반복문 : for, while 활용

백석대학교 강윤희

(문제풀이)반복 및 조건

- for와 함께 자주 사용하는 range함수

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

```
1번 학생 축하합니다. 합격입니다.
3번 학생 축하합니다. 합격입니다.
5번 학생 축하합니다. 합격입니다.
```

- 합격한 학생들의 평균과 불합격 학생들의 평균을 구하시오.

- 합격한 학생들의 평균과 불합격 학생들의 평균을 구하시오.

```
marks=[90,25,67,45,80]
s1=s2=0
cnt1=cnt2=0
for number in range(len(marks)):
    if marks[number] < 60:
        cnt2+=1
        s2+=marks[number]
        continue
    else:
        s1+= marks[number]
        cnt1+=1
        print("%d번 학생 축하합니다. 합격입니다." % (number+1))
print("합격한사람의 평균 ₩t%d" % (s1/cnt1))
print("불합격한사람의 평균 ₩t%d"%(s2/cnt2))
```

while문

조건 참인 동안에 while문 조건 문장을 반복해서 수행

while <조건>:

    <수행할 문장1>

    <수행할 문장2>

    <수행할 문장3>

        …

while문

조건 참인 동안에 while문 조건 문장을 반복해서 수행

while <조건>:

    <수행할 문장1>

    <수행할 문장2>

    <수행할 문장3>

        …

while문

```
treeHit = 0
while treeHit < 10:
    treeHit = treeHit + 1
    print("나무를 %d번 찍었습니다." % treeHit)
    if treeHit == 10:
        print("나무 넘어갑니다.")
```

In [1]:

```
treeHit = 0
while treeHit < 10:
    treeHit = treeHit + 1
    print("나무를 %d번 찍었습니다." % treeHit)
    if treeHit == 10:
        print("나무 넘어갑니다.")
```

```
나무를 1번 찍었습니다.
나무를 2번 찍었습니다.
나무를 3번 찍었습니다.
나무를 4번 찍었습니다.
나무를 5번 찍었습니다.
나무를 6번 찍었습니다.
나무를 7번 찍었습니다.
나무를 8번 찍었습니다.
나무를 9번 찍었습니다.
나무를 10번 찍었습니다.
나무 넘어갑니다.
```

while 문 빠져나오기

```
coffee = 5
money = 300
while money:
    print("돈을 받았으니 커피를 줍니다.")
    coffee = coffee -1
    print("남은 커피은 %d개입니다." % coffee)
    if coffee == 0:
        print("커피가 떨어져  판매를 중지합니다.")
        break
```

In [5]:
```
coffee = 5
money = 300
while money:
    print("돈을 받았으니 커피를 줍니다.")
    coffee = coffee -1
    print("남은 커피은 %d개입니다." % coffee)
    if coffee == 0:
        print("커피가 떨어져 판매를 중지합니다.")
        break
```

```
돈을 받았으니 커피를 줍니다.
남은 커피은 4개입니다.
돈을 받았으니 커피를 줍니다.
남은 커피은 3개입니다.
돈을 받았으니 커피를 줍니다.
남은 커피은 2개입니다.
돈을 받았으니 커피를 줍니다.
남은 커피은 1개입니다.
돈을 받았으니 커피를 줍니다.
남은 커피은 0개입니다.
커피가 떨어져 판매를 중지합니다.
```

```
# coffee.py
coffee = 10
while True:
    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
```

for문과 while문 상호 변환 가능

- for문과 while문은 기본적으로 유사하며, 서로 변환이 가능하다. 하지만 두 구문의 쓰임에는 차이가 있다. For문은 일반적으로 반복 횟수를 정확하게 알고 있고, 반복 횟수가 변하지 않을 때 사용한다. 반면, while문은 반복 실행 횟수가 명확하지 않고 어떤 조건을 만족하면 프로그램을 종료하고자 할 때 사용한다.
- 예를 들어, 학생들의 성적을 채점하는 프로그램을 작성한다고 하자. 이미 학생이 총 몇 명인지 명확하게 알고 있으므로 for문을 사용하는 것이 좋다. 하지만 가위바위보를 한다고 가정했을 때 '이기면 종료하라.'라는 조건을 주면 언제 이길지 모르므로 while문을 사용하는 것이 낫다.

```
for i in range(0.5):
    print (i)
```

(a) 반복 실행 횟수를 명확히 알 때

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

(b) 반복 실행 횟수가 명확하지 않을 때

[for문과 while문 상호 변환]

(프로그램) while 문

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

(프로그램) while 문

- (50점 이상의 총합) 다음은 A학급 학생의 점수를 나타내는 리스트이다. 다음 리스트에서 50점 이상의 점수들의 총합을 구하시오.

A = [20, 55, 67, 82, 45, 33, 90, 87, 100, 25]

(프로그램) 조건문과 반복문

- 입력된 문자열을 역순으로 출력하는 프로그램

```
sentence = "I love you"
reverse_sentence = ' '
for char in sentence:
    reverse_sentence = char + reverse_sentence
print(reverse_sentence)
```

Loop	reverse_sentence[1]	reverse_sentence[2]	char
0		I	I
1	I	I	
2	I	I I	I
3	I I	ol I	o
4	ol I	vol I	v
5	vol I	evol I	e
6	evol I	evol I	
7	evol I	y evol I	y
8	y evol I	oy evol I	o
9	oy evol I	uoy evol I	u

(프로그램) 조건문과 반복문

- 십진수를 이진수로 변환하는 프로그램

```
decimal = 10
result = ' '
while (decimal > 0):
    remainder = decimal % 2
    decimal = decimal // 2
    result = str(remainder) + result
print(result)
```

- 십진수 숫자를 2로 계속 나눈 후, 그 나머지를 역순으로 취하면 이진수가 된다.