

Python

주석 #, ctrl + /

ctrl + 1 왼쪽 창 없애기/보이기

ctrl + 4 실행창 없애기/보이기

변수

- 1. 기본

```
#수치적으로 나오는 것들을 더함 = 계산기
#글자에서의 덧셈 = 합쳐라
a="ABC"
b="XYZ"
print(a+b)

#글자 반복
a="ABC"
b=3
print(a*b)
```

- 2. 리스트

```
#리스트 []
# 값의 접근 인덱스(index) : 0번째 부터 시작
#   -인덱스를 활용하여 값을 꺼내올 수 있음
#   -인덱스의 범위를 벗어나는 경우 에러(list index out of range)
#리스트에는 다양한 타입의 데이터들을 함께 담을 수 있음
a= [1,2,3,4,5, "A"] #컬렉션

a[4] = "Z" #리스트 데이터 변경 , 4번째 위치에 Z (수정)

a.append(6) #리스트에 새로운 데이터 추가, 리스트의 맨 뒤에 추가
a.insert(1, "BBB") #리스트 데이터 추가(자리 지정) : insert(몇 번째, 어떤 값)
print(a) #[1, 'BBB', 2, 3, 4, 'Z', 'A', 6]

a.remove("Z") #리스트의 데이터 지우기
del a[0]      #리스트의 데이터 지우기
a.pop(0) #리스트의 데이터 지우기, 맨 뒤의 값 지우기
          #pop(인덱스) : 인덱스 지정 시 해당 위치의 값 삭제

print(a)
print(a[4]) #a의 4번째 값을 꺼내라 : 5

#리스트 중첩 가능
mem =[1,2,3, ["A","B"] ]
print(mem)
print(mem[3][1]) #B
```

```
#리스트에 튜플 넣기
mem =[1,2,3, ("A","B") ]
print(mem[3][1]) #B

#리스트에 딕셔너리 넣기
mem =[1,2,3, {"empno":7733, "ename":"smith"} ]
print(mem[3]["empno"]) #7733
```

- 3. 튜플

```
#튜플() : 리스트의 상수형 -->항상 변하지 않는 수
#한번 세팅하면 끝, 값 바꿀 수 없음(수정, 변형 불가) -->조회 용도
b=(1,2,3,4,5)
print(b)
print(b[2]) # b의 2번째 값을 꺼내라 : 3
```

- 4. 딕셔너리

```
#딕셔너리{키:값, 키:값, 키:값, ....}
d = {"empno":7733, "ename":"smith"}
print(d)
print(d["ename"]) #키 값 꺼내기 [키 명]: smith

d["job"] = "SALES" # 데이터 추가 : 기존에 없는 키면 가능

d["ename"]="SSMITH" # 기존에 있는 키면 기존 값 변경
print(d)

del d["job"] #딕셔너리 삭제
print(d)
```

- 실습

```
#-----title, url, viewcount 출력
ytb={
  "kind": "youtube#videoListResponse",
  "etag": "\"UCBpFjp2h75_b92t44sqraUcyu0/sDA1sG9NGKfr6v5A1PZKSEZdtqA\"",
  "videos": [
    {
      "id": "71CDEYXw3mM",
      "kind": "youtube#video",
      "etag": "\"UCBpFjp2h75_b92t44sqraUcyu0/iYynQR8AtacsFUwwmrVaw4Smb_Q\"",
      "snippet": {
        "publishedAt": "2012-06-20T22:45:24.000Z",
        "channelId": "UC_x5XG10V2P6uZZ5FSM9Ttw",
        "title": "Google I/O 101: Q&A On Using Google APIs",
        "description": "Antonio Fuentes speaks to us and takes questions on working with Google APIs and OAuth 2.0.",
        "thumbnails": {
          "default": {
            "url": "https://i.ytimg.com/vi/71CDEYXw3mM/default.jpg"
          }
        }
      }
    }
  ]
}
```

```

        "medium": {
            "url": "https://i.ytimg.com/vi/7lCDEYXw3mM/mqdefault.jpg"
        },
        "high": {
            "url": "https://i.ytimg.com/vi/7lCDEYXw3mM/hqdefault.jpg"
        }
    },
    "categoryId": "28"
},
"contentDetails": {
    "duration": "PT15M51S",
    "aspectRatio": "RATIO_16_9"
},
"statistics": {
    "viewCount": "3057",
    "likeCount": "25",
    "dislikeCount": "0",
    "favoriteCount": "17",
    "commentCount": "12"
},
"status": {
    "uploadStatus": "STATUS_PROCESSED",
    "privacyStatus": "PRIVACY_PUBLIC"
}
}
]
}

```

#모든 준정형 데이터들은 이와 같이 자료를 꺼내옴

```

print( ytb["videos"][0]["snippet"]["title"] )
print( ytb["videos"][0]["snippet"]["thumbnails"]["default"]["url"] )
print( ytb["videos"][0]["statistics"]["viewCount"] )

```

#-----type: delete의 url

```

nv={
    "urls": [
        {
            "url": "http://www.your-site.com/article-1",
            "type": "update"
        },
        {
            "url": "http://www.your-site.com/article-2",
            "type": "update"
        },
        {
            "url": "http://www.your-site.com/article-3",
            "type": "delete"
        },
        {
            "url": "http://www.your-site.com/article-4",
            "type": "delete"
        }
    ]
}

```

```
print(nv["urls"][2]["url"])
print(nv["urls"][3]["url"])
```

- 타입 변환

```
#변수 타입
#type() : 변수 타입 알아보는 함수
print(type(5),5) #<class 'int'>
print(type("abc"),"abc") #<class 'str'>
print(type(17.4),17.4) #<class 'float'>
print(type([1,2,3]),[1,2,3])
print(type((1,2,3)),(1,2,3))
print(type({"empno":777}),"empno":777})
#true false : 불리언(bool)
print((type(True),True))

#-----캐스팅 : casting : 형변환 : 타입변환
#원하는 타입으로 감싸기 : 타입()
print(3 + int("4"))
print(str(3) + "4") #34
print(4 + float(3)) #7.0 ->타입이 큰 쪽으로 따라감(둘 중 하나만 타입 변환해도 됨)
```

- 슬라이싱

```
#-----슬라이싱(Slicing)
#[시작이상 : 끝미만 : 방향(1정방/-1거꾸로)]
msg = "ABCDE"
print ( msg[0:2]) #AB
print ( msg[3:]) #DE
print ( msg[-2:]) #DE 뒤에서부터 -1

emps=[7733,7822,7922,8000]
print(emps[1:3]) #[7822, 7922]

license_plate = "24가 2210"
print(license_plate[-4:]) #2210

string= '흔썸흔썸흔썸'
print(string[::2]) #흔흔흔

#출수만 출력
nums = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(nums[::2])

string="PYTHON"
print(string[::-1]) #NOHTYP
```

- 함수

```
# -제외하고 출력하기
#변수명.replace
phone_number = "010-1111-2222"
print(phone_number[0:3],phone_number[4:8],phone_number[9:]) #010 1111 2222
```

```

phone_number1 = phone_number.replace("-", " ") #replace(원본문자, 바꿀문자)
print(phone_number1) #010 1111 2222

#좌우 공백 제거
#변수명.strip()
data="  삼성전자  "
data1=data.strip()
print(data1) #삼성전자

#구분자 기준으로 문자열 자르기
#변수명.split(구분자)
#리스트로 반환
a = "hello world"
a.split()
print(a.split()) #['hello', 'world']

tel="010-123-1234"
print(tel.split("-")) #['010', '123', '1234']
print(tel.split("-")[1]) #123

#-----리스트 합치기
lang1 = ["C", "C++", "JAVA"]
lang2 = ["Python", "Go", "C#"]
lang3 = lang1+lang2
print(lang3)

lang1.append(lang2)
print(lang1) #['C', 'C++', 'JAVA', ['Python', 'Go', 'C#']]
# lang1원 안에 lang2 리스트를 그대로 집어넣음(리스트 안에 리스트)

lang1.extend(lang2)
print(lang1) #['C', 'C++', 'JAVA', 'Python', 'Go', 'C#']
#lang1에 extend 했기 때문에 lang1 을 출력해야 함

# len() : 길이 재기
print(len(lang1))

```

조건문

- if문

```
if 조건식:
    실행1
elif 조건식:
    실행2
elif 조건식:
    실행3
else :
    실행4
```

#인덴테이션(들여쓰기) 주의-->Python

```
#90이상이면 A   80이상이면 B   아니면 C
score = 80
```

```
if score >=90:
    print("A")
elif score >=80:
    print("B")
else:
    print("C")
```

```
#중첩 if
#90이상이면서 성별이 남성이면 1 여성이면 2
#80이상이면서 성별이 남성이면 11 여성이면 22
#아니면 0
```

```
score = 80
gen = "남"
```

```
#1)중첩 if
if score >=90:
    if gen == "남" :
        print("1")
    else:
        print("2")
elif score >=80:
    if gen == "남":
        print("11")
    else:
        print("22")
else:
    print("0")
```

```
#2)and로 연결
if score >=90 and gen == "남" :
    print("1")
if score >= 90 and gen == "여":
    print("2")
elif score >=80 and gen == "남":
    print("11")
elif score >=80 and gen == "여":
    print("22")
else:
    print("0")
```

반복문

for 조건식 :
반복실행문

- 1. for 문 : 시작 값과 끝 값이 명확 할 때 사용

```
emp = [7788,7799,7800]
# empno = emp[0]
# empno = emp[1]
# empno = emp[2]

for empno in emp: #emp 안에 있는 것 중에 하나만 꺼내어 empno에 담아라
    print(empno)

#-----
# range(시작번호 이상, 끝번호 미만, 증감분)
# 끝번호 미만*****
# 증감분 기본 값 : 1
#-----
print ( list (range(1,6) ) ) #[1, 2, 3, 4, 5]
#list : 값을 담고 있는 객체
print ( list (range(1,11,2) ) ) #[1, 3, 5, 7, 9]
print ( list (range(10,0,-1) ) ) #[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

#for 문 range 이용 1~10 출력
for i in range(1,11):
    print(i, end="\t")

print()

#-----중첩 for문
for i in range(100,600,100):
    print(i)
    for j in range (1,4):
        print(j, end="\t")
    print()

#-----구구단
dan = 2
gob = 3
print(dan,"*",gob,"=", dan*gob)

# 2단
# 2*1=2 2*2=4 ..... 2*9=18
# 3단
# 3*1=3 3*2=6 ..... 3*9=27

for dan in range(2,10):
    print()
```

```

print(dan, "단")
for gob in range(1,10):
    print(dan, "*", gob, "=", dan*gob, end="\t")
print()

#2,4,6,8단 *5
for dan in range(2,9,2):
    print()
    print(dan, "단")
    for gob in range(1,6):
        print(dan, "*", gob, "=", dan*gob, end="\t")
    print()

#9,7,5,3단 * 9 *8 *7 *6 *5
for dan in range(9,2,-2):
    print()
    print(dan, "단")
    for gob in range(9,4,-1):
        print(dan, "*", gob, "=", dan*gob, end="\t")
    print()

#-----중첩 for 별찍기

# *
# ***
# *****
for g in range(2,-1, -1):
    print(" "*g, end="")
    for i in range(1, 6, 2):
        print("*" * i)

# *
# ***
# *****
for i in range(3):
    print(' ' * (2-i), end='')
    print((i * 2 + 1) * '*')

for n in range(3):
    for space in range(2-n):
        print(' ', end='')
    for star in range(1+n*2):
        print('*', end='')
    print()

# ****
# ***
# **
# *
for j in range(1,5):
    for k in range(1,5):
        #if j=1 k=1 2 3 4
        if j <= k :
            print("*", end = "")

```



```

        else :
            print(" ",end = "")
    print()

#      *
#     **
#    ***
#   ****
for j in range(1,5): #행
    for k in range(4,0,-1): #칸
        #if j=1 k=4 3 2 1
        if j < k :
            print(" ", end = "")
        else :
            print("*",end = "")
    print()

```

- 2. while 문

```

while(조건식) :
    반복실행문

#-----
# do :                -->파이썬에는 없음
#     반복실행문
# while(조건식)

```

```

num = 5
while(num < 10) :
    print("그렇다" , num)
    num=num+1

# 그렇다 5
# 그렇다 6
# 그렇다 7
# 그렇다 8
# 그렇다 9

#while문으로 구구단
dan=2

while(dan<10):
    print(dan, "단")
    gob = 1
    while(gob<10):
        print(dan, "*", gob, "=", dan*gob)
        gob+=1
    dan += 1
    print()

```