



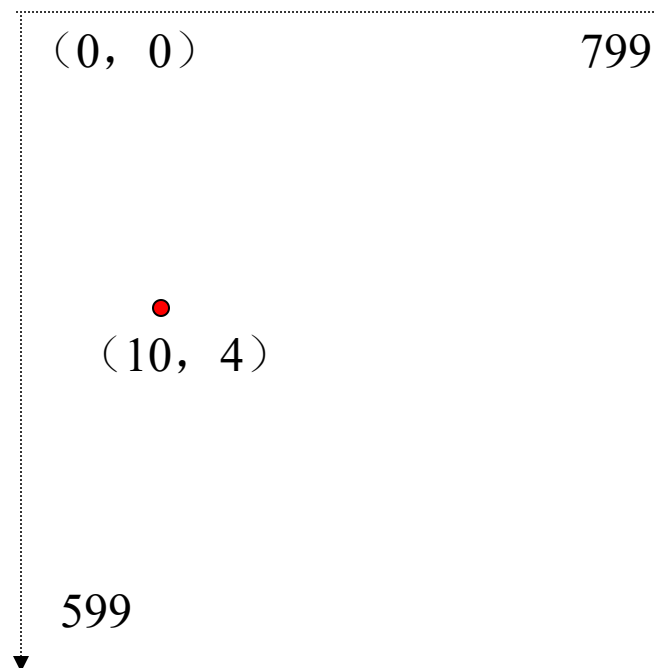
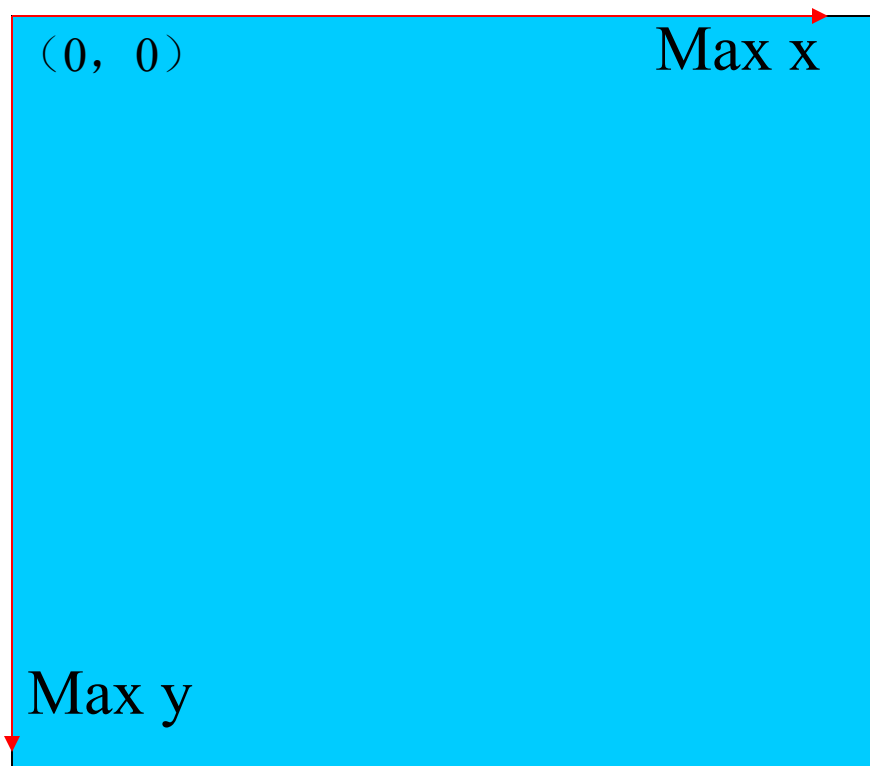
# 计算机图形学

C语言图形程序设计基础-1

谢忠红



- 在图形模式下,以屏幕的左上角为坐标原点 $(0,0)$ 水平方向为x轴, 垂直方向为y轴

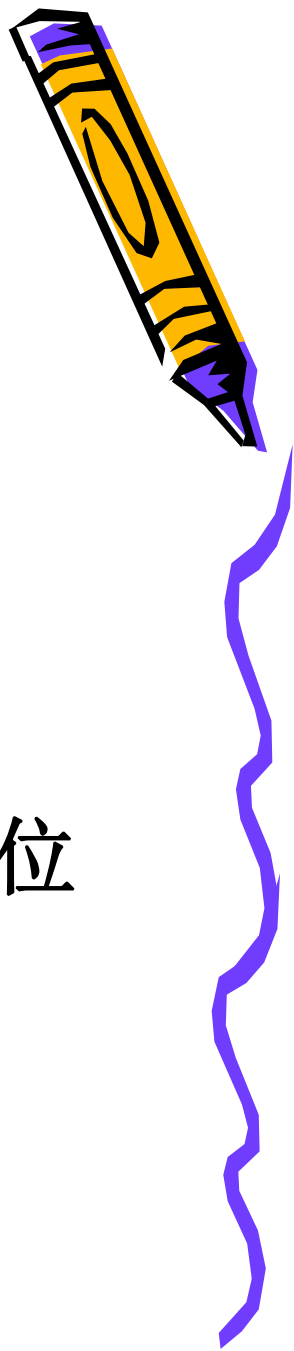


**例：**一个分辨率为800\*600的分辨率的显示屏（800列，600行）

### • 3.1 一个简单的图形程序

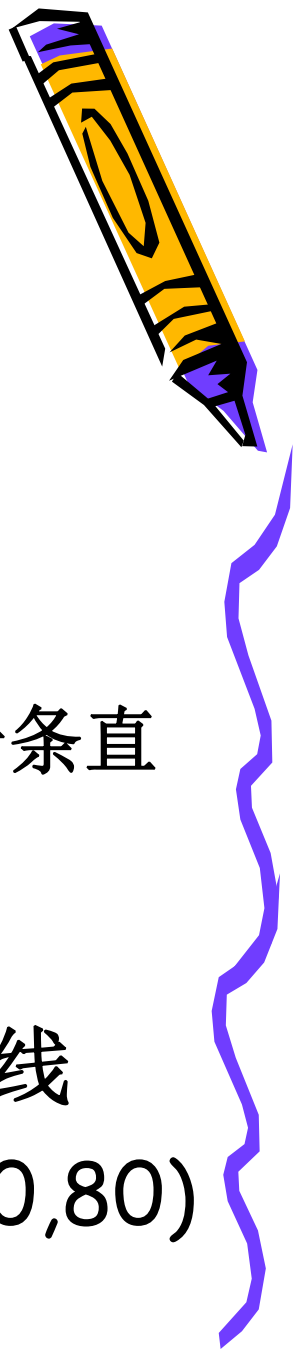
```
• #include <graphics.h>      #include <time.h>
• #include <conio.h>
• void main()
• {char s[2] = {0};
•   srand((unsigned) time(NULL));    // init randomize seed
•   initgraph(640, 480);            // init graph mode
•   int x,y;
•   while(!kbhit())
•   {   for (int i=0; i<479; i++)
•       {   setcolor(GREEN);
•           for (int j=0; j<3; j++)
•           {   x = (rand() % 80) * 8;
•               y = (rand() % 20) * 24;
•               s[0] = (rand() % 26) + 65;
•               outtextxy(x,y,s);
•           }
•           setcolor(0);   line(0,i,639,i);   Sleep(10);
•           if (kbhit())   break;
•       }
•   }
•   closegraph();
• }
```

- 控制画笔位置的函数
- - `moveto(int x ,int y)`
- 作用：将当前位置移到(x,y)
- 例： `moveto(10,20)`
- - `moverel(int deltax,int deltax)`
- 作用：相对移动函数
- 说明：`deltax`和`deltay`。表示相对移动位置

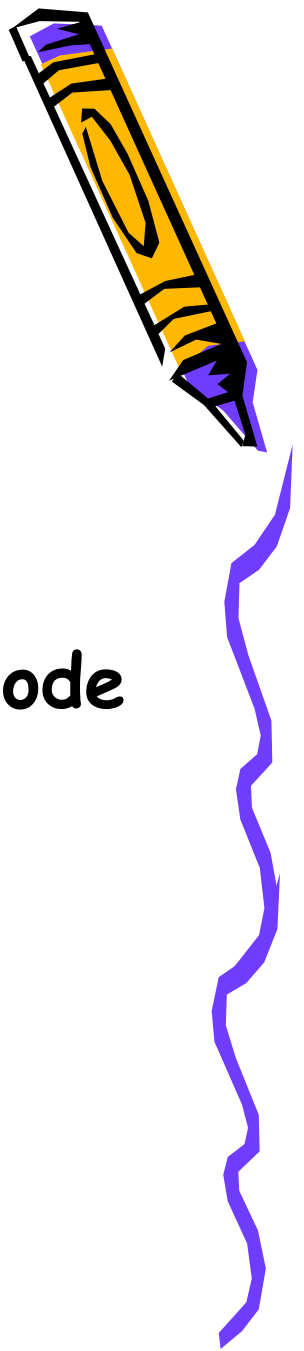


### 3. 直线和线型函数

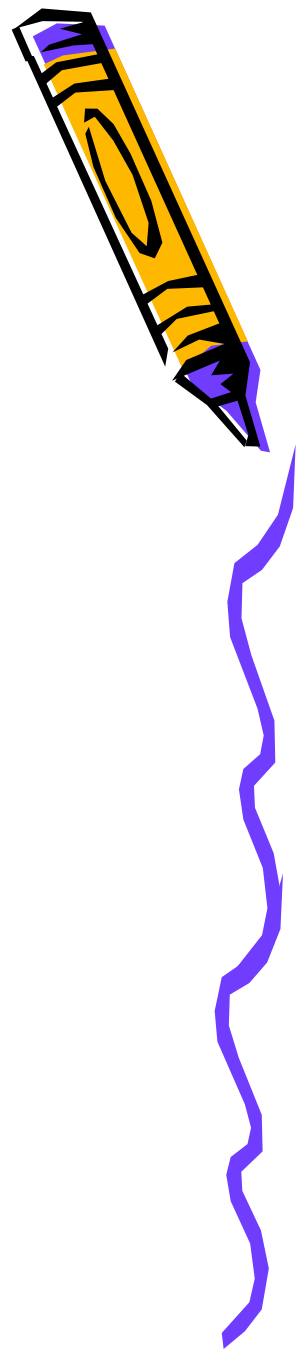
- `-line(int x1,int y1,int x2,int y2)`
- 作用：-在指定的两点之间画直线
- 说明：当前光标依然在 $(x1,y1)$
- `lineto(int x,int y)`
- -作用：从当前光标位置到 $(x,y)$ 之间画一条直线
- `-linerel(int deltax,int deltay)`
- 作用：从当前坐标以相对增量方式画直线
- 问题2：如果 `CP(10,20); linerel(70,80)`  
应在哪两点之间画直线？



- 阅读程序，写出程序运行结果
- 例：`#include "graphics.h"`
- `#include "stdio.h"`
- `main()`
- `{`
- `int cp_x, cp_y;`
- `initgraph(640, 480); // init graph mode`
- `moveto(5,10);`
- `line(5,10,100,50);`
- `cp_x=getx();`
- `cp_y=gety();`
- `}`



- `printf("the cp_x cp_y are %d %d",cp_x,cp_y);`
- `printf("\n") ; //回车`
- `getch() ;`
- `moveto(100,50);`
- `lineto(100,100);`
- `cp_x=getx();`
- `cp_y=gety();`
- `printf("the cp_x cp_y is %d %d",cp_x,cp_y) ;//`
- `getch();`
- `linerel(50,50);`
- `getch();`



## • 圆弧类函数

- `-arc(int x,int y,int startangle,int endangle,int radius)`

- 作用：画圆弧

- 例： `arc(100,100,0,360,50)`

- `arc(100,100,0,180,30)`

- `circle(int x,int y,int radius)`

- 作用：画圆

- `-ellipse(int x,int y,int start,int end ,int xradius,int yradius)`

- 作用：画椭圆

说明：以(x,y)为圆心，长半轴为xradius，短半轴为yradius

- 开始角度为start，终止角度为end





- -**Pieslice**(int x,int y int startangle,int endangle,int radius
- 作用：画扇形
- 例：void draw\_slice
- {int start,end;
- start=0,end=45;
- For (i=0 ;i< 8;i++)
- {**setfillstyle(SOLID\_FILL,i);**
- pieslice(260,200,start,end,100);
- start+=45;
- end+=45;
- }
- Getch();



- **5多边形类函数**
- **rectangle(int left,int top,int right,int bottom)**
- 作用：画矩形
- 例：**rectangle(200,100,500,400)**
- **drawpoly(int numpoints, int \*polypoints)**
- 作用：画多边形
- 说明：用当前画笔颜色画一个多边形，多边形的点数为 **numpoints**, **\*polypoints** 指向一个整型数组，共有 **numpoints\*2** 个整数，每一对整数给出了一个多边形顶点 (x,y) 的坐标。
- 例：**int matrix[10]={10,20,20,60,30,50,20,40,15,20}**
- **Drawpoly(5,matrix)**

