|  |  |
| --- | --- |
| 교육 제목 | **Python 다양한 함수(if) 및 list 관련 활용법** |
| 교육 일시 | 20211101 09:00 ~ 18:00 |
| 교육 장소 | 강의실 오프라인 수업 |
| **교육 내용** | |
| 오전 | # 기존 수업 review  # 리스트와 반복문 진행 완료  # 리스트는 여러 데이터를 순차적으로 나열하여 저장할 수 있는 데이터 타입임.  # append, insert, in, not in, for  # 문자, 숫자, bullean -> 이를 묶어둔 것이 list  # 리스트를 우선 선언하고, 리스트 활용  # insert (position, elements)  # extend 와 append 의 차이점  # : append는 추가 시, 하나의 요소로 추가  # : extend는 각각의 요소를 리스트로써 추가  # del, pop, remove, clear 로 리스트 및 요소 삭제  # del 리스트명, 리스트명.pop(index), 리스트.clear(index), 리스트.remove(index)  # 리스트 요소 추가 및 제거, 변경  a\_list = [1, 2, 3]  b\_list = [4, 5, 6]  # list 요소 추가  a\_list.append("abcd")  print ("list append : ", a\_list) # [1, 2, 3, "abcd"]  a\_list.insert(1,10)  print("list list : ", a\_list) # [1, 10, 2, 3, "abcd"]  # list 에 list 추가  a\_list.extend(b\_list)  print("list extend : ", a\_list) # [1, 10, 2, 3, "abcd", 4, 5, 6]  # list 요소 변경  a\_list [3] = "변경"  print("list 변경 : ", a\_list) # [1, 10, 2, "변경", "abcd", 4, 5, 6]  # list 요소 삭제  del b\_list [1]  print("b\_list del : ", b\_list) # [4, 6]  a\_list.pop(2)  print("a\_list pop : ", a\_list) # [1, 10, "변경", "abcd", 4, 5, 6]  a\_list.remove("abcd")  print("a\_list remove : ", a\_list) # [1, 10, "변경", 4, 5, 6]  a\_list.clear()  print("a\_list clear : ", a\_list) # []  # 내부 요소 확인  print(6 in b\_list) # True  print(6 not in b\_list) #False  # for 반복문 : for 변수 in 반복 자료 :  for i in range(5) :  print("i : ", i)  a\_list = [1, 2, 3, 4, 5]  for element in a\_list :  print("a\_list : {}".format(element))    for char in "hello world!!" :  print("-", char)  list\_of\_list = [[1, 2, 3], [4, 5, 6, 7], [8, 9]]  for sequence in list\_of\_list :  for sequence\_2 in sequence :  print ("모든 요소를 순차적으로 출력 합니다 : ", sequence\_2)  # 딕셔너리는 키값을 통해 리스트 요소를 확인할 수 있는 명령어  # 딕셔너리 [key][index] : 딕셔너리가 리스트일 경우 키를 입력하고, 리스트(요소인)의 index를 입력해주면 출력이 가능함.  # 딕셔너리 선언 방법  # dictionary = {key\_1 : value\_1, ... , key\_n : value\_n} -> 중괄호{} 사용 (선언이 아닌 접근 시에는 대괄호[]로 key를 통해 접근)  # key 는 '', "" 로 정의해주어야 한다.  # 딕셔너리 선언  dictionary = {}  print("요소 추가 이전 : ", dictionary)  dictionary["name"] = "새로운 이름"  dictionary["head"] = "새로운 정신"  dictionary["body"] = "새로운 몸"  print("요소 추가 이후 : ", dictionary)  # del dictionary : dictionary 삭제  # 입력을 받아서 dictionary 확인 및 출력 과정  dictionary\_a = {  "name" : "7D 건조 망고",  "type" : "당절임",  "ingredient" : ["망고", "설탕", "메타중아황산나트륨", "치자황색소"],  "origin" : "필리핀"  }  print(dictionary\_a)  key = input("접근 key를 입력하세요 : > ")  if key in dictionary\_a :  print (dictionary\_a[key])    else :  print ("존재하지 않는 key 입니다.")    # for 문을 이용해 전체적인 dictionary 값만 출력하기  dictionary\_a = {  "name" : "7D 건조 망고",  "type" : "당절임",  "ingredient" : ["망고", "설탕", "메타중아황산나트륨", "치자황색소"],  "origin" : "필리핀",  "items" : {"key1" : "dic test", "value" : "dict value"}  }    # dictionary 내부에 리스트 또는 dictionary 가 있는 경우, 각 값들만 출력 해보기  for value\_1 in dictionary\_a :  if type(dictionary\_a[value\_1]) == dict : # dict : dictionary 의 type 을 의미함.  for value\_sub\_1 in dictionary\_a[value\_1]:  print("값들이 디셔너리로 구성되어 있습니다. 값은 : ", dictionary\_a[value\_1][value\_sub\_1])    elif type(dictionary\_a[value\_1]) == list : # list : list 의 type 을 의미함.  for value\_sub\_2 in dictionary\_a[value\_1]:  print("값들이 리스트로 구성되어 있습니다. 값은 : ", value\_sub\_2)  else :  print(dictionary\_a[value\_1])    # range 정의 : range(시작값, 끝값, 증가분) -> 내부 수치들은 항상 int로 기입하거나, range 앞에 int를 붙여준다.  array = [273, 45, 103, 54, 101]  for i in range(len(array)):  print("{}번째 데이터 : {}".format(i, array[i]))  # 거꾸로 출력해보기  for i\_2 in range(len(array)-1, 0-1, -1):  print("{}번째 데이터 : {}".format(i\_2, array[i\_2]))    # reversed 함수 이용해 거꾸로 출력해보기  print("reversed function")  for i\_3 in reversed(range(len(array))):  print("{}번째 데이터 : {}".format(i\_3, array[i\_3]))  # while 반복문 활용법  # while 표현식 : 표현식이 True이면 실행.  while True :  print(".") # infinite loop!! need to interupt.  array = [273, 45, 103, 54, 101]  i = 0  while i < len(array) :  print("{}st data : {}".format(i, array[i]))  i += 1  # 연습문제 : 빈 리스트를 입력 받아 quit이 loop에 입력되는 시점에 각각의 리스트요소 출력  input\_data = []  while True:  input\_str = input("문자를 입력하세요. (q를 입력하면 종료됩니다.)")    if input\_str.isdecimal():  print("입력된 값이 숫자인 경우 리스트화되지 않습니다.")  continue # un do if is true    elif input\_str == "q" :  print("q가 입력되어 종료됩니다.", "현재 누적 리스트는 다음과 같습니다. : >", input\_data)  break    else :  input\_data.append(input\_str)    # 이름을 입력 받아 리스트에 저장 (이름에 "q"가 입력되면 입력 종료)  # 검색 하고자 하는 이름을 입력 받아 몇 번째 입력된 이름인지 출력.  # enumerate 함수 이용  input\_name = []  while True :  input\_str = input("리스트 요소 입력(종료는 q) >")    if input\_str == "q":  print("q를 입력하여 프로그램 종료")  break  input\_name.append(input\_str)  search\_name = input("검색하고자 하는 이름 입력. >")    for idx, name in enumerate(input\_name) :  if name == search\_name :  print("{}번째 데이터는 {}입니다.".format(idx+1, name))  break |
| 오후 | # 입력된 이름의 갯수만큼 점수 입력 score[] 에 저장  # 저장된 이름과 점수를 출력 : 홍길동 90, 김철수 100  # 딕셔너리로 변환해서 출력  input\_name = []  while True :  input\_str = input("이름 입력(종료는 q) >")    if input\_str == "q":  print("q를 입력하여 프로그램 종료")  break  input\_name.append(input\_str)  score = []  i = 0  while i < len(input\_name):  score.append(int(input("점수 입력 : ")))  i += 1    print("input\_name[] : score[]")  for idx in range(len(input\_name)):  print("{}, {}".format(input\_name(idx), score(idx)))  i += 1  while i < len(input\_name):  dict\_score[input\_name[i]] = score[i]  i += 1    print("dict\_score : ")  for key, value in dict\_score\_items(): # 딕셔너리 출력  pritn("{}, {}".format(key, value))  # 다시 한 번 체킹  # list 내포 [표현식 for 반복자 in 반복자료]  array = []  for i in range(0, 20, 2):  array.append(i \* i)  print(array)    array\_1 = [i\*i for i in range(0, 20, 2)]  print(array\_1)  array\_2 = [i for i in range(100) if i%5 == 0]  print(array\_2)  # 키보드에서 자료를 입력 받아 한글자씩 리스트에 저장한 후  # 문자만 리스트로 출력  input\_str = list(input("input data> :")) # list 화를 위한 명령어 : list ()  str\_list = [char for char in input\_str if type(char) == str] # type으로 함수를 지정하게 되면, 숫자 역시 문자로 인식되어 모든 문자가 출력이 됨.  print(str\_list)  str\_list2 = [char for char in input\_str if char.isalpha()] # 문자임을 검증하는 부분 char.isalpha() 해당 부분이 True면 대입.  print(str\_list2)  # 문자열.join함수 - 문자열로 구성된 리스트, 문자열을 하나의 리스트로 병합하는 함수  join\_char = "::".join(["1", "2", "3"])  print(join\_char)  # 이터레이터  numbers = [1, 2, 3, 4, 5, 6]  r\_number = reversed(numbers) # 위치를 반대로 변경하여 위치만 가지고 있는 붕뜬 상태.  i = 0  while i < len(numbers) :  print(next(r\_number))  i += 1  # def 를 통해 함수를 선언하고 만들어 냄.  # \*후 변수를 기재하면 가변 변수(입력을 할 수 있는 변수)로 지정이 가능함.  # sep() : seperate 의 의미  # def 함수명 () : 매개변수 return 있을 수도, 없을 수도  def print\_func():  print("func test")    def print\_n\_times(value, n):  for i in range(n):  print(value)  def print\_times(value, n=3): # 지정 매개 변수 타입  for i in range(n):  print(value)  def print\_var\_times(n, \*values): # 가변 매개 변수 타입, \*을 붙여서 표기하며, 수량에 상관 없이 변수를 넣을 수 있음.  for i in range(n):  for value in values:  print(value)    print\_func()  print\_n\_times("test", 3)  print\_times("test3")  print\_var\_times(6, "redong", "random", "wow", 123, 3423)  # 키보드에서 입력 받는 함수 : 숫자와 부호를 입력 받아 list로 변환하는 함수  # 부호가 '+'면 합을 구하는 함수로  # 부호가 '-'면 합을 구하는 함수로  def input\_data():  input\_list = input("10 + 20의 형식으로 입력").split()  return input\_list    def plus\_func(num\_1, num\_2):  return num\_1 + num\_2  def minus\_func(num\_1, num\_2):  return num\_1 - num\_2  while True :  input\_list = input\_data()    if "q" in input\_list :  print("q를 입력 받아 프로그램을 종료합니다.")  break    if input\_list[1] == "+":  result = plus\_func(int(input\_list[0]), int(input\_list[2]))  print("입력된 값은 다음과 같습니다.", input\_list[0], input\_list[1], input\_list[2])    elif input\_list[1] == "-":  result = minus\_func(int(input\_list[0]), int(input\_list[2]))  print("입력된 값은 다음과 같습니다.", input\_list[0], input\_list[1], input\_list[2])  print("{} {} {} = {}".format(input\_list[0], input\_list[1], input\_list[2], result)) |