

원본 링크 : <https://velog.io/@suminwoooo/파이썬-프로그래밍-기초-자료형2>

3. 제어문

3-1. IF문

- 프로그래밍에서는 주어진 조건을 판단한 후 그 상황에 맞게 처리해야 할 경우가 생긴다.
- 이렇듯 프로그래밍에서 조건을 판단하여 해당 조건에 맞는 상황을 수행하는 데 쓰는 것이 바로 if문이다.
- 기본 구조

```
if 조건문: (필수)
    수행할 문장1
    수행할 문장2

elif 조건문 : (선택)
    수행할 문장1
    수행할 문장2

else: (선택)
    수행할 문장1
    수행할 문장2
```

- 기본 if문 예시

```
a = 1
if a==1:
    print("정답")
else:
    print("오답")
# 정답

a = True
if a: #(is True 생략)
    print("a is True")
else:
    print("a is False")
# a is True
```

- 이처럼 조건식은 다양한 경우에 활용할 수 있다.
- 조건식에 유용하게 활용 가능한 내용
 - x or y : x와 y 둘중에 하나만 참이어도 참

- `x and y` : `x`와 `y` 모두 참이어야 참
- `not x` : `x` 가 거짓이면 참
- `x in 리스트` : `x not in 리스트`
- `x in 튜플` : `x not in 튜플`
- `x in 문자열` : `x not in 문자열`

- 활용 if문 예시

```
# 1
money = True
test = 1
if (money==True) | (1 in [2,3]):
    print("택시를 타고 가라")
else:
    print("걸어 가라")
# 택시를 타고 가라

# 2
money = 2000
card = True
if (money >= 3000) | card:
    print("택시를 타고 가라")
else:
    print("걸어가라")
# 택시를 타고 가라

# 3
pocket = ['paper', 'cellphone']
card = False
if 'money' in pocket:
    print("택시를 타고가라")
elif card:
    print("택시를 타고가라")
else:
    print("걸어가라")
# 걸어가라
```

3-2. FOR문

- for 문은 리스트와 같은 시퀀스(sequence)를 이용해서 원하는 명령을 반복할 때 사용한다.
- 기본 구조

```
for 변수 in 리스트(또는 튜플, 문자열):
    수행할 문장1
    수행할 문장2
```

- 기본 for문 예시

```
test_lst = [1,2,3,4]
for i in test_lst:
    print(i)
# 1
# 2
# 3
# 4
```

- 활용 for문 예시

```
# 1
marks = [90, 25, 67, 45, 80]

number = 1
for mark in marks:
    print('mark:',mark, ' / number:', number)
    number = number +1

# mark: 90 / number: 1
# mark: 25 / number: 2
# mark: 67 / number: 3
# mark: 45 / number: 4
# mark: 80 / number: 5

# 2
marks = [90, 25, 67, 45]

number = 0
for mark in marks:
    print('mark:',mark)
    number = number +1
    print('number:', number)
    if mark >= 60:
        print("%d번 학생은 합격입니다." % number)
    else:
        print("%d번 학생은 불합격입니다." % number)

# mark: 90
# number: 1
# 1번 학생은 합격입니다.
# mark: 25
# number: 2
# 2번 학생은 불합격입니다.
# mark: 67
# number: 3
# 3번 학생은 합격입니다.
# mark: 45
# number: 4
# 4번 학생은 불합격입니다.

# 3
```

```

add = 0
for i in range(1, 11):
    add = add + i
    # add += i
print(add)
# 55

# 4
for i in range(2, 10):      # ①번 for문
    for j in range(1, 10):  # ②번 for문
        print(i*j, end=" ")
    print('')
# 2 4 6 8 10 12 14 16 18
# 3 6 9 12 15 18 21 24 27
# 4 8 12 16 20 24 28 32 36
# 5 10 15 20 25 30 35 40 45
# 6 12 18 24 30 36 42 48 54
# 7 14 21 28 35 42 49 56 63
# 8 16 24 32 40 48 56 64 72
# 9 18 27 36 45 54 63 72 81

# 5
a = [1, 2, 3, 4]
result = []
for num in a:
    result.append(num*3)
print(result)
# [3, 6, 9, 12]

result_test = [num * 3 for num in a]
print(result_test)
# [3, 6, 9, 12]

# 6
result_test2 = [num * 3 for num in a if num % 2 == 0]
print(result_test2)
# [6, 12]

result_test = []
for num in a:
    if num%2==0:
        result_test.append(num*3)
print(result_test)
# [6, 12]

```

3-3. WHILE문

- 반복해서 문장을 수행해야 할 경우 while문을 사용한다.
- 기본 구조

```
while <조건문>:
    <수행할 문장1>
    <수행할 문장2>
    <수행할 문장3>
```

- 기본 while문 예시

```
num = 1
while num <= 100:
    # print(num)
    num = num + 1
print(num)
# 101
```

- 활용 for문 예시

```
# 1
coffee = 10
money = 300
while money:
    print("돈을 받았으니 커피를 줍니다.")
    coffee = coffee - 1
    money = money + 50
    print("남은 커피의 양은 %d개입니다." % coffee)
    if coffee == 0:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
    elif money >= 5000:
        print('할당액끝')
        break
```

```
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 9개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 8개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 7개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 6개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 5개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 4개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 3개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 2개입니다.
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 1개입니다.
```

```
# 돈을 받았으니 커피를 줍니다.
# 남은 커피의 양은 0개입니다.
# 커피가 다 떨어졌습니다. 판매를 중지합니다.
```

- 참고 자료

- <https://wikidocs.net/book/1>- 라이브러리(library)란 다른 프로그램의 구성 요소로 사용하기 위해 미리 만들어 둔 프로그램 조각으로 볼 수 있습니다. 자주 사용하는 기능을 모듈·패키지로 만들어 두었습니다. 한 번 만들어 놓으면 여러 프로그램에서 재사용할 수 있고, 다른 사람과 공유할 수도 있습니다.

- 공식문에서는 파이썬 표준 라이브리에 대해 아래의 같은 내용이 적혀있습니다.

<파이썬 표준 라이브러리>

파이썬 언어 레퍼런스는 파이썬 언어의 정확한 문법과 의미를 설명하고 있지만, 이 라이브러리 레퍼런스 설명서는 파이썬과 함께 배포되는 표준 라이브러리를 설명합니다. 또한, 파이썬 배포판에 일반적으로 포함되어있는 선택적 구성 요소 중 일부를 설명합니다.

파이썬의 표준 라이브러리는 매우 광범위하며, 아래 나열된 긴 목록에 표시된 대로 다양한 기능을 제공합니다. 라이브러리에는 일상적인 프로그래밍에서 발생하는 많은 문제에 대한 표준적인 해결책을 제공하는 파이썬으로 작성된 모듈뿐만 아니라, 파일 I/O와 같은 시스템 기능에 액세스하는 (C로 작성된) 내장 모듈들이 포함됩니다 (이 모듈들이 없다면 파이썬 프로그래머가 액세스할 방법은 없습니다). 이 모듈 중 일부는 플랫폼 관련 사항을 플랫폼 중립적인 API들로 추상화시킴으로써, 파이썬 프로그램의 이식성을 권장하고 개선하도록 명시적으로 설계되었습니다.

- 처음 접한다면, 위의 의미가 어떠한 내용인지 이해하기 쉽지 않습니다.
- 점프 투 파이썬에 라이브러리에 대해 쉽게 설명을 해두었습니다.

전 세계의 파이썬 사용자들이 만든 유용한 프로그램을 모아 놓은 것이 바로 파이썬 라이브러리이다.
"라이브러리"는 "도서관"이라는 뜻 그대로 원하는 정보를 찾아보는 곳이다.
모든 라이브러리를 다 알 필요는 없고 어떤 일을 할 때 어떤 라이브러리를 사용해야 한다는 정도만 알면 된다. 그러기 위해 어떤 라이브러리가 존재하고 어떻게 사용하는지 알아야 한다.
자주 사용되고 꼭 알아 두면 좋은 라이브러리를 중심으로 하나씩 살펴보자.

- 데이터 분석을 하기 위해서는 pandas, numpy, matplotlib 정도만 이해한다면 큰 어려움은 없습니다. 분석을 하다보면 필요한 기능이 생기는데 그때마다 검색을 통해 확인할 수 있습니다.
- 다만, 머신러닝, 딥러닝의 경우 알아야하는 라이브러리가 많기 때문에 필요한 부분을 학습하면 됩니다.
- 참고 자료

- <https://wikidocs.net/book/1>
 - <https://docs.python.org/>