Python Basic

Python 프로그램 설치

- 설치 방법 1
  - Python 3.x 설치

- 설치 방법 2 (recommended)
  - Anaconda 설치

- Anaconda 란 ?
  - Python / R 기반 데이터 사이언스 플랫폼
  - Package 관리가 용이
  - Jupyter Notebook, Spyder 등 기본 프로그램 사용이 용이

Python 특징

- Easy to code and read
  - High level language
  - Interpreted language
  - Dynamically typed
- Large standard library
- Highly extensible
  - easy to import external package & library
- short and powerful coding

- Largest warehouse of machine learning & deep learning libraries

A Python first program

```
a,b = 0, 1
while b < 100:
    print (b)
    a,b = b, a+b
```

- multiple assignment
- no type declaration
- indentation

Multiple assignment

- 두 변수 값을 스위칭하는 코드

```
a = 1
b = 2
t = a
a = b
b = t
```

vs.

```
a,b = 1,2
a,b = b,a
```

Control flow: if, elif

```
x=1
while  x > 0:
    x = int(input("Please enter number:"))
    if x <= 0:
        print ('프로그램을 중단합니다.')
    elif x % 2 == 1:
        print (x, '홀수입니다.')
    else:
        print (x, '짝수입니다')
```

Control flow: for

- use “range()”

```
i=0
While (i<10):
    print (i, i**2)
    i += 1
```

(vs)

```
for i in range(10):
    print (i, i**2)
```

- range() 예
  - range(10)
  - range(1, 11, 2)

함수

```
def compute_two(N):

    sum = 0
    multiply = 1
    for k in range(1, N):
        sum += k
        multiply *= k

    return sum, multiply

a, b = compute_two(11)
```

String

- s = 'hello world'
- concatenate with + or neighbors
  - word = 'Help' + s
- subscripting of strings
  - 'Hello'[2] → 'l'
  - 'Hello'[1:2] → 'el'
- immutable: 각 element의 변경은 불가
  - s[2] = 'x'

list

- list는 순서가 있는 element의 집합
- lists can be heterogeneous. element의 각 타입이 달라도 됨.
  - a = ["mouse", [8, 4, 6], 5]
- Lists can be indexed and sliced:
  - a[1]
  - a[0:2]
- Lists can be manipulated: mutable
  - `a[2] = a[2] + 23`
  - `a[0:2] = [1,12]`
- python의 list는 겉으로 보기에는 배열(array) 같지만, 내부적으로는 연결리스트(linked list)로 구현되어 있음.

Python 실습 문제1

- List 를 입력받아 소팅하는 Bubble sort 함수 bubble_sort(a) 를 정의할 것.

- list 변수 a 에 임의의 1과 1000 사이의 정수 10개를 입력하고, 출력할 것.

- bubble_sort()를 호출하여, 소팅이된 list를 출력할 것.

List comprehension

- 주어진 리스트의 각 element를 변경하여 새로운 리스트를 만들어내는 한 문장의 구조

```
pow2 = []
for x in range(10):
        pow2.append(2 ** x)
```

(VS.)

```
pow2 = [2 ** x for x in range(10)]
```

List comprehension

(1) i*f else* statement

```
numbers = [1, 2, 3, 4, 5]
result = ['even' if num % 2 == 0 else 'odd' for num in numbers]
```

(2) Nested *for vs. zip*

```
A = [1, 2]
B = ['a', 'b']

result = [(x, y) for x in A for y in B]
result = [(x, y) for x, y in zip(X, Y)]
```

tuple

- 투플은 각 element가 순서는 있고, 이름은 없는 구조체
- () 로 표시
  - t = (1, 2, 'a')
- element 접근 시는 []
  - t[1]
- Immutable: cannot assign to individual items
  - Same as string
- 전형적인 예: a list of tuples
  my_list = [(1,a), (2,b), (3,c)]
- MySQL 에서 SQL을 통해서 insert 시 tuple이 활용됨.

dict

- 키(key), 값(value) 쌍을 매핑하여 저장하는 집합. 순서없음.
  - key/value 형태로 데이터를 저장하는 구조
- python에서는 구조체(structure)를 미리 정의해서 사용하지 않고, 주로 dict를 사용.
- {}로 표시
  - d = {"a" : 1, "b":2, "c":3}
- []을 통해서 key에 대한 value값을 접근
  - d['a']
- mutable: 특정 key의 value 변경 가능
  - d['a'] = 100
- 전형적인 예: a list of dict
  - my_list = [{1:'a', 2:'b'}, {1:'c', 3:'e'}]
- MySQL에서 SQL 검색결과의 각 투플을 dict로 반환함.

dict

```
d = {"a" : 1, "b":2, "c":3}
print(d['a'])

for key in d:
    print(key)

for v in d.values():
    print(v)

for k, v in d.items():
    print(k, v)

d['d'] = 4
del d['b']
```

Python 실습문제2

- 주어진 문장에서 alphanumeric 문자를 추출하여, list를 만들어서 출력하라.

- 해당 리스트의 각 문자에 대하여 count 를 저장하는 dict를 만들어서 출력하라.

- 해당 dict를 다시 문자의 count 순으로 소팅하고 출력하라.