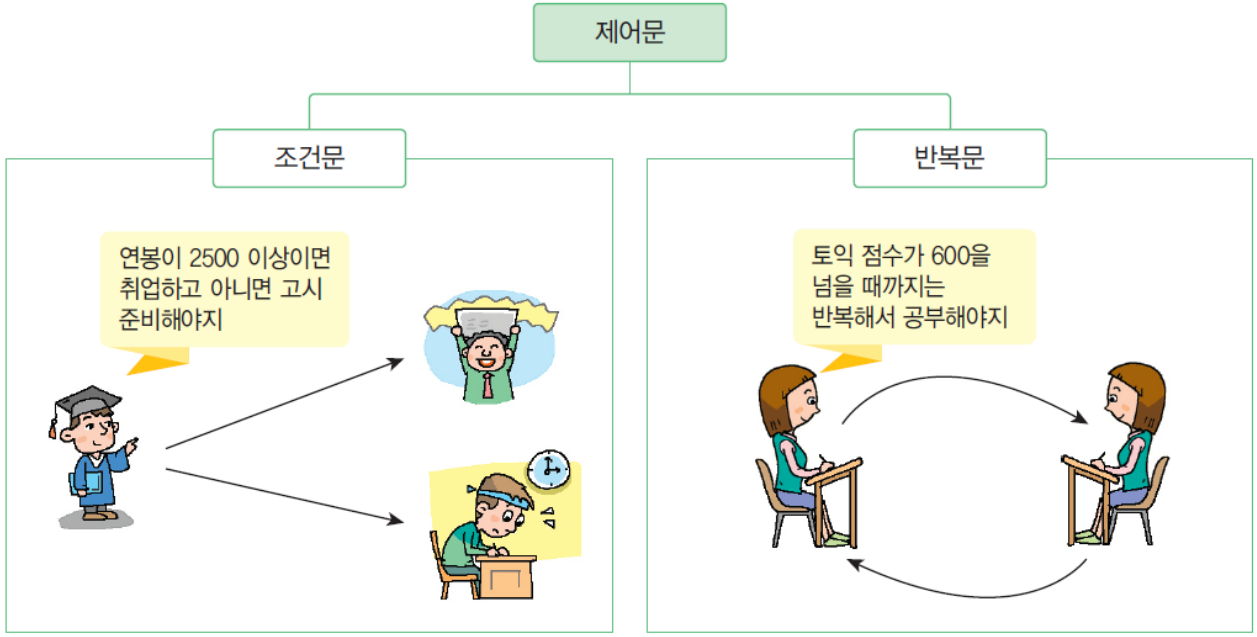
4장 조건문

제어문. p111

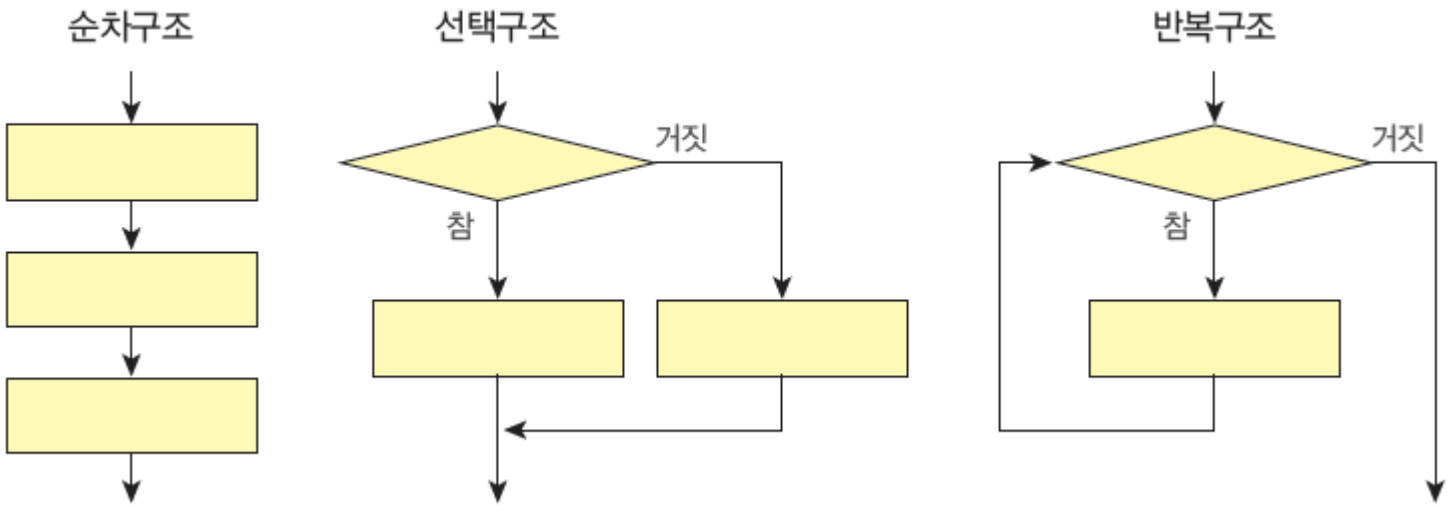
- 제어문(control statement) : 문장들이 실행되는 순서를 제어하는 문장



- 조건문 : 어떤 조건에 따라 문장의 실행 여부가 결정
- 반복문 : 특정 조건에 따라서 문장을 반복하여 실행

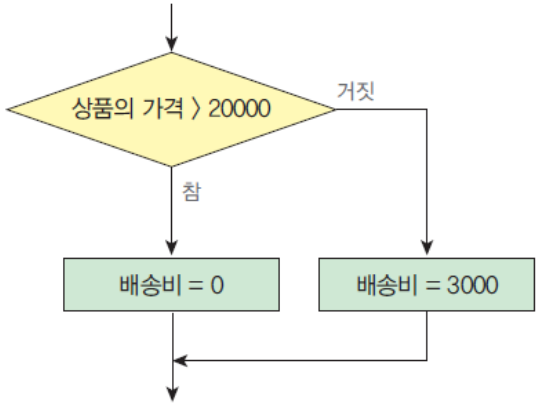
3가지의 제어 구조. p112

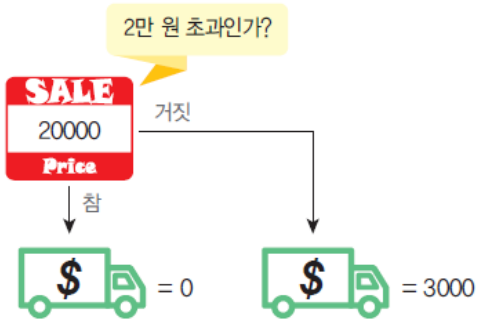
- 프로그램을 작성할 때 사용할 수 있는 3가지의 기본적인 제어 구조



프로그램은 이들 3가지의 구조를 섞어서 만들어진다.

if-else 문. p112





Syntax: if-else 문

형식

```
if 조건식 :
        문장1
else :
        문장2
```

예

```
if price > 20000 :
        shipping_cost = 0
else :
        shipping_cost = 3000
```

- 참이나 거짓으로 계산되는 조건식. 관계 연산자 == != 〈 〉 >= 〈=을 사용한다.
- 콜론(:)은 복합문을 의미한다.
- 조건식이 참이면 실행되는 문장
- 조건식이 거짓이면 실행되는 문장
- else절은 생략될 수도 있다.
- if와 else는 같은 위치여야 한다.

배송비 계산 프로그램. p114

```
# 사용자로부터 상품의 가격을 입력받는다.
price = int(input("상품의 가격: "))

# 배송비를 결정한다.
if  price > 20000 :
                shipping_cost = 0
else :
                shipping_cost = 3000

# 배송비를 출력한다.
print("배송비 = ", shipping_cost)
```

```
상품의 가격: 30000
배송비 =  0
```

블록. p114

- 들여쓰기를 이용하여 문장을 묶는다.

```
if  price > 20000 :
□□□□shipping_cost = 0
□□□□discount = 0.1
else :
    shipping_cost = 3000
```

블록

```
if  price > 20000 :
□□□□shipping_cost = 0
□□□□□discount = 0.1
else :
    shipping_cost = 3000
```

들여쓰기가 달라서 동일한 블록이 아니다.

실행 결과

```
IndentationError: unexpected indent
```

else는 없을 수도 있다. p115

- 만약 else 부분이 필요 없다면 생략할 수 있다.

```
shipping_cost = 3000                # 기본적으로 배송비는 3000원이다.
if  price > 20000 :                 # 만약 상품의 가격이 2만원 초과이면
        shipping_cost = 0           # 배송비가 없다.
```

오류 조심: 들여쓰기

파이썬에서는 들여쓰기가 아주 중요하다. **if-else** 문에서도 들여쓰기가 잘못되면 오류가 발생한다.

```
if  number > 0 :
    print("양수")
else :
    print("0 또는 음수")
```

0 1 들여쓰기 레벨

프로그래밍 힌트: 중복 방지. p116

- 프로그래밍할 때 항상 신경 써야 하는 것은 중복된 문장을 없애는 것이다. 만약 if-else 문에 중복된 문장이 있다면 밖으로 꺼내면 된다.

```
if  price > 20000 :
                shipping_cost = 0
                print("배송비 = ", shipping_cost)
else :
                shipping_cost = 3000
                print("배송비 = ", shipping_cost)
```

```
if  price > 20000 :
                shipping_cost = 0
else :
                shipping_cost = 3000
print("배송비 = ", shipping_cost)
```

프로그래밍 힌트: pass 키워드. p116

- 프로그래밍을 하다보면 일단 전체의 골격을 만들어두고 나중에 세부 사항을 채우는 경우도 많다. 이 때 사용할 수 있는 것이 pass이다. pass 키워드는 “나중에 채워넣겠음”을 의미한다.

```
if  price > 60000 :
            pass        # 아직 미구현 상태이다.
else :
            pass        # 아직 미구현 상태이다.
```

논리 연산자. p117

- 복잡한 조건을 표현할 때 사용

상품의 가격이 2만원 초과, 그리고 "파이썬" 카드이면
-> 배송료가 없음

(상품의 가격이 2만원 초과이다) and ("파이썬" 카드이면)
-> 배송료가 없음

연산	의미
x and y	and 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
x or y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
not x	not 연산, x가 참이면 거짓, x가 거짓이면 참

논리 연산자. p118

- 진리표

A	B	A and B
True	True	True
True	False	False
False	True	False
False	False	False

A	B	A or B
True	True	True
True	False	True
False	True	True
False	False	False

A	not A
True	False
True	True

```
price = int(input("가격을 입력하시오: "))
card = float(input("카드 종류를 입력하시오: "))

if price > 20000 and card == "python" :
                print("배송료가 없습니다.")
else
                print("배송료는 3000원입니다.")
```

```
가격을 입력하시오: 30000
카드 종류를 입력하시오: python
배송료가 없습니다.
```

```
가격을 입력하시오: 30000
카드 종류를 입력하시오: java
배송료는 3000원입니다.
```

드모르간의 법칙. p118

- 인간은 일반적으로 **not** 연산자가 적용된 수식을 이해하기가 어렵다. 논리 학자 드모르간(De Morgan)의 이름을 딴 드모르간의 법칙을 사용하여 이러한 논리식을 단순화할 수 있다.

$$\neg(P \vee Q) \iff (\neg P) \wedge (\neg Q),$$

$$\neg(P \wedge Q) \iff (\neg P) \vee (\neg Q),$$

```
if not (country == "한국" and province != "제주") :
        shipping_cost = 8000
```

```
if country != "한국" or province == "제주" :
        shipping_cost = 8000
```

팁: 가독성. p119

- 프로그램은 다른 사람도 쉽게 읽을 수 있어야 한다. 가독성 있는 코드를 제공하려면 논리식에서 변수를 True나 False와 비교하면 안 된다.

```
if full == False :
        print("가득 차지 않았습니다.)
```

```
If not full  :
        print("가득 차지 않았습니다.)
```

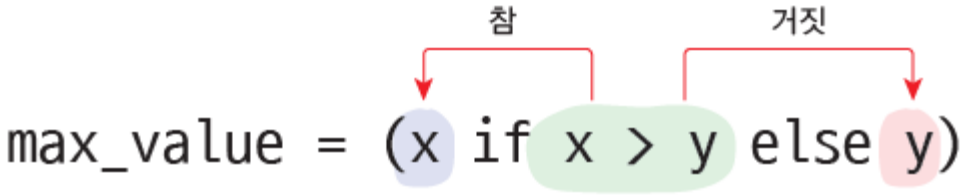
- **참고사항 : 수학과 프로그래밍**

  수학에서는 변수 x의 값이 .....

  If 0 <= x <= 100 :

  If x >= 0 and x <= 100 :

조건 연산자. p120




```
shipping_cost = ( 0 if price >= 20000 else 3000 )
```

```
absolute_value = (x if x > 0 else -x)          // 절대값 계산
max_value = (x if x > y else y)                // 최대값 계산
min_value = (x if x < y else y)                // 최소값 계산
```

- 반드시 조건 연산자를 감싸는 괄호가 있어야 한다. 괄호가 없으면 할당 연산자가 먼저 계산된다.

```
x = int(input("첫 번째 수 ="))
y = int(input("두 번째 수 ="))
max_value = (x if x > y else y)
min_value = (y if x > y else x)
print("큰 수=", max_value, "작은 수=", min_value)
```

```
첫 번째 수 =10
두 번째 수 =20
큰 수= 20 작은 수= 10
```

Lab 산술 퀴즈 프로그램. p122

- 초등학생들을 위하여 산수 퀴즈를 발생시키는 프로그램을 작성해보자. 부울 변수도 사용해본다.

```
25 + 78 = 103
True
```

```
25 + 78 = 100
False
```

```
##
# 이 프로그램은 산수 문제를 출제한다.
#
import random

x = random.randint(1, 100)
y = random.randint(1, 100)

answer = int(input(f"{x} + {y} = "))

# 부울 변수에 결과를 저장하고 출력한다.
flag = (answer == (x+y))
print(flag)
```

Lab 동전 던지기 게임. p123

- 동전을 던지기 게임을 작성해보자.

```
동전 던지기 게임을 시작합니다.
뒷면입니다.
게임이 종료되었습니다.
```

```
import random

print("동전 던지기 게임을 시작합니다.")
coin = random.randrange(2)

if coin == 0 :
        print("앞면입니다.")
else :
        print("뒷면입니다.")
print("게임이 종료되었습니다.")
```

0부터 1 사이의 난수를 생성하는 문장이다.
randint(0, 1)과 같다.

Lab 로그인 프로그램. p124

- 사용자로부터 아이디를 받아서 프로그램에 저장된 아이디와 일치하는 지 여부를 출력하는 프로그램을 작성해보자.

```
아이디를 입력하시오: ilovepython
환영합니다.
```

```
아이디를 입력하시오: iloveruby
아이디를 찾을 수 없습니다.
```

```
id = "ilovepython"
s = input("아이디를 입력하시오: ")

if s == id:
        print("환영합니다.")
else:
        print("아이디를 찾을 수 없습니다.")
```

중첩 if 문. p125

- if 문 안에 다른 if 문이 들어갈 수도 있다.

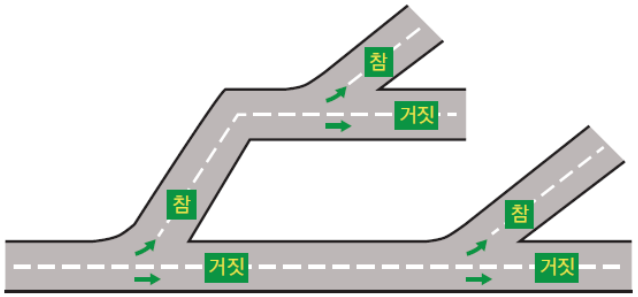
Syntax: 중첩 if 문

형식

```
if 조건식1 :
    if 조건식2 :
        문장1
    else :
        문장2
else :
    if 조건식3 :
        문장3
    else :
        문장4
```

조건식1이 참일 때 실행

조건식1이 거짓일 때 실행



배송비 계산 프로그램. p126

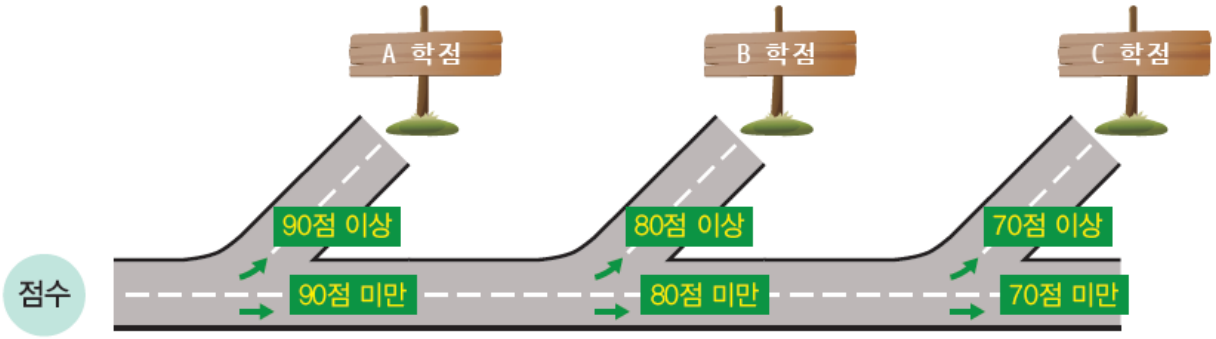
- 배송지가 한국이면 다음과 같이 배송비가 결정된다. - ”상품의 가격이 2만원 이상이면 배송비는 없고 그렇지 않으면 3000원의 배송비가 붙는다.“
- 배송지가 미국이면 다음과 같이 배송비가 결정된다. - ”상품의 가격이 10만원 이상이면 배송비는 없고 그렇지 않으면 8000원의 배송비가 붙는다.“

```
# 사용자로부터 상품의 가격을 입력받는다.
country = input("배송지(현재는 korea와 us만 가능): ")
price = int(input("상품의 가격: "))

# 배송비를 결정한다.
if  country == "korea" :
        if  price >= 20000 :
                shipping_cost = 0
        else :
                shipping_cost = 3000
else :
        …
        …
```

```
배송지(현재는 korea와 us만 가능): us
상품의 가격: 120000
배송비 = 0
```

연속 if 문. p126



- 연속 if 문에서는 순서가 아주 중요하다. – p127. 예제 참조

학점 결정 프로그램. p128

```
# 성적을 받아서 학점을 결정하는 프로그램

score = int(input("성적을 입력하시오: "))

if score >= 90 :
        print("학점 A")
elif score >= 80 :
        print("학점 B")
elif score >= 70 :
        print("학점 C")
elif score >= 60 :
        print("학점 D")
else :
        print("학점 F")
```

```
성적을 입력하시오: 88
학점 B
```

Lab 지진 상황 출력하기. p129

- 사용자로부터 지진의 리히터 규모를 받아서 그 영향을 출력하는 프로그램을 작성

리히터 규모	영향
2.0 미만	지진계에 의해서만 탐지 가능합니다.
2.0–3.9	물건들이 흔들리거나 떨어집니다.
4.0–6.9	빈약한 건물에 큰 피해가 있습니다.
7.0–7.9	지표면에 균열이 발생합니다.
8.0–9.0	대부분의 구조물이 파괴됩니다.

```
리히터 규모를 입력하시오: 5.2
빈약한 건물에 큰 피해가 있습니다.
```

Lab 오늘의 운세 출력하기. p130

- 조건문을 이용하여서 오늘의 운세를 알려주는 프로그램을 개발해보자. 난수를 발생하여서 난수에 해당하는 운세를 출력한다.

```
##
#            이 프로그램은 오늘의 운세를 출력한다.
#
import random

print("행운의 매직볼로 오늘의 운세를 출력합니다. ")
answers = random.randint(1, 8)
if answers == 1:
    print("확실히 이루어집니다.")
elif answers == 2:
    print("좋아 보이네요")
elif answers == 3:
    print("믿으셔도 됩니다.")
elif answers == 4:
    print("저의 생각에는 no입니다.")
else:
    print("다시 질문해주세요.")
```

```
행운의 매직볼로 오늘의 운세를 출력합니다.
확실히 이루어집니다.
```

Lab 도형 그리기. p131

- 터틀 그래픽을 이용하여 사용자가 선택하는 도형을 화면에 그리는 프로그램을 작성해보자. 도형은 “사각형”, “삼각형”, “원” 중의 하나이다.

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

s = input( "도형을 입력하시오: ")

if s == "사각형" :
    w = int(turtle.textinput("","가로: "))
    h = int(turtle.textinput("","세로: "))
    color = input(“색상”)
    t.pencolor(color)
    t.forward(w)
    t.left(90)
    t.forward(h)
    t.left(90)
    …
    …
```

Mini Project 가위, 바위, 보 게임. p132

- **(직접 해보자)** 컴퓨터와 가위, 바위, 보 게임을 하는 프로그램을 작성하라. 컴퓨터는 사용자에게 알리지 않고 가위, 바위, 보 중 에서 임의로 하나를 선택한다. 사용자는 프로그램의 입력 안내 메시지에 따라서, 3개 중에서 하나를 선택하게 된다. 사용자의 선택이 끝나면 컴퓨터는 누가가 무엇을 선택하였 고 누가 이겼는지, 비겼는지를 알려준다.

```
선택하시오(1: 가위 2:바위 3:보) 1
컴퓨터의 선택(1: 가위 2:바위 3:보) 2
컴퓨터가 이겼음
```

연습문제. p134

Programming. p137