# 파이썬을 활용한 인공지능 체험하기

2주: 조건문/반복문과 기타 제어문

한양대학교 ERICA 소프트웨어학부

조상욱 교수

swcho3@hanyang.ac.kr



# 반복문

### 01. 왜 조건을 따져봐야 하나요?



- 조건을 따져서 서로 <mark>다른 동작</mark>을 하도록 하는 것은 컴퓨터 프로그램에서 아주 중요!
- 프로그래밍에서 조건을 따지는 방법
   : 질문(question)한 후 결정(decision)하는 형태.

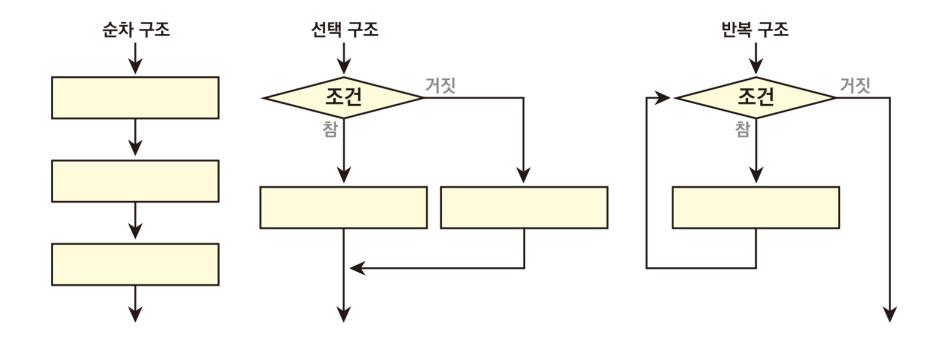
### 왜 조건을 따져봐야 하나요?



순차 구조(sequence): 명령어들이 순차적으로 실행되는 구조

선택 구조(selection) : 둘 중의 하나의 명령을 선택하여 실행되는 구조

반복 구조(iteration) : 동일한 명령이 반복되면서 실행되는 구조



### 왜 조건을 따져봐야 하나요?



프로그램에서 선택 구조가 필요한 예

- ① 게임에서 철수가 정답을 맞혔으면 철수의 점수가 1만큼 증가한다.
- ② 작성한 파일을 저장하지 않았으면 저장할 것인지를 물어보는 메시지를 출력한다.
- ③ 파일이 하드 디스크에 없으면 오류 메시지를 출력한다.

# 참과 거짓 그리고 관계 연산자



1 + 12 = 13	 (1)
11 - 1 = 9 ······	 (2)
한라산은 제주도에 있다.	 (3)
수학책은 두껍다	 (4)
베토벤은 음악 천재이다.	 (5)

수학시간에배운 명제를 떠올려보세요~

# 참과 거짓 그리고 관계 연산자





그럼, 파이썬으로 다음과 같은 문장을 어떻게 작성할 것인지 생각해봅시다.

# 참과 거짓 그리고 관계 연산자

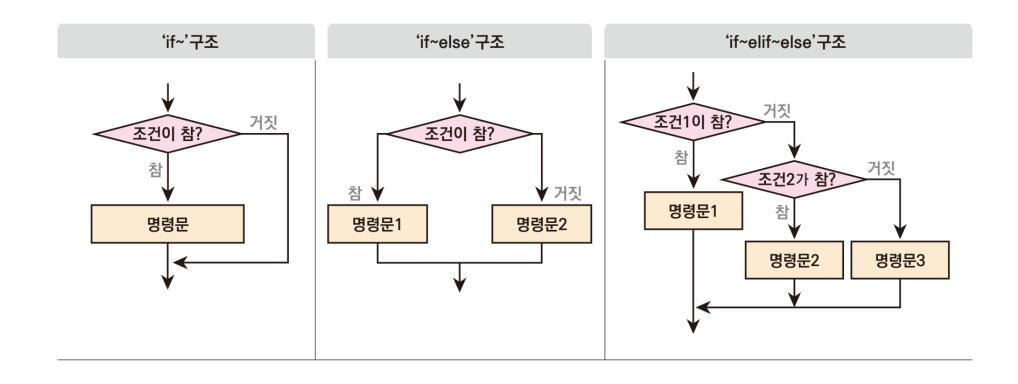


- 조건식: 조건에 해당하는 부분을 수식으로 나타낸 것
- 관계 연산자(relational operator)

연산	의미	예	결괏값
x == y	x와 y가 같다	100 == 1	False
x != y	x와 y가 다르다(같지 않다)	100 != 1	True
x > y	x가 y보다 크다	100 > 1	True
x < y	x가 y보다 작다	100 < 1	False
x >= y	x가 y보다 크거나 같다	100 >= 1	True
x <= y	x가 y보다 작거나 같다.	100 <= 1	False

# 조건문(만약에~)





### 조건문(만약에~)





score가 60 이상이면 "합격입니다! / 수고하셨습니다"를 출력하는 프로그램을 작성해 봅시다.

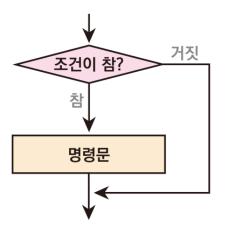
### 코드

score = 90 print("합격입니다!") #반드시 들여 쓰세요. print("수고하셨습니다.")

#score에 90을 저장합니다. if score >= 60: #마지막에 콜론(:)을 잊지 마세요.

### 실행 결과

합격입니다! 수고하셨습니다.

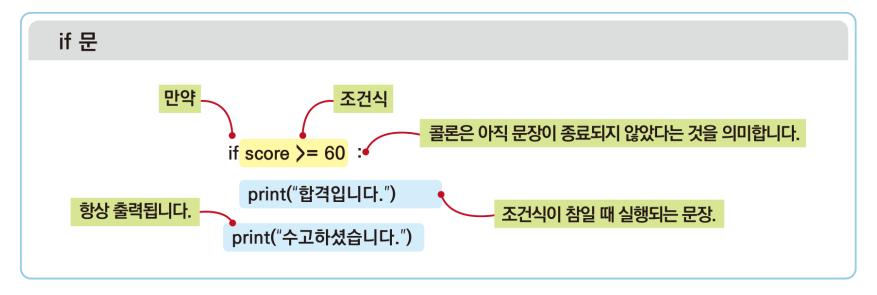


# if와 if~else





if문을 살펴보겠습니다.

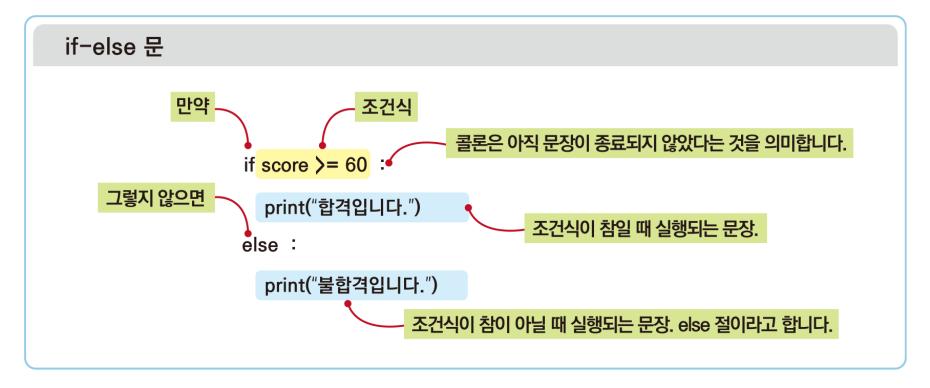


# if와 if~else





if-else 문을 살펴보겠습니다.



# if를 좀 더 이해시켜 줄 예제





우리들의 번역기 "마마고"

#### 코드

```
language = int(input("언어를 선택하세요(1=한국어, 2=영어, 3=프랑스어, 4=독일어)"))

if language == 1: # 콜론()을 잊지 마세요.
print("안녕")

if language == 2:
print("Hello")

if language == 3:
print("Bonjour")

if language == 4:
print("Guten morgen")
```

#### 실행 결과 1

언어를 선택하세요(1=한국어, 2=영어, 3=프랑스어, 4=독일어)1 안녕

#### 실행 결과 2

언어를 선택하세요(1=한국어, 2=영어, 3=프랑스어, 4=독일어)4 Guten morgen

### if-else를 좀 더 이해시켜 줄 예제





영화 관람 나이 제한 검사

### 코드

```
age = int(input("나이를 입력하시오: "))

if age >= 15:
  print("영화를 관람할 수 있습니다.")
else:
```

print("영화를 관람할 수 없습니다.")

#### 실행 결과 1

나이를 입력하시오: 19 영화를 관람할 수 있습니다.

#### 실행 결과 2

나이를 입력하시오: 14 영화를 관람할 수 없습니다.



홀수-짝수 판별 프로그램

### 코드

```
num = int(input("정수를 입력하시오: "))
if num % 2 == 0:
    print("짝수입니다.")
else:
    print("홀수입니다.")
```

#### 실행 결과 1

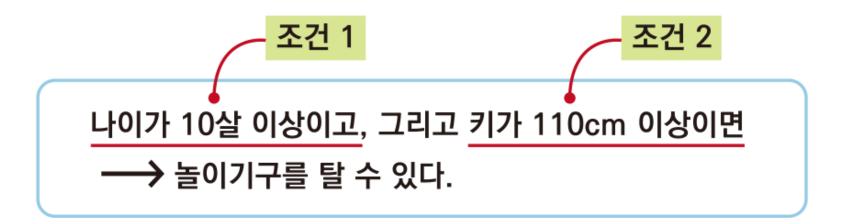
정수를 입력하시오: 10 짝수입니다.

### 실행 결과 2

정수를 입력하시오: 23 홀수입니다.

### 복잡한 조건의 표현-논리 연산자





# 복잡한 조건의 표현-논리 연산자

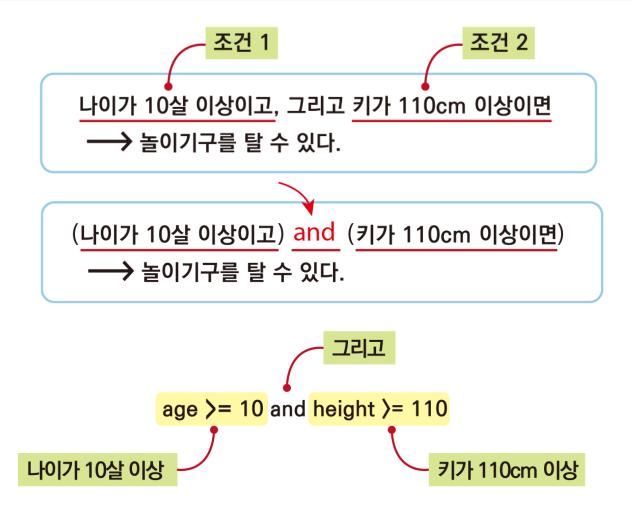


• 논리 연산자(logical operator)는 여러 개의 조건을 조합하여 참인지 거짓인지를 따질 때 사용

연산	의미
x and y	x와 y가 모두 참이면 참, 하나라도 거짓이면 거짓
x or y	x나 y 중에서 하나만 참이면 참, 모두 거짓이면 거짓
not x	x가 참이면 거짓, x가 거짓이면 참

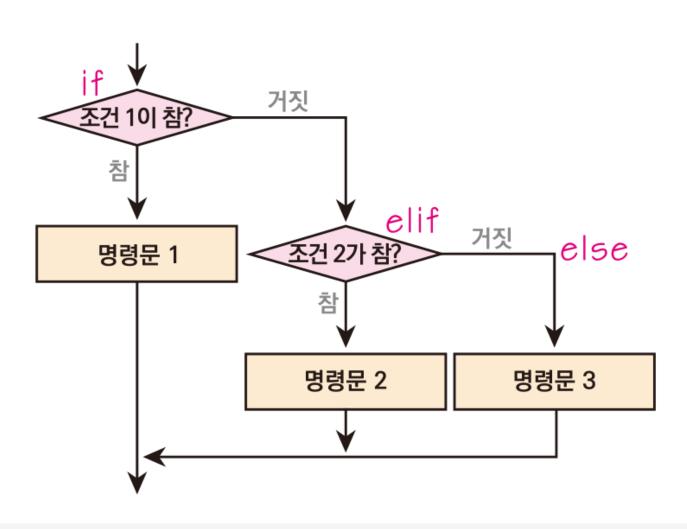
# 복잡한 조건의 표현-논리 연산자





# if-elif-else





### if-elif-else





if-elif-else가 사용되는 예제를 살펴보겠습니다.

### 코드

```
num = int(input("정수를 입력하시오: "))
if num > 0:
    print("양수입니다.")
elif num == 0:
    print("0입니다.")
else:
    print("음수입니다.")
```

#### 실행 결과 1

정수를 입력하시오: **10** 양수입니다.

### 실행 결과 2

정수를 입력하시오: 0 0입니다.

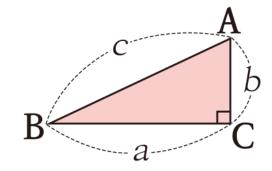
### 실행 결과 3

정수를 입력하시오: -10 음수입니다.

# 실습1. 직각삼각형 판별하기



사용자로부터 세 변의 길이 a, b, c를 입력받아 피타고라스의 정리를 이용하여 직각삼각형을 판별하는 프로그램을 작성해 봅시다. (단, c 〈a + b입니다.)



$$a^2 + b^2 = c^2$$

#### 실행 결과 1

변a의 길이: 3

변b의 길이: 5

변c의 길이: 6

직각삼각형이 아닙니다.

### 실행 결과 2

변a의 길이: 3

변b의 길이: 4

변c의 길이: 5

직각삼각형입니다



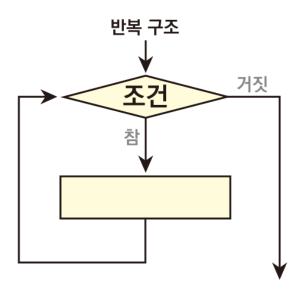
# 반복문

# 01. 왜 반복이 중요할까요?



- 루프(loop)
- 동일한 작업을 여러 번 수행할 때 무엇이 더 간결할까?

### 똑같은 문장 여러 번 쓰기 vs 반복 구조



# 01. 왜 반복이 중요할까요?





반복 구조를 이용하면 프로그램이 더 간결해집니다.

### 같은 작업을 여러 번 나열하여 작성

print("방문을 환영합니다!")

print("방문을 환영합니다!")

print("방문을 환영합니다!")

print("방문을 환영합니다!")

print("방문을 환영합니다!")

### 반복 구조를 이용하여 작성한 코드

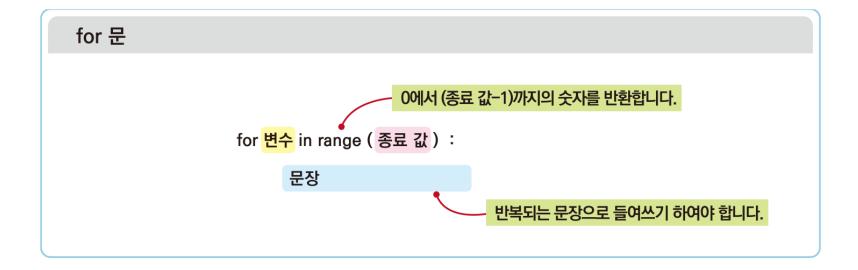
for i in range(5): print("방문을 환영합니다!")

### 02. 횟수 제어 반복-for



• 횟수 제어 반복: 반복의 횟수를 미리 아는 경우에 사용, for 루프

많은 언어에서 횟수 제어 반복 에 for를 키워드로 사용합니다..



### 02. 횟수 제어 반복-for





예를 들어서 "방문을 환영합니다!" 문장을 5번 반복하여 출력해보겠습니다.

### 코드

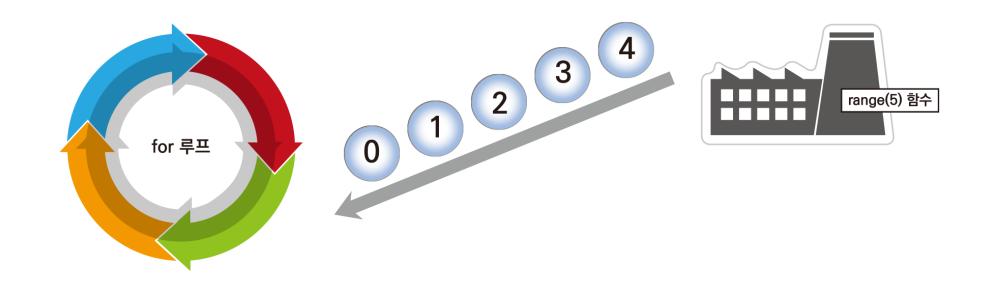
for i in range(5): # 끝에 콜론(:)이 있음 print("방문을 환영합니다.") # 들여쓰기하세요.

### 실행 결과

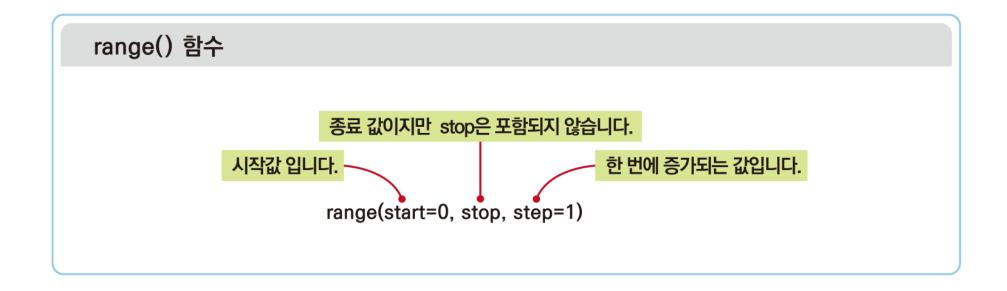
방문을 환영합니다. 방문을 환영합니다. 방문을 환영합니다. 방문을 환영합니다. 방문을 환영합니다. 방문을 환영합니다.



- range() 함수: 숫자들을 생산하는 공장
- range(5)











만약 1부터 시작하여서 5까지 반복하고 싶다면 어떻게 하면 될까요?

### 코드

for i in range(1, 6, 1):
 print(i, end=" ")

### 실행 결과

12345





만약 10부터 시작하여서 1까지 반복하고 싶다면 어떻게 하면 될까요?



for i in range(10, 0, -1):
 print(i, end=" ")

### 실행 결괴

10987654321

### 04. 횟수 제어 반복을 좀 더 이해시켜 줄 예제





1부터 100까지의 합을 구하는 프로그램

1에서 100까지의 자연수의 합을 구하여 출력하는 프로그램을 작성해 봅시다.

#### 코드

```
sum = 0
for i in range(1, 101):
    sum += i
print("1부터 100까지의 합은",sum,"입니다.")
```

### 실행 결과

1부터 100까지의 합은 5050 입니다.

### 04. 횟수 제어 반복을 좀 더 이해시켜 줄 예제





팩토리얼 계산 프로그램

팩토리얼 n!은 1부터 n까지의 정수를 모두 곱한 것을 의미합니다. 즉, 다음과 같습니다.

n 값을 입력하면 n!를 계산하는 프로그램을 작성해 보세요.

#### 코드

```
n = int(input("정수를 입력하시오: "))
fact = 1

for a in range(1, n + 1):
  fact = fact * a

print(n, "!은", fact, "이다.")
```

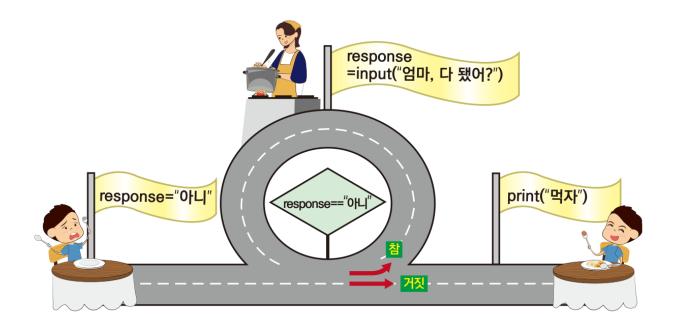
#### 실행 결과

정수를 입력하시오: 10 10!은 3628800 이다.

# 05. 조건 제어 반복-while

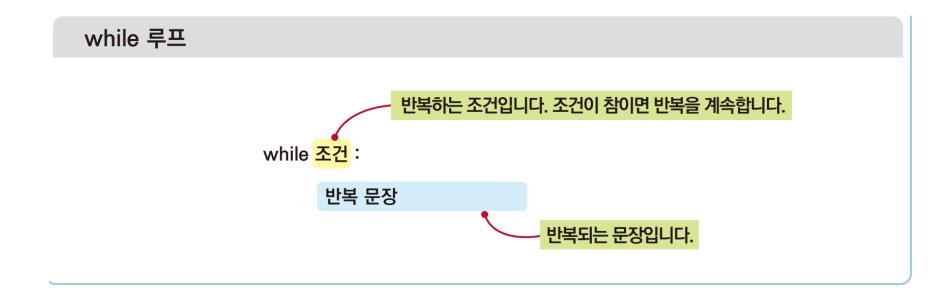


- 어떤 조건이 만족되는 동안 반복
- 반복의 횟수는 모르지만, 반복의 조건은 알고 있는 경우에 주로 사용하는 반복



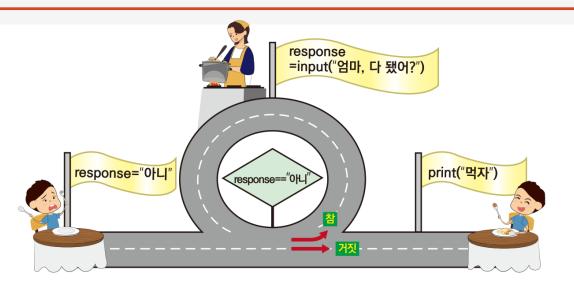
# 05. 조건 제어 반복-while





# 05. 조건 제어 반복-while







위의 상황을 프로그램으로 작성해 보면 다음과 같습니다.

### 코드

```
response = "아니"
while response == "아니":
  response = input("엄마, 다됐어? ");
print("먹자")
```

#### 실행 결과

엄마, 다됐어? 아니 엄마, 다됐어? 아니 엄마, 다됐어? 그래 먹자

### 06. 조건 제어 반복을 좀 더 이해시켜 줄 예제





1부터 100까지의 합을 구하는 프로그램

1에서 100까지의 자연수의 합을 구하여 출력하는 프로그램을 작성해 봅시다.

#### 코드

```
count = 1
sum = 0
while count <= 100:
    sum = sum + count
    count = count + 1
print("1부터 100까지의 합은",sum,"입니다.")
```

### 실행 결과

1부터 100까지의 합은 5050 입니다.

### 06. 조건 제어 반복을 좀 더 이해시켜 줄 예제





로그인 프로그램

### 코드

password = ""
while password != "pythonisfun":
 password = input("암호를 입력하시오: ")
print("로그인 성공")

### 실행 결과

암호를 입력하시오: idontknow 암호를 입력하시오: 12345678

암호를 입력하시오: pythonisfun

로그인 성공



### 07. 무한 반복



- 반복이 무한히 발생하는 것 = 무한 루프(infinite loop)
- 특정 조건에서 그 무한 반복을 멈추는 것을 생각해줘야 함



무한 반복을 이용하여 신호등 프로그램을 작성해 보겠습니다.

### 코드

print('전진!!')

```
sign = True

while sign:
  light = input('신호등 색상을 입력하시오: ')
  if light == 'blue':
    sign = False
```

### 실행 결과

신호등 색상을 입력하시오: red 신호등 색상을 입력하시오: red 신호등 색상을 입력하시오: blue 전진!!

# 08. break와 continue



• break : 강제로 반복 중지

```
코드
while True:
light = input('신호등 색상을 입력하시오: ')
if light == 'blue':
break
print('전진!!')
```

### 실행 결과

신호등 색상을 입력하시오: red 신호등 색상을 입력하시오: red 신호등 색상을 입력하시오: blue 전진!!

# 09. break와 continue



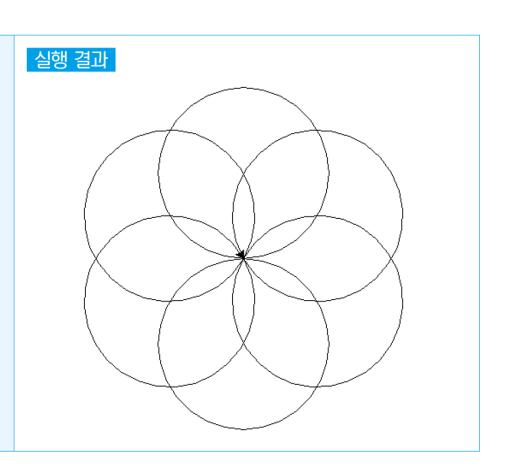
• continue : 해당 차례를 건너뛰고 새롭게 다음 차례의 반복을 수행



# 실습. 코드를 줄여보아요.



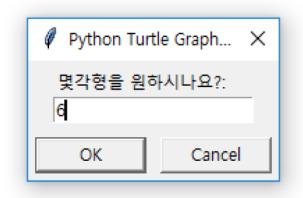
```
코드
import turtle
t = turtle.Turtle()
t.circle(100) # 반지름이 100인 원을 그림
t.left(60) # 60도만큼 터틀을 왼쪽으로 회전
t.circle(100)
t.left(60)
t.circle(100)
t.left(60)
t.circle(100)
t.left(60)
t.circle(100)
t.left(60)
t.circle(100)
```

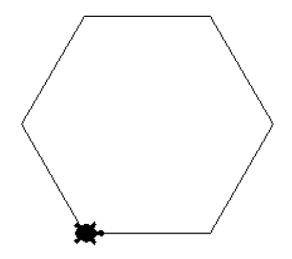


# 실습. n각형 그리기



• 사용자로부터 정수 n을 입력받아서 한 변의 길이가 100인 정 n각형을 그리는 프로그램을 작성해 보세요.





### 실습. 모든 약수 구하기



- 약수: 어떤 수를 나누어 떨어지게 하는 수를 그 수의 약수 라고 함. 나머지를 '0'으로 하는 수
- 사용자로부터 어떤 자연수를 입력 받아 약수를 모두 출력 하는 프로그램을 작성해 보세요.

### 실행 결과

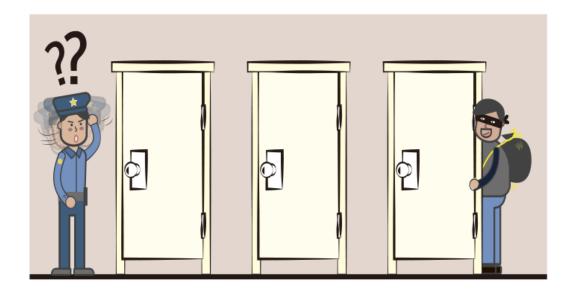
자연수 입력: 6

1236

# 실습. 범인 찾기 게임



- 3개의 방 중 한 곳에 숨어 버린 범인을 찾아라!
- 범인이 숨은 방을 맞추면 100점 추가 후 게임 종료.
- 하지만 틀리면 범인은 다른 방에 숨고 10점 감점 후 다시 맞추기.



# 감사합니다.