|  |  |
| --- | --- |
| 교육 제목 | Python |
| 교육 일시 | 21.09.17 |
| 교육 장소 | 영우글로벌러닝 / idiscord비대면 교육 |
| **교육 내용** | |
| 오전 | 복습  input\_str = ("문자를 입력하세요 6글자 이상 > ")  print(input\_str[ 1 : 6 ]) #index는 0부터 시작  print(input\_str[ -3 : ])  print(input\_str[-1])    format 함수에 대해서  format은 문자열로 출력하는 것.  숫자 -> 문자    # format() 함수 : 문자열 편집  print("입력된 숫자는 {}".format(10)) # 함수3(함수2(함수1()))  a\_str = "입력된 숫자는 {}".format(30)  print(a\_str, "format data type : ", type(a\_str))  "입력된 숫자는 {}와 {}입니다".format(10,20)    a\_str  len(a\_str) # 문자열의 길이를 구하는 함수  input\_str = "입력된 숫자는 {}와 {}입니다"  input\_str.format(30,40)    input\_data = input("두 수를 입력하세요 > ")  input\_num1 = input\_data[0]  input\_num2 = input\_data[2]  print(input\_str.format(input\_num1, input\_num2))    format 함수의 다양한 기능  # 정수  output\_a = "{:d}".format(52)    # 정수  output\_a = "{:d}".format(52)    # 특정 칸에 출력하기  output\_b = "{:5d}".format(52)  output\_c = "{:10d}".format(52)    # 빈칸을 0으로 채우기  output\_d = "{:05d}".format(52)  output\_d = "{:05d}".format(-52)    print("# 기본")  print(output\_a)  print("특정 칸에 출력하기")  print(output\_b);  print(output\_c)  print("# 빈 칸을 0으로 채우기")  print(output\_d);  print(output\_e);    부동 소수점 출력의 다양한 형태  -float  :g -> 소수점 0.1은 보이지만 0.0에서 .0은 안보이게 하는것    float자료형 기본  output\_a = "{:f}".format(52.273)  output\_b = "{:15f}".format(52.273)  output\_c = "{:+15f}".format(52.273)  output\_d = "{:+015f}".format(52.273)    print(output\_a)  print(output\_b)  print(output\_c)  print(output\_d)    소수점 아래 자릿수 지정하기  output\_a="{:15.3f}".format(52.273)  output\_a="{:15.2f}".format(52.273)  output\_a="{:15.1f}".format(52.273)    print(output\_a)  print(output\_b)  print(output\_c)    의미 없는 소수점 제거하기  output\_a = 52.0  output\_b = "{:g}".format(output\_a)  print(output\_a)  print(output\_b)    upper() 와 lower() 와 strip()  # 문자열 관련 함수 : upper() , lower () , strip() , lstrip() , rstrip()  a\_str = " Hello python programming "  print(a\_str.upper()) # 대문자로  print(a\_str.lower()) # 소문자로  print("-{}-".format(a-str))  print("-{}-".format(a\_str.strip()))  print("-{}-".format(a\_str.lstrip()))  print("-{}-".format(a\_str.rstrip()))    print(a\_str.lsalnum()) # alpha or number 인지 확인    in 연산자  print("안녕" in "안녕하세요")    split() 함수  # 1. 문자 입력  input\_str = input("두 수를 입력 > ")  # 2. 문자를 분리 (두 개의 숫자(문자)로), # 3. 문자를 숫자로 변환  split\_str = input\_str.split()  input\_num1 = float(split\_str[0])  input\_num2 = float(split\_str[1])    # 4. 계산해서 결과 출력  print(input\_num1, " / " , input\_num2, " = ", "{:7.3f}".format(input\_num1/input\_num2))  "{} / {} = {:7.3f}".format(input\_num1 , input\_num2,input\_num1/input\_num2)    불 자료형과 if조건문  불리언 비교 연산자 논리 연산자  if조건문  프로그래밍 언어에는 기본적인 자료형으로 참과 거짓을 나타내는 값이 있으며,  이를 불(boolean)이라고 한다.  True와 False 값만 가질 수 있다.  = 하나는 대입  == 같다 (비교 연산자)  x = 10  under\_20 = x < 20  print("under\_20:", under\_20)  print("not under\_20:",not under\_20)    # 문) 키보드에서 두 수를 입력받아 두 수가 같으면 True를 다르면 False를 출력하세요  input\_str = input(" 두 수를 입력 > ")  split\_num = input\_str.split()  print(int(split\_num[0]) < and int(split\_num[1] < 10 ) # 둘 다 true면 true 아니면 false  print(int(split\_num[0]) < or int(split\_num[1] < 10 ) # 둘 중의 하나만 true면 true    if조건문  : 클론을 반드시 써줘야 인식을 한다.  들여쓰기가 들어간다  if True:  print("정말 true 입니다")  들여쓰기는 탭으로 많이 사용  # if 조건문:  # 들여쓰기로 문장 처리  if True :  print("조건문 처리 ")  print("조건문 처리 111")  if False:  print("조건문 처리 222")  print("조건문 처리 333")  print("조건문 나옴")    # 문) 1. 두 수를 입력받아 두 수가 0보다 크면 투 수의 합을 출력  num1 = int(input(" 숫자 입력 > "));  num2 = int(input(" 숫자 입력 > "))  if num1 >0 and num2 > 0 :  print(num1, " + ", num2, " = " , num1+num2)  if num1 > num2:  print(num1, " - ", num2, " = " , num1-num2)  if num1 < num2:  print(num2, " - ", num1, " = " , num2-num1)  if num1 !=0 and num2 !=0 :  if num1 > num2:  print(num1, " % ", num2, " = ", num1%num2)  if num1 < num2:  print(num2, " % ", num1, " = ", num2%num1) |
| 오후 | else / elif / if / false / pass    # 입력을 받습니다.  number = input ("정수 입력> ")  number = int(number)    #조건을 사용합니다  if number % 2 == 0 :  # 조건이 참일 때, 즉 짝수 조건  print("짝수입니다")  else:  # 조건이 거짓일 때, 즉 홀수 조건  print("홀수입니다")      elif 조건이 3개 이상일때 if 조건문과 else 구문 사이에 입력    if 조건문 ->  elif 조건문 ->  else 구문    # if, if ~ else, if ~ elif ~ else  # 성적을 입력 받아 등급을 부여해서 출력  # 60미만은 F, 61~70 D, 71~80 C, 81~90 B, 91~100 A    score = int(input("성적 입력 : "))    if score >90 :  print('A')  elif 80 < score <= 90 :  print('B')  elif 70 < score <= 80 :  print('c')  elif 60 < score <= 70 :  print('D')  #나중에 처리  pass  else :  print('F')      #문) 숫자 연산자기호 숫자를 입력받아 연산자 기호가 '+' 이면 두 숫자의 합을  # '-' 이면 두 수의 차를, '\*'이면 두 수의 곱을, 아니면 "기호 오류" 문자 출력    input\_str = input("계산식 입력 :> ")  num1 = int(input\_str.split()[0])  buho = input\_str.split()[1]  num2 = int(input\_str.split()[2])  comp\_var = "{} {} {} = ".format(num1,buho,num2)  if buho == '+':  print(comp\_var, num1+num2)  elif buho == '-':  print(comp\_var, num1-num2)  elif buho == '\*':  print(comp\_var, num1\*num2)  else :  print("기호 오류")      리스트  리스트 연산자 : 연결(+), 반복(\*), len()  리스트에 요소 추가 : append, insert  리스트에 요소 제거 : del  리스트 내부에 있는지 확인 in / not in  for반복문    리스트는 인덱스가 중요하다.    array = [273, 32, 103, "문자열", True, False]  print(array)  [273, 32, 103, "문자열", True, False]    ## list : [ , , ....] -> 여러개의 데이터 집합  # list 추가 : list명.append(), list명.insert(index, 추가데이터), list명.extend(추가할리스트)  # list 삭제 : list명.pop(index), list명.pop(2:index), del list명[index]  # list 값으로 삭제 : list명.remove(값)  # list의 모든 값 삭제 : list명.clear()  a\_list = [ "abc", "test", 123 , True, 78, "python", [1,2,3]]  a\_list[0] # "abc"  a\_list[0][0] # 'a'  a\_list[-1][2] # 't' 3    b\_list = [1,2,3]; c\_list = [4,5,6]  print(b\_list + c\_list) # list 연산 + 두 개의 리스트 결합, \* 반복  print(b\_list \*3)    a\_list.append("add") # list의 마지막에 요소 추가  print(a\_list)  a\_list.insert(2, "insert data") # 원하는 위치에 데이터 추가  print(a\_list)  len(a\_list)  print(b\_list)  b\_list.extend(c\_list) # b\_list에 c\_list의 값을 추가 기존의 데이터 변경  print(b\_list)  print(b\_list.pop())  b\_list  del a\_list[-1]  a\_list    # 문) 여러 개의 데이터를 입력 받아 리스트로 저장한 후  # 1. 마지막 데이터를 출력 후 삭제  # 2. 3번째 인덱스의 값을 출력  # 3. 추가로 여러 개의 데이터를 입력 받아 기존의 리스트에 추가 후 출력  # 4. 처음 입력한 데이터의 리스트 변수를 clear  f\_list = input("input data :").split() # 자료 입력  print(f\_list, len(f\_list)) # 입력한 자료 리스트 출력, 갯수 출력  print(f\_list.pop()) # 1. 마지막 데이터를 출력 후 삭제  print(f\_list[3]) # 2,3번째 인덱스의 값을 출력    f.list.extend(input("input data : ").split()) # 두번째 자료 입력 후 기존의 list에 추가  print(f\_list)  print(f\_list.clear())      리스트에 요소 추가하기  append() 와 insert()    리스트에 요소 제거하기  del() 와 remove(값) list.clear()    리스트 내부에 있는지 확인하기  in / not in    for 반복문  print("출력")  print("출력")  print("출력") -> for i in range(100):  print("출력") print("출력") #"출력"을 100회 반복 출력  print("출력")    for + 변수 + in + 범위 (반복)    a\_list = [ "abc", "test", 123 , True, 78, "python", [1,2,3]]  # 반복문 for 변수 in 반복자료:  # 처리문  for i\_var in a\_list:  print(i\_var)    for i in range(5): # 5회 반복  print("반복", i)    for char in "Hello":  print(char, ' - ')    #키보드로 입력을 받아 리스트에 저장한 후 입력한 자료가 숫자이면 합을 구해서 출력하세요  # 1. input 자료를 리스트로 추가  # 2. 리스트의 자료를 하나씩 비교 ( 숫자인지 ) 반복문 실행  # 3. 숫자이면 합을 구함  # 4. 반복문 끝난 후 합을 출력  a\_list = input("input data : ").split()  total = 0  for var in a\_list:  if var.isnumeric(): # 입력된 자료가 숫자인지 확인  total += int(var) # 숫자이면 합계 구함  print("total : ", total)      # 숫자를 입력받아 해당하는 숫자의 구구단을 출력 하세요  # 3  # 3 \* 1 = 3  # 3 \* 2 = 6    input\_num = int(input("input number : "))  for i in range(9):  print("{} \* {} = {}".format(input\_num, i+1, input\_num \* (i+1)))    # 1. 숫자와 문자열의 다양한 기능 : format() " { } 문자 { } ".format(1,2)  # 2. 문자열 구성 파악하기 is함수명() -> True 또는 False  # 3. 문자열 자르기 : split() -> 문자열.split()  # 4. 조건문 : if 조건식: , if ~ else, if~ elif 조건식 : ~ elif . . . ~ else : 조건식 없음  # 5. 날짜, 시간 함수 : datetime, import 의미, import datetime as dt -> alias명을 사용  # 6. list -> 관련함수 append(), extend(), pop(), insert(), remove(), clear(), del  # 7. 반복문 -> for 변수 in 리스트: |