

또 만나서 반가워요
여러분 

오늘은 Python을 활용해 볼 예정입니다!

엄청 유명한 프로그래밍 언어인 만큼
이미 파이썬을 아는 분들이 많으리라 생각합니다!

미리 파이썬을 학습해 보신 분은
잠시 쉬어가는 시간으로,

처음 배우시는 분은 차근차근 배워가는 시간이
되어 보도록 해요!

그래서 왜 파이썬을 배울까요?



1. 배우기 쉽고
2. 생산성이 높고
3. 다양한 분야에서 쓰이기 때문!

오늘 파이썬을 활용하며 복습할 내용들은

변수
&
자료형

함수

조건문

리스트

딕셔
너리

반복문

오늘 파이썬으로 만들어 볼 프로그램들은?

1. 구구단 출력해보기



변수
&
자료형

조건문

반복문

특정 구구단을 출력하는 프로그램을 만들어 주세요

주의점

1. 소수나, 1~9 사이의 정수가 아니면 다시 입력받아야 합니다!
2. 출력 형태는 $n * 1 = 9$ / $n * 2 = 18 \dots$ 의 형태로 출력 해주셔야 합니다!

변수 & 자료형

변수란?

프로그래밍을 할 때 다양한 데이터를 만들고, 저장해야 합니다!
이 때, 데이터를 저장하는 공간을 “변수”라고 부릅니다

변수를 만들고 사용하는 방법은?

예)

```
name = “최준형”  
print(name)
```

```
number = 1  
print(number + 1)
```

변수
&
자료형

“1” / 1 / 1.0

이 세가지는 뭐가 다를까요??

이 세 가지는 모두 “자료형”이 다른 형태입니다.

“(큰따옴표), “(작은따옴표)로 묶인 값들은 문자열(string, str)

소수점이 붙지 않은 수들은 정수(int)

소수점이 붙은 수들은 실수(float)



이외에도 값이 없음을 표현하는 None

참과 거짓을 표현하는 True, False 도 있습니다!

조건문

조건문을 알기 전에 조건식(논리식)을 알아 보도록 해요!

논리식은 결과 값이 True, False로 나뉘는 식을 의미합니다!

예) - age라는 변수에 12라는 int형이 저장되어 있는 상황

`age > 15 //False`

`age == 12 //True`

`Age != 12 //False`

`Age < 15 //True`

위에서 알 수 있듯이, 논리식에 사용되는 논리 연산자엔
>, >=, <, <=, ==, != 등이 있습니다!

조건문

추가적인 연산자로 'and'와 'or'가 있는데요 잘 따라와 주세요!

and: 양쪽의 조건식이 모두 참이어야 참

예) 조건식1 and 조건식2

조건식1,2의 결과가 모두 참이어야 참, 그 이외엔 거짓

조건식1	조건식2	전체 결과
True	True	True
True	False	False
False	True	False
False	False	False

조건문

추가적인 연산자로 'and'와 'or'가 있는데요 잘 따라와 주세요!

Or: 한쪽의 조건식만 참이어도 참

예) 조건식1 or 조건식2

조건식1,2의 결과가 둘중의 하나라도 참이라면 참, 둘 다 거짓일 시 거짓

조건식1	조건식2	전체 결과
True	True	True
True	False	True
False	True	True
False	False	False

조건문

조건문은 이렇게 작성됩니다!

```
myname = "choi"

if(myname == "kim"):
    print("당신의 성씨는 김씨입니다.")
elif (myname == "choi"):
    print("당신의 성씨는 김씨가 아닌 최씨입니다.")
else:
    print("당신의 성씨는 김씨, 최씨가 아닙니다")
```

조건문

복합 조건문은 이렇게 되겠죠?

```
number = 10

if(number == 3 or number == 6 or number == 9):
    print("당신의 숫자는 3,6,9중에 하나입니다.")

if(number % 2 == 0 and number % 5 == 0):
    print("당신의 숫자는 2와 5의 배수입니다.")
```


반복문

파이썬의 반복문은 while문과 for문이 있습니다.

While문을 먼저 보자면, 조건이 맞는 동안 계속 도는 if문과 같다고 보시면 됩니다.

```
myage = 5

while(myage < 20):
    print("응애")
    myage += 1
```

```
myage = 5

while(myage < 20):
    print("응애")
    #myage += 1
    #myage가 변하지 않는다면? 조건이 계속 True이므로 무한반복!
```

While문을 쓸 땐 무한 반복에 빠지기 쉬우므로 조심해 주세요!

반복문

For문은 추후 배열 리스트, 딕셔너리 같은 여러 개를 모아둔 주머니에서 하나씩 꺼내 반복해보는 형태입니다

```
for 꺼낸것의이름 in 리스트:
    #이 영역에서 꺼낸것의이름을 사용하시면
    #리스트에 있던 것들이 매 반복마다 하나씩 뽑혀서
    #꺼낸것의이름의 변수에 값이 할당됩니다
```

```
week = ["월", "화", "수", "목", "금", "토", "일"]
```

```
for day in week:
    print(day)
```

변수
&
자료형

조건문

반복문

특정 구구단을 출력하는 프로그램을 만들어 주세요

주의점

1. 1~9 사이의 정수가 아니라면 다시 입력받아야 합니다!

2. 출력 형태는 $n * 1 = 9$ / $n * 2 = 18 \dots$ 의 형태로 출력 해주셔야 합니다!

프로그램 출력 예시

```
python3 gugudan.py
```

몇단을 보실 예정인가요?: 15

구구단의 범위를 벗어났습니다.

몇단을 보실 예정인가요?: 9

9 x 1 = 9

9 x 2 = 18

9 x 3 = 27

9 x 4 = 36

9 x 5 = 45

9 x 6 = 54

9 x 7 = 63

9 x 8 = 72

9 x 9 = 81

2. 계절별 음식, 옷 골라서 출력해보기!



리스트

딕셔
너리

함수

계절에 맞는 음식, 옷을 출력해주는 프로그램을 만들어 봅시다!

주의점

1. 출력 형태는 예시를 맞춰주세요!

리스트

순서가 있고, 중복이 가능한 집합의 형태를 가진 자료입니다.

[] (대괄호)를 사용해서 생성 가능합니다.

안에 들어가는 자료형은 아무거나 가능합니다

```
mylist = [1, 1.0, True, None, [1, 2, 3, 4], "abc"]
```

리스트

안의 내용물을 꺼내고 싶으면

리스트이름[몇번째 있는지]를 적어줍니다
(컴퓨터에서 숫자는 항상 0부터 시작합니다 -
Starting from Zero!)

```
fruit = ["apple", "banana", "grape"]
```

```
fruit[0] #"apple"
```

```
fruit[2] #"grape"
```

```
fruit[3] #ERROR!!!!
```


딕셔
너리

여러가지 요소들이 키(key)와 값(value)로 묶인 형태입니다.

{키:값, 키:값, ... } 의 형태로 선언됩니다.

```
seasonFruitDictionary = {"봄" : "딸기",
                        "여름" : "수박",
                        "가을" : "사과",
                        "겨울" : "귤"}
```

딕셔
너리

안의 내용물을 꺼내고 싶으면

딕셔너리이름[키]를 적어줍니다.

```
seasonFruitDictionary = {"봄" : "딸기",
                          "여름" : "수박",
                          "가을" : "사과",
                          "겨울" : "귤"}
```

```
seasonFruitDictionary["봄"] #딸기
seasonFruitDictionary["여름"] #수박
seasonFruitDictionary["가을"] #사과
seasonFruitDictionary["겨울"] #귤
```

```
seasonFruitDictionary["비"] #ERROR!!!
```



함수는 특정 기능을 하는 기능단위입니다.

Def 키워드를 사용해서 함수를 정의합니다

파라미터는 함수가 실행될 때 외부에서 데이터를 받아오는 용도로 사용됩니다.
(개수는 상관 없습니다 - 없어도 됩니다)

Return 키워드를 사용하면 함수를 그 자리에서 중지 및 특정 값을 반환합니다.

```
def functionName(parameter1,parameter2):
    ## 실행문
    a = 1
    b = 2
    print(a,b)

    return 123
```



함수는 특정 기능을 하는 기능단위입니다.

Return 키워드는 함수가 그 값으로 바뀐다고 생각하시면 편합니다.

```
def myFunction(number):
    result = number + 10
    return result

print(myFunction(20)) #print(30)으로 변한다고 생각!
```

리스트

딕셔
너리

함수

계절에 맞는 음식, 옷을 출력해주는 프로그램을 만들어 봅시다!

주의점

1. 출력 형태는 예시를 맞춰주세요!

프로그램 출력 예시

```

❏ ➤ ~/l/s/pythonbasic ➤ python3 weatherAnswer.py
현재 계절은 어느 계절인가요? - 봄, 여름, 가을, 겨울: 봄
봄 에 맞는 옷은 코트 입니다.
봄 의 제철 과일은 딸기 입니다.
❏ ➤ ~/l/s/pythonbasic ➤ |
    
```

배대 코드

```
def printWeather(wdict,w):
    #여기에 코드를 작성해 주세요
    pass
```

```
wdict = {"봄" : ["코트", "딸기"],
        "여름" : ["반팔", "수박"],
        "가을" : ["긴팔", "사과"],
        "겨울" : ["패딩", "감귤]}
```

```
w = input("현재 계절은 어느 계절인가요? - 봄, 여름, 가을, 겨울: ")
```

```
printWeather(wdict,w)
```

3. 데이터베이스를 만들어 보자…?



데이터 베이스란?

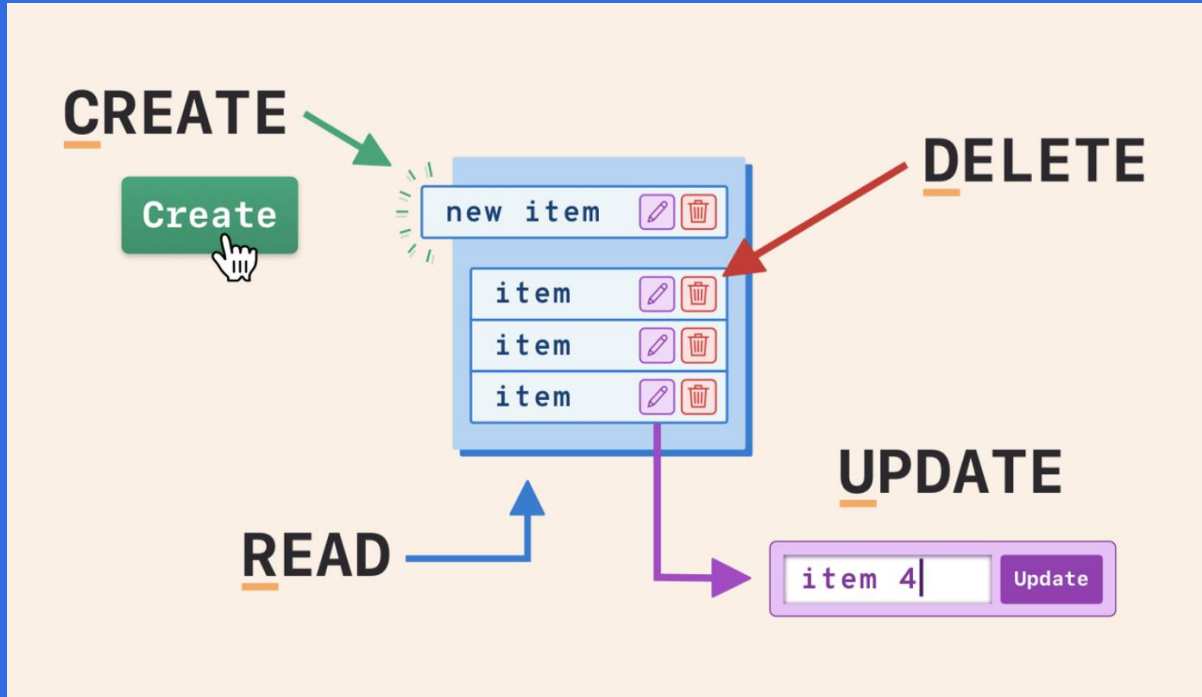
여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합이다. - 위키백과

이번 시간에는 사용자
데이터베이스를
만들어 볼 예정입니다!

target_date | target_time | server_time | cpu_index | cpu | device

target_date	target_time	server_time	cpu_index	cpu	device
2016-12-26	02:29:30	1482737402	2	12	
2016-12-26	02:32:29	1482737582	1	13	
2016-12-26	02:32:29	1482737582	2	19	
2016-12-26	02:35:29	1482737762	1	14	
2016-12-26	02:35:29	1482737762	2	19	
2016-12-26	02:38:29	1482737942	1	16	
2016-12-26	02:38:29	1482737942	2	18	
2016-12-26	02:41:30	1482738123	1	13	
2016-12-26	02:41:30	1482738123	2	19	
2016-12-26	02:41:30	1482738302	1	15	
2016-12-26	02:44:29	1482738302	2	18	
2016-12-26	02:44:29	1482738482	1	15	

그래서 어떻게?



CRUD를 통해서!

Create
Read
Update
Delete

이 네가지
기능을 가진
데이터베이스를
만들어 보아요!

그래서 어떻게?

Create : 사용자 데이터를 추가할 수 있는 기능을 만들어 주세요!

Update: 사용자 데이터를 수정할 수 있는 기능을 만들어 주세요!

Read: 사용자 데이터를 읽어 올 수 있는 기능을 만들어 주세요!

Delete: 사용자 데이터를 삭제 할 수 있는 기능을 만들어 주세요!

딕셔너리를 활용 해 보자!

```
dic = {"a":1}

#딕셔너리에 값을 추가할때?
dic["키"] = "값"

#딕셔너리의 값을 수정할때?
dic["키"] = "변경된 값"

#딕셔너리의 값을 삭제할때?
del dic["키"]
#또는
dic.pop("키")

#딕셔너리의 키로 반복문을 쓰고 싶다면?
for key in dic:
    key
```

옆의 힌트를 잘 보시고
보여드리는 형태와
비슷하게
사용자의 이름과 나이를
관리하는
데이터베이스를 만들어
주세요!

예시

```
python3 database.py
나이 데이터베이스에 오신 걸 환영합니다.

어떤 작업을 수행하시겠습니까? 1)생성하기 2)목록보기 3)수정하기 4)삭제하기 5) 프로그램 끝내기: 2

-----나이데이터베이스-----
choi : 15

어떤 작업을 수행하시겠습니까? 1)생성하기 2)목록보기 3)수정하기 4)삭제하기 5) 프로그램 끝내기: 1

이름을 입력 해 주세요: jun
나이를 입력 해 주세요: 25

어떤 작업을 수행하시겠습니까? 1)생성하기 2)목록보기 3)수정하기 4)삭제하기 5) 프로그램 끝내기: 3

-----나이데이터베이스-----
choi : 15
jun : 25

나이를 수정할 이름을 적어주세요: jun
수정할 나이를 적어주세요: 17

어떤 작업을 수행하시겠습니까? 1)생성하기 2)목록보기 3)수정하기 4)삭제하기 5) 프로그램 끝내기: 4

-----나이데이터베이스-----
choi : 15
jun : 17

삭제할 사용자 이름을 입력해주세요: choi

어떤 작업을 수행하시겠습니까? 1)생성하기 2)목록보기 3)수정하기 4)삭제하기 5) 프로그램 끝내기: 2

-----나이데이터베이스-----
jun : 17

어떤 작업을 수행하시겠습니까? 1)생성하기 2)목록보기 3)수정하기 4)삭제하기 5) 프로그램 끝내기: 5
```



오늘도 수고하셨습니다! 🐼