

# 이번 장에서 만들 프로그램 (1)

## ■ 간단한 계산기 프로그램 소개



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help

===== RESTART: C:/파이썬코드/05-p1.py =====
첫번째 수를 입력하세요 : 12
계산할 연산자를 입력하세요 : /
두번째 수를 입력하세요 : 5
12 / 5 = 2.400000 입니다.
>>>

===== RESTART: C:/파이썬코드/05-p1.py =====
첫번째 수를 입력하세요 : 2
계산할 연산자를 입력하세요 : **
두번째 수를 입력하세요 : 4
2 ** 4 = 16 입니다.
>>> |
```

사용자가 입력한 값

사용자가 입력한 값

Ln: 747 Col: 4

그림 5-1  
간단한 계산기 프로그램

# 이번 장에서 만들 프로그램 (2)

## ■ 종합 계산기 프로그램 소개

입력한 수식대로 계산되는 기능과 입력한 두 수 사이에 있는 모든 수의 합계를 내는 기능을 가진 종합 계산기

```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help

===== RESTART: C:/파이썬코드/05-p2.py =====
1. 수식 계산기 2.두수 사이 합계 : 1
*** 수식을 입력하세요 : 3*4/2-5
3*4/2-5 결과는 1.0 입니다.
>>>

===== RESTART: C:/파이썬코드/05-p2.py =====
1. 수식 계산기 2.두수 사이 합계 : 2
*** 첫번째 숫자를 입력하세요 : 1
*** 두번째 숫자를 입력하세요 : 10
1+...+10는 55입니다.
>>>
```

그림 5-2  
종합 계산기 프로그램

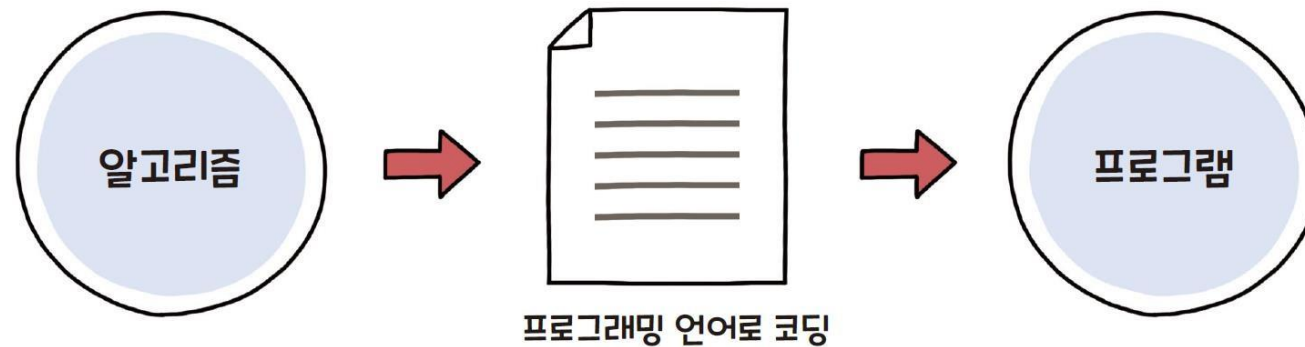
# 알고리즘이란?(1)

## ■ 알고리즘의 개념

특정 업무를 수행하기 위한 절차 또는 명령어의 집합

주어진 문제를 해결하는 데 필요한 방법과 절차가 순서대로 기술한 것

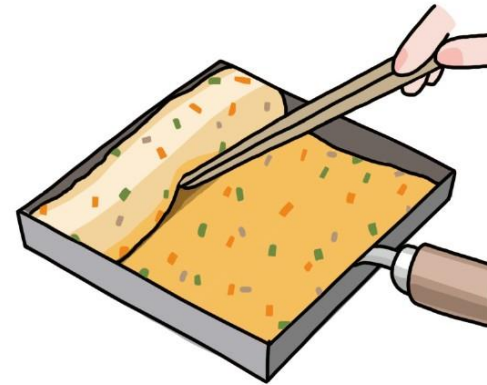
즉 우리가 평소 사용하는 컴퓨터 프로그램은 알고리즘을 프로그래밍 언어를 이용하여 코드로 변환한 것을 의미함



## 알고리즘이란?(2)

### ■ '야채 계란말이' 알고리즘의 예시

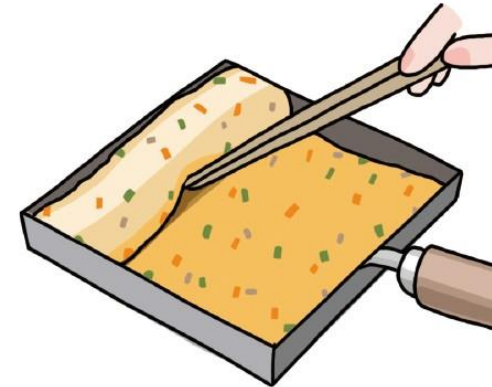
- ① 냉장고에서 '야채'와 '계란'을 꺼낸다.
- ② 야채를 깨끗이 씻고 잘게 다진다.
- ③ 계란과 야채를 그릇에 넣고 섞는다.
- ④ 가스 불을 켜다.
- ⑤ 후라이팬에 기름을 두른다.
- ⑥ 계란 물은 부은 후 어느 정도 익으면 돌돌 만든다.
- ⑦ 계란말이가 완성되면 먹기 좋게 자른 후 그릇에 담는다.



# 알고리즘이란?(3)

## ■ 더 완벽한 '야채 계란말이' 알고리즘의 예시

- ① 냉장고에 '야채'와 '계란'이 있으면 꺼낸다.
- ② 만약 재료가 없다면 마트에 가서 구입한다.
  - ②-1 시간적 여유가 있다면 저렴하게 구입할 수 있는 대형 마트에서 구입한다.
  - ②-2 시간적 여유가 없다면 집에서 가까운 슈퍼에서 구입한다.
- ③ 야채를 깨끗이 씻고 잘게 다진다.
- ④ 계란과 야채를 그릇에 넣고 섞는다.
- ⑤ 가스 불을 켜다.
- ⑥ 후라이팬에 기름을 두른다.
- ⑦ 계란 물은 부은 후 어느 정도 익으면 돌돌 만든다.
- ⑧ 계란말이가 완성되면 먹기 좋게 자른 후 그릇에 담는다.
- ⑨ 케첩을 좋아하면 뿌리고 싫어하면 생략한다.





# 알고리즘이란?



# 알고리즘이란?(4)

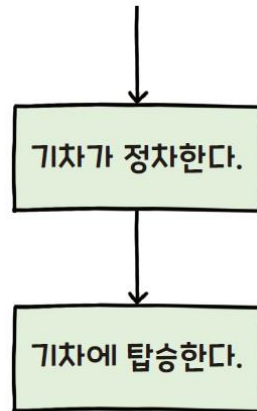
## ■ 알고리즘 설계

순차 구조 : 문제를 해결하기 위해 가장 효율적인 방법을 찾아내는 과정

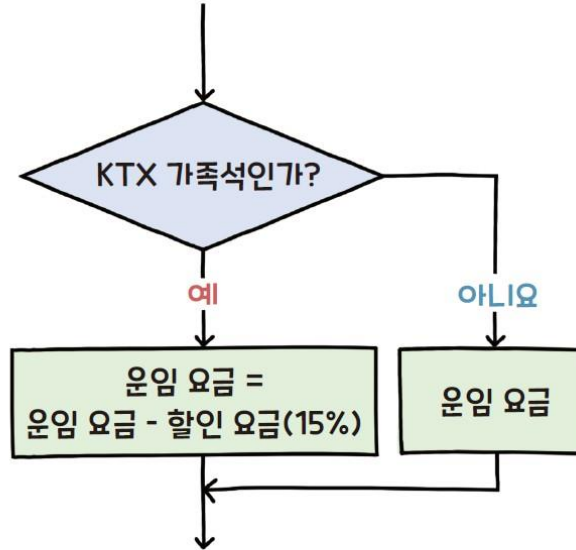
선택 구조 : 특정 조건을 만족하는지 여부에 따라 다음 명령을 선택하여 실행

반복 구조 : 어떤 문제를 해결하기 위한 같은 동작을 반복하여 실행

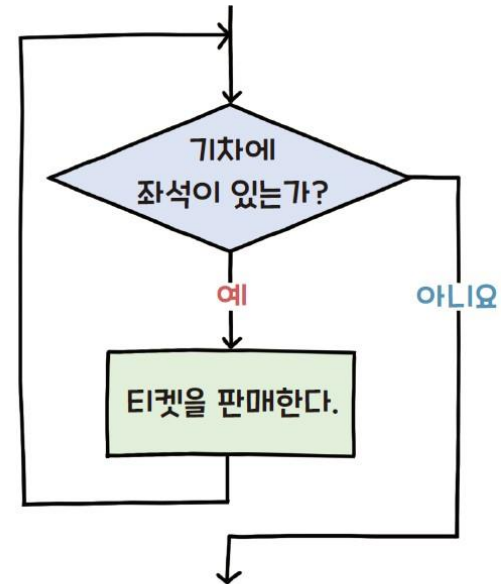
- 기차 탑승 과정의 제어 구조 예시



순차 구조



선택 구조



반복 구조

# 순서도란?(1)

## ■ 순서도의 개념

다양한 기호를 사용하여 알고리즘을 표현한 것

순서도 역시 명확하고 실현 가능한 흐름으로 작성해야 함





# 순서도란?(2)

## ■ 순서도 기호

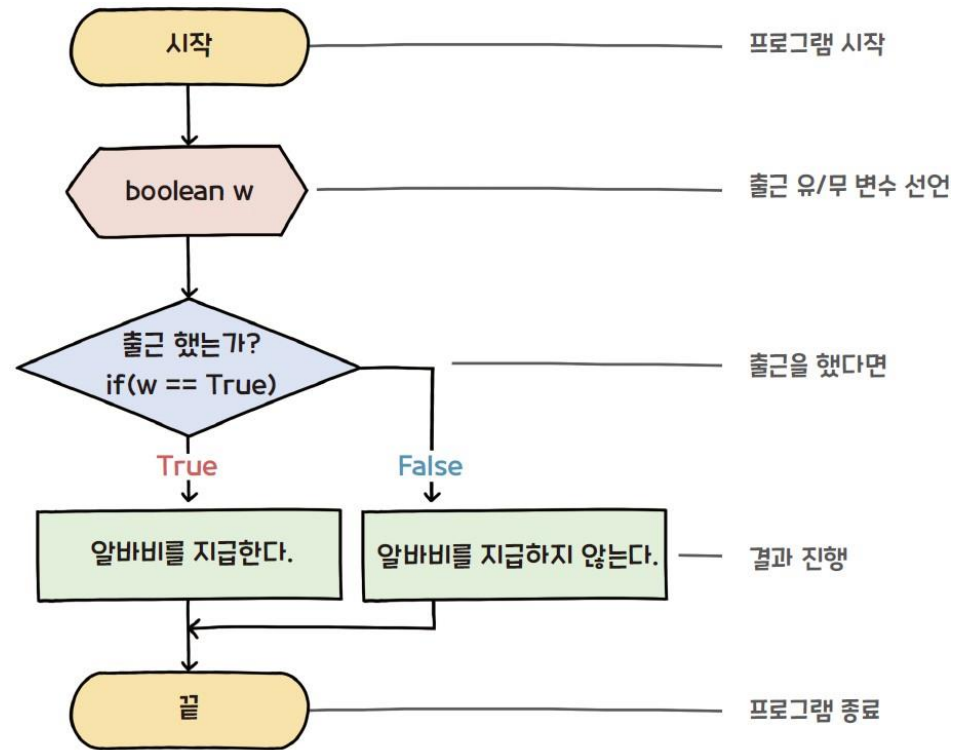
기호	이름	설명
	흐름선	작업의 흐름을 나타낸다.
	단말	순서도의 시작과 끝을 나타낸다.
	준비	작업 단계 시작 전 준비를 나타낸다.
	처리	처리해야 할 작업을 명시한다.
	판단	조건 연산을 나타낸다. 연산 결과(True, False)에 따라 흐름선이 선택된다.
	입출력	데이터의 입력과 출력을 나타낸다.
	서브루틴	다른 곳에 정의된 서브프로그램을 호출한다.



# 순서도 작성(1)

## ■ 아르바이트비 지급 순서도

출근을 했다면 아르바이트비를 지급하고, 출근하지 않았다면 지급하지 않음

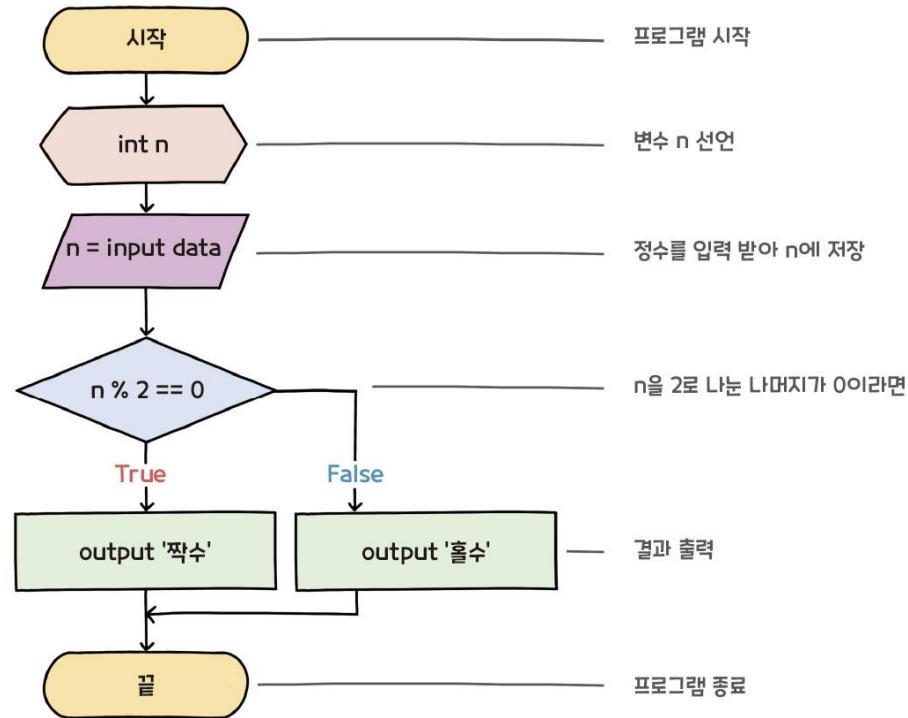


# 순서도 작성(2)

## ■ 홀짝 구분하기

입력한 숫자가 홀수인지 짝수인지 구분하는 순서도

사용자가 입력한 정수를 2로 나눈 나머지가 '0'이면 '짝수', '1'이면 '홀수'

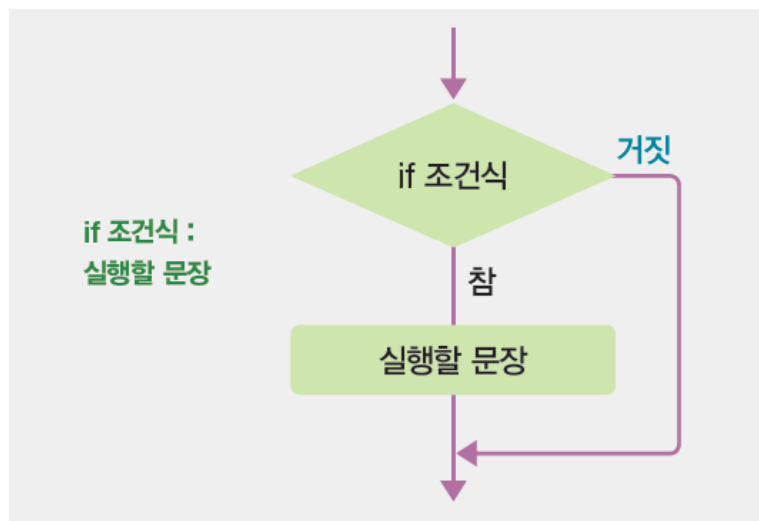


# if문으로 참과 거짓을 가려봅시다 (1)

## ■ 기본 if문

if 조건식 : 에서 조건식이 참이면 실행할 문장이 처리되고, 거짓이면 아무것도 실행하지 않고 프로그램을 종료

그림 5-3  
if문의 형식과 순서도



```
a=99
if a<100 :
    print("100보다 작군요.")
```

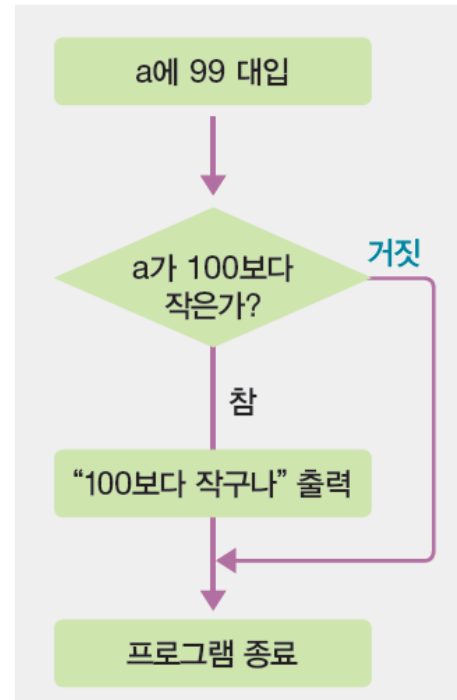
출력 결과

100보다 작군요.

# if문으로 참과 거짓을 가려봅시다 (2)

## ■ 실행 과정

그림 5-4  
if문의 실행 과정



# if문으로 참과 거짓을 가려봅시다 (3)

## 소스코드 5-1

(파일명 : 05-01.py)

```
1 a=200
2
3 if a<100 :
4     print("100보다 작군요")
5 print("거짓이므로 이 문장은 안 보이겠죠?")
6
7 print("프로그램 끝")
```

## 출력 결과

```
거짓이므로 이 문장은 안 보이겠죠?
프로그램 끝
```

- 들여 쓰기를 하지 않아 실행 하지 않아야 할 5행까지 실행 됨.
- 다음과 같이 줄 바꿈을 수정하여 실행





# if문으로 참과 거짓을 가려봅시다 (4)

소스코드 5-2  
(파일명 : 05-02.py)

```
1 a=200
2
3 if a<100 :
4     print("100보다 작군요")
5     print("거짓이므로 이 문장은 안보이겠죠?")
6
7 print("프로그램 끝")
```

출력 결과

프로그램 끝

- 들여쓰기 잘못으로 오류 발생한 경우

```
if a<100 :
    print("100보다 작군요")
    print("거짓이므로 이 문장은 안보이겠죠?")
```



# if문으로 참과 거짓을 가려봅시다 (5)

## ■ if~else 문

참일 때 실행하는 문장과 거짓일 때 실행하는 문장이 다를 때 사용



그림 5-5  
if~else문의 형식과 순서도

# if문으로 참과 거짓을 가려봅시다 (6)

```
1 a=200
2
3 if a<100 :
4     print("100보다 작군요.")
5 else :
6     print("100보다 크군요.")
```

출력 결과

100보다 크군요.

소스코드 5-3  
(파일명 : 05-03.py)

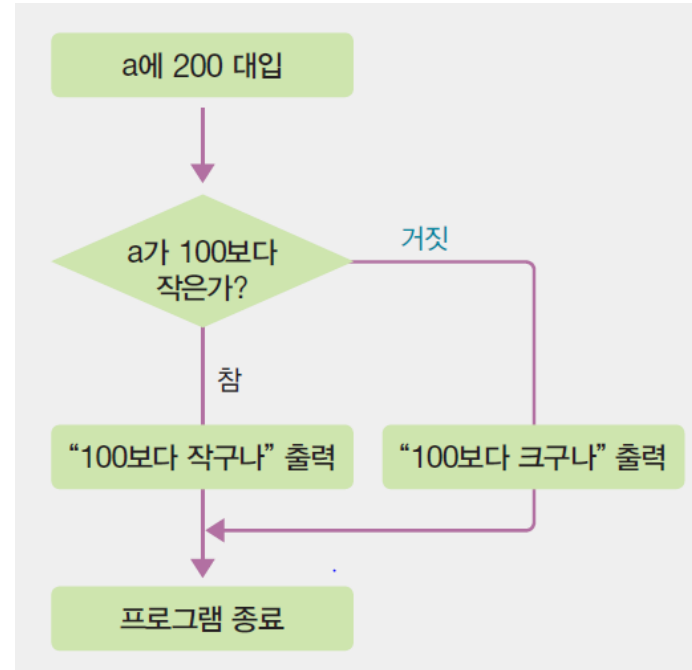


그림 5-6  
[소스코드 5-3]의 실행 과정

# if문으로 참과 거짓을 가려봅시다 (7)

**소스코드 5-4**  
(파일명 : 05-04.py)

```
1  a=200
2
3  if a<100 :
4      print("100보다 작군요.")
5      print("참이면 이 문장도 보이겠죠?")
6  else :
7      print("100보다 크군요.")
8      print("거짓이면 이 문장도 보이겠죠?")
9
10 print("프로그램 끝!")
```

**출력 결과**

```
100보다 크군요.
거짓이면 이 문장도 보이겠죠?
프로그램 끝!
```

## if문으로 참과 거짓을 가려봅시다 (8)

### ■ 입력한 숫자가 짝수인지 홀수인지 계산하는 프로그램

**소스코드 5-5**  
(파일명 : 05-05.py)

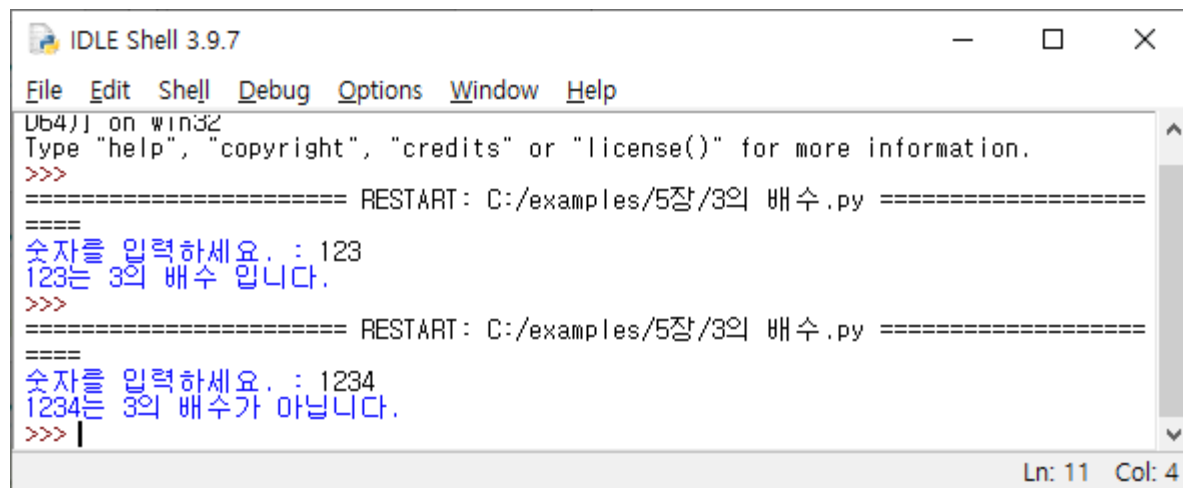
```
1 a=int(input("정수를 입력하세요 : "))
2
3 if a%2==0 :
4     print("짝수를 입력했군요.")
5 else :
6     print("홀수를 입력했군요.")
```

**출력 결과**

정수를 입력하세요 : 125 ← 사용자가 입력한 값  
홀수를 입력했군요.

# 파이썬 연습문제\_1

- 입력한 숫자가 3의 배수인지를 판단하는 프로그램을 작성하세요.



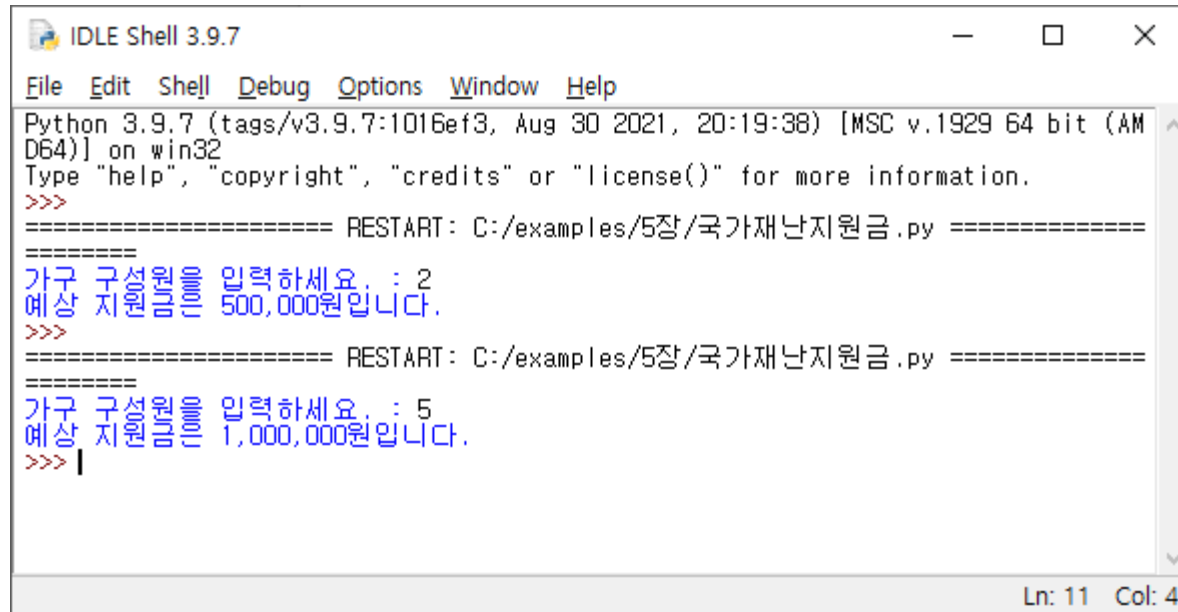
```
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 Shell on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/examples/5장/3의 배수.py =====
>>>
숫자를 입력하세요. : 123
123은 3의 배수 입니다.
>>>
===== RESTART: C:/examples/5장/3의 배수.py =====
>>>
숫자를 입력하세요. : 1234
1234는 3의 배수가 아닙니다.
>>> |
```

Ln: 11 Col: 4



## 파이썬 연습문제\_2

- 국가 재난지원금은 한 가구 구성원 1명당 250,000원씩 최대 4인까지 수령할 수 있습니다. 가구 구성원을 입력하면 수령하게 될 예상 재난지원금을 출력하도록 작성해봅시다.

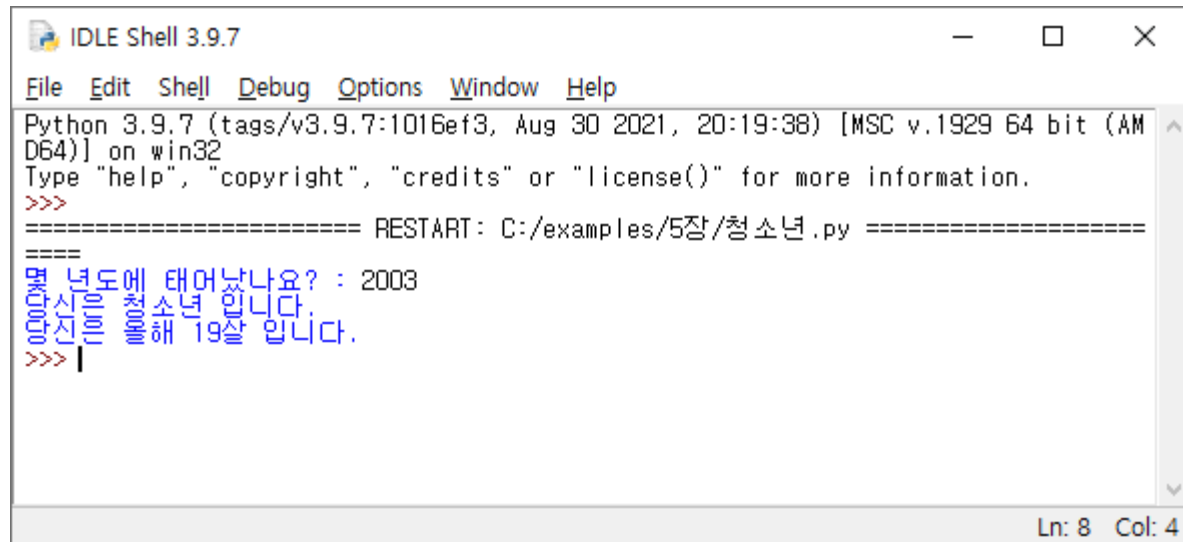


```
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/examples/5장/국가재난지원금.py =====
=====
가구 구성원을 입력하세요. : 2
예상 지원금은 500,000원입니다.
>>>
===== RESTART: C:/examples/5장/국가재난지원금.py =====
=====
가구 구성원을 입력하세요. : 5
예상 지원금은 1,000,000원입니다.
>>> |
```

Ln: 11 Col: 4

## 파이썬 연습문제\_3

- 출생 연도를 물어보고 입력받아 나이를 계산한 후 나이가 15세 이상이고 20세 미만이면 “당신은 청소년입니다.”라고 출력하는 프로그램을 만들어보시오.



```
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/examples/5장/청소년.py =====
>>>
몇 년도에 태어났나요? : 2003
당신은 청소년입니다.
당신은 올해 19살 입니다.
>>> |
```

Ln: 8 Col: 4

# if문 안에 if문이 들어가면 어떻게 될까요?(1)

## ■ if ~ else ~ if ~ else문

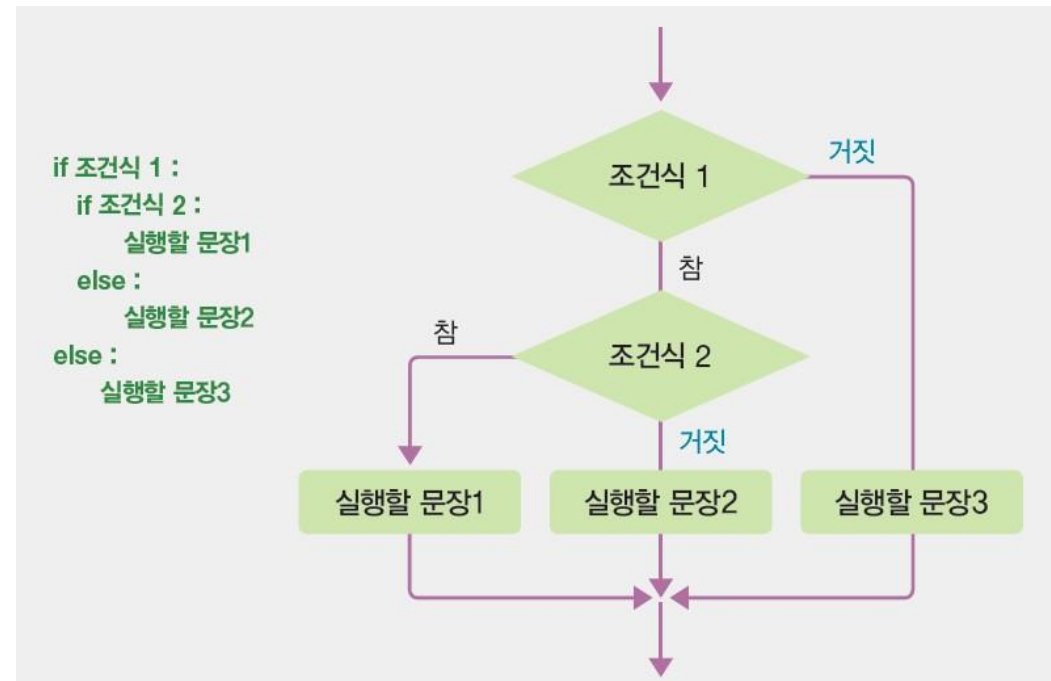


그림 5-7

중첩 if문의 형식과 순서도

## if문 안에 if문이 들어가면 어떻게 될까요?(2)

소스코드 5-6  
(파일명 : 05-06.py)

```
1 a=75
2
3 if a>50 :
4     if a<100 :
5         print("50보다 크고 100보다 작군요.")
6     else :
7         print("와~~ 100보다 크군요.")
8 else :
9     print("에고~ 50보다 작군요..")
```

출력 결과

50보다 크고 100보다 작군요.

- 3행에서 a가 50보다 크면 참이므로 들여쓰기가 된 부분(4행~7행)의 내용을 실행
- 그 안에서 a가 100보다 작아서 5행을 출력

# if문 안에 if문이 들어가면 어떻게 될까요?(3)

- 중첩 if문의 실제 사례

**소스코드 5-7**  
(파일명 : 05-07.py)

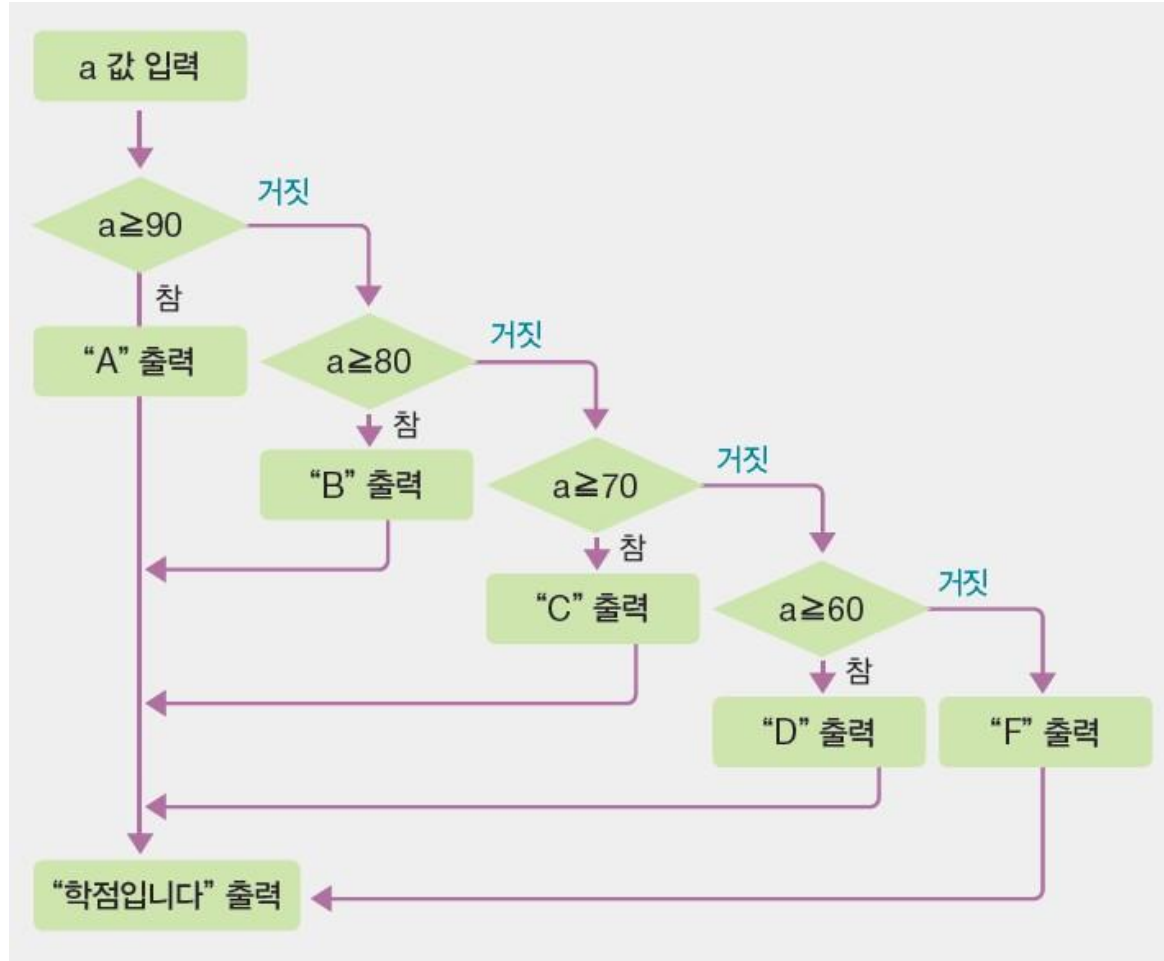
```
1 score=int(input("점수를 입력하세요 : "))
2
3 if score>=90 :
4     print("A")
5 else :
6     if score>=80 :
7         print("B")
8     else :
9         if score>=70 :
10            print("C")
11        else :
12            if score>=60 :
13                print("D")
14            else :
15                print("F")
16
17 print("학점입니다. ^^")
```

**출력 결과**

```
점수를 입력하세요 : 85 ← 사용자가 입력한 값
B
학점입니다. ^^
```

# if문 안에 if문이 들어가면 어떻게 될까요?(4)

그림 5-8  
[소스코드 5-7]의 실행 과정





# if문 안에 if문이 들어가면 어떻게 될까요?(5)

## ■ if ~ elif ~ else문

```
1 score=int(input("점수를 입력하세요 : "))
2
3 if score>=90 :
4     print("A")
5 elif score>=80 :
6     print("B")
7 elif score>=70 :
8     print("C")
9 elif score>=60 :
10    print("D")
11 else :
12    print("F")
13
14 print("학점입니다. ^^")
```

**소스코드 5-8**  
(파일명 : 05-08.py)



# if문 안에 if문이 들어가면 어떻게 될까요?(6)

## ■ 간단한 계산기 프로그램 완성

소스코드 5-9

(파일명 : 05-09.py)

```
1  ## 변수 선언 부분
2  a, b, ch=0, 0, ""
3
4  ## 메인(main) 코드 부분
5  a=int(input("첫 번째 수를 입력하세요 : "))
6  ch=input("계산할 연산자를 입력하세요 : ")
7  b=int(input("두 번째 수를 입력하세요 : "))
8
9  if ch=="+" :
10     print(" %d+%d=%d 입니다. " % (a, b, a+b))
11 elif ch=="-" :
12     print(" %d-%d=%d 입니다. " % (a, b, a-b))
```

## if문 안에 if문이 들어가면 어떻게 될까요?(7)

```
13 elif ch=="*" :
14     print(" %d * %d = %d 입니다. " % (a, b, a * b))
15 elif ch== "/" :
16     print(" %d / %d = %f 입니다. " % (a, b, a / b))
17 elif ch=="%" :
18     print(" %d %% %d = %d 입니다. " % (a, b, a % b))
19 elif ch=="//" :
20     print(" %d // %d = %d 입니다. " % (a, b, a // b))
21 elif ch=="**" :
22     print(" %d ** %d = %d 입니다. " % (a, b, a ** b))
23 else :
24     print(" 알 수 없는 연산자 입니다. ")
```



# If문을 응용해봅시다 (1)

## ■ 리스트와 함께 사용

리스트(List)는 여러 개를 한곳에 담아놓은 것

```
fruit=['사과', '배', '딸기', '포도']  
print(fruit)
```

출력 결과

```
['사과', '배', '딸기', '포도']
```

'리스트이름.append(항목)' 함수

```
fruit.append('귤')  
print(fruit)
```

출력 결과

```
['사과', '배', '딸기', '포도', '귤']
```



## If문을 응용해봅시다 (2)

리스트 확인은 if문을 사용함

```
if '딸기' in fruit :  
    print ("딸기가 있네요 ^^")
```

출력 결과

```
딸기가 있네요 ^^
```

'if 항목 in 리스트 :' 는 리스트 안에 항목이 있으면 True를 반환



# If문을 응용해봅시다 (3)

- 0부터 9까지 숫자 중에서 리스트 안에 없는 숫자를 찾아내는 프로그램

**소스코드 5-10**  
(파일명 : 05-10.py)

```
1 import random
2
3 numbers=[]
4 for num in range(0, 10) :
5     numbers.append(random.randrange(0, 10))
6
7 print("생성된 리스트", numbers)
8
9 for num in range(0, 10) :
10     if num not in numbers :
11         print ("%d 숫자는 리스트에 없네요." % num)
```

**출력 결과**

```
생성된 리스트 [5, 8, 8, 7, 8, 1, 9, 0, 0, 4]
2 숫자는 리스트에 없네요.
3 숫자는 리스트에 없네요.
6 숫자는 리스트에 없네요.
```



## If문을 응용해봅시다 (4)

- 'random.randrange(시작, 끝)' 함수는 '시작'부터 '끝-1'까지 숫자 중에서 임의의 숫자를 하나 반환
- 1행 : randrange() 함수를 사용하기 위해 필요
- 4행과 9행 : 0부터 9까지 총 10회를 반복.
- 5행 : random.randrange(0,10)은 0부터 9까지 숫자 중에서 임의의 숫자를 반환
- 7행 : 생성된 리스트 출력, 이 숫자들은 실행할 때마다 다름.
- 9행 : 0부터 9까지 num에 넣은 후 10행에서 numbers 리스트에 그 숫자가 없다면 11행에서 숫자가 없다는 메시지를 출력



# If문을 응용해봅시다 (5)

## ■ 종합 계산기 프로그램 완성

소스코드 5-11  
(파일명 : 05-11.py)

```
1  ## 변수 선언 부분
2  select, answer, numStr, num1, num2=0, 0, "", 0, 0
3
4  ## 메인(main) 코드 부분
5  select=int(input("1. 수식 계산기  2.두 수 사이 합계  : "))
6
7  if select==1 :
8      numStr=input(" *** 수식을 입력하세요  : ")
9      answer=eval(numStr)
10     print(" %s 결과는 %5.1f 입니다. " % (numStr, answer))
11 elif select==2 :
12     num1=int(input(" *** 첫 번째 숫자를 입력하세요  : "))
13     num2=int(input(" *** 두 번째 숫자를 입력하세요  : "))
14     for i in range(num1, num2+1) :
15         answer=answer+i
16     print(" %d+...+%d는 %d입니다. " % (num1, num2, answer))
17 else :
18     print("1 또는 2만 입력해야 합니다.")
```

## If문을 응용해봅시다 (6)

- 실행 후에 1을 입력하면 수식을 계산하는 계산기로 작동
- 8행 : 수식 자체를 입력
- 9행 : eval(수식) 함수는 수식을 계산해주는데 아주 유용한 함수
- 실행 후에 2를 입력하면 두 숫자 사이의 모든 수의 합계를 구함
- 14행 : for문 안에 range( )로 입력한 숫자 사이를 반복
- 17행 : 1, 2 외의 숫자를 입력하면 처리하는 오류 메시지

