|  |  |
| --- | --- |
| 교육 제목 | python 4 |
| 교육 일시 | 2021. 09. 24 |
| 교육 장소 | 영우글로벌러닝 |
| **교육 내용** | |
| 오전 | counter=0  #함수를 선언합니다.  def fibonacci(n):  #어떤 피보나치 수를 구하는지 출력합니다.  print("fibonacci({})를 구합니다.".format(n))  global counter  counter+=1  #피보나치 수를 구합니다.  if n==1:  return 1  if n==2:  return 1  else:  return fibonacci(n-1)+fibonacci(n-2)  #함수를 호출합니다.  fibonacci(10)  print("---")  print("fibonacci(10) 계산에 활용된 덧셈 횟수는 {}번 입니다.".format(counter))  -------------------------------------------------------------------------------------  #피보나치 수열 계산 프로그램  #메모 변수를 만듭니다.  dictionary = {  1: 1,  2: 2  }  count=0  #함수를 선언합니다. - 피보나치 함수  def fibonaci(n):  global count  count+=1  if n in dictionary:  #메모 되어 있으면 메모된 값 리턴  return dictionary[n]  else:  #메모 되어 있지 않으면 값을 구함  output=fibonaci(n-1)+fibonaci(n-2)  dictionary[n]=output  return output  #함수를 호출합니다.  print("fibonaci(10) : ", fibonaci(30), count)  ------------------------------------------------------------------------  #평탄화 함수 작성  def flatten(data):  #최종 결과를 저장할 리스트 작성  output=[]  #리스트 각각의 item을 가져와서 작업  for item in data:  #데이터의 타입이 list이면 flatten 함수 호출  #아니면 output에 리스트 추가  if type(item)==list:  output+=flatten(item)  else:  output.append(item)  return output  #리스트가 포함된 리스트 작성  example=[[1,2,3],[4,[5,6]],7,[8,9]]  print("example original : ", example)  print("example flatten : ", flatten(example)) # 평탄화 함수 호출  --------------------------------------------------------------------------  c=(1,2,3)  print(c)  # c[0]=5 -> tuple error  #값 교환  a,b=10, 20  print("befor: ", a,b)  a,b=b,a  print("after : ", a,b)  -----------------------------------------------------------------  def minus\_func(num1, num2):  return num1-num2  def trans\_func(num1, num2):  if num1<num2:  return (num2, num1)  else:  return (num1, num2)  input\_numbers=input("두 수 입력 : ").split()  (num1, num2)=trans\_func(int(input\_numbers[0]),int(input\_numbers[1]))  print(minus\_func(num1, num2))  ----------------------------------------------------------------------------------  #함수를 선언합니다.  #def power(item):  # return item\*item  #def under\_3(item):  # return item<3  power=lambda x: x\*x #lambda 매개변수 : 리턴 값  under=lambda x:x<3  #변수를 선언합니다.  list\_input\_a=[1,2,3,4,5]  #map() 함수를 사용합니다. map(함수, 리스트) -> 결과는 리스트  output\_a=map(power, list\_input\_a)  #output\_a=map(lambda x: x\*x, list\_input\_a) 함수를 만들지 않고 바로 사용 가능  print("map() 함수의 실행결과")  print("map(power, list\_input\_a) : ", output\_a)  print("map(power, list\_input\_a) : ", list(output\_a))  print()  #filter() 함수를 사용합니다.  #filter(함수, 리스트) ->  output\_b=filter(under\_3, list\_input\_a)  print("filter() 함수의 실행결과")  print("filter(under\_3, list\_input\_a) : ", output\_b)  print("filter(under\_3, list\_input\_a) : ", list(output\_b))  ------------------------------------------------------------------------------------  # file을 이용한 데이터 저장  # open, read or write, close  # open(파일명, 모드) => file 객체 반환  # w -> 기존의 파일이 존재하면 기존 데이터 삭제 후 새로 생성,  # 존재하지 않으면 새로 생성  # a -> 마지막에 추가  # r -> 읽기만 가능, 존재하지 않으면 에러  file = open("python.txt", "w") # file open  file.write("Hello python programing") # file write  file.close() # file close  #file.write("test") -> error 발생  file=open("python.txt","r")  contents=file.read()  print("file read data : ", contents)  file.close()  -----------  with open("python.txt", "a") as file:  file.write("Hello programing python!!!!!")  -----------------------------------------------------------------------------  #100명의 키와 몸무게를 이름과 같이 파일에 저장하기  import random  # 한글 리스트 생성  hanguls=list("가나다라마바사아자차카타파하")  with open("info.txt", "w") as file:  for i in range(100):  name=random.choice(hanguls)+random.choice(hanguls)  weight=random.randrange(40,100)  height=random.randrange(140, 200)  file.write("{},{},{}\n".format(name, height, weight))  with open("info.txt", "r") as file:  for line in file:  (name, height, weght)=line.split(',')  print("{} : {}, {}".format(name, height, weght))  # human\_data=file.read() |
| 오후 | #제너레이터 함수 만들기 : yield 키워드를 넣으면 일반 함수가 제너레이터 함수가 됨  def test():  print("함수 호출 111")  yield 1  print("함수 호출 222")  yield 2  print("함수 호출 333")  yield "abd"  output=test()  print("시작 00 : ", next(output)) # next() 함수를 사용해야 제너레이터 함수를 호출한다.  print("시작 11 : ", next(output))  print("시작 22 : ", next(output))  #next(test()) # 더이상 진행할 부분이 없으면 error  --------------------------------------------------------------------------------  numbers=list(range(1,10+1))  print("# 홀수만 추출하기")  print(list(filter(lambda x: x%2==1,numbers)))  print()  print("# 3 이상, 7 미만 추출하기")  print(list(filter(lambda x: 3<=x<7, numbers)))  print()  print("# 제곱해서 50 미만 출력")  print(list(filter(lambda x: x\*x<50, numbers)))  --------------------------------------------------------------------  #조건문으로 예외 처리하기  PI=3.14  #사용자의 입력이 숫자인지  #예외처리 try: 실행할 구문 입력 except: 예외 처리 구문 입력  try:  input\_number = int(input("정수 입력 > ")) # 예외가 발생 가능 코드  print("원의 반지름 : ",input\_number)  print("원의 둘레 : ", input\_number\*2\*PI)  print("원의 넓이 : ", input\_number\*input\_number\*PI)  except:  print("정수를 입력하지 않음")  ------------------------------------------------------------------------------  try:  input\_number = int(input("정수 입력 > ")) # 예외가 발생 가능 코드  except:  print("정수를 입력하지 않음")  else:  print("원의 반지름 : ",input\_number)  print("원의 둘레 : ", input\_number\*2\*PI)  print("원의 넓이 : ", input\_number\*input\_number\*PI)  finally:  print("무조건 실행")    # try + except  # try + except + else  # try + except + finally  # try + except + else + finally  # try + finally  ----------------------------------------------------------------------------------  def write\_func(filename, text):  try:  file=open(filename, 'w')  return # return이 중간에 존재하면  file.write(text) # 실행 안됨  except:  print("file error")  # except Exception as error:  # print("file error", error)  finally: # 중간에 return이 되어도 무조건 실행  print("try end")  file.close()    write\_func("file\_close.txt","Hello file")  print("file.closed : ", file.closed)  -------------------------------------------------------------------------------  filename=input("파일명 입력 : ")  try:  file=open(filename, "r")  priint(file.read())  except Exception as err:  print("file err", err)  file=open(filename, "w")  file.write("New file")  finally:  file.close()  print("file.closed : ", file.closed)  ------------------------------------------------------------------------------  # try:  # except ValueError as e:  # except IndexError as e:  # except Exception as e:  #변수를 선언합니다.  list\_number=[52,273,32,72,100]  # try except 구문으로 예외를 처리합니다.  try:  #숫자를 입력받습니다.  number\_input=int(input("정수 입력 "))  #리스트의 요소를 출력합니다.  print("{}번째 요소: {}".format(number\_input, list\_number[number\_input]))  # print("value error ", list\_number.index(number\_input)) # index error  except ValueError as exception:  # ValueError가 발생하는 경우  print("정수를 입력해 주세요!")  print(type(exception), exception)  except IndexError as exception:  # IndexError가 발생하는 경우  print("리스트의 인덱스를 벗어났어요!")  print(type(exception), exception)  except Exception as exception:  # 이외의 예외가 발생한 경우  print("미리 파악하지 못한 예외가 발생했습니다.")  print(type(exception), exception)  -----------------------------------------------------------------------------------  number=10  if number>0:  raise NotImplementedError # raise -> 에러를 발생시켜 강제로 종료시키고 싶을때  else:  pass # pass하면 아무것도 안나온다.  --------------------------------------------------------------------------------  # 자료 입력(이름, 성적), 이름에 'end'입력되면 입력 종료  # 파일 open 함수 호출  filename=input("파일명 입력") # 파일명 입력  file=file\_open(file.name,"a") # 파일 open 함수 호출  file\_write(file) # 파일에 저장하는 함수 호출  file.close() # 파일 close  # 파일에서 자료를 가져와서 변수에 저장  names, scores= [], []  with file\_open(file\_name, "r") as file:  for item in file:  values=item.split(',')  names.append(values[0])  scores.append(int(values[1]))  # 검색할 이름 입력  name=input("find name > ")  # 자료 검색 : 존재하면 출력, 없으면 'Not found' 출력  if name in names:  print("{} : {}".format(name, scores[names.index(name)]))  else:  print("Not Found")  ------------------------------------------------------------------------------  def file\_open(file\_name, file\_mode):  file=open(file\_name, file\_mode)  return file  def file\_write(file): # 자료 입력(이름, 성적), 이름에 'end'입력되면 입력 종료  while True:  name=input("이름 입력 : 'end' 입력시 종료 ")  if name=='end':  break    while True:  try:  score=int(input("성적 입력 "))  break  except:  print("숫자 입력")  context=name+','+str(score)+'\n'  file.write(context) # file에 자료 저장 |