**Python기초 (DAY 1)**

김기태 강사(010-4536-0187) [리눅스, 윈도우서버, Python기초]

윈도우키+r키--> \\192.168.51.100 접속--> 주말오후\_공유

편하게 사용하기 위해서 Visual Studio Code 프로그램 사용

Visual Studio Code 프로그램에서 좌측 맨아래 아이콘클릭

Code Runner, Python, Python Extended, Python for VSCode 설치하고 리로드

Visual Studio Code 프로그램에서 Python 소스파일 만들 때 .py 확장자 반드시 입력

Visual Studio Code 프로그램에서 코딩한 거 확인할 때 Ctrl + Alt + N 키 입력해서 확인

※ print = 출력 명령어

■ print("내용") = 내용출력

☞ ex) print("반가워요~~ 파이썬 ^^") = “반가워요~~ 파이썬 ^^” 출력

■ print(‘내용’) = 내용출력

☞ ex) print(‘Python 첫날 입니다’) = ‘반가워요~~ 파이썬 ^^’ 출력

■ print(숫자 연산기호 숫자) = 연산기호 계산해서 출력

■ print("숫자 연산기호 숫자") = “숫자 연산기호 숫자” 계산하지 않고 “”안의숫자를 문자로 처리해서 문자 그대로 출력

☞ ex) print("덧셈 결과 : ",123 + 123); --> 덧셈 결과 : 246 (“”끝나고 ,넣는 것 주의 꼭!!! 넣어야함)

※ 기본적인 것

■ 문장맨처음# = 현재 줄 주석(삭제)

■ ‘’‘ = 범위주석(범위삭제)

☞ ex)

'''

print("반가워요~~ 파이썬 ^^"); print("hello")

print('Python 첫날 입니다'); print("Python을 시작해 볼까요")

'''

■ 첫 명령어 끝나고; = 명령어를 연속으로 입력할 때

■ 명령어 마지막에; = 명령어 끝낸다는 기호인데 다른 프로그래밍 언어에서는 꼭 ; 넣어줘야 하지만 Python에서는 넣지 않아도 문제없지만 ;넣는 것 습관 만들기!!!

※ ESCAPE 문자

■ \n = New line(새로운 줄로 이동)

■ \t = Tab(탭 크기만큼 이동) (한Tab에 8글자=8칸 단위) (8글자 다채우고 \t 넣으면 다음Tab 공백 그다음Tab에 입력)

■ \“내용\” = “내용” 출력 함

■ \‘내용\’ = ‘내용’ 출력 함

■ \+공백은 사용가능하지만 \+문자는 사용할수 없기 때문에(에러) \\+문자로 사용

※ 진법

10진수 : 0 ~ 9

2진수 : 0 , 1 Python 표현식 : 0b

8진수 : 0 ~ 7 Python 표현식 : 0o

16진수 : 0 ~ 9 , A ~ F Python 표현식 : 0x

■ print(Python 표현식숫자) = 진수를 10진수로 자동계산해서 출력

☞ ex) 125(12) --> 1111101(2) --> 175(8) --> 7D(16)

☞ ex) print(125); --> 125

☞ ex) print(0b1111101); --> 125

☞ ex) print(0o175); --> 125

☞ ex) print(0x7D); --> 125

■ print(bin(숫자)) = 숫자(10진수)를 2진수로 계산해서 출력

■ print(oct(숫자)) = 숫자(10진수)를 8진수로 계산해서 출력

■ print(hex(숫자)) = 숫자(10진수)를 16진수로 계산해서 출력

※서식 문자

■ %d : 10진 정수

☞ ex) print("%d"%123);

☞ ex) #print("%d %d"%123); #에러 (인자개수가 부족함)

☞ ex) #print("%d"%(123 , 321)); #에러 (서식문자 개수가 부족함)

☞ ex) print("%d %d"%(123 , 321));

☞ ex) print("%d + %d = %d"%(123 , 321 , 123 + 321));

☞ ex) print("10진 정수 : %d"%123); --> 10진 정수 : 123

☞ ex) print("10진 정수 : %d"%0o173); --> 10진 정수 : 123

☞ ex) print("10진 정수 : %d"%0x7b); --> 10진 정수 : 123

☞ ex) print("정수 표현 : %d"%123); --> 정수 표현 : 123

☞ ex) print("정수 표현 : %d"%123.623); --> 정수 표현 : 123 [소수점 반올림x 버림o]

☞ ex) print("정수 표현 : %d %d"%(123,456)); --> 정수 표현 : 123 456

■ %o : 8진 정수

☞ ex) print("8진 정수 : %o"%123); --> 8진 정수 : 173

☞ ex) print("8진 정수 : %o"%0o173); --> 8진 정수 : 173

☞ ex) print("8진 정수 : %o"%0x7b); --> 8진 정수 : 173

■ %x : 16진 정수

☞ ex) print("16진 정수 : %x"%123); --> 16진 정수 : 7b

☞ ex) print("16진 정수 : %x"%0o173); --> 16진 정수 : 7b

☞ ex) print("16진 정수 : %x"%0x7b); --> 16진 정수 : 7b

■ %f : 실수(소수점이 붙은 수)

☞ ex) print("실수 표현 : %f"%456); --> 실수 표현 : 456.000000 [소수점 6자리까지 표현]

☞ ex) print("실수 표현 : %.2f"%456.456); --> 실수 표현 : 456.45 [%.2f로 인해서 소수점 2번째 자리 까지만 표현 반올림o 버림x]

☞ ex) print("실수 표현 : %f %f"%(456.456 , 123.123)); --> 실수 표현 : 456.456000 123.123000 [소수점 6자리까지 표현]

■ %c : 단일 문자

☞ ex) print("문자 표현 : %c %c"%("한","글")); --> 문자 표현 : 한 글 [“”‘’상관X 열고 닫는 것 만 같으면 됨]

☞ ex) print("문자 표현 : %c %c"%('표','현')); --> 문자 표현 : 표 현

■ %s : 문자열

☞ ex) print("문자열 표현 : %s"%"안녕"); --> 문자열 표현 : 안녕

☞ ex) print("문자열 표현 : %s\t%s"%('문자열',"표현방식")); -->

문자열 표현 : 문자열 표현방식 [“”‘’상관X 열고 닫는 것 만 같으면 됨]

※서식 지정

■ %d : 10진 정수

☞ ex) print("기본 값 : %d"%123); --> 기본 값 : 123

☞ ex) print("설정 값 : %5d"%123); --> 설정 값 : 123 [5칸에서 오른쪽 정렬]

☞ ex) print("설정 값 : %05d"%123); --> 설정 값 : 00123 [5칸에서 오른쪽 정렬 나머지 칸 에는 0출력]

☞ ex) print("설정 값 : %5d%5d"%(123,123)); --> 설정 값 : 123 123 [5칸에서 오른쪽 정렬]

☞ ex) print("설정 값 : %-5d%-5d"%(123,123)); --> 설정 값 : 123 123 [5칸에서 왼쪽 정렬(음수 - 이기 때문에)]

■ %f : 실수(소수점이 붙은 수)

☞ ex) print("기본 값 : %f"%123.45678); --> 기본 값 : 123.456780 [소수점 6자리이기 떄문에 마지막에 0]

☞ ex) print("설정 값 : %10.3f"%123.45678); --> 설정 값 : 123.457 [10칸에서 오른쪽 정렬, 소수점 3번째 까지 표현하고 반올림]

☞ ex) print("설정 값 : %2.1f"%123.45678); --> 설정 값 : 123.5 [2칸에서 오른쪽 정렬이지만 출력하는 것이 5칸이기 때문에 무시하고 그냥 실행, 소수점 1번째 까지 표현 반올림]

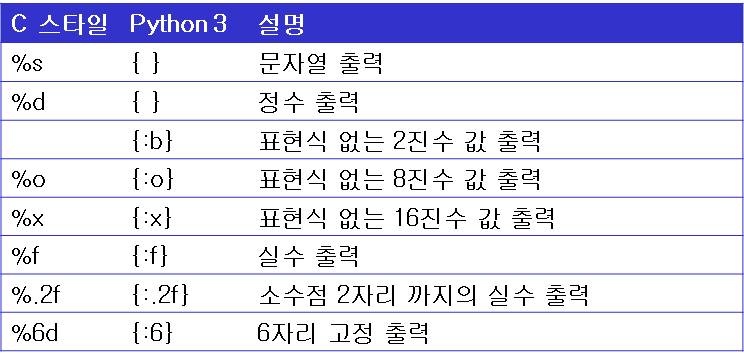
☞ ex) print("설정 값 : %.2f"%123.45678); --> 설정 값 : 123.46 [소수점 2번째 까지 표현하고 반올림]

■ %s : 문자열

☞ ex) print("기본 값 : %s"%"python test"); --> 기본 값 : python test

☞ ex) print("설정 값 : %20s"%"python test"); --> 설정 값 : python test [python test = 11글자 20칸에서 오른쪽 정렬]

※서식 문자 확장 (파이썬 3.x 버전 서식문자)

☞ ex) print('%s님의 나이는 %d 입니다.'%('김동환',38)); --> 김동환님의 나이는 38입니다.

☞ ex) print('{}님의 나이는 {} 입니다.'.format('김동완',38)); --> 김동환님의 나이는 38입니다. [% 넣지말고 .format 넣어야함 %넣으면 인식하지 못함]

☞ ex) print("%10s , %-10s"%('오른쪽','왼쪽')); --> 오른쪽 , 왼쪽 [양수 : 오른쪽 정렬, 음수 : 왼쪽 정렬]

☞ ex) print("{:>10} , {:<10}".format('오른쪽','왼쪽')); --> 오른쪽 , 왼쪽 [> : 왼쪽이 비어 있음(우측 정렬), < : 오른쪽이 비어 있음(좌측 정렬)]

☞ ex) print('{:^10}'.format('가운데')); --> 가운데 [^ : 가운데 정렬 정확하게 계산이 되지 않을 때는 왼쪽의 빈칸이 더 적음]

☞ ex) print('{:,}'.format(1000000)); --> 1,000,000 [, : 1000단위에 , 표시] [문자열일 경우 에러 생김]