7. 반복 구조

1. 반복 구조의 정의

- 반복 구조의 개념
  - 특정 명령문 명령문을 여러 번 실행하는 구조
    - 반복 횟수를 조건에 따라 결정
    - loop, iterate, repeat 유의어 존재

- 반복 출력 프로그램

☞ ' 저는 파이썬을 잘 다룰 수 있습니다 ' 를 5번 출력하는 프로그램을 작성하시오.

```
print("저는 파이썬을 잘 다룰 수 있습니다.")
print("저번 도로 출력")
print("저는")
print("저는")
print("저는")
print("저는")
print("저는")
```

- 반복 구조의 구분방식
  - 구분방식: 조건 제어 반복
    - while 반복-제속-조건 :
      ↳ 코드 블록
  - 구분방식: 제수 제어 반복
    - for 제수-제어-변수 in 시퀀스 :
      ↳ 코드 블록
- 반복 구조의 장점요약
- 조건 제어 반복

☞ 조건 제어 반복

- 반복 출력 프로그램(while)

☞ ' 저는 파이썬을 잘 다룰 수 있습니다 ' 를 5번 출력하는 프로그램을 작성하시오.

```
print("저는 파이썬을 잘 다룰 수 있습니다.")
print("저번 도로 출력")
count = 1
while count <= 5 :
    print("저는")
    count = count + 1
```

- 반복 구조의 공유

```
print("저는 파이썬을 잘 다룰 수 있습니다.")
print("저번 도로 출력")
count = 1
while count <= 5 :
    print("저는")
```

- 파이썬 및 개인 프로그램
  - 주제의 사용자가 입력한 값 만큼의 합을 구하는 프로그램을 작성하시오.
- 반복 구조 설계 전략
  - 1. 반복되어야 하는 명령 블록을 작성
  - 2. 반복되어야 하는 명령문을 언제부터 반복 반복할 지인
    - while True:
      명령 블록
  - 3. 반복-제속-조건을 작성하고 반복 구조를 제어하기 위해 반복 횟수 명령문을 추가
    - while 반복-제속-조건:
      명령 블록
      반복 제어 명령문
- 반복 구조 설계 전략 작성

```
sum = sum + i

while True:
    sum = sum + i

sum = 0
i = 1
last = int(input("어디까지 더할까요: "))
while i <= last:
    sum = sum + i
    i = i + 1
```

- 구구단 출력 프로그램

☞ 사용자가 입력한 단의 구구단 출력

```
출력할 나을 입력하세요: 9
9 X 1 = 9
9 X 2 = 9

9 X 9 = 81
```

2. 리스트

- 원물 계산 프로그램 개선
  - 한식원은 10이고 놀이가 1, 5, 14, 26, 31인 원물의 부피와 겉넓이를 각각 출력하시오.

```
반지름 10 , 높이 1 원뿔
부피: 104.6666666666666 놀이: 345.4
겉넓이 10 , 높이 1 원뿔
부피: 523.3333333333333 놀이: 471.0
                  :
반지름 10 , 높이 31 원뿔
부피: 3244.6666666666665 놀이: 1287.4
```

- 리스트의 개념
  - 순서대로 값이 집합해를 저장할 수 있는 데이터 타입
    - 인덱 시퀀스에 연속된 저장 값인 접근 수단 제공
    - 개별 원소의 값을 수정, 추가, 삭제 가능
    - 원소(element)의 나열을 저장할 수 있는 시퀀스 타입 중 하나
  - 인덱 리스구조
    - 리스트, 튜플, 딕셔너리 등
- 리스트의 구성

리스트의 구성

☞ 원소들의 순서를 표현

- 리스트의 생성

리스트의 생성

☞ 구문형식

```
list(원소시퀀스)
[원소 시퀀스]
```

→인용부호 [ ] 를 사용하여 표현
→원소는 쉼표(,)로 나열

```
[1, 4, 14, 26, 31]   hei_list = [1, 4, 14, 26, 31]
beds = ['181.78', 'dark brown', 'black']
```

- list(원소 시퀀스)
- [원소 시퀀스]
- 인덱스 연산자
  - 시퀀스 타입의 원소에 접근하는 연산자
    - 원소 연산자 [ ] 사용
    - 원소에 부여된 인덱스 번호로 지정

인덱스 연산자

☞ 시퀀스 타입의 원소에 접근하는 연산자
→원소 연산자 [ ] 사용
→원소에 부여된 인덱스 번호로 지정

```
print(hei_list[2])   hei_list[4] = 45
```

- 제수 제어 반복의 사용

제수 제어 반복의 사용

☞ 구문형식

for 제수-제어-변수 in 시퀀스 :
↳ 코드 블록

→제수-제어 변수에 시퀀스 사용
→반복 시 제수-제어 변수에 시퀀스의 원소 할당

```
hei_list = [1, 4, 14, 26, 31]
for hei in hei_list :
    print(hei)
```

- 제수-제어 변수에 시퀀스 사용
- 반복 시 제수-제속-변수에 시퀀스의 원소 할당
- 원물 계산 프로그램 개선

원물 계산 프로그램 개선

☞ 반지름은 100이고 높이가 1, 5, 14, 26, 31인 원물의 부피와 겉넓이를 각각 출력하시오.

```
반지름 10 , 높이 1 원뿔
부피: 104.6666666666666 놀이: 345.4
반지름 10 , 높이 5 원뿔
부피: 523.3333333333333 놀이: 471.0
                  :
반지름 10 , 높이 31 원뿔
부피: 3244.6666666666665 놀이: 1287.4
```

3. 반복 구조의 확장

- 원물 계산 프로그램 개선

원물 계산 프로그램 개선

☞ 반지름과 높이가 (10, 1), (20, 5), (30, 14)인 원물의 부피와 겉넓이를 각각 출력하시오.

```
반지름 30 , 높이 14 원뿔
부피: 3244.6666666666665 놀이: 1287.4
반지름 20 , 높이 5 원뿔
부피: 2093.3333333333333 놀이: 1413.0
반지름 30 , 높이 14 원뿔
부피: 13185.6666666666665 놀이: 3244.6
```

- 리스트 생성 방법
  - 리스트 내 원소에 규칙성이 있는 경우 생성 자동화를 위해 range 사용 가능

리스트 생성 자동화

☞ 리스트 내 원소에 규칙성이 있는 경우 생성 자동화를 위해 range 사용 가능

range(a, b, k)

반복 시 기본값 a 연속 시 기본값 1

→ 파이썬 3.0부터 지연 값이라 1이 증가시키며 시퀀스 생성

- range(a, b, k)
- range 함수의 사용

range 함수의 사용

```
range(1, 100, 1)   range(1, 100)
1 2 3 4 5 ··· 99
rad_list = range(10, 40, 10)
rad_list  10 20 30
```

- 반복 출력 프로그램 개선

반복 출력 프로그램 개선

☞ ' 저는 파이썬을 잘 다룰 수 있습니다 ' 를 5번 출력하는 프로그램을 작성하시오.

```
print("저는 파이썬을 잘 다룰 수 있습니다.")
print("저번 도로 출력")
for count in range(1, 6) :
    print(end)
```

- 원물 계산 프로그램 개선

☞ 반지름과 높이가 (10, 1), (20, 5), (30, 14)인 원물의 부피와 겉넓이를 각각 출력하시오.

```
반지름 30 , 높이 14 원뿔
부피: 3244.6666666666665 놀이: 1287.4
반지름 20 , 높이 5 원뿔
부피: 2093.3333333333333 놀이: 1413.0
반지름 30 , 높이 14 원뿔
부피: 13185.6666666666665 놀이: 3244.6
```

- zip 함수

```
for rad, hei in zip(rad_list, hei_list):
```

- 구구단 출력 프로그램 개선

구구단 출력 프로그램 개선

☞ 구구단표 출력

구구단표

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

- 중첩 반복 구조
  - 반복 구조 내 다른 반복 구조를 내포한 형식

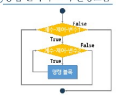
중첩 반복 구조

☞ 반복 구조 내 다른 반복구조를 내포한 형식
☞ 구문형식

for 제수-제어-변수 in 시퀀스 :
↳ for 제수-제어-변수 in 시퀀스 :
↳ 코드 블록

- 중첩 반복 구조의 장점요약

중첩 반복 구조의 실행흐름



- 구구단 출력 프로그램 개선

구구단 출력 프로그램 개선

☞ 구구단표 출력

구구단표

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

- format 함수
- 데이터를 양식에 맞춰 형식화

format 함수

☞ 데이터를 양식에 맞춰 형식화

format("구구단표", ">10s")

대상 데이터 양식 지정자

→형식 지정자

기 | 정 | 동

정렬방법 | 최소 폭 | 데이터 타입
(d:정수, s:문자열)

- format 함수의 사용

format 함수의 사용

☞ 문자열 형식

format("구구단표", ">10s")

10

구구단표

☞ 정수 형식

format(1234, "<10d")

10

1 2 3 4

```
print(format("구구단표", ">20s"))

# 2. 이중 출력
print("  |", end="")
for j in range(1, 10):
    print("  %d |", j, end="")
print("  |")
print("-"*40)
print("-"*40)
# 3. 이중이 본 출력
for i in range(1, 10):
    print(i, "|", end="")
    for j in range(1, 10):
        # end="" : 줄바꿈 없이 진행
        print( format(i*j ", ">4d"), end="")
    print()
```

- 중첩 반복 구조의 공유

중첩 반복 구조의 공포

```
for i in range (1,10):
    for j in range (1,10):
        for k in range (1,10):
            i * j * k
```