세종사이버대학교

파이썬기초

3주차 - 3교시

반복문

학습내용

- 반복문의 이해와 종류
- 반복문의 활용

학습목표

- 파이썬에서 제공하는 반복문의 특징과 종류를 알아보고 문법에 맞게 코드를 작성할 수 있다.
- 반복문을 이용하여 반복 처리가 필요한 상황에서 반복문을 선택해서 코드를 작성할 수 있다.

생각해 봅시다

똑같은 처리를 여러 번 해야 할
상황에 대해 생각해보고 어떻게 처리하면
좋을지에 대해 생각해봅시다.

01

반복문의 이해와 종류

1 | 반복문의 이해와 종류

01 문장의 반복

“ 같은 코드와 같은 명령을
반복적으로 실행해야 한다면
어떻게 하면 좋을까?! ”

1~1000까지의 숫자를 출력하거나
혹은 같은 처리를 1만 번 해야 한다면?

1 | 반복문의 이해와 종류

01 문장의 반복

- 파이썬은 이처럼 같은 작업을 **반복처리** 할 수 있는 반목문을 제공함

> 반복문의 종류 : while 문, for 문

2 | While 문

01 while 문

while expression

반복 실행할 코드

- while 반복문은 **expression 조건이 참**인 동안 실행함
- 반복 실행할 횟수를 모르는 경우 **무한 반복**으로 처리함
- while 반복의 명령문은 **들여쓰기**로 지정함

- 반복 실행하는 명령문은 들여쓰기 될 때 시작되고 들여쓰기가 되지 않은 명령문을 만나면 종료됨

2 | While 문

01 while 문

In [4] :
```
num = 0
while num < 5 :
        print(num)
        num += 1
print('프로그램을 종료합니다.')
```

결과(Console)
```
0
1
2
3
4
프로그램을 종료합니다.
```

2 | While 문

02 break

break

반복문 밖으로 벗어나도록 하는 데 사용되는 키워드

- 중첩 반복문은 내부 반복문 안에 break가 있는 경우 내부 반복문을 벗어남

break

무한 반복 사용시 특정 조건에 대해 반복을 벗어나야 하는 경우에 사용

2 | While 문

02 break

In [5] :

```
#무한반복에서 특정 조건에서 반복문 탈출
num = 0
while True :
    if num > 4 :
        break
    print(num)
    num += 1
print('프로그램을 종료합니다.')
```

결과(Console)

```
0
1
2
3
4
프로그램을 종료합니다.
```

2 | While 문

03 continue

continue

특정 조건에서 아래 처리 구문을 실행시키지 않도록
처리할 때 사용

- 진행되는 반복 구문에서 **continue**를 만나면
  그 아래 구문은 반복되지 않고 조건식으로 이동

2 | While 문

03 continue

In [6] :

```
#1~10 사이의 숫자 중 홀수만 출력
num = 1
while num <= 10 :
        if num%2==0:
                num += 1
                continue
        print(num)
        num += 1
print('프로그램을 종료합니다.')
```

결과(Console)

```
1
3
5
7
9
프로그램을 종료합니다.
```

3 | For 문

01 for 문

for value in sequence

반복 블록

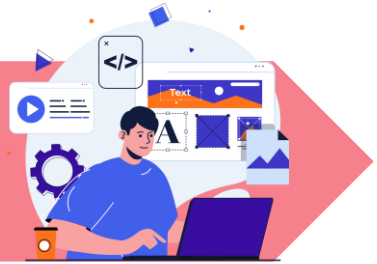
- **특정 횟수만큼 반복하고 싶을 때** for 반복문을 사용함
- **range() 함수**를 사용하여 반복 횟수를 지정하는 방법을 많이 사용함

3 | For 문

01 for 문

- 일반적으로 리스트, 튜플 또는 문자열과 같은 **반복 가능한 객체**에서 사용

for 반복문
반복 블록을 정의하고 해당 코드가 지정한 조건에 맞을 때 블록을 실행



3 | For 문

02 range ( ) 함수

- 순차적인 정수의 범위를 나타내기 위해 사용

함수	사용 예
range(x)	0부터 x-1까지 정수의 순차적 범위
range(x,y)	x부터 y-1까지 정수의 순차적 범위
range(x,y,z)	x부터 y-1까지 z씩 증가하는 정수의 순차적 범위
range(x,y,-z)	x부터 y+1까지 z씩 감소하는 정수의 순차적 범위

3 | For 문

02 range ( ) 함수

In [12] :
```
range(5)  # 0 ~ 5-1 순차적 범위의 정수
```

out [12] :
```
range(0, 5)
```

In [13] :
```
for i in range(5):
    print(i)
```

```
0
1
2
3
4
```

3 | For 문

02 range ( ) 함수

In [16] :
```
range(2, 5)  # 1 ~ 5-1 순차적 범위의 정수
```

out [16] :
```
range(2, 5)
```

In [17] :
```
for i in range(2, 5):
    print(i)
```

```
2
3
4
```

3 | For 문

02 range ( ) 함수

```
In [20] : range(1, 10, 2)  # 1 ~ 10-1 사이의 2씩 증가한 순차적 범위의 정숫값
out [20] : range(1, 10, 2)
```

```
In [20] : for i in range(1, 10, 2):
              print(i)
```

```
1
3
5
7
9
```

3 | For 문

02 range ( ) 함수

In [20] :
```
range(10, 1, -2)  # 10 ~ 1+1 사이의 2씩 감소한 순차적 범위의 정숫값
```

out [20] :
```
range(10, 1, -2)
```

In [20] :
```
for i in range(10, 1, -2):
    print(i)
```

```
10
8
6
4
2
```

4 | 시퀀스 자료형과 반복문

01 시퀀스 자료형

시퀀스 자료형(sequence types)

리스트, 튜플, range, 문자열처럼 **값이 연속적**으로 이어진 자료형

- list: [10, 20, 30, 40]
- tuple : (10, 20, 30, 40)
- range : range()
- str : '안녕하세요

4 | 시퀀스 자료형과 반복문

02 문자열과 반복문

In [26] :
```
str = '안녕 하세요!'
for s in str:
print(s)
```

```
안
녕

하
세
요
!
```

4 | 시퀀스 자료형과 반복문

03 리스트와 반복문

In [27] :
```
list = [10, 20, 30, 40]
for item in list:
print(item)
```

```
10
20
30
40
```

4 | 시퀀스 자료형과 반복문

04 튜블과 반복문

In [28] :
```
tuple_temp = (10, 20, 30, 40)
for item in tuple_temp:
print(item)
```

```
10
20
30
40
```

02

반복문의 활용

1 | 코딩해보기

01 1 부터 100까지의 합을 구하는 프로그램 작성

In [33] :
```
sum = 0
for i in range(1,101):
        sum += i
print('1~100 까지의 합 :', sum)
```

```
1~100 까지의 합 : 5050
```

1 | 코딩해보기

01 1 부터 100까지의 합을 구하는 프로그램 작성

In [32] :
```
num = 1
sum = 0
while num<101:
        sum += num
        num += 1
print('1~100 까지의 합 :', sum)
```

```
1~100 까지의 합 : 5050
```

1 | 코딩해보기

02 for 문을 이용하여 1부터 10까지를 곱한 결과를 출력하는 프로그램 작성

In [34] :
```
result = 1
for i in range(1,11):
        result *= i
print('1~10 까지의 곱 :', result)
```

```
1~10 까지의 곱 : 3628800
```

1 | 코딩해보기

02 for 문을 이용하여 1부터 10까지를 곱한 결과를 출력하는 프로그램 작성

In [35] :

```
num = 1
result = 1
while num<11:
        result *= num
        num += 1
print('1~10 까지의 곱 :', result)
```

1~10 까지의 곱 : 3628800

1 | 코딩해보기

03 구구단의 짝수 단(2,4,6,8)만 출력하는 프로그램 작성

- 단, 2단은 2x2까지, 4단은 4x4까지 , 6단은 6x6까지, 8단은 8x8 까지 출력

In [41] :
```
for i in range(2,10,2):
    for j in range(1, i+1) :
        print(i ,"X", j, "=", i*j)
```

```
2 X 1 = 2
2 X 2 = 4
4 X 1 = 4
4 X 2 = 8
4 X 3 = 12
4 X 4 = 16
6 X 1 = 6
6 X 2 = 12
6 X 3 = 18
6 X 4 = 24
6 X 5 = 30
6 X 6 = 36
8 X 1 = 8
8 X 2 = 16
8 X 3 = 24
8 X 4 = 32
8 X 5 = 40
8 X 6 = 48
8 X 7 = 56
8 X 7 = 64
```

1 | 코딩해보기

04 while문을 무한 루프로 구성하여 작성하는 예제

- 1부터 시작해서 모든 짝수와 3의 배수를 더해서 그 합이 언제 1000이 넘어서는지, 그리고 1000이 넘어선 값은 얼마가 되는지 계산하여 출력

In [45] :

```
num = 1;
sum = 0;
while True:

    # 모든 짝수와 3의 배수를 찾고 더하기
    if num%2==1 or num%3==0 :
      sum += num
      #print(num, sum)
```

1 | 코딩해보기

04 while문을 무한 루프로 구성하여 작성하는 예제

- 1부터 시작해서 모든 짝수와 3의 배수를 더해서 그 합이 언제 1000이 넘어서는지, 그리고 1000이 넘어선 값은 얼마가 되는지 계산하여 출력

In [45] :

```
        if(sum > 1000) :
            break

        num += 1

print('마지막 숫자 ', num, '을 더했을 때,')
print('처음으로 1000을 넘은 숫자', sum, '이 되었습니다.')
```

```
마지막 숫자 55 을 더했을 때,
처음으로 1000을 넘은 숫자 1054 이 되었습니다.
```

Q1

산술 연산자 중 나머지를 구하는 연산자는?

1. %
2. /
3. //
4. **

Q1

산술 연산자 중 나머지를 구하는 연산자는?

1. %
2. /
3. //
4. **

정답

1번

해설

나머지를 구하는 연산자는 % 연산자입니다.

Q2

이 코드의 결과는?

보기

```
age = 20
print( age >= 19 )
```

1. None
2. False
3. True
4. null

Q2

이 코드의 결과는?

보기

```
age = 20
print( age >= 19 )
```

1. None
2. False
3. True
4. null

정답

3번

해설

비교연산의 결과는 논리값 (True / False) 값을 반환합니다.

Q3

for 반복문에 사용 가능한 sequence 자료형이 아닌 것은?

1. 리스트
2. tuple
3. 문자열
4. 숫자

Q3

for 반복문에 사용 가능한 sequence 자료형이 아닌 것은?

1. 리스트
2. tuple
3. 문자열
4. 숫자

정답

4번

해설

시퀀스 자료형의 종류에는 문자열, 리스트, 튜플, range()가 있습니다.

파이썬의 연산자

- 연산이란 **새로운 데이터**를 만드는 것이고 연산자는 **연산을 위한 기호**를 의미함
- 새로운 결과 데이터에 따라 파이썬에서 다양한 연산자를 제공함

- 산술연산자
- 관계연산자
- 논리연산자
- 비트연산자
- 대입연산자
- 멤버연산자
- 식별연산자

프로그램의 흐름 제어

- 프로그램 내부에서 **특정 조건**에 따라 결과를 다르게 만들 수 있도록 조건식을 제공함
- 파이썬에서 제공하는 조건문
  - if
  - if ~ ~else
  - if ~ elif ~ else

반복문

- 프로그램에서 여러 번 **반복적으로** 처리해야 할 코드를 반복문을 이용해서 처리함
- 파이썬에서 제공하는 반복문
  - while
  - for

반복문

- while 반복문은 특정조건에 만족할 때 반복문을 사용하고, **무한 반복**을 이용할 수 있음
- for 반복문은 **시퀀스 자료형을 반복**하는데 사용함
  - 문자열
  - 리스트
  - 튜플
  - range( )

세종사이버대학교

파이썬기초

NEXT

함수