

Python

<지난주 while 명령어 설명>

```
coffee = 10
```

```
while True: → 무한 루프
```

```
    money = int(input("돈을 넣어 주세요: "))
```

```
    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 중지 합니다.")
```

```
        break → 무한 루프 빠져나오는 명령어
```

■ for 문

for 변수 in 리스트(또는 튜플, 문자열):

```
    수행할 문장1
```

```
    수행할 문장 2
```

```
...
```

```
a = [1,2,3,4,5]
```

```
a[-1]
```

```
5
```

```
len(a)
```

```
5
```

for(i=0; i<len(a); i++) → 간단하게 for i in a로 표현

```
test_list = ['one', 'two', 'three']
```

```
for i in test_list:
```

```
    print(i)
```

요소가 있는 동안만 값을 집어넣음

이중리스트를 활용한 for문

```
a = [(1,2), (3,4), (5,6)]
```

```
for i, j in a:
```

```
    print(i, j)
```

```
a = [(1,2), (3,4), (5,6)]
```

```
for i in a:
```

```
    print(i, "-->", type(i))
```

※ tuple의 요소값과 일반 index 값의 비교

```
(first, last) = (1,2)
```

```
first
```

```
1
```

```
last
```

```
2
```

```
first, last = (3,4)
```

```
first
```

```
3
```

```
last
```

```
4
```

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

```
number = 0
```

```
for mark in marks:
```

```
    number = number + 1 → number += 1로 쓸 수도 있음.
```

```
    if mark >= 60:
```

```
        print("%d번 학생은 합격입니다." % number)
```

```
    else:
```

```
        print("%d번 학생은 불합격입니다." % number)
```

for 변수 in list or tuple같은 자료형

number = number+1 → 학생번호(누적연산)

더미 데이터 방지로 number = 0이라는 초기값 부여.

```
a = []
```

a = [1,2,3,4,5] → 대입이 되어서 더미 데이터가 소거됨

■for문과 continue

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

```
number = 0
```

```
for mark in marks:
```

```
    number += 1
```

```
    if mark < 60:
```

```
        continue
    print("%d번 학생은 합격입니다." % number)
```

■ range 함수

```
a = range(10)
```

a

range(0, 10) → $0 < a < 10$ 의 범위를 지정할 때 사용함

```
a = [1,2,3,4,5,6,7,8,9,10]
```

```
b = range(1,11)
```

a

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

b

```
range(1, 11)
```

```
a = [1,2,3,4,5,6,7,8,9,10]
```

```
b = range(1,11)
```

```
for i in a:
```

```
    print(i)
```

```
for i in b:
```

```
    print(i)
```

→ print할 시 1 2 3 4 5 6 7 8 9 10이 나옴.

b값 변수를 안주고 range(1, 11)을 넣어도 값이 똑같이 나옴.

★ print("apple") → print를 하면 기본적으로 개행문자가 있어서 엔터키 친 것 같이 다음줄로 내려감

print(i, end= ' ')로 공백을 넣은상태에서 입력이 되기 때문에 1 2 3 ...형태로 됨

print() -> 개행문자 형태로 엔터키 친 것(줄바꿈)과 같음

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

```
for number in range(len(marks)):
```

```
    if marks[number] < 60:
```

```
        continue
```

```
    print("%d번 학생 축하합니다. 합격입니다." % (number+1))
```

index값을 이용해서 작업을 함.

■ range를 이용해서 구구단 만들기

```
for i in range(2,10):
```

```
for j in range(1,10):
    print(i,"*", j, "=", i*j, end=' ')
print()
```

■ 리스트 내포

```
a=[1, 2, 3, 4]
result=[]
for(num*3) num in a:
    result.append
print(result)
```

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

→ 리스트 안에 결과 값을 넣기 때문에 리스트 내포라고 함

```
a=[1, 2, 3, 4]
result=[num * 3 for num in a if num % 2 ==0]

print(result)
```

<연습 문제>

Q1 다음 코드의 결과값은 무엇일까?

```
# a = "Life is too short, you need python"
# if "wife" in a: print("wife")
# elif "python" in a and "you" not in a: print("python")
# elif "shirt" not in a: print("shirt")
# elif "need" in a: print("need")
# else: print("none")
```

-> shirt라는 결과값이 나옴

첫 번째 조건: "wife"라는 단어는 a 문자열에 없으므로 거짓이다.

두 번째 조건: "python"이라는 단어는 a 문자열에 있지만 "you" 역시 a 문자열에 있으므로 거짓이다.

세 번째 조건: "shirt"라는 단어가 a 문자열에 없으므로 참이다.

네 번째 조건: "need"라는 단어가 a 문자열에 있으므로 참이다.

가장 먼저 참이 되는 것이 세 번째 조건이므로 "shirt"가 출력된다.

if 또는 else if(elif)는 먼저 참인 것을 우선으로 표시함

Q2 while문을 사용해 1부터 1000까지의 자연수 중 3의 배수의 합을 구해 보자.

```
# result = 0
# i = 1
# while i <= 1000:
#     if i % 3 == 0: -> 3으로 나누어 떨어지는 수는 3의 배수
#         result += i
#     i += 1
# print(result) -> 166833
```

Q3 while문을 사용하여 다음과 같이 별(*)을 표시하는 프로그램을 작성해 보자.

```
# i = 0
# while True: -> while문 수행 시 1씩 증가
#     i += 1
#     if i > 5: break -> i 값이 5보다 크면 while문을 벗어난다.
#     print('*' * i) -> i 값 개수만큼 *를 출력한다.
```

Q4 for문을 사용해 1부터 100까지의 숫자를 출력해 보자.

```
for i in range(1, 101):
#     print(i, end=' ')
# print()
```

Q5

A 학급에 총 10명의 학생이 있다. 이 학생들의 중간고사 점수는 다음과 같다.

[70, 60, 55, 75, 95, 90, 80, 80, 85, 100]

for문을 사용하여 A 학급의 평균 점수를 구해 보자.

```
# A = [70, 60, 55, 75, 95, 90, 80, 80, 85, 100]
# total = 0
# for score in A:
#     total += score -> A학급의 점수를 모두 더한다.
# average = total/(len(A)) -> 평균을 구하기 위해 총 점수를 총 학생 수로 나눈다.
# print(average) -> 평균 79가 출력된다.
```

Q6

리스트 중에서 홀수에만 2를 곱하여 저장하는 다음 코드가 있다.

```
# numbers = [1, 2, 3, 4, 5]
# result = []
# for n in numbers:
#     if n % 2 == 1:
#         result.append(n*2)
# 위 코드를 리스트 내포(list comprehension)를 사용하여 표현해 보자.
```

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