

# numpy 이해와 데이터 생성

송실대학교  
베어드교양대학  
강의선 교수  
iami86@ssu.ac.kr

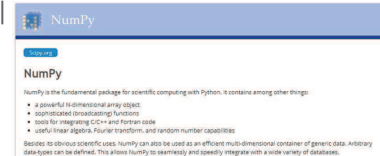
## numpy 의 장점

- 데이터를 생성할 수 있다.
  - 특정 패턴의 수열, 랜덤 수, 특정 분포에 근거한 데이터, 수학 함수 데이터 (삼각 함수 등)
- 많은 데이터를 쉽고 빠르게 처리할 수 있다.
  - 리스트로 하는 것보다 수행 속도가 훨씬 빠름
  - 코드도 훨씬 짧음
- 복잡한 연산을 수행할 수 있다.
  - 통계, 선형 대수(행렬 연산 등), 푸리에 연산 등

## numpy 란 무엇인가?

<https://numpy.org/>

- 넘파이(Numpy)
  - 왜 Numpy 인가? Number 혹은 Numerical 그리고 Python
  - 수치 연산, 과학 연산을 위한 파이썬 외부 라이브러리
  - 데이터분석, 시각화, 머신러닝 등의 작업에 필수적
  - 벡터, 행렬 등의 자료구조 및 연산 지원
- 제공하는 기능들
  - 통계 함수들 : 최대, 최소, 평균, 중간값, 분산, 표준편차, n분위수
  - 수학 함수들 : 삼각함수, 로그함수 등
  - 벡터 및 행렬 연산 : 행렬의 곱, 역행렬, 전치행렬(Array 라는 이름으로 제공)
  - 공학수학, 선형대수학 등



## numpy 설치하기

**pip install numpy 명령 수행**

- 아나콘다 설치를 설치하였다면...
- 데이터 분석에 필수적인 numpy 라이브러리가 설치됨



## numpy가 왜 필요할까?

### ■ 문제1

- 1부터 20까지 3씩 증가되는 수열(리스트)을 생성하시오.
- 리스트의 모든 항목 값에 10을 더하시오.
- 리스트의 모든 항목 값에 2를 곱하시오.

### ■ 문제2

- 0.2부터 2.5까지 0.3씩 증가되는 수열을 생성하시오.
- 수열의 모든 항목 값에 10을 더하시오.
- 수열의 모든 항목 값에 2를 곱하시오.

5

## numpy 왜 필요할까? : 문제 1

### ■ 문제 1

- 1부터 20까지 3씩 증가되는 수열(리스트)를 생성하기 / 생성된 값을 출력하기
- 수열의 모든 값에 10을 더하기 / 생성된 값을 출력하기
- 수열의 모든 값에 2를 곱하기 / 생성된 값을 출력하기

6

## numpy 왜 필요할까? (문제1)

- 1부터 20까지 3씩 증가되는 리스트를 생성하기
- 수열의 모든 값에 10을 더하기
- 수열의 모든 값에 2를 곱하기

```
In [70]: x = list( range(1, 20, 3) )
```

```
In [71]: x
```

```
Out [71]: [1, 4, 7, 10, 13, 16, 19]
```

```
In [72]: for i in range(0, len(x)) :  
          x[i] = x[i] + 10
```

```
In [73]: x
```

```
Out [73]: [11, 14, 17, 20, 23, 26, 29]
```

```
In [74]: for i in range(0, len(x)) :  
          x[i] = x[i] * 2
```

```
In [75]: x
```

```
Out [75]: [22, 28, 34, 40, 46, 52, 58]
```

7

## numpy 왜 필요할까? (문제2)

### ■ 문제2

- 0.5부터 2.5까지 0.3씩 증가되는 수열(리스트)를 생성하기
- 수열의 모든 값에 10을 더하기
- 수열의 모든 값에 2를 곱하기

### ■ 문제1로 문제2를 해결할때 문제점

- range() 함수로는 실수에 대한 수열을 생성 불가능
- 반복문을 사용해야 하는 번거로움
- 지저분한 소스코드

8

## numpy 왜 필요할까? (문제2)

- 0.5부터 2.0까지 0.3씩 증가되는 Array 생성
- 수열의 모든 값에 10을 더하기
- 수열의 모든 값에 2를 곱하기

```
In [1]: import numpy as np
In [8]: x = np.arange(0.5, 2.0, 0.3)
In [9]: x
Out[9]: array([0.5, 0.8, 1.1, 1.4, 1.7])
In [10]: x = x + 10
In [11]: x
Out[11]: array([10.5, 10.8, 11.1, 11.4, 11.7])
In [12]: x = x * 2
In [13]: x
Out[13]: array([21. , 21.6, 22.2, 22.8, 23.4])
```

9

## 문제 : numpy 왜 필요할까? (점수 평균)

- 5명의 학생이 특강수업에 참여하고 있다.
- 이 수업은 중간고사 50%, 기말고사 50%의 비율로 최종 점수가 계산된다.
  - 중간고사 성적 : [90, 80, 70, 60, 50]
  - 기말고사 성적 : [80, 70, 60, 50, 40]
- 계산하기
  - 중간고사와 기말고사를 평균한 결과를 구하시오. [85, 75, 65, 55, 45]
  - 전체적으로 +7의 가산점을 적용하시오. [92, 82, 72, 62, 52]

10

## 잠깐! 리스트 복습하기

```
>>> a = [10, 20, 30]
>>> b = [11, 22, 33]
>>>
>>> a[0] + b[0]
21
>>>
>>> (a[1] + b[1]) / 2
21.0

>>> a + b
[10, 20, 30, 11, 22, 33]
>>> a * 2
[10, 20, 30, 10, 20, 30]
>>> b * 0.5
Traceback (most recent call
```

리스트는 실제 조금이라도 복잡한 연산에서는 활용하기 어렵다.

11

## numpy 사용 안하는 경우

- 아래의 기본 코드를 입력하시오.

```
중간 = [90, 80, 70, 60, 50]
기말 = [80, 70, 60, 50, 40]
최종 = []
```
- 중간, 기말성적의 평균 점수를 최종 List에 저장하시오.
- 최종 성적에 +7 가산점을 적용하시오.

12

## numpy 사용 안하는 경우

```
중간 = [90, 80, 70, 60, 50]
기말 = [80, 70, 60, 50, 40]
최종 = []
```

```
n = len(중간)
for i in range(n):
    v = (중간[i] + 기말[i]) / 2
    최종.append( v )
```

```
for i in range(n):
    최종[i] = 최종[i] + 7

print( 최종 )
```

13

## numpy 사용

```
import numpy as np
```

```
중간 = [90, 80, 70, 60, 50]
기말 = [80, 70, 60, 50, 40]
```

```
mid = np.array( 중간 )
fin = np.array( 기말 )
```

```
score = (mid + fin)/2 + 7
```

```
print( score )
```

```
import numpy as np
```

```
중간 = [90, 80, 70, 60, 50]
기말 = [80, 70, 60, 50, 40]
```

```
mid = np.array( 중간 )
fin = np.array( 기말 )
```

```
score = mid + fin
score = score / 2
score = score + 7
```

```
print( score )
```

14

## numpy Array 생성하기

```
import numpy as np
```

### 1차원 행렬

- 리스트를 만든 후 Array로 변환하여 생성

type( data1 ) 실행  
type( arr1 ) 실행

```
In [2]: data1 = [1,2,3,4,5]
data1
```

```
Out [2]: [1, 2, 3, 4, 5]
```

```
In [4]: # numpy를 이용하여 array 정의하기
# 1. 위에서 만든 python list를 이용
arr1 = np.array(data1)
arr1
```

```
Out [4]: array([1, 2, 3, 4, 5])
```

15

## numpy Array 생성하기

### 2차원 행렬

- 2차원 리스트 생성 후 array로 변환

```
In [21]: arr4 = np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
arr4
```

```
Out [21]: array([[ 1,  2,  3],
 [ 4,  5,  6],
 [ 7,  8,  9],
 [10, 11, 12]])
```

16

수고하셨습니다.