

행렬의 데이터타입, 깊이, 몇 차원?

```
OpenCVExample.cpp* x
OpenCVExample (전역 범위)
1 #include "opencv2/opencv.hpp"
2
3 using namespace std;
4 using namespace cv;
5
6 int main()
7 {
8     Mat A1(1, 2, DataType<uchar>::type); // row, col, type
9
10    A1.at<uchar>(0, 0) = 1; // A1.at() == A1[]
11    A1.at<uchar>(0, 1) = 2;
12
13    cout << "A1" << A1 << endl;
14    cout << "depth=" << A1.depth() << ", " << "channels=" << A1.channels() << endl;
15    //depth : 자료형 크기. 예를 들면 char은 1byte이므로 2^0 이므로 depth = 0;
16    //channel : 행렬의 차원을 뜻함. 여기서는 1차원이므로 channel = 1;
17
18
19    return 0;
20 }
21
```

Point 템플릿 클래스

```
5
6 int main()
7 {
8     Point2f pt1(0.1f, 0.2f), pt2(0.3f, 0.4f); //실수형 좌표(2차원)
9     //or Point2f pt1 = Point2f(0.1f, 0.2f)
10
11    Point pt3 = (pt1 + pt2)*10.0f; //정수형 좌표(2차원)
12
13    cout << "pt3 : " << pt3 << endl;
14    cout << "x : " << pt3.x << ", y: " << pt3.y << endl; // .x, .y를 통한 각 좌표 확인
15
16    cout << "pt1&pt2 내적 : " << pt1.dot(pt2) << endl; // 두 좌표의 내적 계산.
17    // result < 0 : 예각
18    // result > 0 : 둔각, = 0이면 직각.
19
20    cout << "원점 <-> pt1 사이의 거리 " << norm(pt1) << endl; // 원점에서의 거리 계산.
21
22    Rect rect(0, 0, 10, 10); //Left_X, Left_Y, width, height
23    Point pt(5, 5);
24
25    if (pt.inside(rect)) // 해당 좌표가 해당 사각형 내부의 점인지 확인.
26        cout << " pt is inside in rect" << endl;
27    else
28        cout << " pt is not inside in rect" << endl;
29    return 0;
30 }
31
```

```
OpenCVExample (선택 범위)
2
3 using namespace std;
4 using namespace cv;
5
6 int main()
7 {
8     Point3f pt1(1.0f, 0.0f, 0.0f);
9     Point3f pt2(0.0f, 1.0f, 0.0f);
10
11    cout << "pt1&pt2 내적 : " << pt1.dot(pt2) << endl; // 두 좌표의 내적 계산.
12
13    cout << "원점 <-> pt1 사이의 거리 " << norm(pt1) << endl; // 원점에서의 거리 계산.
14
15    cout << "pt1에서 pt2의 외적 : " << pt1.cross(pt2) << endl;
16    cout << "pt2에서 pt1의 외적 : " << pt2.cross(pt1) << endl;
17
18    return 0;
19 }
20
```

//내적 : 두 벡터가 만드는 평면-> 스칼라 값.

//외적 : 두 벡터가 만드는 평면이 아닌 위치에 두 벡터에 직교하는 제3의 벡터가 존재한다 -> 벡터 값

Size 클래스

```
OpenCVExample (전역 범위)
1  #include "opencv2/opencv.hpp"
2
3  using namespace std;
4  using namespace cv;
5
6  int main()
7  {
8      Size rectangle1(30, 20), rectangle2(20, 20);
9
10     Size Plus = rectangle1 + rectangle2;
11     cout << "각 면적의 가로 세로 길이를 더한 면적 : " << Plus << endl;
12     //Plus.width = 가로 , Plus.height = 세로
13
14     cout << "rectangle1의 넓이 : " << rectangle1.area() << endl;
15
16     return 0;
17 }
18
```

Rect 클래스

```
OpenCVExample (전역 범위)
1  #include "opencv2/opencv.hpp"
2
3  using namespace std;
4  using namespace cv;
5
6  int main()
7  {
8      Rect rt1(0, 0, 10, 10), rt2(5, 5, 10, 10);
9
10     Point pt1(5, 5);
11     Size sz1(20, 20);
12
13     Rect rt3 = rt1 + pt1; // 중심좌표 변경
14     Rect rt4 = rt1 + sz1; // width, height값 변경
15
16     cout << "rt1 : " << rt1 << endl;
17     cout << "rt3 : " << rt3 << endl;
18     cout << "rt4 : " << rt4 << endl;
19     /*
20     rt1: [10 x 10 from(0, 0)]
21     rt3: [10 x 10 from(5, 5)]
22     rt4: [30 x 30 from(0, 0)]
23     */
24     return 0;
25 }
26
```

//Rect rt(bottom_Left_X, bottom_Left_Y, width, height)

```
OpenCVExample (전역 범위)
1 #include "opencv2/opencv.hpp"
2
3 using namespace std;
4 using namespace cv;
5
6 int main()
7 {
8     Rect rt1(10, 10, 50, 40), rt2(5, 5, 10, 10);
9
10    Point pt1 = rt1.tl(); // top-left 좌표
11    Point pt2 = rt1.br(); //bottom-right 좌표
12
13    cout << pt1 << " , " << pt2<<endl;
14
15    Point pt3(20, 20);
16    if (rt1.contains(pt3)) // Point.inside()와는 인자가 반대.
17        cout << "pt3 is inside in rt1" << endl;
18
19    Rect rt3 = rt1 & rt2;
20    Rect rt4 = rt1 | rt2;
21    cout << "rt1과 rt2 겹치는 사각형: " << rt3 << endl;
22    cout << "rt1과 rt2 를 포함하는 최소크기 사각형: " << rt4 << endl;
23
24    return 0;
25 }
26
```

```
OpenCVExample.cpp* x
OpenCVExample (전역 범위)
5
6 int main()
7 {
8     Rect rt1(100, 100, 320, 240), rt2(200, 200, 320, 240);
9
10    Rect rt3 = rt1 & rt2;
11    Rect rt4 = rt1 | rt2;
12
13    //그림그리기
14
15    Mat img(600, 800, CV_8UC3); // 8bit 깊이의 uchar 자료형의 3채널 자료형
16
17    namedWindow("ImageExample", WINDOW_AUTOSIZE);
18
19    rectangle(img, rt1, Scalar(255, 0, 0), 2); // 그림그릴곳, 그릴 것, 색상, 두께
20    rectangle(img, rt2, Scalar(0, 255, 0), 2);
21    rectangle(img, rt3, Scalar(255, 255, 0), 2);
22    imshow("ImageExample", img); //imshow("그릴 윈도우 창 이름", 그림이 그려진 행렬 변수)
23    waitKey();
24    circle(img, Point(150, 150), 30, Scalar(255, 255, 220), 2);
25
26    imshow("ImageExample", img);
27    waitKey();
28
29    return 0;
30 }
31
```

