

BigData

인공지능, 딥러닝, 머신러닝의 개요

TIOBE INDEX - 현 상황의 프로그래밍 언어 사용 빈도

Python 설치

IDLE 바탕화면 바로가기

<Jupyter Notebook>

웹 브라우저에서 파이썬 코드를 작성하고 실행까지 해볼 수 있는 주피터 노트북(jupyter notebook)

대화형 모드 - 한 줄 씩

<사칙 연산>

```
a=100
```

```
b=50
```

```
result = a + b
```

```
print(result)
```

```
150
```

```
print(a, '+', b, '=', result)
```

```
100 + 50 = 150
```

```
result = a - b
```

```
print(a, '-', b, '=', result)
```

```
100 - 50 = 50
```

```
result = a*b
```

```
print(a, '*', b, '=', result)
```

```
100 * 50 = 5000
```

```
result = a/b
```

```
print(a, '/', b, '=', result)
```

```
100 / 50 = 2.0
```

Python 역시 객체지향 프로그래밍의 Class 있음.

```
import os
```

```
a=100
```

```
b=50
```

```
result = a + b
```

```
print(result)
```

```
150
```

```
print(a, '+', b, '=', result)
```

```
100 + 50 = 150
```

```
result = a - b
```

```
print(a, '-', b, '=', result)
```

```
100 - 50 = 50
```

```
result = a*b
```

```
print(a, '*', b, '=', result)
```

```
100 * 50 = 5000
```

```
result = a/b
```

```
print(a, '/', b, '=', result)
```

100 / 50 = 2.0

```
os.system("pause")
```

100

50

100 + 50 = 10050

Traceback (most recent call last):

File "C:\Users\admin\Desktop\Python\source\Code02-02.py", line 5, in <module>

```
result = a - b
```

TypeError: unsupported operand type(s) for -: 'str' and 'str'

기본적으로는 자료형을 선언하지 않아도 자료형을 파악하지만 input()은 String으로 간주를 함.

<거북이를 이용해서 그림 그리기>

함수선언부터 시작해서 Class개념이 적용되었음.

```
def screenLeftClick(x,y):
```

```
global r, g, b ...
```

형태로 변수와 함수를 선언

<데이터형>

List = Java Array 개념(배열)

리스트와 배열의 차이점은 리스트는 서로 다른 데이터형도 하나로 묶을 수 있고, 배열은 동일한 데이터형만 묶을 수 있다.

```
aa = []
```

```
for i in range(0, 4) : - Java for~each와 같은 개념
```

```
aa.append(0)
```

```
hap = 0
```

```
for i in range(0, 4) :
```

```
aa[i] = int(input(str(i+1) + "번째 숫자 : " )) - Java Wrapper Class와 같은 개념
```

```
hap = aa[0] + aa[1] + aa[2] + aa[3]
```

```
print("합계 ==> %d" % hap)
```

* List는 자료를 추가 할 수 있음.

유연성 확보

1차원, 2차원, 3차원 List

데이터 분석 행렬에도 사용

Tuple, Dictionary

Tuple : List와 사용법이 비슷하면서도 다르다 List는 []를 사용 Tuple은 ()로 생성

값을 수정 할 수 없고, 읽기만 가능해 읽기 전용 자료를 저장할 때 사용(상수는 튜플 사용)

JS에서 Const와 let과 같은 유사개념

Dictionary는 사전을 생각하면 된다.

Dictionary는 두 값이 한 쌍으로 묶인 자료구조이다.

자료 구조 {}로 구성 Key와 Value를 사용하여 1쌍으로 구성

{apple:사과}와 같이 사용
JSON 형태와 비슷하게 사용

<Python에서의 Class> - <https://wikidocs.net/85> 참조

```
class Singer:                                # 가수를 정의하겠느니라...
    def sing(self):                            # 노래하기 메서드
        return "Lalala~"
        taeji = Singer()                      # 태지를 만들어랏!
        taeji.sing()                          # 노래 한 곡 부탁드립니다~
        'Lalala~'
```

이와 같이 사용한다.

<GUI와 파일 입출력>

App

Web, Mobile - HTTP를 통신 기반으로 하는 프로그램

PC App - 네트워크가 필요 없는 프로그램

<Data 분석>

MongoDB - NoSQL 활용

CRUD 적용 - SQL

NoSQL은 CRUD 사용 X

NoSQL 중 대표적으로 쓰는 것이 MongoDB

<MongoDB 설치>

MongoDB Community Server에서 다운로드

Compass 설치

mongodb://localhost:27017 확인

환경변수 등록해줘야 함

C:\Program Files\MongoDB\Server\[버전]\bin

환경변수 등록 후 아래 같이 CMD에서 입력

C:\Users\admin>mongod --version

db version v7.0.0

Build Info: {

"version": "7.0.0",

"gitVersion": "37d84072b5c5b9fd723db5fa133fb202ad2317f1",

"modules": [],

"allocator": "tcmalloc",

"environment": {

"distmod": "windows",

"distarch": "x86_64",

"target_arch": "x86_64"

}

}

```
C:\Users\admin>mongos --version
mongos version v7.0.0
Build Info: {
  "version": "7.0.0",
  "gitVersion": "37d84072b5c5b9fd723db5fa133fb202ad2317f1",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```

<MongoDB 구조>

Document, Collection, DB

DB 안에 Collection(SQL에서는 Table 개념) 안에 Document가 있는 구조

Document(SQL에서의 Row개념) - JSON과 유사한 BSON(Binary JSON) 구조로 되어 있음.

```
{
  name: "Karoid",
  age: 26,
  status: "B",
  groups: ["family", "sports"],
}
```

이와 같이 BSON구조의 도큐먼트를 사용함

<Anaconda 설치>

Node.js 같은 경우 버전이 다양하기 때문에 버전 관리를 해야 함

Data Tool Kit - Anaconda

Container - Docker, K8S(Kubernetes) 등 시스템 구축을 위해서 버전관리를 하는 프로그램을 의미

Python Data Science ToolKit = Anaconda

ipynb = 쥬피터 노트북 확장자