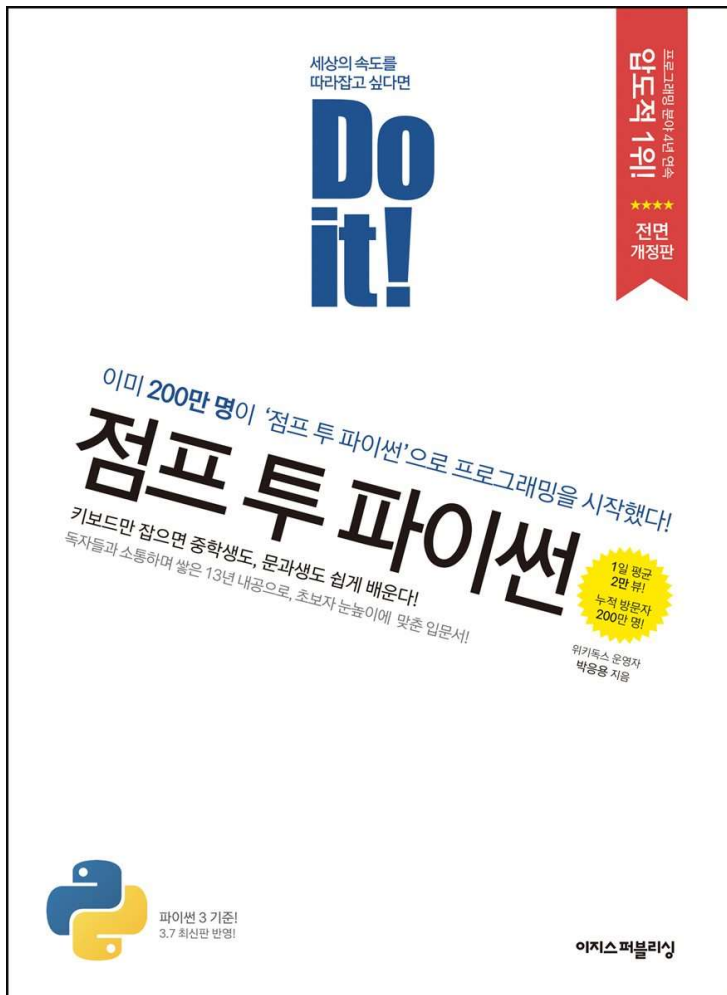


WISET 여대학원생 공학연구팀제

2021.06.06



2장. 파이썬 프로그래밍 기초, 자료형

2-1. 숫자형

2-2. 문자열 자료형

2-3. 리스트 자료형

2-4. 튜플 자료형

3장. 프로그램의 구조를 쌓는 제어문

3-1. if문

3-2. while문

3-3. for문

```
import pandas as pd
wiset=pd.read_csv("C:/Users/SAMSUNG/OneDrive/바탕 화면/wiset.csv")
wiset
```

	HJH	LSB	JYJ
0	1	2	3
1	4	5	6
2	7	8	9

객체(Object)
= 어떤 데이터 또는 정보가
저장되어 있는 것

변수(Variable)
= 객체를 저장한 공간

객체의 유형은 무엇일까 ?

자료형 (DataType)

파이썬 자료형





An Introduction to Sequence Data

```
a = 'I love Python' # String
b = [1, 2, 3, 4, 5]  # List
c = (1, 2, 3, 4, 5)  # Tuple
d = 10               # int
e = 10.3             # float
```

Indexing

Slicing

Concatination

Repetition

Member Check

Length Info

조건문(if-elif-else)

1) 조건이 하나일 때

If (조건):
 (조건에 맞는 출력)
else :
 (조건에 맞는 출력)

a = 3 , b = 4

```
>>> if a > b:
...     print('a')
... else:
...     print('b')
...
```

만약 a가 b보다 크면
'a'를 출력한다.
그렇지 않으면
'b'를 출력한다.

조건문(if-elif-else)

2) 조건이 여러개일 때

If (조건):
 (조건에 맞는 출력)
elif (다른 조건):
 (조건에 맞는 출력)
elif (다른 조건):
 (조건에 맞는 출력)
...
else :
 (조건에 맞는 출력)

```
>>> c = 15 * 5
>>> d = 15 + 15 + 15 + 15 + 15
>>> if c > d:
...     print('c is greater than d')
... elif c == d:
...     print('c is equal to d')
... elif c < d:
...     print('c is less than d')
... else:
...     print('I don\'t know')
```


반복문(while)

While문의 기본 구조

While [조건문]:
 [수행부분]

While 반복문은 [조건문]이
True 인 경우 내부의 수행 부분을 진행하고
False 인 경우 while문을 빠져나간다.

```
>>> num = 1
>>> while num <= 100:
...     print(num)
...     num = num + 1
... 
```

1
2
3
.
.
.
.
100

반복문(for)

for문의 기본 구조

for 변수 in 범위 :
반복해서 실행할 명령

```
country = ['korea', 'china', 'japan']

for c in country:
    print(c)

"""
korea
china
japan
"""
```

```
for i in range(1,11):
    if i % 2 == 0:
        print(f'{i}는 짝수입니다.')
    else:
        print(f'{i}는 홀수입니다.')
```

출력 결과:

```
1는 홀수입니다.
2는 짝수입니다.
3는 홀수입니다.
4는 짝수입니다.
5는 홀수입니다.
6는 짝수입니다.
7는 홀수입니다.
8는 짝수입니다.
9는 홀수입니다.
10는 짝수입니다.
```

range(1,11,1)
=> 1부터 10까지 10개

연구에서 쓰일 부분

#5. After that, we will add a new dimension to the tensors:

```
if len(x_train.shape) == 3:  
    x_train = x_train.unsqueeze(1)
```

```
print(x_train.shape)
```

```
if len(x_val.shape) == 3:  
    x_val = x_val.unsqueeze(1) •
```

```
print(x_val.shape)
```

```
torch.Size([60000, 1, 28, 28])
```

```
torch.Size([10000, 1, 28, 28])
```

```
for xb, yb in train_dl:  
    #move batch to cuda device  
    xb = xb.type(torch.float).to(device)  
    yb = yb.to(device)  
    #get model output  
    out = model(xb)  
    #calculate loss value  
    loss = loss_func(out, yb)  
    print(loss.item()) # 총합 계산  
    break
```

124.3841323852539

연구에서 쓰일 부분

```
img_tr_np[0]
```

```
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      ],  
[0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      ],  
[0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      ],  
[0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      , 0.      , 0.      ,  
0.      , 0.      , 0.      ], dtype=float32)
```

Reference

파이썬 자료형

<https://bodhi-sattva.tistory.com/58>

<https://analyticsstory.com/31>