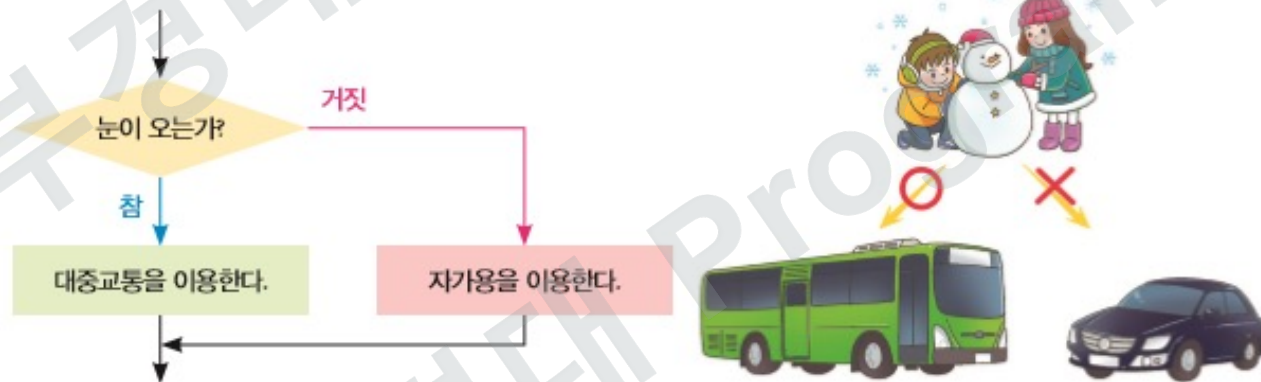


조건문

- 우리가 문제를 해결할 때 어떤 조건에 따라서 두 개 또는 여러 개의 실행 경로 가운데 하나를 선택해야 하는 경우가 종종 있다.



if-else 문

전체적인 구조



조건식

```
if score >= 60 :
```

콜론은 아직 문장이 종료되지 않았다는 것을 의미한다.

```
    print("합격입니다.")
```

조건식이 참일 때 실행되는 문장, then 절이라고 한다.

```
else :
```

```
    print("불합격입니다.")
```

조건식이 참이 아닐 때 실행되는 문장, else 절이라고 한다.

'그렇지 않으면'이라는 의미이다.

관계 연산자

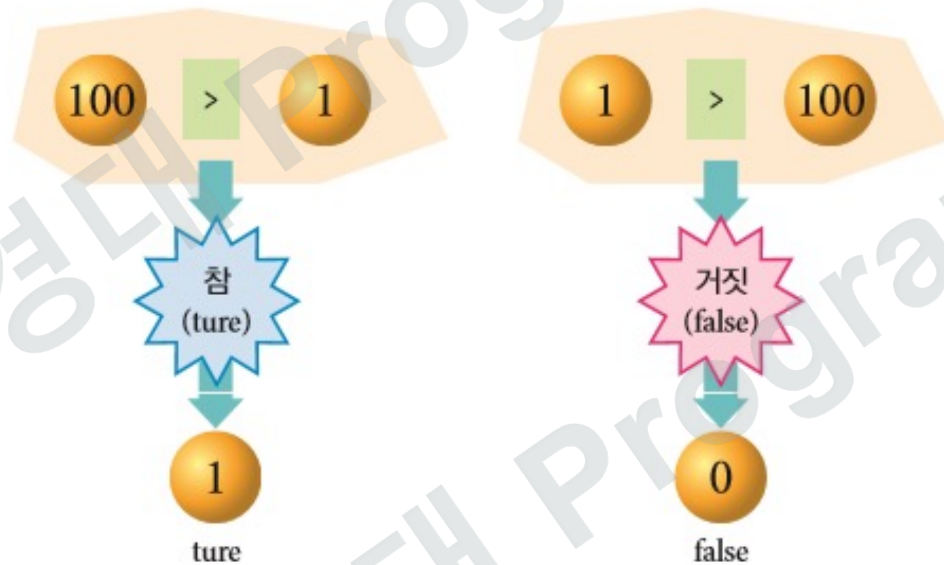
- 관계 연산자(relational operator)는 두 개의 피연산자를 비교하는데 사용된다. 예를 들면 "변수 x가 변수 y보다 큰지"를 따지는데 사용된다.

표 3.1 관계 연산자

연산	의미
$x == y$	x와 y가 같은가?
$x != y$	x와 y가 다른가?
$x > y$	x가 y보다 큰가?
$x < y$	x가 y보다 작은가?
$x >= y$	x가 y보다 크거나 같은가?
$x <= y$	x가 y보다 작거나 같은가?

관계 연산자

- 관계 수식은 참이나 거짓이라는 값을 생성한다



예제

```
>>> age = 19
>>> if( age >= 19) :
    print("마트에서 주류를 구입할 수 있습니다.")
else :
    print("조금 기다리세요!")
```

마트에서 주류를 구입할 수 있습니다.

else는 생략할 수 있다!

```
>>> food = '스테이크'
>>> if food == '스테이크':
    print('내가 제일 좋아하는 음식!')
    print(10 * food)
```

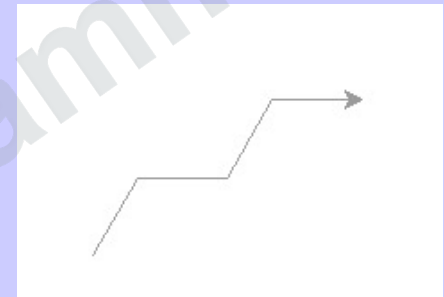
내가 제일 좋아하는 음식!

스테이크스테이크스테이크스테이크스테이크스테이크스테이크스테이크
스테이크스테이크

Lab: 터틀 그래픽

- 사용자로부터 명령어를 받아서 터틀을 제어해보자. 즉 사용자가 "left"를 입력하면 왼쪽으로 회전하게 하고 사용자가 "right"를 입력하면 오른쪽으로 회전하게 하자.

왼쪽=left, 오른쪽=right:left
왼쪽=left, 오른쪽=right:right
왼쪽=left, 오른쪽=right:left
왼쪽=left, 오른쪽=right:right
왼쪽=left, 오른쪽=right:



Lab: 수하물 비용 계산

- 항공사에서는 짐을 부칠 때, 20kg이 넘어가면 20,000원을 내야한다고 하자. 20kg 미만이면 수수료는 없다. 사용자로부터 짐의 무게를 입력받고 사용자가 지불하여야 할 금액을 계산하는 프로그램을 작성해보자.

짐의 무게는 얼마입니까? 18
짐에 대한 수수료는 없습니다.
감사합니다.

짐의 무게는 얼마입니까? 30
무거운 짐은 20,000원을 내셔야 합니다.
감사합니다.

Lab: 수하물 비용 계산

- 키보드에서 입력받은 정수가 홀수인지 짝수인지를 말해주는 프로그램을 작성하여 보자. 홀수와 짝수는 어떻게 구별할 수 있는가?

정수를 입력하시오: 2
입력된 정수는 짝수입니다.

Lab: 수하물 비용 계산

- 사용자로부터 두개의 정수를 입력받아서 둘 중에서 큰 수를 출력하는 프로그램을 작성하여 보자.

첫 번째 정수: 10
두 번째 정수: 20
큰 수는 20

Lab: 수하물 비용 계산

- 인터넷 쇼핑몰에서 물건을 구입할 때, 구입액이 10만원 이상이면 5%의 할인을 해준다고 하자. 사용자에게 구입 금액을 물어보고 최종적으로 할인 금액과 지불 금액을 출력하는 프로그램을 작성해보자.

구입 금액을 입력하시오: 100500
지불 금액은 95475.0 입니다.

Lab: 문자열의 중간문자

- 문자열의 중앙에 있는 문자를 출력한다. 예를 들어서 문자열이 "weekday"이라면 중앙의 문자는 "k"가 된다. 하지만 만약 문자열이 짝수개의 문자를 가지고 있다면 중앙의 2개의 글자를 출력한다. 예를 들어서 "string" 문자열에서는 "ri"를 반환한다.

문자열을 입력하시오: *weekday*
중앙글자는 *k*

문자열을 입력하시오: *string*
중앙글자는 *ri*

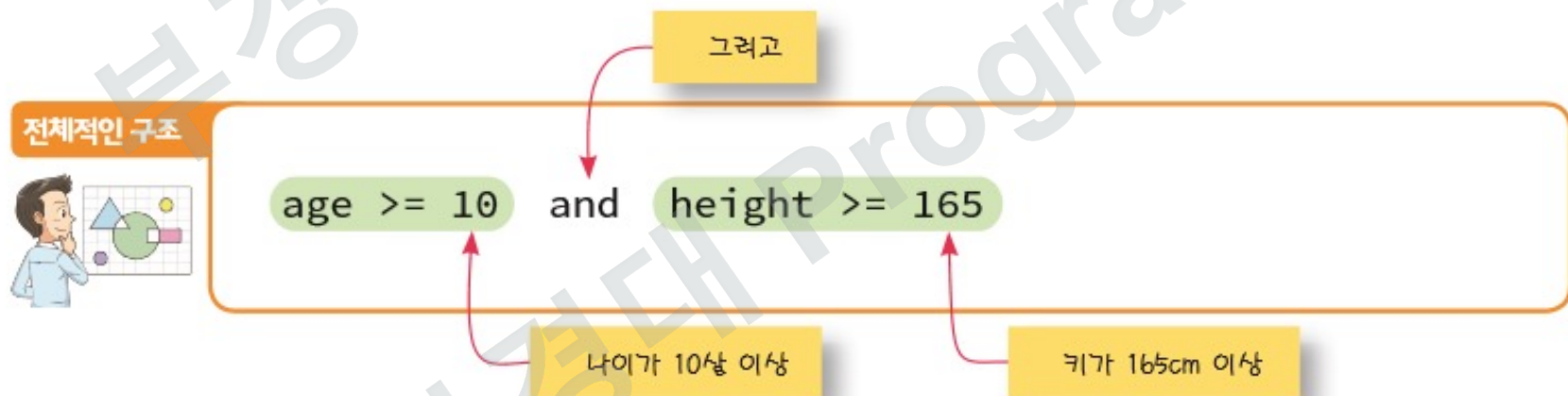
논리 연산자


- 논리 연산자(logical operator)는 여러 개의 조건을 조합하여 참인지 거짓인지를 따질 때 사용한다. AND(논리곱), OR(논리합), NOT(논리부정)이 있다.

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

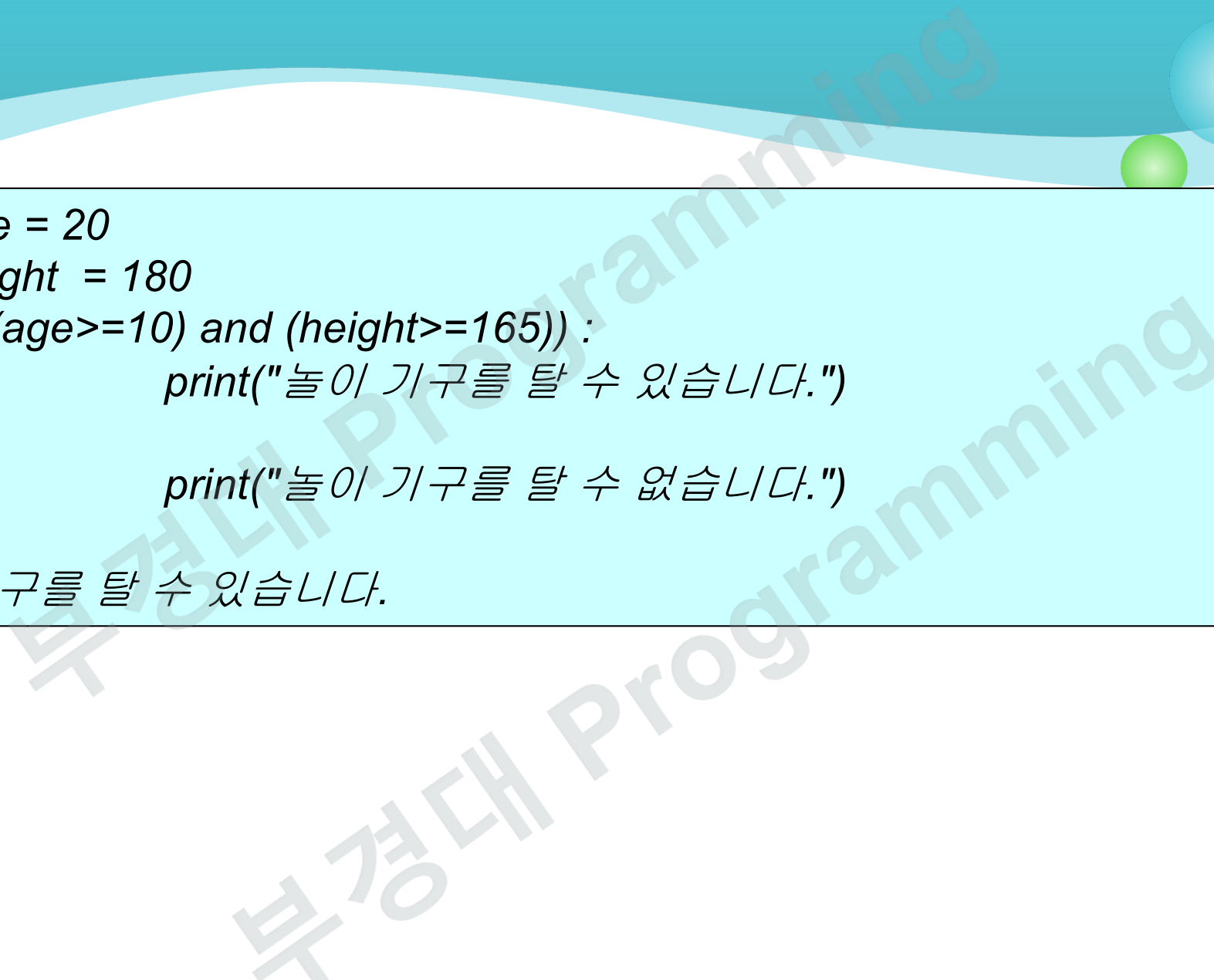
- 논리 연산자(logical operator)는 여러 개의 조건을 조합하여 참인지 거짓인지를 따질 때 사용한다. AND(논리곱), OR(논리합), NOT(논리부정)이 있다.
- 놀이공원에서 놀이기구를 탈 수 있는 조건을 논리 수식으로 작성하여 보면 다음과 같다.





```
>>> age = 20
>>> height = 180
>>> if( (age>=10) and (height>=165)) :
        print("놀이 기구를 탈 수 있습니다.")
else :
        print("놀이 기구를 탈 수 없습니다.")

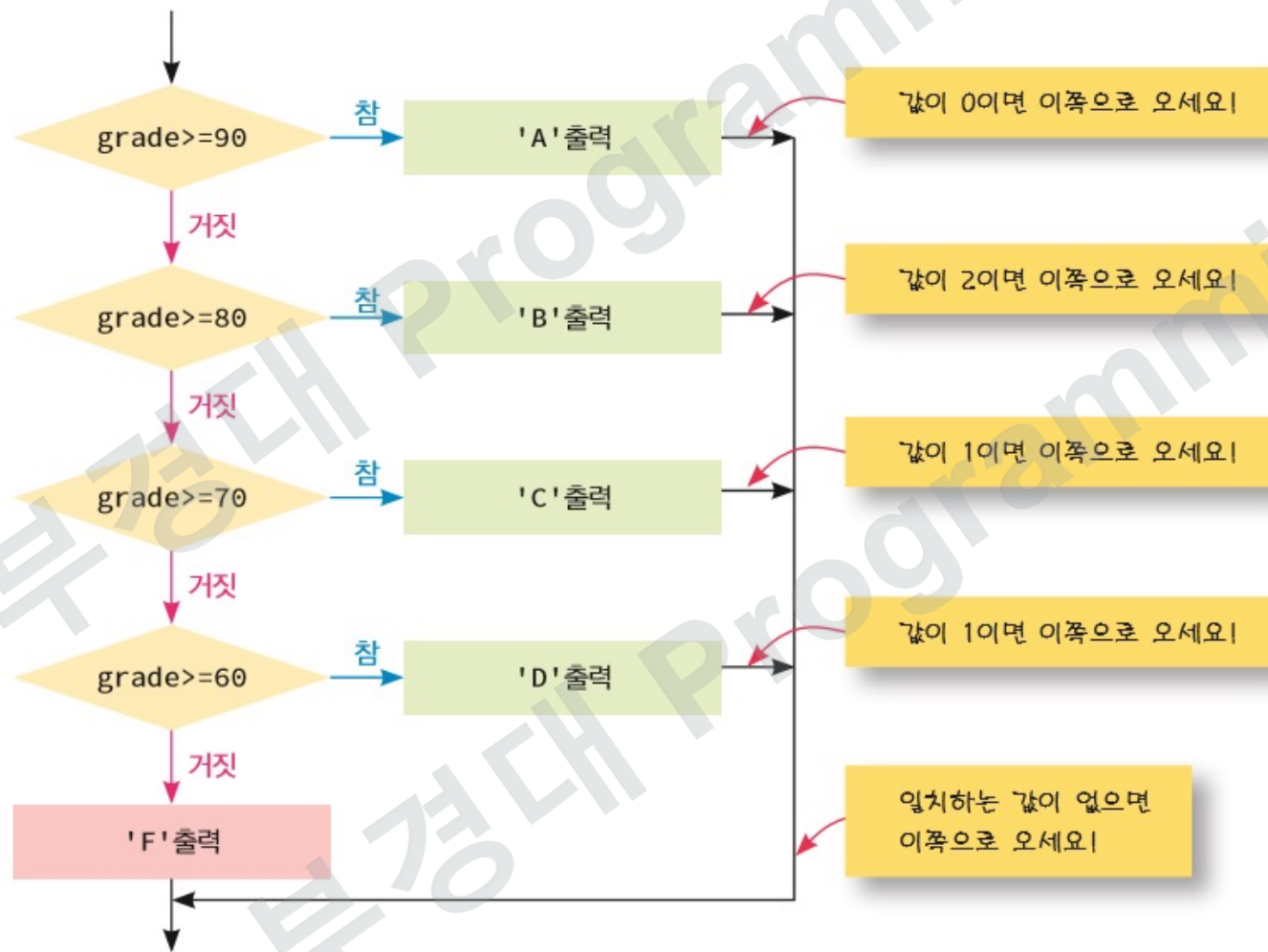
놀이 기구를 탈 수 있습니다.
```



Lab: 졸업 학점 검사하기

- 어떤 대학교를 졸업하려면 적어도 140학점을 이수해야 하고 평점이 2.0은 되어야 한다고 하자. 이것을 파이썬 프로그램으로 검사해보자. 사용자에게 이수학점수와 평점을 물어보고 졸업 가능 여부를 출력하는 프로그램을 작성해보자.

이수한 학점수를 입력하시오: 120
평점을 입력하시오: 2.3
졸업이 힘듭니다!



예제 소스

```
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: 90

학점 A

Lab: 음수, 0, 양수 구별하기

- 사용자로부터 정수를 받아서 음수, 0, 양수 중의 하나로 분류하여 보자.

정수를 입력하시오: -10
입력된 정수는 음수입니다.

정수를 입력하시오: 0
입력된 정수는 0입니다.

중첩 if-else 문

- 우리는 if-else 문장 안에 다른 if-else 문장을 넣을 수 있다. 이것을 중첩(nesting)된 if-else문이라고 부른다.

전체적인 구조



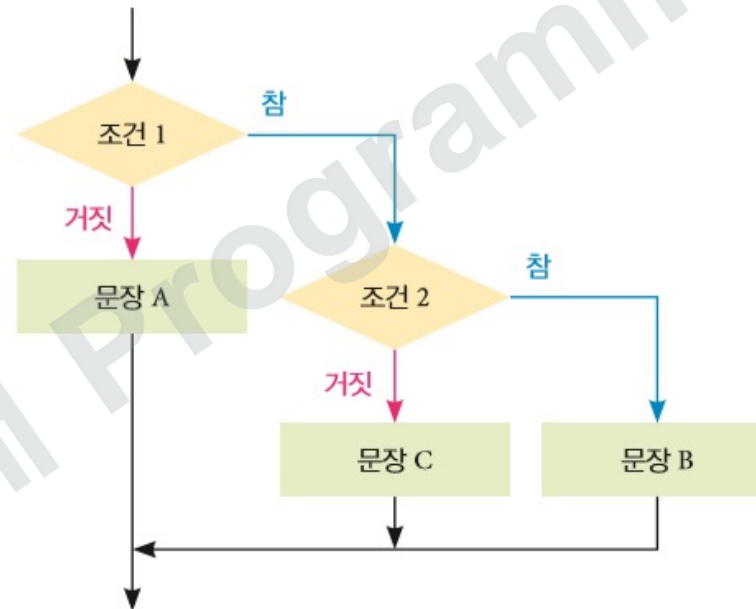
```
if 조건1 :  
    문장_A
```

```
else:
```

```
    if 조건2 :  
        문장_B
```

```
    else:
```

```
        문장_C
```



예제

- 예를 들어보자. 마트에서 사과가 신선하면 사과를 사기로 한다. 만약 사과가 개당 1000원 미만이면 10개를 산다. 하지만 사과가 개당 1000원 이상이면 5개만 산다.

```
appleQuality = input("사과의 상태를 입력하시오: ")
applePrice = int(input("사과 1개의 가격을 입력하시오: "))
if appleQuality == "신선":
    if applePrice < 1000:
        print("10개를 산다")
    else:
        print("5개를 산다")
else:
    print("사과를 사지 않는다.")
```

Lab: 음수, 0, 양수 구별하기

- 앞에서 사용자가 입력한 수가 양수인지 0인지 음수인지를 구별하여 출력하는 프로그램을 작성해보았다. 여기서는 중첩 if-else문을 사용하여 동일한 기능을 하는 프로그램을 작성해보자.

정수를 입력하시오: -10
입력된 정수는 음수입니다.

정수를 입력하시오: 0
입력된 정수는 0입니다.

Lab: 아이디 검사

- 아이디를 입력받아서 등록된 아이디인지를 검사하는 프로그램을 작성해보자. 등록된 아이디를 리스트(list)에 저장하도록 한다. 아이디가 일치하면 패스워드 물어본다.

아이디를 입력하시오: hong
패스워드를 입력하시오: 12345678
환영합니다.

Lab: 달의 일수 출력

- 1년의 각 달의 일수를 출력하는 프로그램을 작성하여보자. 즉 특정 달이 입력되면 그 달의 일수를 출력한다. 여러 가지 방법으로 작성할 수 있겠으나 여기서는 if-else 문을 사용하여 보자.

월을 입력하시오:12
월의 날수는 31

Lab: 윤년 판단

- 입력된 연도가 윤년인지 아닌지를 판단하는 프로그램을 만들어보자. 윤년은 다음의 조건을 만족해야 한다.
 - ⊙ 연도가 4로 나누어 떨어지면 윤년이다.
 - ⊙ 100으로 나누어 떨어지는 연도는 제외한다.
 - ⊙ 400으로 나누어 떨어지는 연도는 윤년이다.

연도를 입력하시오: 2012
2012 년은 윤년입니다.

Lab: 이차 방정식

- 이차 방정식은 추억의 방정식일 것이다. 고등학교에서 아마 가장 중점적으로 학습한 내용이 아닐까 생각된다. 이차 방정식 $ax^2 + bx + c = 0$ 의 근을 계산하는 프로그램을 작성하여 보자.

$A = 1$

$B = -5$

$C = 6$

$x1 = 3.0$

$x2 = 2.0$

문자열과 숫자

- 숫자 100과 문자열 "100"은 어떤 차이가 있을까? 유사해 보이지만 컴퓨터에서는 이것을 상당히 다르게 처리한다. 우리가 조건식을 만들 때도 문자열과 숫자를 구별하여야 한다.

```
>>>s="100"  
>>>if (s==100):  
    print("숫자 100입니다.")  
>>>
```

문자열-> 숫자

```
>>>s="100"  
>>>number=int(s)  
>>>if (number==100):  
    print("숫자 100입니다. ")
```

숫자 100입니다.

```
>>>
```

Lab: 숫자 맞추기 게임

- 이 예제는 프로그램이 가지고 있는 정수를 사용자가 알아맞히는 게임이다. 사용자가 답을 제시하면 프로그램은 자신이 저장한 정수와 비교하여 제시된 정수가 더 높은지 낮은지 만을 알려준다. 정수의 범위를 1부터 100까지로 한정하도록 하자. 그리고 사용자는 단 한 번의 기회만 가진다.

숫자 게임에 오신 것을 환영합니다.

숫자를 맞춰 보세요: 9

너무 큼!

게임 종료

Lab: 가위 바위 보

- 사용자가 가위, 바위, 보 중에서 하나를 선택하고 컴퓨터도 난수로 가위, 바위, 보 중에서 하나를 선택한다. 사용자의 선택과 컴퓨터의 선택을 비교하여서 승패를 화면에 출력한다.

(가위, 바위, 보) 중에서 하나를 선택하세요: 가위
사용자: 가위 컴퓨터: 바위
컴퓨터가 이겼음!

Q & A

