머신 러닝·딥러닝에 필요한 수학 기초 with 파이썬

Introduction to Python

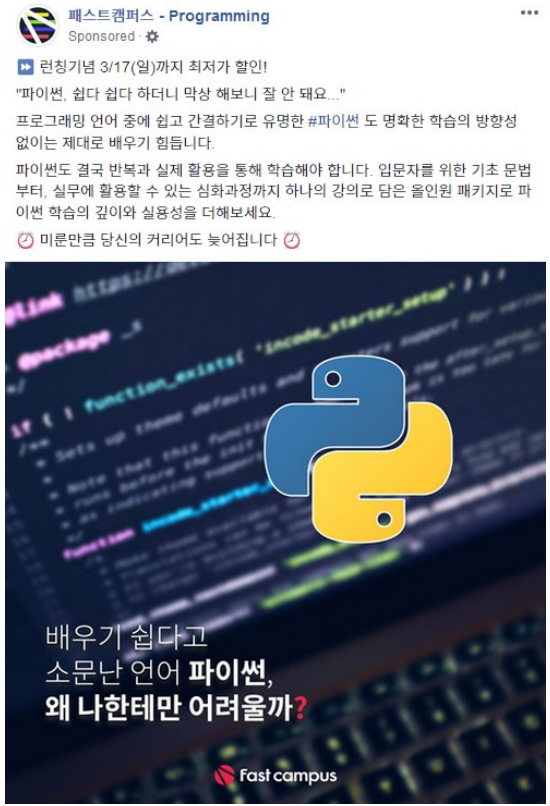
조준우
metamath@gmail.com

개요

- **기초 문법**
  - 머신러닝 알고리즘을 구현하기 위한 최소 기초 문법 소개
- **Numpy**
  - 행렬-벡터 계산을 위한 numpy 라이브러리 소개
- **주요참고문헌**
  - A Byte of Python : python 2.7 기준, Free
  - 처음시작하는 파이썬Introducing Python, 한빛미디어
  - http://cs231n.github.io/python-numpy-tutorial/

왜 Python인가?

- 공짜, 많이 씀
- 쉬운 문법(?)



왜 Python인가? - 맛보기

- 들여쓰기

C/C++

```
int factorial(int x) {
    if(x == 0) {
        return 1;
    }
    else {
        return x * factorial(x - 1);
    }
}
```

```
int factorial(int x) {
if(x == 0) { return 1; } else
{ return x * factorial(x - 1); } }
```

Python

```
def factorial(x) :
    if x == 0 :
        return 1
    else :
        return x * factorial(x - 1)
```

```
def factorial(x) : if x == 0 :
return 1 else :
return x * factorial(x - 1)
```

Python 2 vs 3

- Python 2
  - 2000년 10월 16일 배포
  - python 3.x 가 발표되면서 구분을 위해 python 2.x 버전을 의미
- Python 3
  - 2008년 12월 3일 배포
  - 2.x대 버전의 파이썬과 하위호환성이 없음
  - 처음 시작한다면 무조건 python 3으로 시작
- Version Check
  - `python --version`

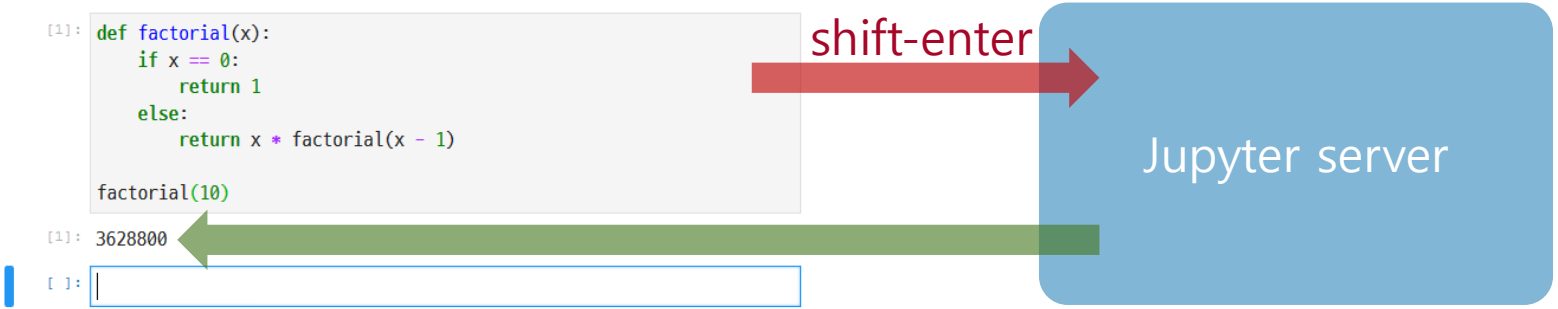
Install Python

- Python 설치
  - python.org에서 언어 패키지만 설치
  - 모든 라이브러리를 pip install로 직접 다 설치
  - 버츄얼 환경 직접 구성

- 3rd party 설치
  - anaconda.org 에서 언어+각종 라이브러리를 패키지 제공
  - 대부분 패키지가 기본 제공, numpy, matplotlib, scipy
  - 버츄얼 환경 기본 제공

개발 환경

- **Jupyter notebook**
  - 서버에 웹브라우저로 접속하여 코드를 입력하고 결과를 HTML로 돌려 받음
  - 셀cell이라 부르는 코드 블록에 코드를 입력하고 shift-enter로 실행



and print

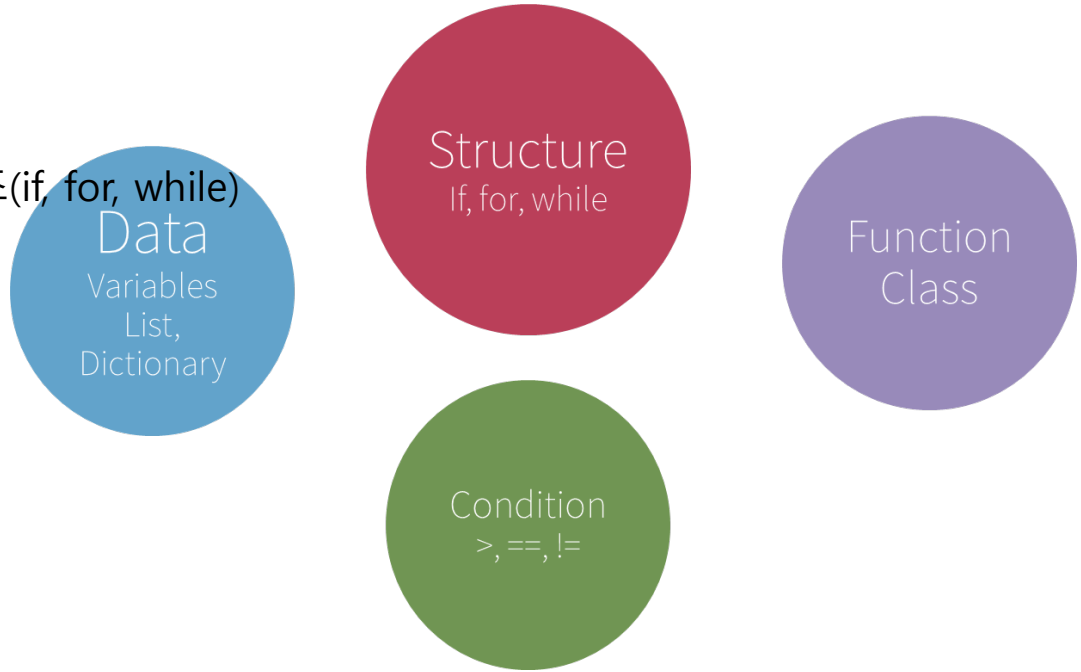
- **주석**
  - '#' 문자 뒤에 따라오는 짧은 문장
  - 코드로 해석되지 않음
  - 내가 만든 코드를 내가 봐도 모르기 때문에 항상 주석을 친절히 달자!
    - `print('Hello world') # Note that print is a statement`

- **print**
  - 변수값을 출력하거나 간단한 디버깅 용도로 사용
  - { }와 format함수를 사용해서 출력 양식을 조정

```
a = 10
print("a is {}".format(a))
# 결과: a is 10
```

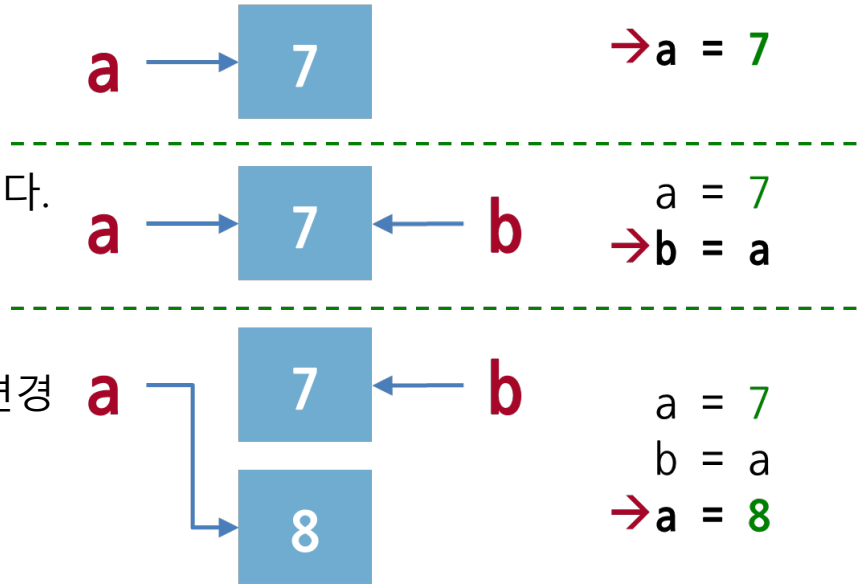
프로그래밍 구조

- 자료를 담는 그릇
  - 변수, 자료구조
- 논리의 흐름
  - 조건(==, !=, <, >), 코드 구조(if, for, while)
- 효율성을 위한 도구
  - 함수, 클래스



변수Variable: 불변과 가변

- **변수** : 자료가 담긴 그릇에 붙인 이름표
  - 변수, 자료구조
- 불변immutable
  - 자료에 변화가 생기면 자료를 다시 만든다.
  - 숫자, 문자열, 튜플
- 가변mutable
  - 자료에 변화가 생기면 그 자료 자체를 변경
  - 리스트, 딕셔너리 같은 것들이 있음



jupyter python.ipynb

Integer is immutable

변수: 부울Booleans

- True 또는 False를 나타내는 자료형
- 조건문 등에 사용

```
# http://cs231n.github.io/python-numpy-tutorial/
t = True
f = False
print(type(t))  # Prints "<class 'bool'>"
print(t and f)  # Logical AND; prints "False"
print(t or f)   # Logical OR;  prints "True"
print(not t)    # Logical NOT; prints "False"
print(t != f)   # Logical XOR; prints "True"
```

```
if t is True :
    print('t is True')
else :
    print('t is not True')
```

NOTE

XOR : (T,T):F, (T,F):T, (F,T):T, (F,F):F

변수: 문자열Strings

- ' ' 또는 " "로 둘러 쌓인 문자
  - " "인 문자열 안에 '가 들어갈 수 있고,
  - ' '인 문자열 안에 "가 들어갈 수 있음
- 강력한 문자열 처리 기능을 제공
  - https://docs.python.org/3.5/library/stdtypes.html#string-methods
- 문자열 포맷팅

```
# 소수점 이하 셋째 자리까지 부동 소숫점 숫자 표기
print('{:.3f}'.format(1.0/3)) # 결과 0.333

# 밑줄(_)로 11칸을 채우고 가운데 정렬(^)하기
print('{:_^11}'.format('hello ' ))   # 결과 ___hello___
```

NOTE

도트연산자 .
프로그램상의 객체(숫자, 문자 등등)의 기능이나 속성에 접근할 때 사용

변수: 숫자Numbers

- 정수integers와 소수floats
  - 정수 : ...,-1,0,1,2,3,...
  - 소수 : 1.2, 1.234,...
- 다음 연산자 적용 가능

연산자	설명	예	결과
+	더하기	5+8	13
-	빼기	90-10	80
*	곱하기	4*7	28
/	부동소수점 나누기	7 / 2	3.5
//	정수나누기	7 // 2	3
%	나머지	7 % 3	1
**	지수	3**4	81

자료구조Containers

- 앞서 배운 기본 자료형을 모아서 관리하는 도구
- 여러 개의 값들이 열거되어 저장되어 시퀀스sequence형 자료 라고도 함
- 튜플tuple, 리스트list, 사전dictionary

자료구조:튜플Tuples

- 여러 개의 객체를 모아 담는 데 사용
- 정의 : 객체를 괄호로 묶어 줌 (,)
- 불변, 수정이 불가능

```
zoo = ('python', 'elephant', 'penguin')
print('Number of animals in the zoo is', len(zoo))
print(zoo[1])

# 결과
Number of animals in the zoo is 3
elephant
```

- 항목 한 개 짜리 튜플

```
singleton = (2) # 그냥 2
singleton = (2 , )
```

NOTE

[] 인덱스 연산자
시퀀스형 자료의 요소에 접근할 때 사용

NOTE

패킹 언패킹

```
numbers = 1, 2, 3    # 패킹
a, b, c = numbers    # 언패킹
```

자료구조:리스트Lists

- 순서대로 정리된 항목들을 담고 있는 자료 구조
- 정의 : 객체를 대괄호로 묶어 줌 [ ]
- 가변, 수정이 가능

```
# This is my shopping list
shoplist = ['apple', 'mango', 'carrot', 'banana']
print(shoplist)
# 결과
['apple', 'mango', 'carrot', 'banana']

# Append
shoplist.append('rice')
print(shoplist)
# 결과
['apple', 'mango', 'carrot', 'banana', 'rice']
```

```
# Replace
shoplist[0] = 'lego'
print(shoplist)
# 결과
['lego', 'mango', 'carrot', 'banana', 'rice']

# Delete
del shoplist[1]
print(shoplist)
# 결과
['lego', 'carrot', 'banana', 'rice']
```

자료구조:중첩Nested

- 튜플, 리스트는 숫자, 문자 뿐 아니라 리스트, 사전 등 모든 자료 구조를 담을 수 있음
- 중첩된 시퀀스형은 연속된 [ ][ ] 로 인덱싱

```
a = (1, 2, 3)
b = [a, 1, 2,]
print(b)
# 결과
[(1, 2, 3), 1, 2]
```

```
print(b[0][1])
#결과
2
```

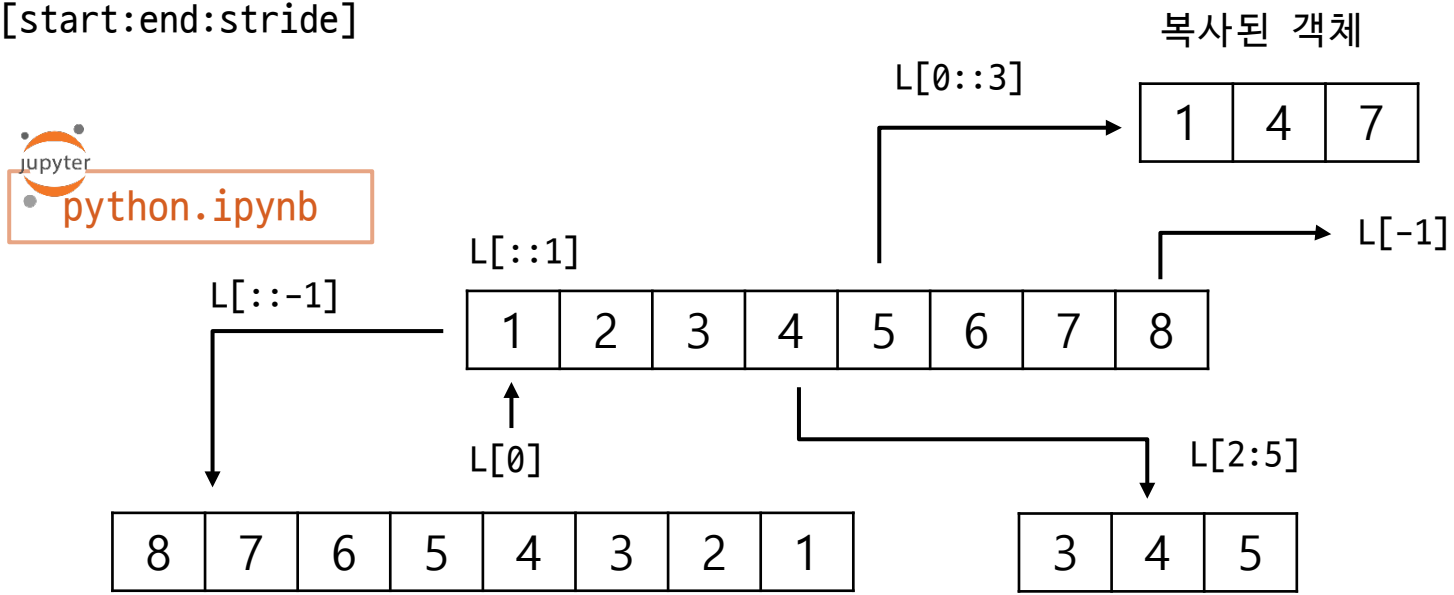
NOTE

리스트를 문자열로 바꾸기

```
mylist = ['Brazil', 'Russia', 'India', 'China']
print(''.join(mylist))
```

슬라이싱Slicing

- [i] 형식으로 요소에 접근하는 방식의 확장
- Minus index : L[-1]
- [start:end:stride]



자료구조:사전Dict.

- 문자 그대로 사전
- 키에 해당하는 값을 저장
- 정의 : {'key':'value' }
- 항목 순서 유지 안됨
- 신경망 가중치를 저장하는 용도

```
ab = {  'Swaroop'    : 'swaroop@swaroopch.com',
        'Larry'      : 'larry@wall.org',
        'Matsumoto'  : 'matz@ruby-lang.org',
        'Spammer'    : 'spammer@hotmail.com'
    }
print(ab)
# 결과
{'Swaroop': 'swaroop@swaroopch.com', 'Matsumoto':
'matz@ruby-lang.org', 'Larry': 'larry@wall.org',
'Spammer': 'spammer@hotmail.com'}
print("Swaroop's address is", ab['Swaroop'])
# 결과
Swaroop's address is swaroop@swaroopch.com
```

코드구조:if else

- 조건을 판별해서 실행하고 싶은 부분만 실행할 때 사용

```
a = -3
if a > 0 :
    print(“positive”)
else :
    print(“0 or negative”)

a = 100
if a = 100 :
    print(“a = 100”)
else :
    print(“a != 100”)

File "<ipython-input-227-e1db640460af>", line 3
    if python = 100:
              ^
SyntaxError: invalid syntax
```

연산자	설명
==	같다
!=	다르다
<	~보다 작다
<=	~보다 작거나 같다
>	~보다 크다
>=	~보다 크거나 같다
in	~안에 있다

코드구조:for else

- 열거형 자료의 각 요소마다 코드 블록을 반복
- 주로 range()함수를 사용해서 반복할 구간을 지정
- 리스트, 사전에 대해서 동일하게 작동

```
for i in range(1, 5):
    print(i)
else :
    print('The for loop is over')

# 결과
0
1
2
3
4
The for loop is over
```

NOTE

range([start,] stop [,stride])
start부터 stop까지 stride씩 이동하면서 숫자를 리턴

함수Function

- 재사용 가능한 이름이 붙여진 프로그램의 조각
- def 키워드를 통해 정의

```
def say_hello():
    # block belonging to the function
    print('hello world')
```

- 매개변수parameters
  - 함수로 넘겨지는 값들의 이름
- 인자arguments
  - 함수에 넘겨준 값
  - 기본 인자 값을 지정할 수 있음

```
def Qr(x, y=1):
    q, r = 0, 0
    while True:
        x -= y
        if x > 0 :
            q += 1
        elif x < 0:
            r = y + x
            break
        else:
            q += 1
            break
    return (q, r)
```

Lambda function

- 이름 없는 한 줄 짜리 함수
- lambda 키워드를 통해 정의

```
p = lambda x, y : x + y

p(3,4)
```

```
def plus(x, y)
    return x + y

plus(3,4)
```