[PYTHON 프로그래밍] 데이터 분석

Terminal에서 cd 경로로 들어가서, jupyter notebook 작성

range(start, end, by)

Start에서 시작해서 end까지 by만큼 증가

1 - 100 사이의 홀수만 출력:

1부터 101 전까지 간격 2씩 출력

1 - 100 사이의 짝수만 출력:

1부터 101전까지 간격 2씩 출력

for문:

for는 반복문이므로 하나씩 반복한다.

**for i in range(1, 101, 2):**

**print(i)**

구구단(2단~9단) 만들어보기

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

**for j in range(1, 10):**

**print(i, “\*”, j, “=”, i \* j)**

피라미드 만들어보기

print(‘\*’ \* 10)

if문

# 1~100 사이의 짝수 홀수 출력

**for i in range(1, 101):**

**if i % 2 == 0:**

**print(i)**

**for i in range(1, 101):**

**if i % 2 == 1:**

**print(i)**

list 자료 구조

# [] 대괄호를 통하여 list자료 구조를 선언

**odd = [1, 3, 5]**

**odd**

# for문을 이용하여 1~100사이의 홀수를 list안에 담기

**odd\_100 = [x for x in range(1, 101, 2)]**

**odd\_100**

# len() : list의 길이 확인

**len(odd\_100)**

**for i in odd\_100:**

**print(i)**

# list slicing

# odd\_100에서 1~10번째 값만 추출

**print('1~10번째 값 : ', odd\_100[0:10])**

**print('1~10번째 값 : ', odd\_100[:10])**

**print(len(odd\_100[0:10]))**

# odd\_100에서 40~50번째 값만 추출

**print('41~50번째 값 : ', odd\_100[40:50])**

**print('41~50번째 값 : ', odd\_100[40:])**

# odd\_100에서 마지막 값 추출

**print('마지막 값 : ', odd\_100[49])**

**print('마지막 값 : ', odd\_100[-1])**

# [start:end:by] 세미콜론 2개를 통한 index slicing

**num = [x for x in range(1, 101)]**

**# num[0:100:2]**

**print(num[::2])**

# 역순 출력

**print(num[::-1])**

# append() : list에 값 추가

**num.append(101)**

**num**

# odd\_100에 방금 추가한 값 확인해보기

**num[-1]**

# del : list 값 삭제

# 괄호안에 해당 index를 통하여 접근

**del num[-1]**

**num**

# 나이를 넣은 list 생성

**age = [22, 21, 23, 25, 24, 20]**

# sort() : list 정렬

**age.sort()**

**print(age)**

**print(age[::-1])**

dictionary 자료 구조

key : value라는 구조로 대응 관계를 표현

ex) {'name' : '양용준', 'school' : '연세대', 'age' : 'secret'}

{}를 통하여 표현하고 ,를 통해 구분

**dic = {'name': '박세희', 'school': '숙명여대', 'age': 23}**

**dic**

# key를 통하여 value 접근

**dic['name']**

**dic['school']**

**dic['age']**

# dictionary에 key, value 추가

# value 수정

**dic['alcohol'] = 1**

**dic['alcohol']**

**dic['age'] = 23**

**dic['age']**

# del dictionary[key] : key를 통해 key:value 한쌍 지우기

**del dic['alcohol']**

**dic**

# dictionary key 모음

**dic.keys()**

# dictionary value 모음

**dic.values()**

# for문을 통하여 key를 통해 각 key에 해당하는 value 접근

**for i in dic.keys():**

**print(dic[i])**

# dictionary key,value 모음

**dic.items()**

# for문을 통하여 key, value 접근

**for k,v in dic.items():**

**print('key', k, '. value :', v)**

# 새로운 dictionary 생성

**dic = {'a': 2, 'b': 3, 'c': 1, 'd': 7, 'e': 5}**

**dic**

**dic.items()**

# sorted(a, key = lambda x : 정렬 기준)

# value를 기준으로 정렬

**sorted(dic.items(), key = lambda x:x[1])**

# Set 자료구조

# 집합 : 데이터 분석에서 Set 자료구조를 사용하는 이유는 중복된 값을 제거 해주기 때문

# set 자료구조 만들기

**set1 = {1, 2, 3, 4, 5, 6, 6, 6}**

**set2 = {4, 5, 6, 7, 8, 9}**

**set1**

# intersection : 교집합

**set1 & set2**

**set1.intersection(set2)**

**set1 & set2 == set1.intersection(set2)**

# union: 합집합

**set1 | set2**

**set1.union(set2)**

**set1 | set2 == set1.union(set2)**

# difference :차집합

**print('set1 - set2 :', set1 - set2)**

**print('set2 - set1 :', set2 - set1)**

**print('set1 - set2 :', set1.difference(set2))**

**print('set2 - set1 :', set2.difference(set1))**

마무리할 때는 Terminal에서 Ctrl + C

<http://localhost:8888/notebooks/python%20%E1%84%80%E1%85%B5%E1%84%8E%E1%85%A9.ipynb#%EC%A7%91%ED%95%A9-:-%EB%8D%B0%EC%9D%B4%ED%84%B0-%EB%B6%84%EC%84%9D%EC%97%90%EC%84%9C-Set-%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0%EB%A5%BC-%EC%82%AC%EC%9A%A9%ED%95%98%EB%8A%94-%EC%9D%B4%EC%9C%A0%EB%8A%94-%EC%A4%91%EB%B3%B5%EB%90%9C-%EA%B0%92%EC%9D%84-%EC%A0%9C%EA%B1%B0-%ED%95%B4%EC%A3%BC%EA%B8%B0-%EB%95%8C%EB%AC%B8>