

# Unit 37.

Status	완료
담당자	
마감일	
완료일	@2022년 12월 1일

## Unit 37. 두 점 사이의 거리 구하기

### 37.1 클래스로 점 구현하기

2차원 평면에서 위치를 표현하려면 x와 y의 값이 필요하므로 Point2D 클래스를 구현하고 x와 y를 속성으로 넣는다.

```
class Point2D:
    def __init__(self, x, y):
        self.x = x
        self.y = y
```

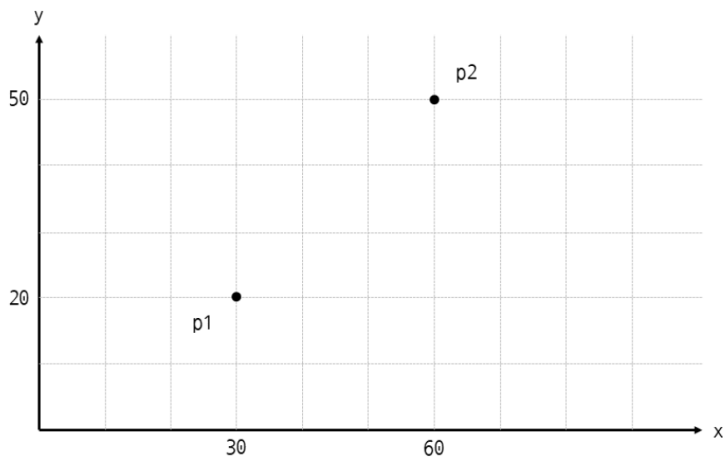
Point2D 클래스로 점 두 개를 만든다.

```
class Point2D:
    def __init__(self, x, y):
        self.x = x
        self.y = y

p1 = Point2D(x=30, y=20) # 점1
p2 = Point2D(x=60, y=50) # 점2

print('p1: {} {}'.format(p1.x, p1.y)) # 30 20
print('p2: {} {}'.format(p2.x, p2.y)) # 60 50
```

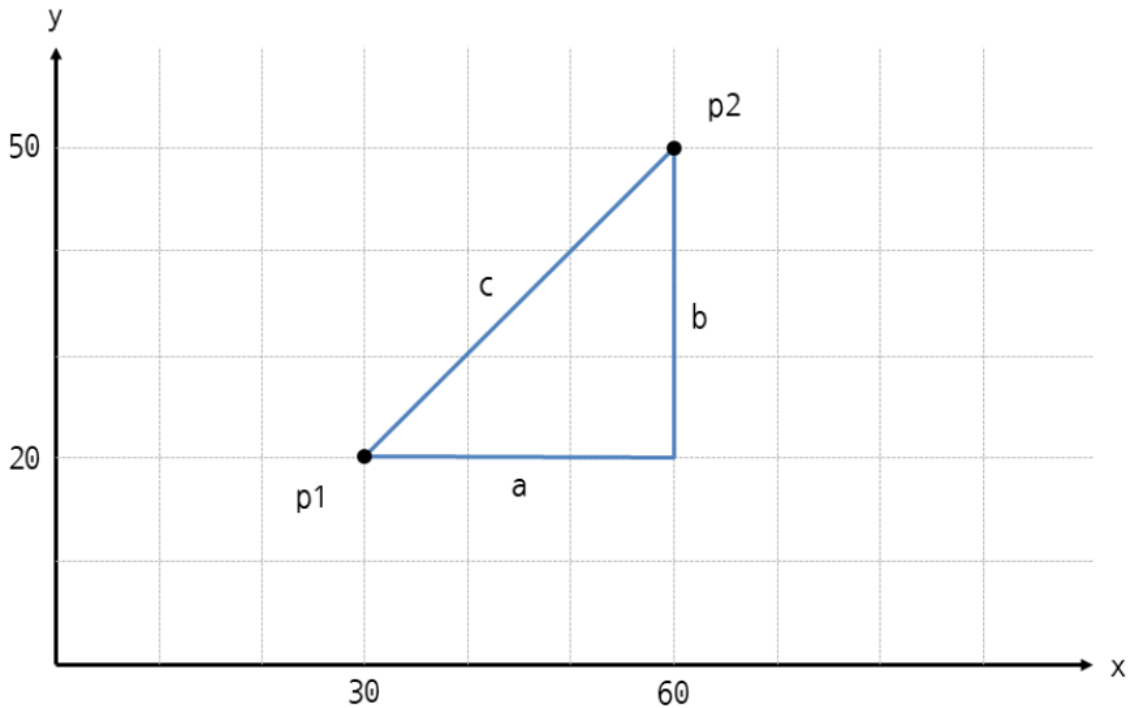
값을 2차원 평면에 표시해보면



### 37.1.2 피타고라의 정리로 두점의 거리 구하기

두 점의 거리를 구하려면 피타고라스의 정리 사용

- 임의의 직각삼각형에서 빗변을 한 변으로 하는 정사각형의 넓이는 다른 두 변을 각각 한 변으로 하는 정사각형의 넓이의 합과 같다.
- $a^2 + b^2 = c^2$



피타고라스의 정리에 대입하려면 선a 와 b의 길이를 구해야 하는데 point2D 클래스의 인스턴스에 두 점의 좌표 정보가 들어있으므로 인스턴스(변수)를 활용

```
a = p2.x - p1.x # 선 a의 길이
b = p2.y - p1.y # 선 b의 길이
```

이제 c의 길이를 계산해야 하는데 그럴려면 제곱근을 구해야 한다.

이때 루트의 기능을 해주는 math 모듈의 sqrt 함수를 사용해준다.

#### • math.sqrt(값)

```
import math

class Point2D:
    def __init__(self, x, y):
        self.x = x
        self.y = y

p1 = Point2D(x=30, y=20) # 점1
p2 = Point2D(x=60, y=50) # 점2

a = p2.x - p1.x # 선 a의 길이
b = p2.y - p1.y # 선 b의 길이

c = math.sqrt((a * a) + (b * b)) # (a * a) + (b * b)의 제곱근을 구함
print(c) # 42.42640687119285
```

여기서는 a의 제곱과 b의 제곱의 합을  $(a * a) + (b * b)$ 처럼 표현했는데 거듭제곱(pow er)을 구하는 pow 함수를 사용해도 된다

#### • math.pow(값, 지수)

```
c = math.sqrt(math.pow(a, 2) + math.pow(b, 2))
```