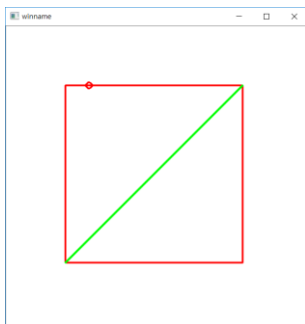


<OpenCV 4.0.1 기준>

도형 그리기

```
1  #include "opencv2/opencv.hpp"
2
3  using namespace std;
4  using namespace cv;
5
6  int main()
7  {
8      Mat dstImage(512, 512, CV_8UC3, Scalar(255, 255, 255)); //그래픽 그릴 영상 데이터
9
10     rectangle(dstImage, Point(100, 100), Point(400, 400), Scalar(0, 0, 255), 2);
11     //사각형그리기(영상, TopLeft, BottomRight, color, thickness)
12     line(dstImage, Point(400, 100), Point(100, 400), Scalar(0, 255, 0), 2);
13     //직선그리기
14
15     Rect imgRect(100, 100, 400, 400); //사각형 객체
16     Point pt1(120, 50), pt2(300, 500); //점 객체
17     clipLine(imgRect, pt1, pt2); // 사각형 imgRect와 pt1-pt2직선이 교차하는 점을 다시 pt1,pt2에 계산
18
19     circle(dstImage, pt1, 5, Scalar(0, 0, 255), 2);
20     //원 그리기
21
22     imshow("winname", dstImage);
23     waitKey();
24
25     return 0;
26 }
27
```

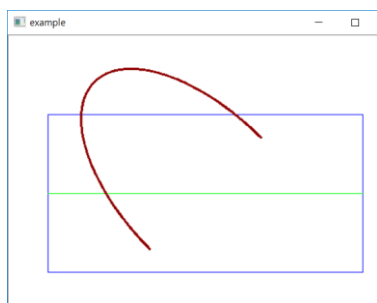
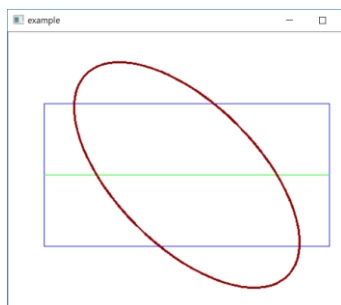


두께 = -1이면 색 채움.

```

1  #include <opencv2/opencv.hpp>
2
3  using namespace std;
4  using namespace cv;
5
6  int main()
7  {
8      Mat dstimg(512, 512, CV_8UC3, Scalar(255, 255, 255));
9
10     Point center(250, 200);
11     Size size(200, 100);
12
13     rectangle(dstimg, Point(center.x - size.width, center.y - size.height), Point(center.x + size.width, center.y + size.height), Scalar(255, 0, 0));
14
15     line(dstimg, Point(center.x - size.width, center.y), Point(center.x + size.width, center.y), Scalar(0, 255, 0));
16
17     ellipse(dstimg, center, size, 45, 0, 360, Scalar(0, 0, 144), 2);
18     //(행렬데이터, 중심좌표, 타원의 가로/세로, 수평축과의 각도, 타원 그리기 시작 각도, 타원 그리기 끝남 각도, 색상, 두께)
19     imshow("example", dstimg);
20     waitKey();
21
22     return 0;
23 }
24

```

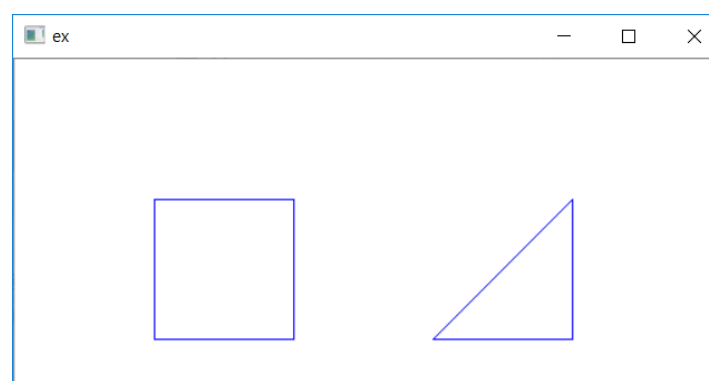


startangle = 90, endangle = 270

```

1  #include "opencv2/opencv.hpp"
2
3  using namespace std;
4  using namespace cv;
5
6  int main()
7  {
8      Mat dstimg(512, 512, CV_8UC3, Scalar(255, 255, 255));
9
10     Point pt1[4] = { Point(100,100), Point(200,100), Point(200,200), Point(100,200) };
11     Point pt2[3] = { Point(300,200), Point(400,100), Point(400,200) };
12
13     const Point *polygons[2] = { pt1, pt2 }; //각 도형별 좌표.
14     int npts[2] = { 4,3 }; // 2개의 다각형의 점 개수
15
16
17     polylines(dstimg, polygons, npts, 2, true, Scalar(255, 0, 0));
18     //여러개를 넣고 싶을 때는 배열 사용. 여기서 2는 그릴 도형 개수.
19     //bool 변수는 true이면 완전한 도형그리기, false면 시작점과 끝점을 연결하지 않음(강 직선의 연속)
20
21     //fillPoly(dstimg, polygons, npts, 2, Scalar(0, 0, 255));
22     //색상채운 다각형
23
24     imshow("ex", dstimg);
25     waitKey();
26
27
28     return 0;
29 }
30

```



문자열 출력

```
1  #include "opencv2/opencv.hpp"
2
3  using 네임스페이스 ...
4
5
6  int main()
7  {
8      Mat dstimg(512, 512, CV_8UC3, Scalar(255, 255, 255));
9
10     //이전에 Point배열을 사용한 것을 Vector로 바꿔봄.
11     vector<vector<Point>> contour(2, vector<Point>{}); // 행 개수, point 벡터배열
12
13     contour[0].push_back(Point(100, 100));
14     contour[0].push_back(Point(200, 100));
15     contour[0].push_back(Point(200, 200));
16     contour[0].push_back(Point(100, 200));
17
18     contour[1].push_back(Point(300, 200));
19     contour[1].push_back(Point(400, 100));
20     contour[1].push_back(Point(400, 200));
21
22     Point *pt1 = (Point*)Mat(contour[0]).data; //Point 포인터로 형 변환
23     Point *pt2 = (Point*)Mat(contour[1]).data;
24
25     Point *polygons[2] = { pt1, pt2 };
26     int npts[2] = { contour[0].size(), contour[1].size() };
27
28     polylines(dstimg, polygons, npts, 2, true, Scalar(0, 0, 255));
29
30     imshow("ex", dstimg);
31     waitKey();
32 }
```

GUI

```
1  #include "opencv2/opencv.hpp"
2
3  using 네임스페이스 ...
4
5
6  int main()
7  {
8      Mat src = imread("image.png", IMREAD_COLOR);
9
10     namedWindow("ex", WINDOW_NORMAL); //윈도우 이름 지정해서 생성
11     imshow("ex", src);
12     waitKey();
13
14     resizeWindow("ex", 500, 300); //("사이즈 재조정할 윈도우 이름", 너비, 높이)
15     imshow("ex", src);
16     waitKey();
17
18     //destroyAllWindows();
19     //위 함수는 모든 윈도우창을 종료하는 함수로, 응용프로그램 종료시 자동 파괴되므로 사용 안해도 됨.
20
21     return 0;
22 }
23
```

```
1      #include "opencv2/opencv.hpp"
2
3      using 네임스페이스 ...
4
5
6      int main()
7      {
8          Mat src = imread("image.png", IMREAD_COLOR);
9
10         namedWindow("ex", WINDOW_NORMAL);
11         resizeWindow("ex", 500, 500);
12         imshow("ex", src);
13
14         int x=100, y = 100;
15         moveWindow("ex", x, y); //윈도우 좌측상단 좌표 옮기기.
16
17         int nkey;
18         while (1) {
19             nkey = waitKey(0);
20             if (nkey == 0x1B) //esc누르면
21                 break;
22             switch (nkey) {
23                 case 'A':
24                     x -= 10;
25                     break;
26                 case 'B':
27                     x += 10;
28                     break;
29             }
30             moveWindow("ex", x, y);
31         }
32     }
```

```

1      #include "opencv2/opencv.hpp"
2
3      using 네임스페이스 ...
4
5
6      int main()
7      {
8          Mat dstimg(512, 512, CV_8UC3, Scalar(255, 255, 255));
9
10         string text = "Hellooooooooooooooooo";
11         int fontFace = FONT_HERSHEY_SIMPLEX;
12         double fontScale = 1;
13         int thickness = 1;
14         int baseLine;
15         Point startpt(50, 50); //문자열 bottom-left 좌표
16
17         putText(dstimg, text, startpt, fontFace, fontScale, Scalar(0, 0, 0));
18
19         Size sz = getTextSize(text, fontFace, fontScale, thickness, &baseLine);
20         //문자열의 너비와 높이
21         //&baseLine은 폰트 스케일에 따라 값이 계산됨. 폰트크기 계산받을 변수의 주소값
22         cout << baseLine <<endl; // 1.0이면 10, 1.5이면 14, 2.0이면 18
23         cout << sz;
24
25         imshow("ex", dstimg);
26         waitKey();
27
28
29         return 0;
30     }
31

```

```

1      #include "opencv2/opencv.hpp"
2
3      using 네임스페이스 ...
4
5
6      void onMouse(int event, int x, int y, int flags, void* param);
7
8      int main()
9      {
10         Mat dstimg(512, 512, CV_8UC3, Scalar(255, 255, 255));
11
12         imshow("winname", dstimg);
13         setMouseCallback("winname", onMouse, (void*)&dstimg);
14
15         waitKey();
16
17         return 0;
18     }
19
20     //마우스이벤트 핸들러 함수로, 이름은 임의지정 가능하나 인수 자료형과 의미는 지정되어 있음.
21     void onMouse(int event, int x, int y, int flags, void* param) //flag는 좌/우/ctrl/alt/shift 눌렀는지 여부확인 상수
22     {
23         Mat* pMat = (Mat*)param;
24         Mat image = Mat(*pMat);
25         switch (event)
26         {
27             case EVENT_LBUTTONDOWN:
28                 if (flags & EVENT_FLAG_SHIFTKEY)
29                     rectangle(image, Point(x - 5, y - 5), Point(x + 5, y + 5), Scalar(255, 0, 0));
30                 else
31                     circle(image, Point(x, y), 5, Scalar(0, 0, 255), 5);
32                 break;
33             case EVENT_RBUTTONDOWN:
34                 circle(image, Point(x, y), 5, Scalar(255, 0, 0), 5);
35                 break;
36             case EVENT_LBUTTONDBLCLK:
37                 image = Scalar(255, 255, 255);
38                 break;
39         }
40         imshow("winname", image);

```



```

1      #include <opencv2/opencv.hpp>
2
3      using namespace cv;
4
5
6      void onChange(int pos, void* param);
7
8      int main()
9      {
10         Mat image[2];
11         image[0] = imread("image.png", IMREAD_GRAYSCALE);
12         if (image[0].empty())
13             return -1;
14         image[1].create(image[0].size(), CV_8U);
15
16         imshow("winname", image[0]);
17
18         int pos = 100;
19         onChange(pos, (void*)image);
20         createTrackbar("control_Bar", "winname", &pos, 255, onChange, (void*)image);
21         //pos는 시작시 값, 255는 최대값
22
23         waitKey();
24
25         return 0;
26     }
27
28     void onChange(int pos, void* param)
29     {
30         Mat* pMat = (Mat*)param;
31         Mat src = Mat(pMat[0]);
32         Mat dst = Mat(pMat[1]);
33
34         int x, y, s, r;
35         int nThreshold = pos;
36         for (x = 0; x < src.rows; x++)
37         {
38             for (y = 0; y < src.cols; y++)
39             {
40                 r = src.at<uchar>(x, y); //화소값 가져오기
41                 if (r > nThreshold) //임계치와 비교
42                     s = 255;
43                 else
44                     s = 0;
45                 dst.at<uchar>(x, y) = s;
46             }
47         }
48         imshow("winname", dst);
49     }
50

```

