

07

Conditional Statement

Conditional Statement

- 조건문이란?
- 조건에 따라 코드 실행을 선택적으로 제어하는 구조
- if, elif, else 키워드 사용

if 조건:

실행할 코드

- 조건이 참(True)이면 들여쓰기된 블록 실행
- 조건이 거짓이면 건너뛴다

Conditional Statement

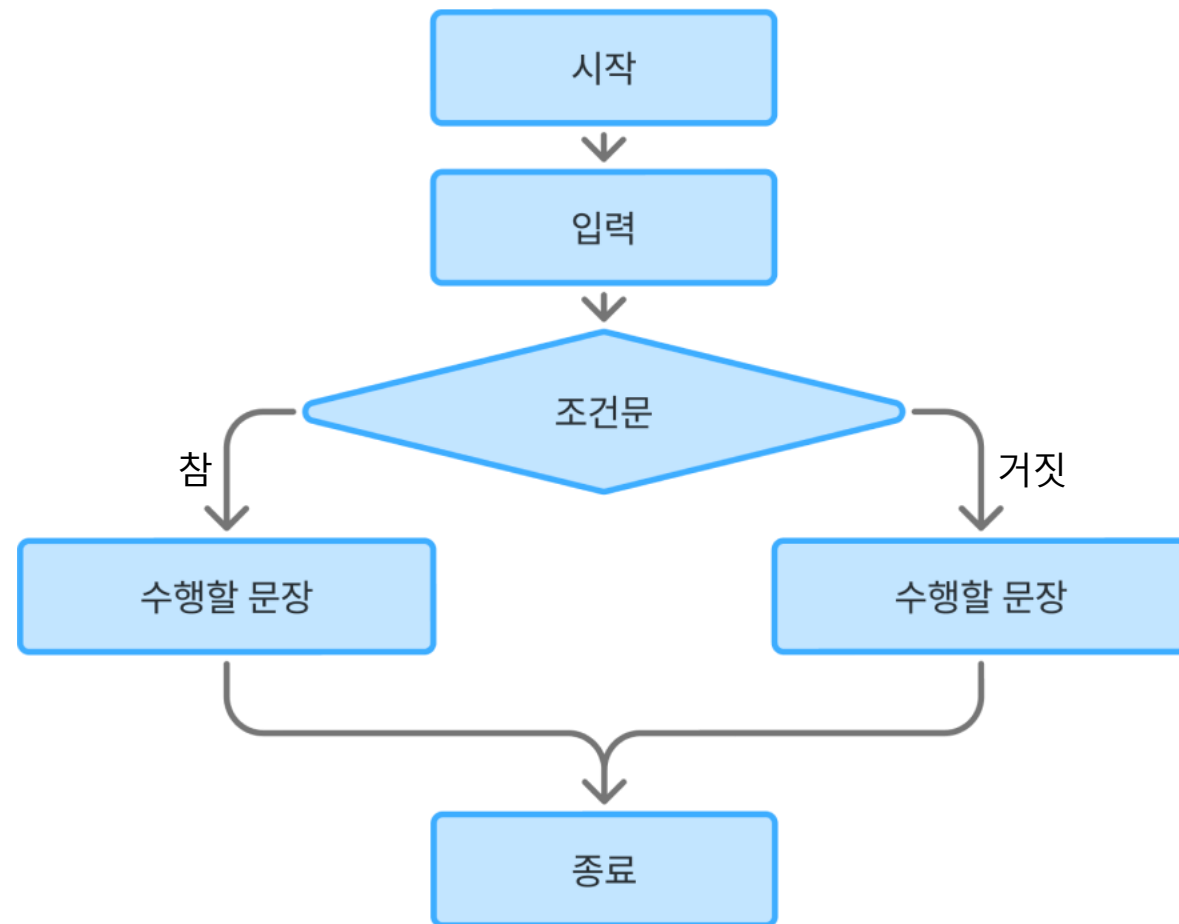
```
>>> num = 10
>>> if num > 5:                # 10 > 5: True
...     print("5보다 큼니다")
...
5보다 큼니다
```

```
>>> num = 3
>>> if num > 5:                # 3 > 5: False
...     print("5보다 큼니다")
...
>>>
```

if-else

- 조건이 참이면 실행1, 거짓이면 실행2

```
if 조건:  
    실행1  
else:  
    실행2
```

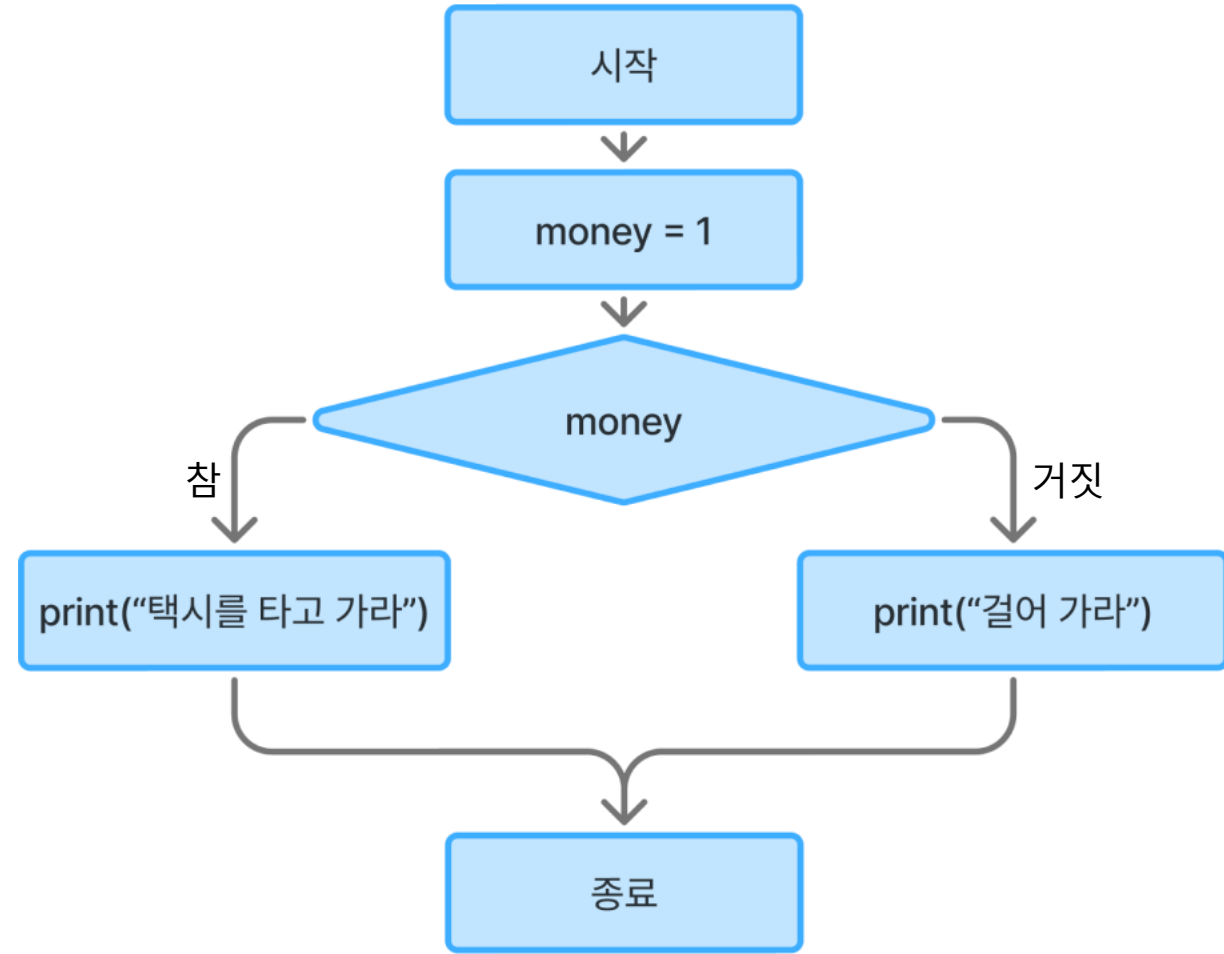


if-else

“돈이 있으면 택시를 타고,
돈이 없으면 걸어 간다.”

```
money = 1
if money:
    print("택시를 타고 가라")
else:
    print("걸어 가라")

택시를 타고 가
라
```

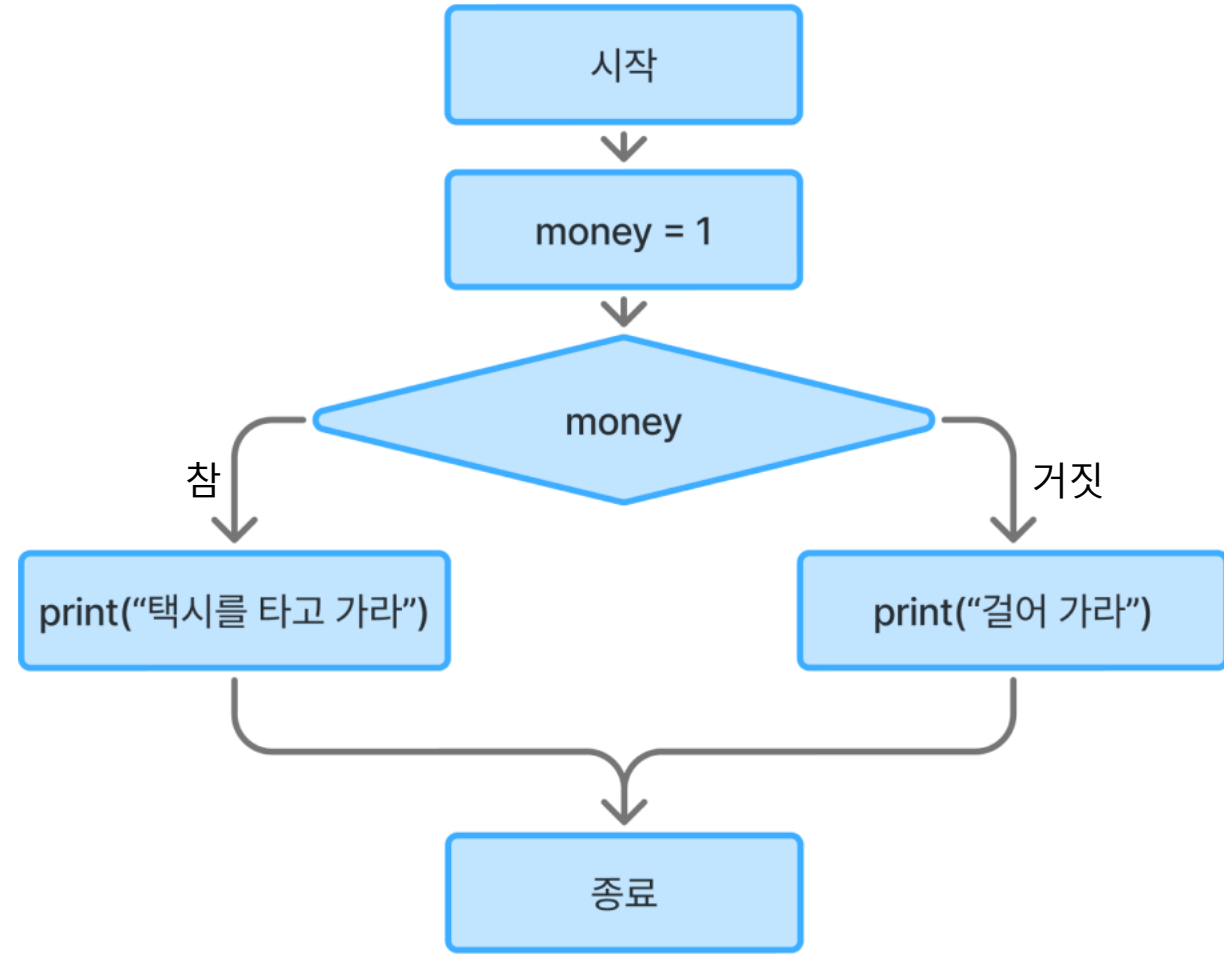


if-else

“돈이 있으면 택시를 타고,
돈이 없으면 걸어 간다.”

```
money = 0
if money:
    print("택시를 타고 가라")
else:
    print("걸어 가라")

걸어 가
라
```



Boolean

자료형	참	거짓
숫자	0이 아닌 숫자	0
문자열	" abc"	""
리스트	[1, 2, 3]	[]
튜플	(1, 2, 3)	()
딕셔너리	{" a" : " b" }	{}

```
>>> print(bool([]))  
False
```

```
>>> print(bool("hi"))  
True
```

if-elif-else

- 여러 조건을 차례대로 검사
- 첫 번째로 참이 되는 조건만 실행

```
if 조건1:
```

```
    실행1
```

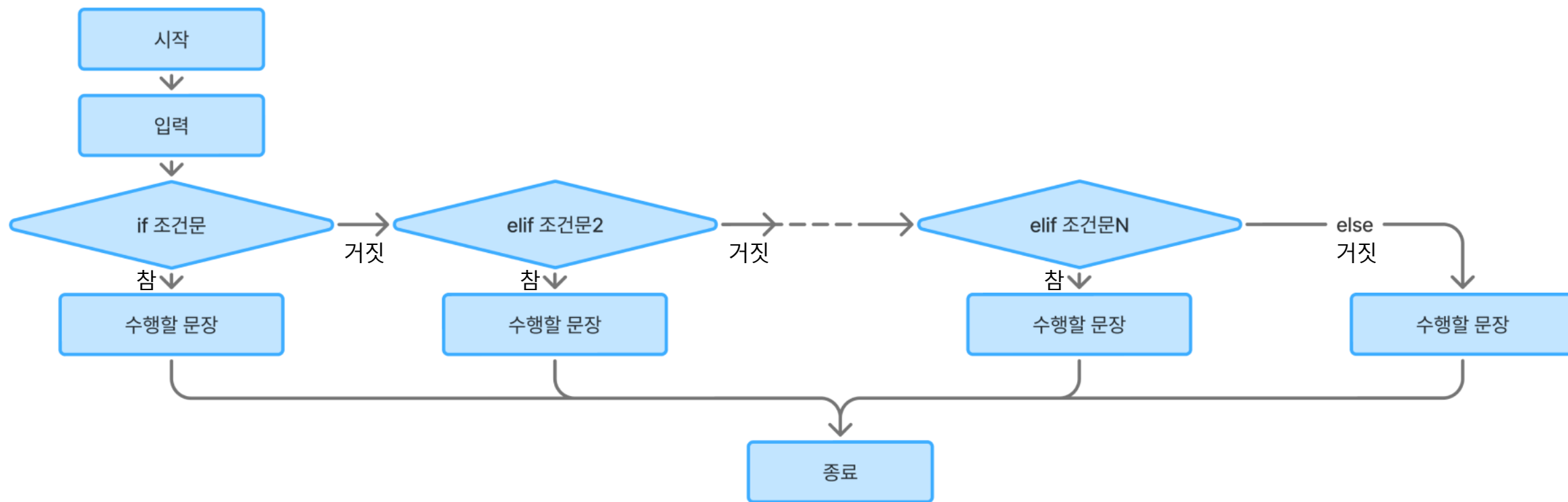
```
elif 조건2:
```

```
    실행2
```

```
else:
```

```
    실행3
```


if-elif-else



Indentation

- 파이썬은 들여쓰기(Indentation)로 코드 블록을 구분
- if, elif, else 뒤에 :를 붙이고, 들여쓰기된 줄이 조건이 참일 때 실행
- 일반적으로 공백 4칸 또는 탭 1칸을 사용

```
>>> if score >= 80:  
... print("Good Score!")  
File "<stdin>", line 2  
    print("Good Score!")  
    ^
```

Indentation Error: expected an indented block after 'if' statement on line 1

Logical / Identity / Membership Operators

Operator	Description	Example
and	두 서술이 모두 참일 때 참 반환	$x < 5$ and $x < 10$
or	두 서술 중 하나라도 참일 때 참 반환	$x < 5$ or $x < 4$
not	서술이 참이면 거짓을, 거짓이면 참을 반환	not($x < 5$ and $x < 10$)
is	두 변수가 같은 객체일 때 참 반환	x is y
is not	두 변수가 같지 않은 객체일 때 참 반환	x is not y
in	특정 값이 객체 안에 존재할 때 참 반환	x in y
not in	특정 값이 객체 안에 존재하지 않을 때 참 반환	x not in y

Logical / Identity / Membership Operators

```
>>> score = 85
>>> if score >= 80 and score < 90:
...     print("B 등급")
...
B 등급
>>> score = 95
>>> if score >= 80 and score < 90:
...     print("B 등급")
...
>>> score = 70
>>> if score >= 80 and score < 90:
...     print("B 등급")
...
>>>
```

Logical / Identity / Membership Operators

- 괄호로 논리 그룹 및 우선순위 명시

```
>>> a, b, c = 1, 3, -1
>>> if (a > 0 and b > 0) or c > 0:
...     print("조건 만족")
... else:
...     print("불충족")
...
조건 만족
```

```
>>> a, b, c = 0, 3, -1
>>> if (a > 0 and b > 0) or c > 0:
...     print("조건 만족")
... else:
...     print("불충족")
...
불충족
```

Conditional Statement

```
age = 20  
gender = "M"
```

```
if age >= 18:  
    if gender == "M":  
        print("성인 남성")
```

성인 남성

```
>>> age, gender = 20, "M"  
>>> if age >= 18:  
...     if gender == "M":  
...         print("성인 남성")  
  
...  
성인 남성
```

input()

- `input(prompt)`: *prompt*를 이용해 화면에 출력할 내용을 전달
- 문자열 형식 반환

쥬피터 노트북에서 셀을 실행했을 때

```
x = input('Enter your name:')  
print('Hello, ' + x)
```

Enter your name:

↓ Alice

```
Enter your name: Alice  
Hello, Alice
```

값을 입력했을 때

input()

```
num = int(input("숫자 입력: "))  
  
if num % 2 == 0:  
    print("짝수")  
else:  
    print("홀수")
```

숫자 입력: 5
홀수

```
num = int(input("숫자 입력: "))  
  
if num % 2 == 0:  
    print("짝수")  
else:  
    print("홀수")
```

숫자 입력: 22
짝수

input()

- if-elif-else 구문을 활용해 간단한 계산기 제작 가능
- 조건문은 선택적 실행 로직을 만들기에 유용

```
a = int(input("a: "))  
b = int(input("b: "))  
op = input("연산자(+,-): ")
```

```
if op == '+':  
    print(a + b)  
elif op == '-':  
    print(a - b)
```

a: 10

b: 3

연산자(+,-):

input()

```
a = int(input("a: "))
b = int(input("b: "))
op = input("연산자(+, -, *, /): ")

if op == '+':
    print(a + b)
elif op == '-':
    print(a - b)
elif op == '*':
    print(a * b)
elif op == '/':
    print(a / b)
else:
    print("not supported operand")
```

```
a: 10
b: 5
연산자(+, -, *, /): *
50
```

7주차 과제

- 07_Conditional_statement.ipynb