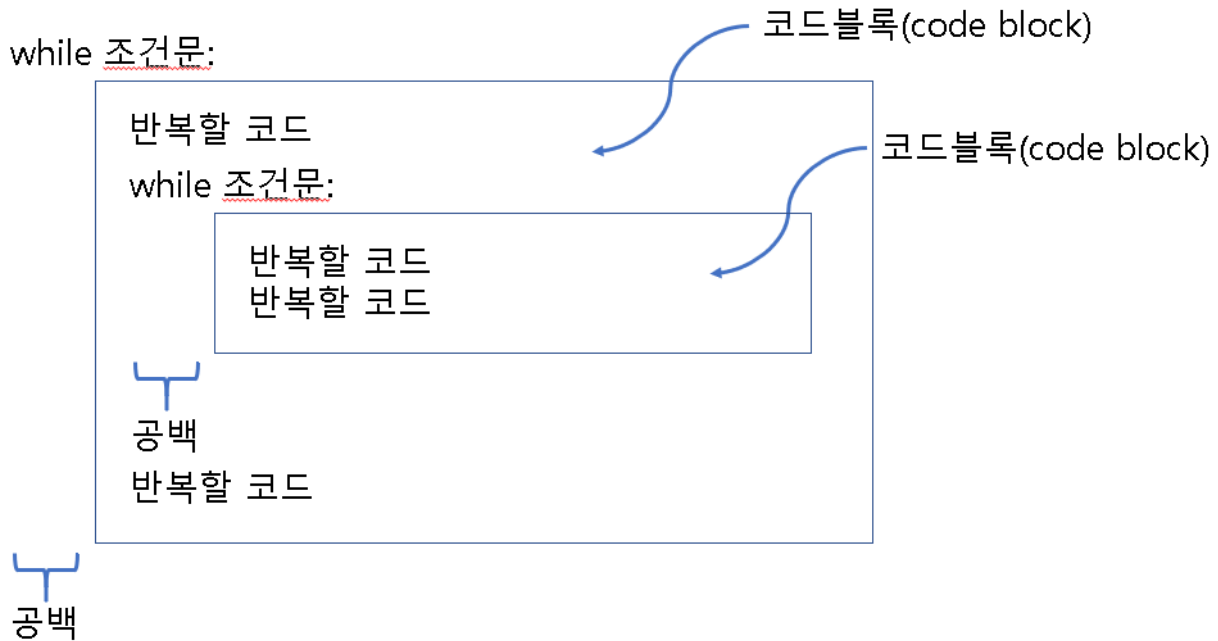


Day 17. 반복문 안에 반복문

반복문 안에 조건문을 넣을 수 있었던 것처럼, 반복문 안에 반복문을 넣을 수도 있습니다.

아래의 구조는 while반복문안에 while반복문을 가지도록 한 구조를 표현하고 있습니다. while반복문 대신에 for in range 반복문이 사용될 수도 있습니다.



반복문 안에 반복문은 언제 사용할까요?

예를 들어 구구단을 출력한다고 생각해 보도록 하겠습니다. 구구단은 1단부터 9단까지 있습니다.

즉 1부터 9까지 반복을 해야하는 거죠. 그리고 각 단별로 1부터 9까지 곱한 결과가 나와야 합니다.

1단도 9번, 2단도 9번, 3단도 9번 9단도 9번 과 같은 식으로 결과를 출력합니다.

문제를 잘 살펴보면 반복 안에 반복이 있는 것을 알 수 있습니다. 이 경우 반복문 안에 반복문을 사용할 수 있습니다.

예제를 통해 알아보도록 하겠습니다.

File메뉴 - New File 메뉴를 선택합니다. 새로운 창이 뜨면 다음과 같은 코드를 입력합니다.

day17.py

```
dan = 1

while dan <= 9 :

    x = 1

    while x <= 9 :

        strDan = str(dan)

        strX = str(x)

        strValue = str(dan * x)

        print(strDan + " * " + strX + " = " + strValue)

        x = x + 1

    dan = dan + 1

print("end")
```

위의 코드를 실행하면 다음과 같은 결과가 나옵니다.

```
1 * 1 = 1

1 * 2 = 2

1 * 3 = 3

1 * 4 = 4

1 * 5 = 5
```

```
1 * 6 = 6
```

```
1 * 7 = 7
```

```
1 * 8 = 8
```

```
1 * 9 = 9
```

```
2 * 1 = 2
```

```
2 * 2 = 4
```

```
2 * 3 = 6
```

```
2 * 4 = 8
```

```
.....
```

```
9 * 6 = 54
```

```
9 * 7 = 63
```

```
9 * 8 = 72
```

```
9 * 9 = 81
```

```
end
```

코드가 좀 복잡해졌습니다. 일단 반복문안에 반복문과 같이 이중으로 되어 있는 코드를 작성할 때는 한번에 모든 것을 작성하기 보다는 바깥쪽의 반복문부터 작성하는 것이 좋습니다.

아래의 코드를 살펴보도록 하겠습니다.

```
dan = 1
```

```
while dan <= 9 :
```

```
print(dan)

dan = dan + 1

print("end")
```

위의 코드를 실행하면 dan 만 출력됩니다. 1부터 9까지 출력되는 코드죠.

print(dan) 이 있는 부분에서는 dan별로 1부터 9까지의 수를 곱한 결과를 출력하도록 바꿔줘야 합니다.

```
x = 1
```

```
while x <= 9 :
```

```
    strDan = str(dan)
```

```
    strX = str(x)
```

```
    strValue = str(dan * x)
```

```
    print(strDan + " * " + strX + " = " + strValue)
```

```
    x = x + 1
```

위의 코드는 x가 1부터 9까지 반복되면서 dan과 x를 곱한 값을 출력하는 코드입니다. 곱셈 식과 그 결과를 문자열로 만들어 출력하기 위해서 str()함수를 이용하여 숫자를 문자로 바꾼 것을 알 수 있습니다.

속제)

소스코드를 지우고, 스스로 구구단을 출력해 보세요. 잘 기억이 안난다면 책의 내용을 살짝 살짝 보셔도 됩니다. 스스로 구구단을 작성해 봤다면, while문이 아니라 for in range를 이용하여 구구단을 출력하도록 코드를 작성해보세요.