

Lesson 8

반복문



목차

1. 반복문
 - while
 - for
2. range() 함수

체중이 10kg이 빠질 때까지 운동을 한다.

반복문(while)

조건식이 참인 동안 계속 실행되는 반복문.

while 조건식:
반복할 명령문

반복해야 하는 횟수나 값의 범위가 명확하지 않은 경우에 주로 사용함.

반복문(while문)

코드

```
# "안녕"을 세 번 출력  
cnt=0  
while cnt<3:  
    print("안녕")  
    cnt+=1
```

안녕
안녕
안녕

코드

```
# "안녕"을 무한 반복 출력  
while True:  
    print("안녕")
```

안녕
안녕
안녕
...



숫자 0, 빈 문자열(""), None, 빈 리스트([]) 등 값이 없는 것은 False

코드

리스트에서 1을 모두 삭제하기

```
li=[1,2,1,1,3,1]
```

```
while 1 in li:
```

```
    li.remove(1)
```

```
print(li)
```

[2,3]

도전!

[문제] 중복되지 않은 5개의 정수 입력받아 출력하기.

```
정수 입력: 1  
정수 입력: 1  
정수 입력: 2  
정수 입력: 3  
정수 입력: 4  
정수 입력: 4  
정수 입력: 5  
{1,2,3,4,5}
```

운동장을 10바퀴 돈다.

반복문(for)

for 변수 **in** 반복 가능한 객체:
반복할 명령문

값의 범위나 횟수가 정해져 있을 때 주로 사용.

코드 `for char in "행복해":
 print(char)`

행
복
해

코드 `for char in enumerate("행복해"):
 print(char)`

(0, "행")
(1, "복")
(2, "해")

1 `for char in "행복해":
 print(char)`

2 `for char in "행복해":
 print(char)`

3 `for char in "행복해":
 print(char)`

enumerate() 함수

- 반복 가능한 객체와 함께 사용.
- 해당 요소의 인덱스와 요소를 튜플로 추출.
- 인덱스는 0부터 표시. 시작 번호 지정 가능.

■ 사용법: `for 변수 in enumerate(반복할 객체):`
반복할 명령문

`for 변수 in enumerate(반복할 객체, 시작 번호):`
반복할 명령문

코드 `for char in enumerate("행복해"):`
`print(char)`

```
(0, '행')
(1, '복')
(2, '해')
```

`for num, char in enumerate("행복해", 1):`
`print(num, char)`

```
1 행
2 복
3 해
```

코드

```
for item in ["가위", "바위", "보"]:  
    print(item)
```

가위
바위
보

코드

```
names=("kim","park","han")  
for name in names:  
    print(name)
```

```
kim  
park  
han
```

코드

```
for item in {"가위", "바위", "보"}:  
    print(item)
```

바위
가위
보



셋(set)은 nonsequence형이므로 실행할 때마다 결과가 다르다

코드

```
person={  
    "name": "은수정",  
    "age": 20  
}
```

```
for key in person:  
    print(key)
```

```
name  
age
```



key변수에는 person의 키가 전달된다.

```
for key in person:  
    print(person[key])
```

```
은수정  
20
```



딕셔너리의 값은 키로 얻을 수 있다.

range() 함수

■ 지정한 범위의 정수를 생성하는 함수.

■ 사용법 :

range(시작값, 종료값, 증감값)

range(종료값)

range(시작값, 종료값)



생성되는 값 => 시작값 ~ **종료값-1**

■ range() 함수 특징

- 시작값 생략시 0부터 시작한다.
- **종료값은 생략할 수 없다.**
- 증감값을 생략하면 1씩 증가한다
- range()함수의 결과는 자료형이 range형이다.



range()함수로 생성된 값을
확인하려면 list(), tuple()로 변경해야
볼 수 있다.

코드

```
print(range(5))
print(type(range(5)))
print(list(range(5)))
print(list(range(1,6)))
print(list(range(0,10,2)))
print(list(range(0,-6,-1)))
```

```
range(0, 5)
<class "range">
[0, 1, 2, 3, 4]
[1, 2, 3, 4, 5]
[0, 2, 4, 6, 8]
[0, -1, -2, -3, -4, -5]
```

for문과 range() 함수

코드

1~5까지 숫자 출력

```
for n in range(1,6):  
    print(n)
```

```
1  
2  
3  
4  
5
```

hello~를 10회 출력

```
for n in range(10):  
    print("hello~")
```

```
hello~  
hello~  
hello~  
hello~  
hello~  
hello~  
hello~  
hello~  
hello~  
hello~
```



변수를 사용하지 않으려면 _(언더바)를 사용하면 됨.

```
for _ in range(10):  
    print("hello~")
```

코드

```
# 1~100까지의 합계 출력
total=0
for n in range(1,101):
    total=total+n    # total+=n

print(f"1~100까지의 합 : {total}")
```

1~100까지의 합 : 5050

코드

입력한 단 출력

```
num=int(input("단 입력: "))
```

```
for i in range(1,10):
```

```
    print(f"{num} * {i} = {num*i}")
```

단 입력: 5

5 * 1 = 5

5 * 2 = 10

5 * 3 = 15

5 * 4 = 20

5 * 5 = 25

5 * 6 = 30

5 * 7 = 35

5 * 8 = 40

5 * 9 = 45

```
2*1= 2 3*1= 3
2*2= 4 3*2= 6
2*3= 6 3*3= 9
2*4= 8 3*4=12
2*5=10 3*5=15
```

코드

```
for i in range(1,6):
    for j in range(2,4):
        print(f"{j}*{i}={i*j:2}",end=' ')
    print()
```