|  |  |
| --- | --- |
| 교육제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정 |
| 교육일시 |  |
| 교육장소 | YGL 학과장 및 자택(디스코드 사용 온라인 학습) |
| **교육내용** | |
| 오전 | 1. **format() 함수로 숫자를 문자열로 변환하기**   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 {} {} 입니다이".format(10,20) -> '입력된 숫자는 30와 50입니다'  이처럼 .”{} {}”.format()을 하면 format에서 출력된 값이 {} 위치로 들어간다. {} 개수가 format에서 나오는 값보다 적으면 오류 발생함.   1. **format() 함수의 정수 출력**   -정수를 특정 칸에 출력하기  output\_a = "{:d}".format(52)  -특정 칸에 출력하기  output\_b = "{:5d}".format(52) <- 5칸  output\_c = "{:10d}".format(52) <- 10칸  - 빈칸을 0으로 채우기  output\_d = "{:05d}".format(52) <- 양수  output\_e = "{:05d}".format(-52) <- 음수  print("# 기본")  print(output\_a)  print("# 특정 칸에 출력하기")  print(output\_b)  print(output\_c)  print("# 빈칸을 0으로 채우기")  print(output\_d)  print(output\_e)   1. **문자열**   -문자열 관련 함수 : upper() : 대문자로 변환, lower() : 소문자로 변환, strip() : 양쪽 공백 제거, lstrip() : 왼쪽 공백 제거, rstrip() : 오른쪽 공백 제거  find() : 왼쪽부터 찾아서 처음 등장하는 위치 찾음  rfind() : 오른쪽부터 찾아서 처음 등장하는 위치 찾음  in 연산자 : 문자열 내부에 어떤 문자열이 있는지 확인하기 위한 연산자. True, False로 나옴  ex) print("안녕" in "안녕하세요") >> True  print("잘자" in "안녕하세요") >> False   1. **조건문**   - = : 대입, == : 같다, != 다르다  예) -not 연산자 조합하기  x = 10  under\_20 = x < 20  print("under\_20 :", under\_20)  print("not under\_20 :", not under\_20)  - if ~조건문 : 조건에 따라 코드 실행하거나 실행하지 않게 할 때 사용하는 구문임. 줄바꿈 후 들여쓰기(tab)꼭 해야함  문)  1. 두 수를 입력받아 두 수가 0보다 크면 두 수의 합을 출력  2. 두 수 중 큰 수에서 작은 수를 뺀 결과 출력  3. 두 수 중 0이 없으면 작은 수를 큰 수로 나눈 나머지 값 출력  num1 = int(input(" 숫자 입력 >"));  num2 = int(input(" 숫자 입력 >"))  if num1 >0 and num2 >0 :  print("문1",num1, "+", num2, "=", num1+num2)  if num1 > num2:  print("문2", num1, "-", num, "=", num1-num2)  if num1 < num2:  print("문21", num2, "-", num1, "=", num2-num1)  if num1 !=0 and num2 !=0 :  if num1 > num2:  print("문3",num1, " % ", num2, "=", num1%num2)  if num1 < num2:  print("문31",num2, " % ", num1, "=", num2%num1)   * 숫자 입력 >3   숫자 입력 >4  문1 3 + 4 = 7  문21 4 - 3 = 1  문31 4 % 3 = 1  위 문제를 if ~ else로 변환  if num1 > num2:  print("문2", num1, "-", num, "=", num1-num2)  else:  print("문21", num2, "-", num1, "=", num2-num1)  if num1 !=0 and num2 !=0 :  if num1 > num2:  print("문3",num1, " % ", num2, "=", num1%num2)  else:  print("문31",num2, " % ", num1, "=", num2%num1)   * 문21 4 - 3 = 1   문31 4 % 3 = 1   1. **날짜/시간 출력하기**   -날짜/시간 관련 기능 가져오기 : import datetime  -현재 날짜/시간 구하기 now = datetime.datetime.now()  -출력 now = dt.datetime.now()  print(now.year,"년")  print(now.month, "월")  print(now.day, "일")  print(now.hour, "시")  print(now.minute, "분")  print(now.sencond, "초")  #datetime 패키지 사용 import 패키지명, import 패키지명 as 약어(alias명) #패키지명.함수명 #import datetime as dt |
| 오후 | 1. **if~else와 elif 구문**   \* else 구문 : if 조건문 뒤에 사용, if 조건문의 조건이 거짓일 때 실행되는 부분이다.  0은 false로 출력된다  if 표현식:  처리1  else:  처리2  예) # if, if ~ else, if ~ elif ~...else  # 성적을 입력 받아 등급을 부여해서 출력  # 60미만은 F, 61~70 D, 71~80 C, 89~90 B, 91~100 A  score = float(input("성적 입력 >"))  if 90 < score :  print("A")  elif 80 < score:  print("B")  elif 70 < score:  print("C")  elif 60 < score:  print("D") # print("D")를 나중에 채우려면 비워두는 대신 pass를 입력한다.  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("함 입력해보셈 :>")  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")  split() => 공백으로 구분하기때문에 input 값에 띄어쓰기 꼭 넣어야함  같은 원리로 split(",") 일 경우엔 ,를 넣어야함  ex) 5 + 9, 10 \* 5 ...   1. **raise NotImplementError**   pass 키워드를 입력해 놨어도 잊어버리는 경우가 많음 raise 키워드와 미구현 상태를 표현하는 NotImplementError를 조합해 raise NotImplementError를 사용하면 아직 구현하지 않은 부분이라는 오류를 강제로 발생시킬 수 있다. ListList : [ ]에 자료를 쉼표로 구분해서 입력. 내부에 넣는 자료는 '요소' 영어로 'element'라고 부름.Index : 리스트 기호인 [ ] 안에 들어간 숫자list 추가 : list명.append(), list명.insert(index, 추가 데이터), list명.extend(추가할 리스트) list 삭제 : list명.pop() > 마지막 요소 삭제, list명.pop(2:index) del list명[index]  list 값으로 삭제 : list명.remove(값)  list의 모든 값 삭제 : list명.clear() 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  z\_list = input("데이터 입력 > ").split() #자료 입력  print("z\_list 111 : ", z\_list, len(z\_list)) # 입력한 자료 리스트 출력, 갯수 출력  print(z\_list.pop()) # 마지막 데이터 출력, 삭제  print("z\_list 222 : ", z\_list[3]) # 3번째 인덱스 값 출력  z\_list.extend(input("데이터 입력2 >").split()) # 두번째 자료 입력 후 기존 리스트 추가  print("z\_list 333 : ",z\_list)  z\_list.clear()  print("z\_list 444 : ", z\_list)   1. For 반복문 : 컴퓨터에 반복 작업을 지시   for ‘변수’ ‘실행범위’ 문1) 키보드로 입력을 받아 리스트에 저장한 후 입력한 자료가 숫자이면 합을 구해서 출력하시오  a\_list = input("입력 > ").split() <- input 자료를 리스트로 추가  total = 0  for var in a\_list:  if var.isnumeric(): <- 입력된 자료가 숫자인지 확인, isnumeric() : 숫자일경우 true 나옴.  total += int(var) <- 숫자이면 합계 구함  print("total : ", total)  문2)숫자를 입력받아 해당하는 숫자의 구구단을 출력 하세요  input\_num = int(input("숫자 입력 : "))  for i in range(9): <- '9번 반복할 것'  print("{} \* {} = {}".format(input\_num, i+1, input\_num \* (i+1))) <- i는 0부터 시작하기때문에 1을 더해서 9번 반복해주면 1~9가 됨 9월 17일 총정리  1. 숫자와 문자열의 다양한 기능 : .format(), "{} 문자 {}".format(1,2) 2. 문자열 구성 파악하기 is함수명() -> True 또는 False 3. 문자열 자르기 : split() 4. 조건문 : if 조건식:, if ~else, if ~ elif 조건식: ... ~ 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: |