파이썬 프로그래밍

조건문과 반복문

코드블록이란?

```
흐름제어문A:
    print("hello")
    num = 10
    num += 5
    print(num)

흐름제어문B:
    print("hello")
    print(1+1)

print("Python!")
```

파이썬은 들여쓰기로 **코드블록**을 정의

조건, 함수와 같은 **실행 흐름 제어문**에서 해당 블록의 시작과 끝을 나타낸다.

흐름제어문A가 작동할 경우 들여쓰기가 된 부분(코드블록)들만 실행된다.

흐름제어문B가 작동하지 않을 경우 들여쓰기가 된 부분(코드블록)은 무시하고 흐름제어문B와 같은 라인의 `print()`가 실행된다.

조건문이란?

특정한 조건을 만족할 때 코드를 실행시키는 명령어.

주로 `if`, `elif`, `else`가 사용된다.

- if: 주어진 조건에 대해 참일 때 코드 실행
- elif: 이전 조건이 거짓일 때, 현재 조건을 체크 후, 참이라면 코드 실행
- else: 모든 이전 조건이 거짓일 때 코드 실행

if 문

```
age = 18

if age < 20:
  print("학생입니다")
  print(age, "살 입니다.")

if -1:
  print("true")
  if 3:
    print("true")

if True:
  print("true")
```

조건*(age가 20 미만인가?)*에 대해 참이기에 들여쓰기가 된 코드 블록이 실행.

연산자의 결과가 참인 경우나
연산의 결과가 **0**이 아닐 경우,
명시적으로 **True**를 적을 경우에도 실행.

if문의 코드 블록 안에 if문이 들어갈 수 있다.

elif 문

```
age = 16

# 조건1
if age<10:
  print("어린이")

# 조건2
elif age>=10 and age<20:
  print("학생")
  if True:
    print("true")

# 조건3
elif age >= 20:
  print("성인")
```

조건1이 거짓이기에 조건2를 확인한다.

조건2가 참이기에 조건3은 확인하지 않는다.

기준이 되는 if문과 같은 코드블록에 있어야 한다.

elif의 코드 블록 안에 if문이 들어갈 수 있다.

else 문

```
age = 30

# 조건1
if age<10:
  print("어린이")

# 조건2
elif age>=10 and age<20:
  print("학생")

else:
  print("성인")
  if True:
    print("true")
```

모든 조건이 거짓일 때 실행된다.

else도 elif처럼 기준이 되는 if문과 같은 라인에 있어야 한다.

else의 코드 블록 안에 if문이 들어갈 수 있다.

예제

짝수 홀수 판별기

2로 나눈 나머지가 1이라면 홀수
2로 나눈 나머지가 0이라면 짝수

연산 순서가 명확하지 않을 때는 괄호로 묶어서 순서를 명확하게 하는 것이 좋다.

```
num = 13

# 2로 나눈 나머지가 1 -> 홀수
if (num % 2) == 1:
  print("홀수")

# 2로 나눈 나머지가 0 -> 짝수
elif (num % 2) == 0:
  print("짝수")

else:
  print("정수를 입력해주세요.")
```

예제

짝수 홀수 판별기

모든 자연수는 2로 나누면
나머지는 0 혹은 1이다.

`num % 2` 의 결과가 1일 경우 -> 홀수
`num % 2` 의 결과가 0일 경우 -> 짝수

```
num = 13

if (num % 2):
  print("홀수")
else:
  print("짝수")
```

예제

윤년 판별기

윤년은 연도가 4의 배수이면서, 100의 배수가 아닐 때 또는 400의 배수일 때이다.

윤년 규칙을 수학적으로 표현한다면
(n % 4 == 0) and ((n % 100 != 0) or (n % 400 == 0))

예제

윤년 판별기

```
if year % 4 == 0:
    if year % 100 == 0:
        if year % 400 == 0:
            # 400으로 나누어떨어지면 윤년이다.
            print(f"{year}년 윤년 O")
        else:
            # 100으로 나누어떨어지지만
            # 400으로 나누어떨어지지 않으면 윤년이 아니다.
            print(f"{year}년 윤년 X")
    else:
        # 4로 나누어떨어지고
        # 100으로 나누어떨어지지 않으면 윤년이다.
        print(f"{year}년 윤년 O")
else:
    # 4로 나누어떨어지지 않으면 윤년이 아니다.
    print(f"{year}년 윤년 X")
```

예제

윤년 판별기

연산의 결과가 True 이면 코드 블록을 실행하는 것이기에

이렇게 최적화 가능.

```
if (year % 400 == 0) or (year % 4 == 0 and year % 100 != 0):
    print(f"{year}년 윤년 O")
else:
    print(f"{year}년 윤년 X")
```

반복문이란?

특정한 조건을 만족할 때, 동일한 코드 블록을 반복하는 명령어

주로 for, while이 사용된다.

- `for` : 반복 횟수가 정해진 경우
- `while` : 반복의 조건이 있는 경우, 반복 횟수가 명시적이지 않은 경우

for 문

```
# 기본 문법
for 변수 in 연속된데이터집합:
  # 코드 블록
```

변수는 연속된 데이터 집합의 각 요소가 들어갈 공간을 나타낸다.

연속된 데이터 집합은 리스트, 튜플, 문자열 등이 포함될 수 있다.

> 변수가 데이터집합 안에 속해있다면, 아래 코드 블록을 실행하고, 변수는 데이터집합의 다음 요소로 넘어간다.

for 문

```
fruits = ["사과", "바나나", "오렌지"]

for 변수 in fruits:
  print(변수)
```

fruits 배열의 첫 번째 요소인
*사과*가 변수에 들어가고,
코드 블록 실행.

사과 출력 후, *사과*의 다음 요소인
*바나나*가 변수에 들어간다.

for 문

```
array = [1, 2, 3, 4, 5, 6]

for i in array:
  print(i)
```

array의 첫 번째 인덱스인 1이
변수 i에 들어가고, 코드 블록 실행.

1 출력 후, array의 두 번째 요소인 2가
변수 i에 들어가고, 코드 블록 실행.

“ i는 index의 약자로 반복문 등에서 index를 나타낼 때 변수를 주로 i로 설정한다. ”

for 문

```
start = 1
end = 15
step = 1

for i in range(start, end, step):
  print(i)
```

`range(a, b)` : a부터 b-1까지 정수를 생성 (a, b는 정수)

`range(1, 10)` 와 `[1, 2, 3, 4 ... 8, 9]` 가 같다고 해석해도 무방.

step이 음수라면, 역순 출력

> 일반적으로 for문은 반복 횟수가 정해진 경우에 사용되기에 range() 함수를 사용하는 경우가 대부분이다.

예제

별 찍기

반복문을 이용하여 우측 결과 출력

```
# for 예제 1번

*
**
***
****
*****
```

```
# for 예제 2번

  *
 ***
*****
 ***
  *
```

예제

별 찍기

```python
# for 예제 1번

for i in range(1,6):
    print("*" * i)
```

```python
# for 예제 2번

# 상단 삼각형 출력
for i in range(1, 4):
  print(" " * (3 - i) + "*" * (2 * i - 1))

# 하단 삼각형 출력
for i in range(2, 0, -1):
  print(" " * (3 - i) + "*" * (2 * i - 1))
```

while 문

```
# 기본 문법
while 조건:
  # 코드 블록

while True:
  print("hello!")

a = 5

while a < 10:
  print(a)
  a += 1
```

주어진 조건이 참일 때 코드 블록을 반복해서 실행.

연산, 연산자 등의 결과가 참이기만 한다면 무한히 반복하기에 반복 횟수를 제어해서 무한 루프를 반복한다.

while 문

```
num = 1

while True:
  if num == 3:
    continue

  if num > 30:
    break

  print(num)
  num += 1
```

`continue` : 현재 반복 단계를 넘기고,
다음 반복 단계로 넘어가도록 한다.

`break` : 반복문을 즉시 종료 후,
반복문 코드 블록을 빠져나가도록 한다.

예제

입력 받은 숫자가 '소수'인지 확인

- 정수를 입력받고, 해당 숫자가 소수인지 판별하는 프로그램을 만들어라.
- 조건: 사용자가 `0`을 입력하기 전까지 입출력을 반복해야 한다.

예제 - 입력 받은 숫자가 '소수'인지 확인

1. 반복문에 진입
    1. 숫자 입력
    2. 입력 숫자가 0이라면 ...
    3. 입력 숫자가 1이라면 ...
    4. 입력이 2 이상인 경우
        1. 나누어 떨어진다면 소수 아님 체크
    5. 체크 여부를 바탕으로 소수인지 출력

```python
while(True):
    num = int(input("숫자 입력: "))
    is_prime = True

    if num == 0:
        break
    elif num == 1:
        print("소수 X")
        continue
    else:
        for i in range(2, num):
            if num % i == 0:
                is_prime = False

    if is_prime == True:
        print("소수 O")
    else:
        print("소수 X")
```

```python
import math

while True:
    num = int(input("숫자 입력: "))
    is_prime = True

    if num == 0:
        break
    elif num == 1:
        print("소수 X")
        continue
    elif num == 2:
        print("소수 O")
        continue
    elif num % 2 == 0:
        print("소수 X")
        continue

    for i in range(3, int(math.sqrt(num)) + 1, 2):
        if num % i == 0:
            is_prime = False
            break

    if is_prime:
        print("소수 O")
    else:
        print("소수 X")
```

파이썬 프로그래밍

조건문과 반복문