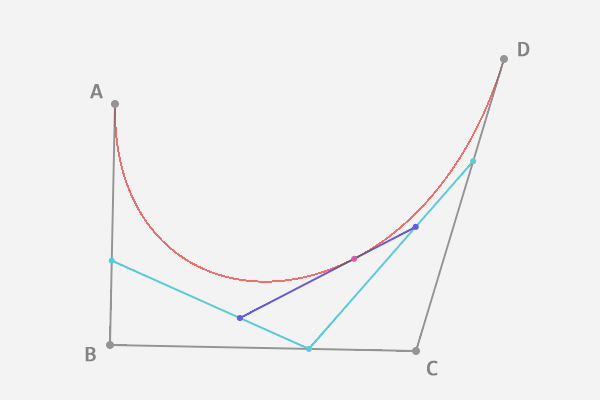
如果你实在想象不出这个过程，没关系，看动画！

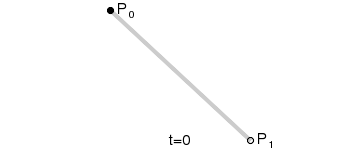
回过头来看这条贝塞尔曲线，为了确定曲线上的一个点，需要进行两轮取点的操作，因此我们称得到的贝塞尔曲线为二次曲线（这样记忆很直观，但曲线的次数其实是由前面提到的伯恩斯坦多项式决定的）。

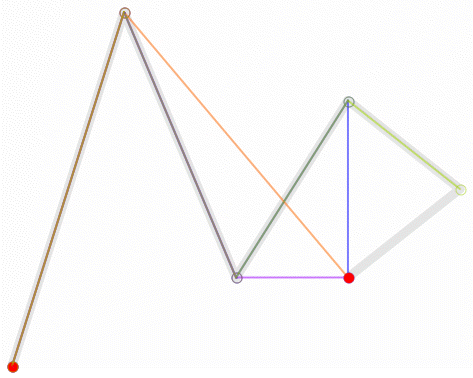
### 三次贝塞尔曲线

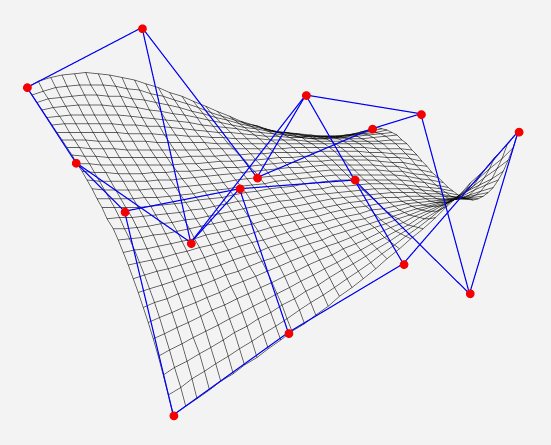
当控制点个数为 4 时，情况是怎样的？

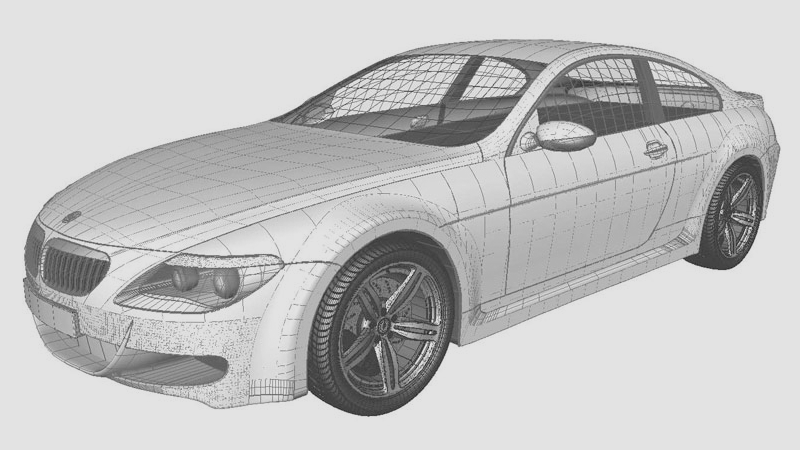
步骤都是相同的，只不过我们每确定一个贝塞尔曲线上的点，要进行三轮取点操作。如图，AE:AB = BF:BC = CG:CD = EH:EF = FI:FG = HJ:HI，其中点 J 就是最终得到的贝塞尔曲线上的一个点。

这样我们得到的是一条三次贝塞尔曲线。

看过了二次和三次曲线，更高次的贝塞尔曲线大家应该也知道要怎么画了吧。那么比二次曲线更简单的一次（线性）贝塞尔曲线存在吗？长什么样？根据前面的介绍，只要稍作思考，想必你也能猜出来了。哈！就是一条直线~

能画曲线也能画直线，是不是很厉害？要绘制更复杂的曲线，控制点的增加也仅仅是线性的。这一特点使其不光在工业设计领域大展拳脚，就连数学基础不好的人也可以比较容易地掌握，比如大多数平面美术设计师们。

上面介绍的内容并不足以展示贝塞尔曲线的真正威力。推广到三维空间的[贝塞尔曲面](http://en.wikipedia.org/wiki/B%C3%A9zier_surface)，以及更进一步的[非均匀有理 B 样条（NURBS）](http://en.wikipedia.org/wiki/Non-uniform_rational_B-spline)，早已成为当今计算机辅助设计（CAD）的行业标准，不论是我们平常用到的各种产品，还是在电影院看到的精彩大片，都少不了它们的功劳。



动态绘制贝塞尔曲线的[在线演示](http://myst729.github.io/bezier-curve)

– 完 –

本站专栏文章皆为原创，转载请注明出处（带有 前端乱炖 字样）和本文的显式链接(<http://www.html-js.com/article/1628>)，本站和作者保留随时要求删除文章的权利！