

CHAPTER_02_Control Flow

01 if

- 조건문 : 우리가 다양한 시나리오를 처리할 수 있는 코드를 작성할 수 있게 해준다.
- 조건문을 배우면 다양한 상황을 다룰 수 있는 더 복잡하고 유용한 프로그램을 작성할 수 있다.
- if, else, elif는 모두 조건부로 코드를 실행할 수 있도록 도와준다.

조건문 예시

ATM 에 내 카드를 넣으면 비밀번호를 요구한다. ATM 코드에는 만약 비밀번호가 정확 하다면, 사용자에게 돈을 주고 그렇지 않다면 사용자에게 에러를 보여준다.

```
# ATM 비밀번호 설정
from re import _FlagType

correct_password = "1234"

# 사용자로부터 입력 받은 비밀번호
user_password = input("ATM 비밀번호를 입력하세요: ")

# 비밀번호 확인
if user_password == correct_password:
    print("비밀번호가 정확합니다. 돈을 지급합니다.")
    # 여기에 돈을 지급하는 코드를 추가할 수 있습니다.
else:
    print("비밀번호가 정확하지 않습니다. 에러를 표시합니다.")
```

02 Else & Elif

- elif, else는 모두 꼭 항상 사용하지 않아도 된다.
- if, else, elif는 모두 조건부로 코드를 실행할 수 있도록 도와준다.

example 1

```
password_correct = True
# password_correct = False

if password_correct:
    print("here is your money")
    # else블럭에는 if의 조건이 false일때 실행될 코드가 될 것이다.
else:
    print("wrong password")
```

example 2

```
winner = 10

if winner > 10:
    print("winner is greater than 10")
# elif는 코드에 또 다른 대안과 조건을 넣을 수 있도록 해준다.
elif winner < 10:
    print("winner is less than 10")
# else는 조건이 true가 아닐 때, 즉 오직 대안만 제공할 수 있도록 해준다.
# else는 모든 경우가 false일 때 실행된다.
else:
    print("winner is 10")
```

03 And & Or

- `build-in` 함수 (=> 이미 준비되어 있는 함수)

```
age = input("how old are you?")
```

```
간혹재/비공-화본/python/02_control_flow/01_if.py
how old are you?
```

- `type` 함수 (=> 변수의 type을 알려주는 함수)

```
type(type(age))
```

- 만약 타입을 변경하고 싶다면?

```
your_age = int(input("how old are you?"))
```

`int`는 `user`가 작성한 `string`을 받고 `int`는 그 `string`을 숫자로 바꿔준다.

음주 나이 계산기 (1)

```
# 18세 미만이면 술을 마실 수 없다.
if your_age < 18:
    print("you can't drink")
# 그런데 만약 user가 18세 이상이고 and 동시에 35세 미만인 경우
# and 사용은 양쪽(앞, 뒤) 모두 true 이어야 한다.
elif your_age >= 18 and your_age <= 35:
    print("you drink beer!")
else:
    print("go ahead!")
```

음주 나이 계산기 (2)

```

if your_age < 18:
    print("you can't drink")
elif your_age >= 18 and your_age <= 35:
    print("you drink beer!")
# or는 하나의 true만 있어도 된다.
elif your_age == 60 or your_age == 70:
    print("birthday party!")
else:
    print("go ahead!")

```

🍷 And & Or 🍷

```

True and True == True
False and True == False
True and False == False
False and False == False

```

```

True or True == True
True or False == True
False or True == True
False or False == False

```

04 Python Standard Library

🔊 Python 카지노 만들기

: 컴퓨터가 숫자 하나를 선택하고 user도 숫자 하나를 선택할 것

: user가 숫자를 정확하게 맞췄다면 이기고, 아니면 지는 것

```

from random import randint

user_choice = int(input("choose number:"))
pc_choice = randint(1, 50)

if user_choice == pc_choice:
    print("you won!")
elif user_choice > pc_choice:
    print("lower! Computer chose", pc_choice)
elif user_choice < pc_choice:
    print("higher!")

```

- 주석 포함 코드

```

from random import randint
# input 함수는 사용자에게 input을 요청하고, 사용자가 키보드로
# 입력한 문자열을 반환할 것이다.
user_choice = int(input("choose number:"))
# pc_choice = 50 을 랜덤 숫자로 만들어주기(function을 module에서 가져오기)
pc_choice = randint(1, 50)

```

```

if user_choice == pc_choice:
    print("you won!")
elif user_choice > pc_choice:
    print("lower! Computer chose", pc_choice)
elif user_choice < pc_choice:
    print("higher!")

```

■ 공식 문서 참고하기

```
random.randint(a, b)
```

: 이 function 은 random 한 정수 n 을 반환한다.

: 첫번째 parameter a 은 n 보다 작거나 같고, n 은 두번째 parameter b 보다 작거나 같다.

05 While

- while 은 멈추지 않고 계속 동작한다.
- if 와 차이점 : while 은 조건문이 false 가 될때까지 코드를 계속 실행한다.
- if 는 한 번만 실행되고 멈춘다.

```

distance = 0

while distance < 20:
    print("I'm running:", distance, "km")
    distance = distance + 1

```

06 Python Casino

user 가 이길때까지 반복하는 코드

```

from random import randint

print("welcome to python casino")
pc_choice = randint(1, 50)

playing = True

# while 조건문 부분(playing) 이 True 일때만 동작한다
while playing:
    user_choice = int(input("choose number:"))
    if user_choice == pc_choice:
        print("you won!")
        # user가 이긴다면 게임은 끝
        playing = False
    elif user_choice > pc_choice:

```

```
print("lower! Computer chose", pc_choice)
elif user_choice < pc_choice:
    print("higher!")
```

🔗 Recap

- `from random import randint` => 모듈안에 있는 함수를 가져오기 위해 작성
- 선택한 모듈 : `random`
- `random` 은 많은 함수들을 가져올 수 있다.
- 그중에서 `randint` 라고 불리는 것을 가져왔고 이것은 랜덤하게 정수를 만들어낸다
- `randint(1, 50)`

`argument` 2개를 넣어 함수를 사용

이 함수는 a보다 크고, b보다는 작은 숫자를 만들어 준다

- `while` 은 계속 반복하고 `if` 는 한번만 실행이된다