







- >> if
- > if else
- > if elif



★ 반복 for in

- >> for var in iterable:
  - statement



반복 while

- while condition:
  - statement



- ▼ 조건에 따른 if 문을 이해하고 구현할 수 있다.
- ☑ 반복가능한 개체에서 순차적으로 항목을 꺼내 반복하는 for 문을 이해하고 구현할 수 있다.
- ☑ 조건에 따라 반복하는 while 문을 이해하고 구현할 수 있다.

# LESSON 01

# 조건 if



#### ☑ 조건 if



#### ⊸ 조건 if



#### ☑ 블록

```
# %% if
     credits = 10
     if credits < 12:</pre>
         print('한 학기 수강 학점이 부족합니다.')
[25] V 0.0s
                                                                  Python
  한 학기 수강 학점이 부족합니다.
     # 한 줄로 작성하지 말도록 권고
     if credits < 12: print('한 학기 수강 학점이 부족합니다.')
[26] 			 0.0s
                                                                  Python
  한 학기 수강 학점이 부족합니다.
```

필요하다면 가능



## ⊸ 조건과 else:

## **❷** 조건이 True, False

```
speed = 55
  if 40 <= speed <= 60:
      print(f'현재 속도 {speed}는 적정 합니다.')
                                                                   Python
현재 속도 55는 적정 합니다.
  temperature = 34
  if 33 <= temperature:</pre>
      print(f'폭염에 주의하세요.')
  else:
      print(f'기온 {temperature}도 입니다.')
✓ 0.0s
                                                                   Python
폭염에 주의하세요.
```



## → if elif



```
age = 20
      if age >= 70:
          print('노인')
      elif age >= 20:
          print('성인')
      elif age >= 12:
          print('청소년')
      elif age >= 6:
          print('어린이')
      elif age >= 1:
          print('유아')
      else:
          print('영아')
[29] 		0.0s
                                                                           Python
  성인
```



#### ⊸ 조건 연산자



#### ♪ 반환 값을 대입도 가능



#### ⊸ 조건의 반복



```
point = 85
      grade = 'C' if point < 80 else 'B' if point < 90 else 'A'</pre>
      print(grade)
[32] V 0.0s
      age = 5
      age_grade = '노인' if age >= 70 else \
                  '성인' if age >= 20 else \
                  '청소년' if age >= 12 else \
                  '어린이' if age >= 6 else \
                  '유아' if age >= 1 else \
                  '영아'
      print(age_grade)
[33] V 0.0s
                                                                             Python
   유아
```



#### → 한 줄에 사용하는 if

## 특별히 간략히 표현하려는 경우



#### ⊸ 블록이 간단한 경우



#### ☑ 한 줄에 여러 문장 가능

```
credits = 15
     if credits < 12: print('학점이 부족합니다.'); print('최소 학점: 12')
     else: print(f'신청 학점: {credits}'); print('학점이 적정합니다.')
[37] 		 0.0s
  신청 학점: 15
  학점이 적정합니다.
     num = 1
     if num == 0: print('해'); print('달'); print('별');
     elif num == 1: print('눈'); print('코'); print('잎');
     elif num == 2: print('소'); print('말'); print('양');
     else: print('꽃'); print('비'); print('씨');
  눈
  코
  잎
```

LESSON 02

# 반복 for in과 while





## → for in



#### for var in iterable

```
for i in [1, 2, 3]:
        print(i)
     else:
        print('리스트의 모든 항목에 대해 반복 종료!')
[39] 		 0.0s
  리스트의 모든 항목에 대해 반복 종료!
     for i in [1, 2, 3]:
        print(i)
```



#### → 반복가능 객체



```
fruits = ("apple", "banana", "grape")
      for one in fruits:
        print(one)
   ✓ 0.0s
                                                                                 Python
   apple
   banana
   grape
      languages = {'Swift', 'Python', 'Go', 'Java'}
      for lang in languages:
           print(lang)
[43] 			 0.0s
   Swift
   Go
   Java
   Python
```



## ⊸ 사전과 문자열 반복



#### ☑ 키가 항목으로 반복

```
languages = {'Swift': 2014, 'Python': 1991, 'Go': 2009, 'Java': 1995}
     for lang in languages:
          print(lang)
[44] 			 0.0s
  Swift
  Python
  Go
  Java
     for lang in languages:
          print(languages[lang])
[45] 		 0.0s
  2014
  1991
  2009
  1995
     for c in 'python':
          print(c, end=' ')
  python
```



## ¬ range(start, step, stop)

# [start, stop), next = prev + step

```
# %% type range()
      r = range(6) # 0, 1, 2, 3, 4, 5
      print(r, type(r))
  range(0, 6) <class 'range'>
      print(r.start, r.step, r.stop)
[49] 		 0.0s
  0 1 6
      list(range(6))
  [0, 1, 2, 3, 4, 5]
      tuple(range(6))
[52] V 0.0s
   (0, 1, 2, 3, 4, 5)
```



## ⊸ 규칙적인 반복





## → for \_ in range





## → while





## ⊸ break



## **魦** 바로 반복 종료 후 다음 문장으로

```
# %% break
   i = 1
   while i <= 10:
       print(f'6 * {i} = {6*i}')
       if i == 3:
           break
       i = i + 1
                                                                            Python
6 * 2 = 12
6 * 3 = 18
```



## **⊸** continue





#### ⊸ 7단 출력



☑ 반복되는 값이 저장된 변수



#### ⊸ 구구단 출력

## **❷** 제어변수 i, j

```
# %%
for i in range(2, 10):
    for j in range(1, 10):
        print(f'{i} * {j} = {i*j:2d} ', end=' ')
    print()
                                                                        Python
```

## SUMMARY

# 학습정긴





•••



- >> if
- if else
- if elif

#### 🌣 반복 for in

- >> for var in iterable:
  - > statement

#### 🌣 반복 while

- > while condition:
  - > statement







•••

## ☼ 반복에서 break, continue

- break
  - > 반복을 종료
- > continue
  - > 다음 반복으로 이동



