

조건으로 따져 실행하자

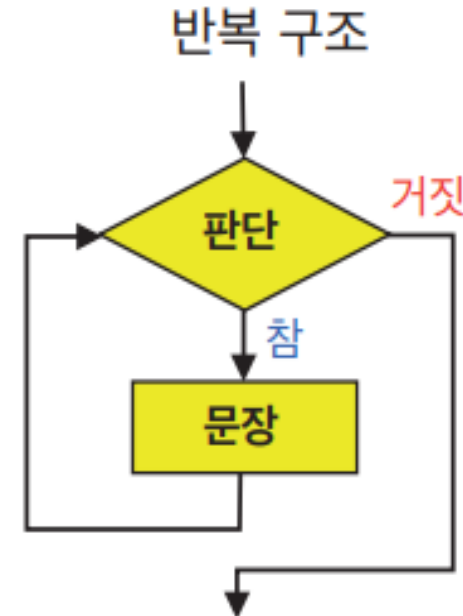
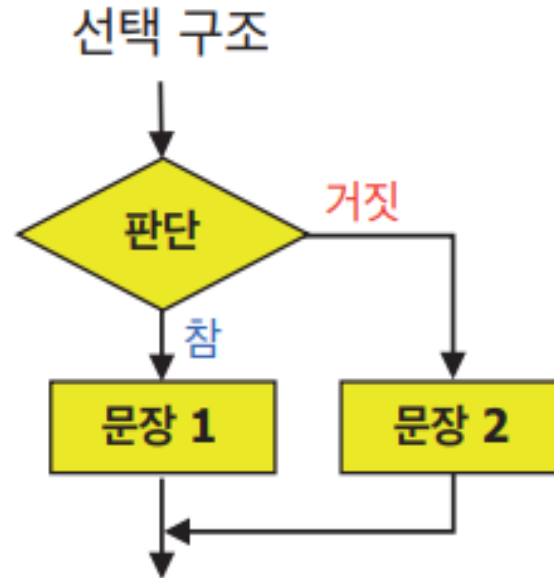
# 목차

---

1. 제어구조
2. 기본 if 문
3. 다중 if문/ 중첩 if문
4. if문 응용

# 제어 구조

1. 순차 구조(sequence) : 명령들이 순차적으로 실행되는 구조
2. 선택 구조(selection) : 여러 개중 하나의 명령문을 선택하여 실행하는 구조
3. 반복 구조(iteration) : 동일한 명령이 반복되면서 실행되는 구조

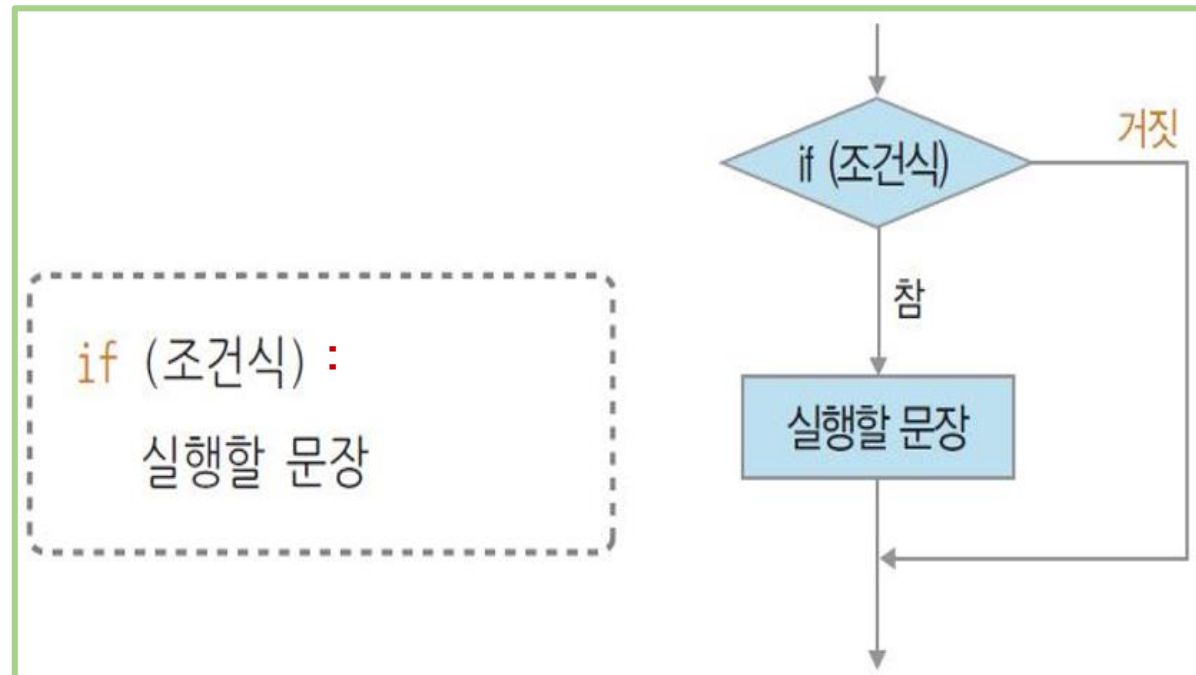


# 기본 if 문

## 1. 조건식

- 어떤 **조건**을 만족하는지 그렇지 않은지를 판정하는 식

## 2. 조건식은 참 또는 거짓의 값을 갖는 **부울**형으로 평가



# 기본 if 문

```
score = int(input("성적을 입력: "))
```

```
if score >= 85: 1
```

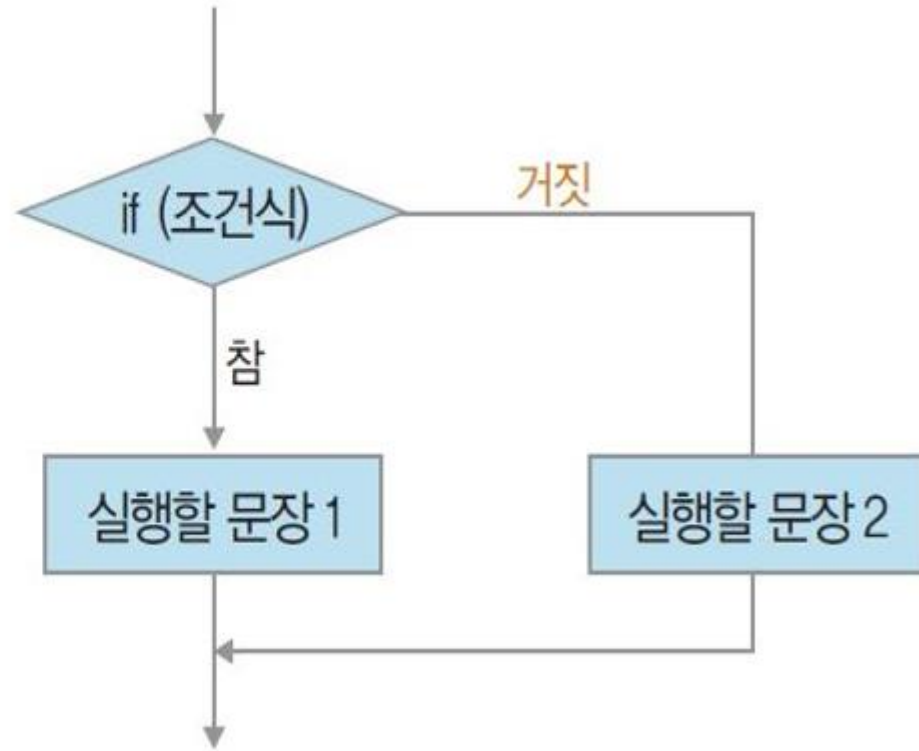
```
2 ←→ print("합격입니다.")  
      print("축하^^")
```

## [주의 사항]

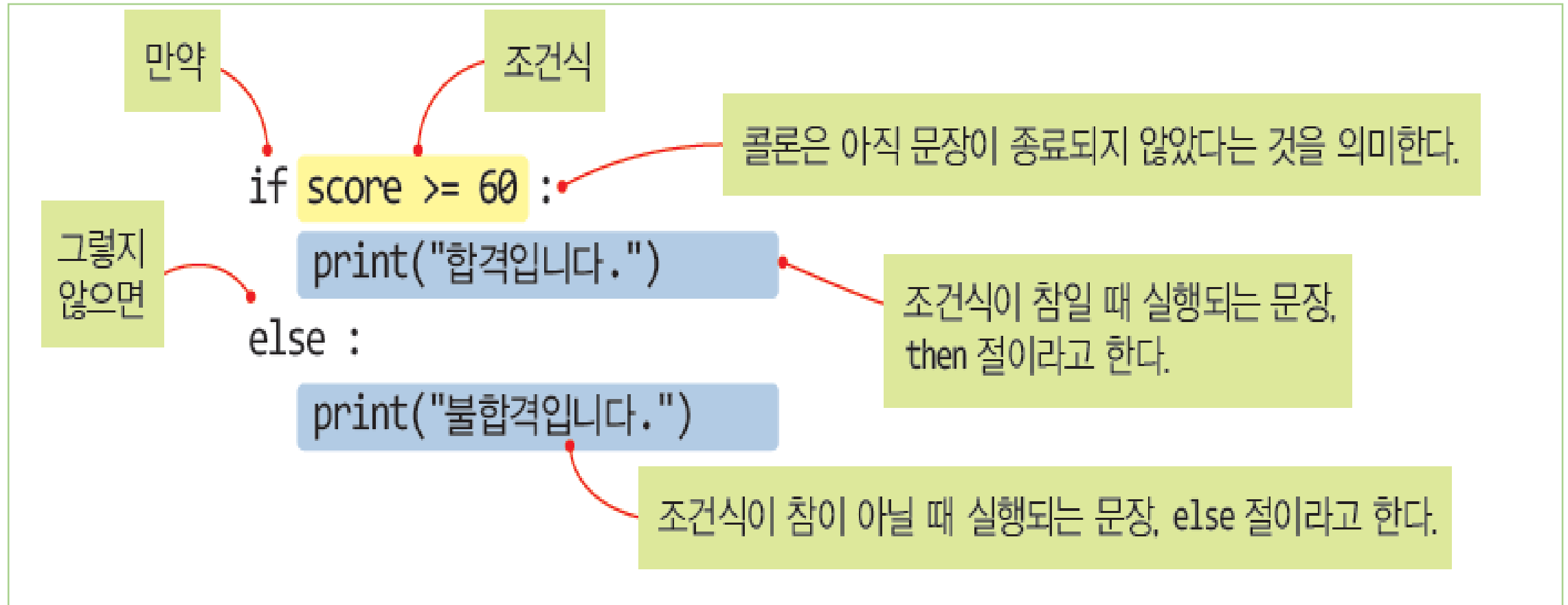
1. if 조건 뒤에 : (콜론기호) 반드시 쓸 것
2. 들여쓰기 (indentation) - 4칸의 들여쓰기

# if - else 문

```
if (조건식):  
    실행할 문장 1  
else :  
    실행할 문장 2
```



# if - else 문



# if - else 문

```
score = int(input("성적을 입력하시오: "))
```

```
if score >= 90:
```

```
    print("합격입니다.")
```

```
    print("장학금도 받을 수 있어요 ")
```

```
else:
```

```
    print("불합격입니다.")
```

```
    print("파이팅하세요")
```



# 도전

---

키보드로 숫자를 입력 받아 **짝수**인지 **홀수**인지 검사하는 프로그램을 작성하세요

**if ~ else** 구문 사용

# 도전

1

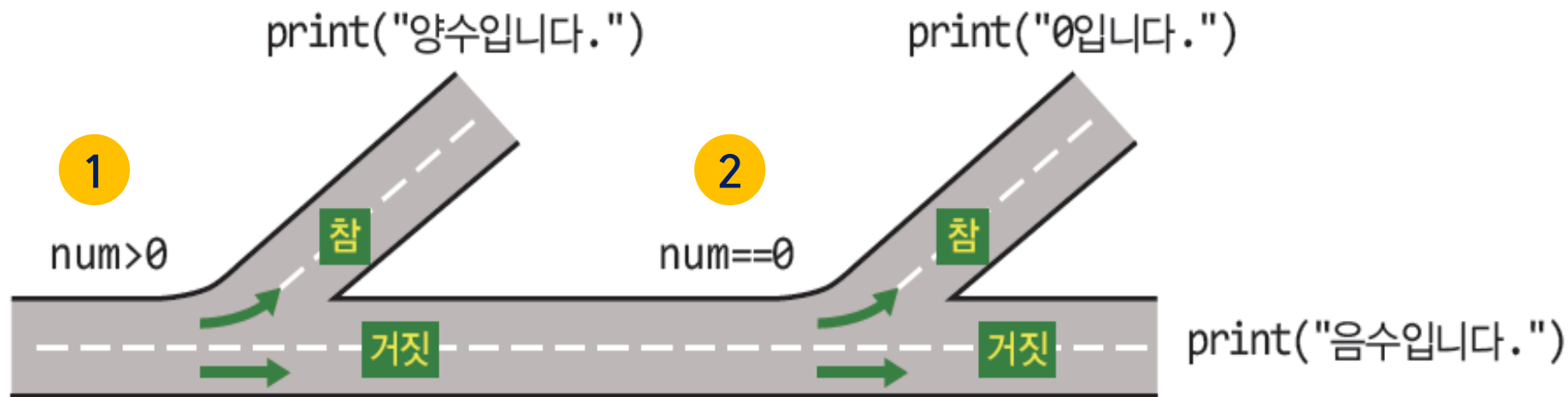
키보드로 숫자를 입력 받아 짝수인지 홀수인지 검사하는 프로그램을 작성하세요

```
num = int(input( " 숫자를 입력하시오: "))

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

# 조건이 거짓일때 연속하여 다른 조건을 검사 ( 다중 if)

1. if - else 문에서 조건이 거짓일 때, 연속하여 다른 조건을 검사할 수 있다
2. **elif** 키워드 사용 ( else if 약어)



사용자에게 정수를 입력받아 양수인지, 0인지, 음수인지 판별

# 도전

키보드로 숫자를 입력 받아 양수인지 음수인지 0인지 판별하는 프로그램을 작성하세요

```
num = int(input("정수를 입력: "))

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

(elif : else if )

# 도전

1~999사이의 숫자를 입력 받아, 입력 받은 숫자가  
몇 자리 정수인지 판단하는 프로그램을 작성하세요  
(elif 사용)

1~999 사이의 숫자를 입력 :345

입력한 정수 345는 3자리 숫자

1~999 사이의 숫자를 입력 :35

입력한 정수 35는 2자리 숫자

```
number = int(input("1~999 사이의 숫자를 입력 :"))

if number >= 100:
    print(" 입력한 정수 %d는 3자리 숫자" %number)
elif number >= 10:
    print(" 입력한 정수 %d는 2자리 숫자" % number)
else:
    print(" 입력한 정수 %d는 1자리 숫자" % number)
```

```
number = input("1~999 사이의 숫자를 입력 :")

if len(number) == 3:
    print(" 입력한 숫자 %s는 3자리 숫자" % number)
elif len(number) == 2:
    print(" 입력한 숫자 %s 2자리 숫자" % number)
else:
    print(" 입력한 숫자 %s는 1자리 숫자" % number)
```

# 도전

숫자(문자)를 입력받아, 입력 받은 숫자(문자)가  
몇 자리 수(문자)인지 출력하는 프로그램을 작성하세요

숫자 입력 : **23456789**  
입력한 데이터는 **23456789**는 8자리 숫자(문자)입니다

숫자 입력 : **54321**  
입력한 데이터는 **54321**는 5자리 숫자(문자)입니다



---

```
number = input( " 숫자 입력 :")
```

```
length = len(number)
```

```
print("입력한 데이터는 %s는 %d자리 숫자입니다" %(number,length))
```

# 도전

키보드로 부터 jumsu를 입력 받아  
Jumsu가 90점 ~ 100점 이면 "A등급" 출력  
80점 ~ 89점 이면 "B등급" 출력  
70점 ~ 79점 이면 "C등급" 출력  
69이하이면 "F등급" 출력 (elif 사용)

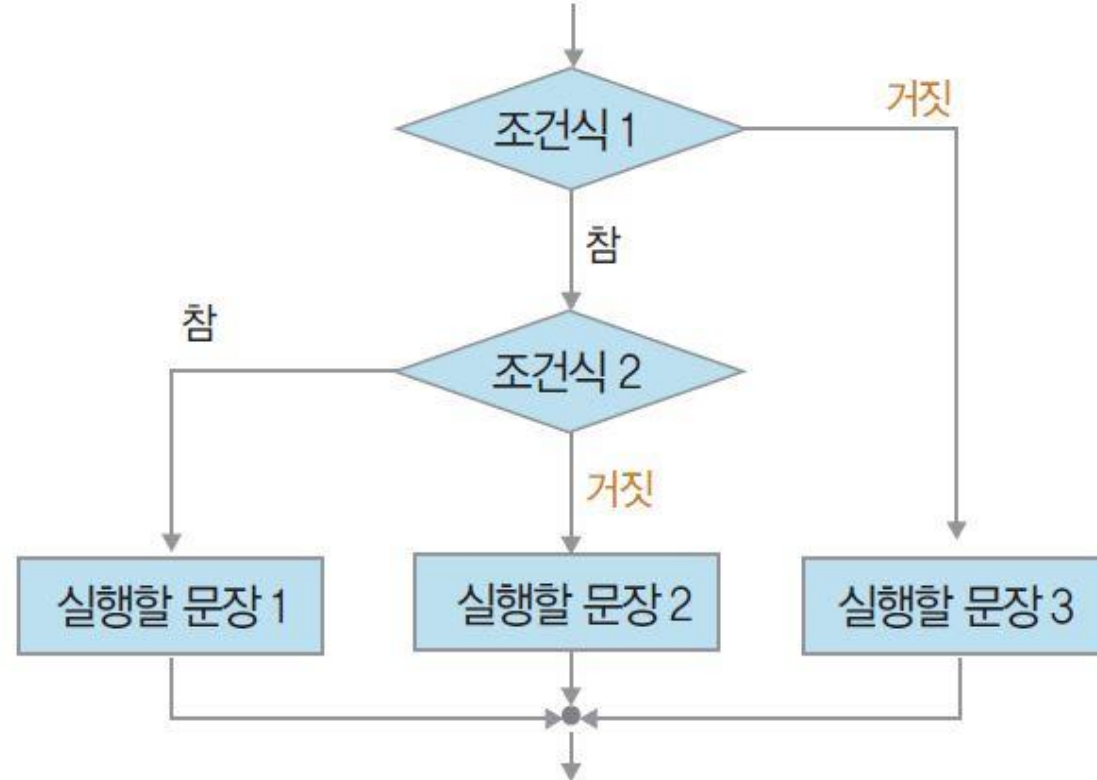
# 도전

```
jumsu = int(input("점수를 입력: "))
```

```
if jumsu >=90:  
    print("A 등급")  
elif jumsu >=80:  
    print("B 등급")  
elif jumsu >=70:  
    print("C 등급")  
else:  
    print("F 등급")
```

# 중첩 if

```
if (조건식 1){  
    if (조건식 2)  
        실행할 문장 1.  
    else  
        실행할 문장 2.  
}  
else  
    실행할 문장 3.
```



## 중첩 if

키보드로 숫자를 입력 받아 양수인지 음수인지 0인지 판별하는 프로그램을 작성하세요

```
num = int(input("정수를 입력하시오: "))
```

```
if num >= 0:
```

```
    if num == 0:
```

```
        print("0입니다.")
```

```
    else:
```

```
        print("양수입니다.")
```

```
else:
```

```
    print("음수입니다.")
```

# pass

- ❖ 문법(구문)상 필요하지만 코드를 실행하고 싶지 않을 때 사용
- ❖ 아무 일도 하지 않게 설정 (빈 코드 블록의 자리 표시자)

```
data = ['paper', 'money', 'cellphone']
```

```
if 'money' in data:
```

```
    pass
```

```
else:
```

```
    print("카드사용")
```

# 도전 - 윤년 판단

연도를 입력 받아, 입력 받은 연도가  
윤년인지 판단하는 프로그램을 작성하세요

연도를 입력하시오: 2012  
2012 년은 윤년입니다.

- ✓ 연도가 4로 나누어 떨어지면서  
100으로 나누어 떨어지는 않는 연도는 윤년이다.
- ✓ 400으로 나누어 떨어지는 연도는 윤년이다.



# 도전 - 윤년 판단

- ✓ 연도가 4로 나누어 떨어지면서 100으로 나누어 떨어지는 않는 연도는 윤년이다.
- ✓ 400으로 나누어 떨어지는 연도는 윤년이다.

```
year = int(input("연도를 입력하시오: "))

if (year % 4 == 0 and year % 100 != 0):
    print(year, "년은 윤년입니다.")
elif year % 400 == 0:
    print(year, "년은 윤년입니다.")
else :
    print(year, "년은 윤년이 아닙니다.")
```



# 도전 - 윤년 판단

- ✓ 연도가 4로 나누어 떨어지면서 100으로 나누어 떨어지는 않는 연도는 윤년이다.
- ✓ 400으로 나누어 떨어지는 연도는 윤년이다.

```
year = int(input("연도를 입력하시오: "))
```

```
if ( (year % 4 == 0 and year % 100 != 0) or year % 400 == 0):
```

```
    print(year, "년은 윤년입니다.")
```

```
else :
```

```
    print(year, "년은 윤년이 아닙니다.")
```

# 다양한 터틀 그래픽 명령어

명령	하는 일
<code>begin_fill()</code> ... <code>end_fill()</code>	<code>begin_fill()</code> 과 <code>end_fill()</code> 사이의 코드에 나타난 부분을 색칠한다. ... 으로 표시된 부분에는 터틀의 좌표나 모양을 기술할 수 있다.
<code>color(c)</code>	터틀의 색깔을 변경한다. c 값으로 'red', 'green', 'blue', 'black', 'gray', 'pink',...등의 여러가지 색상을 선택할 수 있다.
<code>shape(s)</code>	터틀의 모양을 변경한다. s 값으로는 'arrow', 'turtle', 'circle', 'square', 'triangle', 'classic' 등이 있다.
<code>shapese(s),</code> <code>shapese(w, h)</code>	터틀의 크기를 변경한다.
<code>pos(), position()</code>	터틀의 현재 위치를 구한다.
<code>xcor()</code>	터틀의 x 좌표를 구한다.
<code>ycor()</code>	터틀의 y 좌표를 구한다.
<code>heading()</code>	터틀이 현재 바라보는 각도를 구한다.

# 도형 그리기

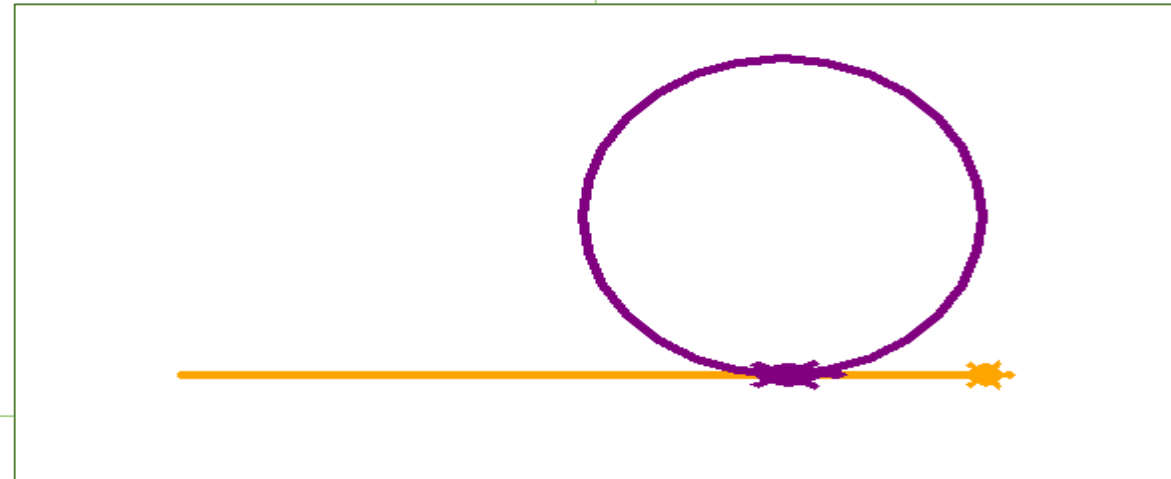
```
import turtle
t = turtle.Turtle()
turtle.setup(width = 500, height = 500)

t.shape("turtle"); t.color("orange")
t.width(5); t.speed(1)

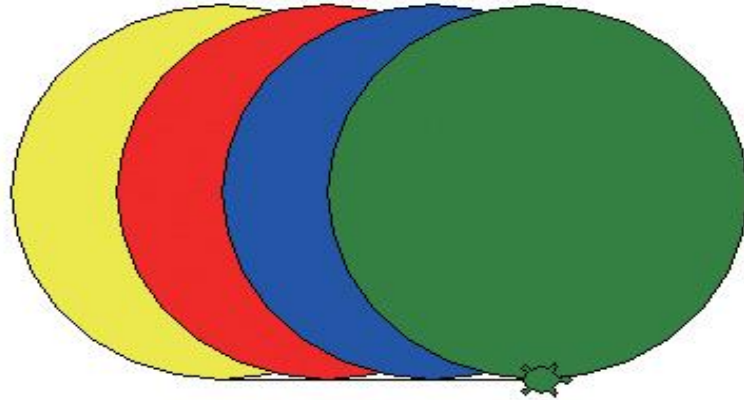
t.fd(400); t.stamp(); t.backward(100); t.speed(10)

t.color("purple")
t.shapesize(1,2)
t.circle(100)

turtle.done()
```



# 도형 그리기



리스트에 색상을 아래와 같이 저장

```
c_list = [ "orange", "cyan", "yellowgreen" ]
```

거북이의 채우기 색상으로 설정하고 원을 그려 보자.

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

c_list = [ "orange", "cyan", "yellowgreen" ]

t.fillcolor(c_list[0])

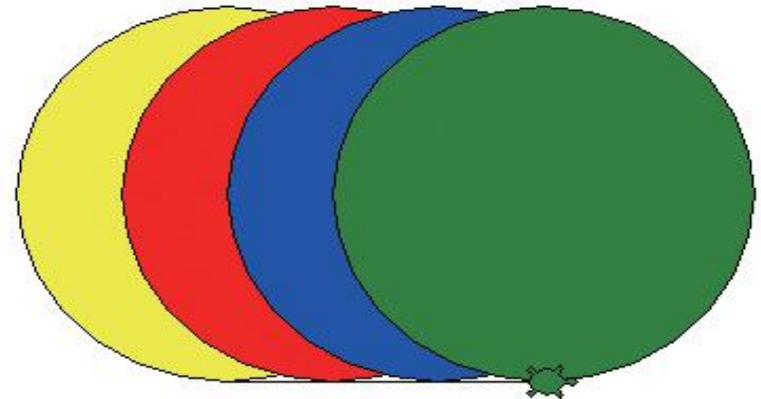
t.begin_fill()
t.circle(100)
t.end_fill()

turtle.done()
```

```
import random  
number = random.randint(0,2)
```

```
t.fd(50)  
t.fillcolor(c_list[number])
```

```
t.begin_fill()  
t.circle(100)  
t.end_fill()
```



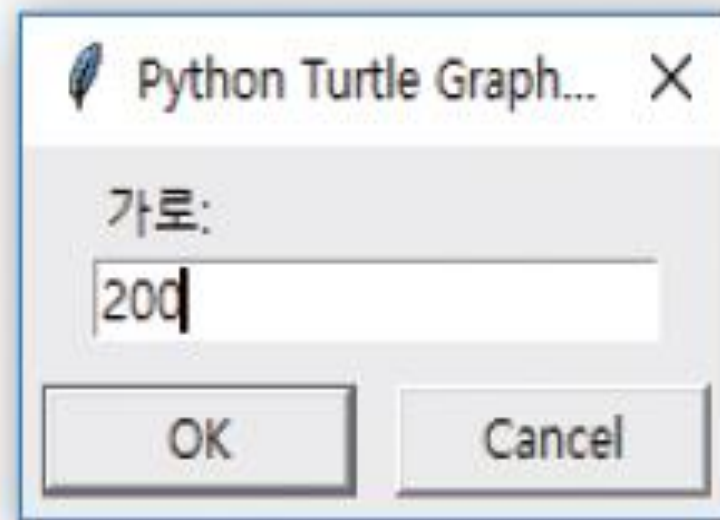
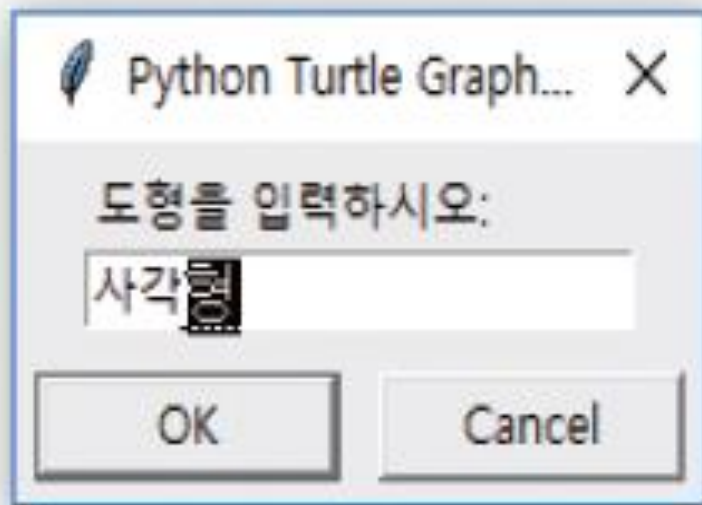
```
c_list = [ "orange", "cyan", "yellowgreen" ]
```

거북이의 채우기 색상은 임의로 설정하고 원을 그려 보자.

# 도전 - 도형 그리기

터틀 그래픽을 이용하여 키보드로부터 **도형 종류**와  
도형의 **한 변의 길이**를 입력받아  
그리는 프로그램을 작성하세요

도형은 “사각형”, “삼각형”, “원” 중의 하나이다.

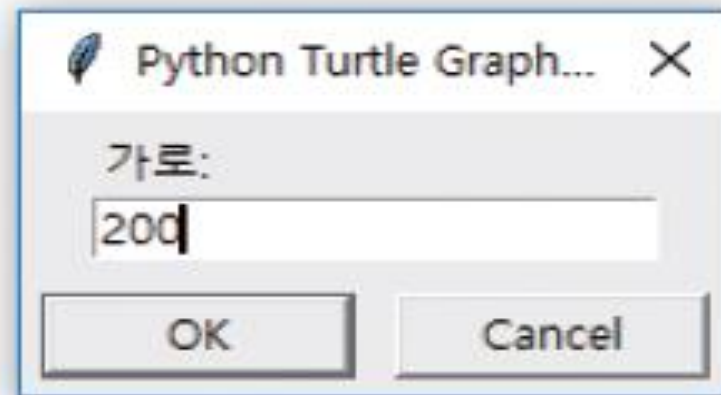
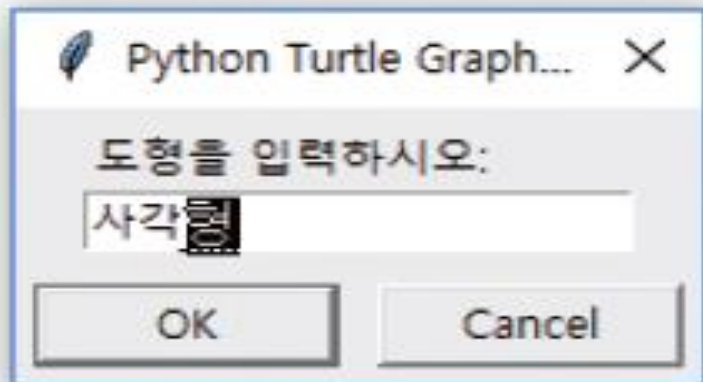


Textinput()

# 도전 - 도형 그리기

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

s = turtle.textinput("도형입력", "도형을 입력하시오: ")
size = turtle.textinput("", "길이: ")
w = int(size)
```





# 도전 - 도형 그리기

```
if s == "사각형" :  
    t.fd(w)  
    t.lt(90)  
    t.fd(w)  
    t.lt(90)  
    t.fd(w)  
    t.lt(90)  
    t.fd(w)  
    t.lt(90)
```

```
if s == "삼각형" :  
    t.fd(w); t.lt(120)  
    t.fd(w); t.lt(120)  
    t.fd(w); t.lt(120)  
  
if s == "원" :  
    t.circle(w)  
  
turtle.done()
```