-파이썬의 입력 : input(’~~’) #원하는 자료형으로 캐스팅

-파이썬의 출력 : print(”{},{}”.format(1,2)) print(”%s %d”%(1,2) print(f”{name},{add}”)

리스트

-[] 이용. 0~-1까지, [from : to : step ] 슬라이싱

-min(),max(),len(),append(),extend() #append는 그대로 extend는 요소 별 추가

-index(찾고자하는 값,검색위치),count(찾고자하는값)

★컴프리헨션

Result = [num for num in numbers if num%2==0]

=>numbers에 있는 요소중 짝수만 result 리스트에 추가

Result = [num\*\*2 for num in numbers]

=>numbers 리스트에있는 요소들을 제곱해서 result 리스트에 추가

-remove(data),del list[idx] pop()

-sort() sort(reverse=True)

-copy() 등을 이용

튜플

=>() 이용 새성, 읽기전용

딕셔너리

-hash함수같다 {} 사용

-dic.keys() dic.values() =>list로 return

셋

-중복허용 x

함수

A =10

def my\_fun(a,b,\*c,d=10,\*\*e) : #순서>튜플>키워드>딕셔너리

“doc string”

global A #함수내에선 지역변수를 쓰지만,global를 통해 전역변수를 사용

언패킹

\*튜플 \*\*딕셔너리

람다식

lambda a(input) : a\*\*2(기능)

lambda a(input) : a%2==0 (T/F)

Def my\_func(func,list) : #매개변수로 쓰이는 func에 주로 람다식 활용

If(func)

Result.append(func(list))

객체

Class Student (”상속할 객체) :

def \_\_init\_\_(self(필수),…) :

super().\_\_init\_\_()

**pandas**

Member = pd.read\_csv(’경로’ , header =None, names = [])

Name = member[’name’] #name 열을 시리즈로 가져옴 .to\_frame()

Member = member.loc[0] #0번 인덱스 행을 시리즈로 가져옴

Member.loc[0:5] =>0~5번 행의 모든 열을 가져옴

Member.loc[:, [’name’ : ‘email’] ] #모든행의 name ~ email까지 열을 가져옴

Member.iloc[0:4] #0~3번까지의 행을 가져옴

Member.iloc[0 , 0:3] #0행의 2열까지 가져옴

Member.loc[Member[’age’]>30] #인덱스 대신 조건

Member[member[’1’]==’1’] #member[’1’]==’1’ 프레임중 1열의 값이 1인(bool) 행 포함

Pd.concat([a,b] , axis=1) #axis=1 =>열 통합 axis = 0 행 통합

Member.groupby([’sp’]) #sp가 같은 것끼리 그룹화