

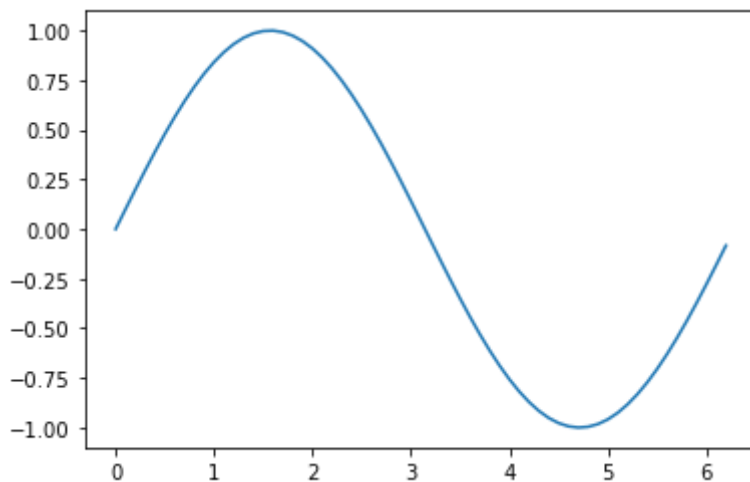
numpy를 이용한 데이터 생성 및 통계

arange()를 이용한 Sin 함수 그리기

```
In [1]: import numpy as np
import matplotlib.pyplot as plt

xlist = np.arange(0, 3.14*2, 0.1)
ylist = np.sin(xlist)

plt.plot(xlist, ylist)
plt.show()
```

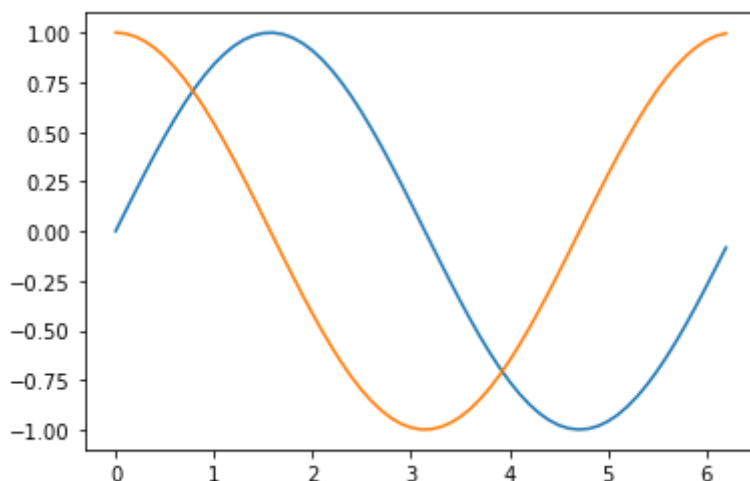


Sin, Cos 함수 그리기

```
In [2]: x = np.arange(0, 3.14*2, 0.1)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1)
plt.plot(x, y2)

plt.show()
```



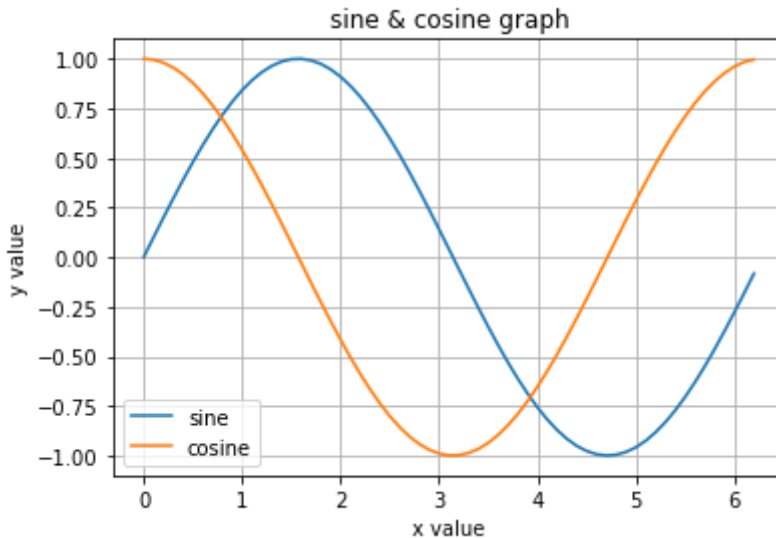
Sin, Cos 자세히 그리기

```
In [10]: x = np.arange(0, 3.14*2, 0.1)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1, label="sine")
plt.plot(x, y2, label="cosine")

plt.title("sine & cosine graph")
plt.xlabel("x value")
plt.ylabel("y value")
plt.grid(True)
plt.legend()

plt.show()
```



np.linspace 함수 이해하기

```
In [11]: a = np.linspace(0, 25, 30) # (시작값, 종료값, 등급의 개수(정수)), 0 부터 25까지 30등분
a
```

```
Out[11]: array([ 0.          ,  0.86206897,  1.72413793,  2.5862069 ,  3.44827586,
        4.31034483,  5.17241379,  6.03448276,  6.89655172,  7.75862069,
        8.62068966,  9.48275862, 10.34482759, 11.20689655, 12.06896552,
        12.93103448, 13.79310345, 14.65517241, 15.51724138, 16.37931034,
        17.24137931, 18.10344828, 18.96551724, 19.82758621, 20.68965517,
        21.55172414, 22.4137931 , 23.27586207, 24.13793103, 25.          ])
```

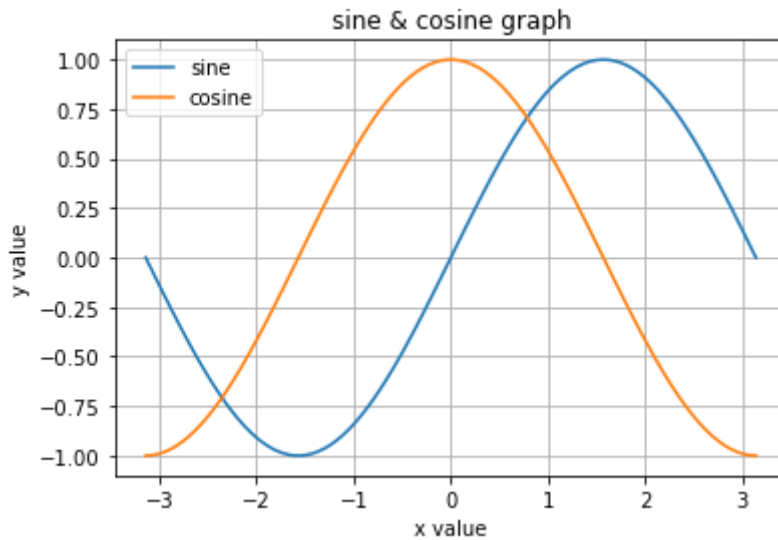
np.linspace() 활용

```
In [12]: x = np.linspace(-np.pi, np.pi, 100)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1, label="sine")
plt.plot(x, y2, label="cosine")

plt.title("sine & cosine graph")
plt.xlabel("x value")
plt.ylabel("y value")
plt.grid(True)
plt.legend()

plt.show()
```



numpy를 활용하지 않는 경우 기본 통계

```
In [14]: scores = []
for i in range(5):
    val = int(input("%d번 성적입력 -> " % (i+1)))
    scores.append(val)

print("\n입력된 전체 점수")
print(scores, "\n")

# 합계, 평균 구하기
sumR = sum(scores)
aveR = sumR / len(scores)

# 분산, 표준편차
r = 0
for i in range(5):
    r += (scores[i] - aveR) ** 2
varR = r / len(scores)
stdR = varR**(1/2)

print("합계: %10d" % sumR)
print("평균: %10.2f" % aveR)
print("분산: %10.2f" % varR)
print("편차: %10.2f" % stdR)
```

```
1번 성적입력 -> 1
2번 성적입력 -> 2
3번 성적입력 -> 3
4번 성적입력 -> 4
5번 성적입력 -> 5
```

```
입력된 전체 점수
[1, 2, 3, 4, 5]
```

```
합계:      15
평균:      3.00
분산:      2.00
편차:      1.41
```

numpy를 활용하는 경우 기본 통계

```
In [15]: scores = []
```

```

for i in range(5):
    val = int(input("%d번 성적입력 -> " % (i+1)))
    scores.append(val)

print("\n입력된 전체 점수")
print(scores, "\n")

sumR = np.sum(scores)
aveR = np.mean(scores)
varR = np.var(scores)
stdR = np.std(scores)

print("합계: %10d" % sumR)
print("평균: %10.2f" % aveR)
print("분산: %10.2f" % varR)
print("편차: %10.2f" % stdR)

```

1번 성적입력 -> 1
 2번 성적입력 -> 2
 3번 성적입력 -> 3
 4번 성적입력 -> 4
 5번 성적입력 -> 5

입력된 전체 점수
 [1, 2, 3, 4, 5]

합계: 15
 평균: 3.00
 분산: 2.00
 편차: 1.41

random 함수들: 난수 생성

- np.random.rand(a, b)
 - 0.0 ~ 1.0사이의 실수형 난수
 - (a, b): 난수로 이루어진 2차원 array 생성
 - b 생략시 난수로 이루어진 1차원 array 생성

In [18]: np.random.rand(5)

Out[18]: array([0.07997921, 0.02063843, 0.20661276, 0.66902775, 0.49723523])

In [17]: np.random.rand(2, 3)

Out[17]: array([[0.16069165, 0.66423984, 0.94562798],
 [0.39323419, 0.7656587 , 0.30842225]])

- np.random.randint(a, b, size = (x, y))
 - a부터 b까지의 정수형 난수 생성
 - size(x, y): 2차원 array로 생성

In [19]: np.random.randint(2, 10, size = 10)

Out[19]: array([9, 3, 8, 2, 9, 3, 7, 4, 3, 8])

In [20]: np.random.randint(1, 11, size = (4, 7))

```
Out[20]: array([[10,  9,  5,  6,  5,  9,  2],
                [ 9,  4,  2,  1,  6,  2,  3],
                [ 8,  2,  5,  8,  7, 10,  5],
                [ 8,  1,  4,  5,  4,  7,  7]])
```

- `np.random.normal(a, b, c)`
 - 정규분포를 갖는 난수 생성
 - a: 평균, b: 표준편차, c: 생성할 개수

```
In [21]: np.random.normal(55, 5, 10)
```

```
Out[21]: array([57.92461365, 50.93258824, 64.39624542, 54.67956491, 54.05794442,
                63.80522077, 65.61537938, 53.0276501 , 59.25791835, 55.69588876])
```