

03 제어문

03-1 _ if 문

- 조건을 판단하여 해당 조건에 맞는 상황 수행

```
if 조건문:
    수행할 문장1
    수행할 문장2
    ....
else:
    수행할 문장A
    수행할 문장B
    ....
```

조건문을 테스트해서 참이면 if 문 바로 다음 문장(if블록)들을 수행하고, 조건문이 거짓이면 else문 다음 문장(else블록)들을 수행. **else문은 if문 없이 독립적으로 사용할 수 없다.**

들여쓰기

- if조건문: 바로 아래 문장부터 if 문에 속하는 모든 문장에 들여쓰기 적용
- 들여쓰기는 언제나 같은 너비로 해야함
- 조건문 뒤에는 꼭 콜론(:) 붙이기

조건문이란?

- if조건문에서 '조건문'이란 참과 거짓을 판단하는 문장
- 비교 연산자

비교 연산자	설명
$x < y$	x가 y보다 작다
$x > y$	x가 y보다 크다
$x == y$	x와 y가 같다
$x != y$	x와 y가 같지 않다
$x >= y$	x가 y보다 크거나 같다
$x <= y$	x가 y보다 작거나 같다

- 연산자 and, or, not

연산자	설명
$x \text{ or } y$	x와 y들 중에 하나만 참이면 참이다
$x \text{ and } y$	x와 y 모두 참이어야 한다
$\text{not } x$	x가 거짓이면 참이다

- x in s, x not in s

in	not in
x in 리스트	x not in 리스트
x in 튜플	x not in 튜플
x in 문자열	x not in 문자열

다양한 조건을 판단하는 elif

elif는 개수에 제한 없이 사용 가능

- 조건부 표현식

```
if score >= 60:
    message = "success"
else:
    message = "failure"
```

조건부 표현식 적용,

```
message = "success" if score >=60 else "failure"
```

- 조건문이 참인 경우 if 조건문 else 조건문이 거짓인 경우

03-2 while문

while문의 기본 구조

반복해서 문장을 수행해야 할 경우 while 문 사용, **반복문** 임

- 조건이 참인 동안에 while문 아래의 문장이 반복해서 수행됨

while문 만들기

```
prompt="""
1. Add
2. Del
3. List
4. Quit

Enter number: """
```

```
number=0
while number != 4:
    print(prompt)
    number=int(input())
```

1. Add
2. Del
3. List
4. Quit

Enter number:

while문 강제로 빠져나가기

- break문 사용

```
coffee=10
while True:
    money=int(input("돈을 넣어주세요: "))
    if money == 300:
        print("커피를 줍니다")
        coffee = coffee - 1
    elif money > 300:
        print("거스름돈 %d를 주고 커피를 줍니다." % (money -300))
        coffee = coffee - 1
    else:
        print("돈을 다시 돌려주고 커피를 주지 않습니다.")
        print("남은 커피의 양은 %d개 입니다." % coffee)
    if coffee == 0:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
```

while문의 맨 처음으로 돌아가기

- continue문 사용

```
a=0
while a<10:
    a=a+1
    if a%2 ==0 : continue
    print(a)
```

무한 루프

- 무한루프는 while문으로 구현
-

03-3 for문

for문의 기본 구조

```
test_list = ['one', 'two', 'three']
for i in test_list:
    print(i)
```

for 변수 in 리스트(또는 튜플, 문자열):

수행할 문장1

수행할 문장2

....

for문과 continue문

for문 안의 문장을 수행하는 도중에 continue문을 만나면 for문의 처음으로 돌아감

```
# marks2.py
marks = [90, 25, 67, 45, 80]
number = 0
for mark in marks:
    number = number+1
    if mark < 60 : continue
    print("%d번 학생 축하합니다. 합격입니다." % number)
```

for문과 함께 자주 사용하는 range 함수

- range함수: 숫자 리스트를 자동으로 만들어줌

```
a= range(10)
a
range(0, 10)
```

range(10): 0부터 10 미만의 숫자를 포함하는 range객체

range(시작 숫자, 끝 숫자) * 끝 숫자는 포함되지 않음!

리스트 내포 사용하기

리스트 안에 for문을 포함하는 리스트 내포를 사용하면 좀 더 편리하고 직관적인 프로그램 만들 수 있음

```
a = [1,2,3,4]
result = [num*3 for num in a]
print(result)
[3,6,9,12]
```

- [표현식 for 항목 in 반복 가능 객체 if 조건]

연습문제

Q1.

shirt 출력

Q2.

```
result = 0
i=1
while i <= 1000:
    if i % 3 == 0:
        result += i
    i += 1

print(result)
```

Q3.

```
i = 0
while i <= 5:
    print("***i)
    i += 1
```

Q4.

```
a=0
for i in range(1,101):
    a += 1
    print(a)
```

Q5.

```
#for문 이용, 학급의 평균 점수 구하기

A = [70, 60, 55, 75, 95, 90, 80, 85, 100]
number = 0
for score in A:
    number += score
average=number/10
print(average)
```

Q6.

```
numbers = [1, 2, 3, 4, 5]
result = [n*2 for n in numbers if n%2 ==1]
print(result)
```

