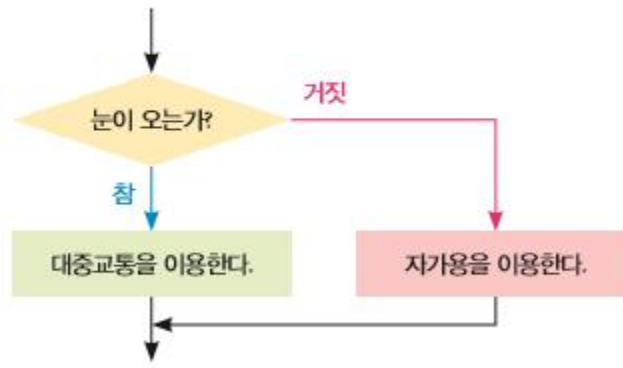


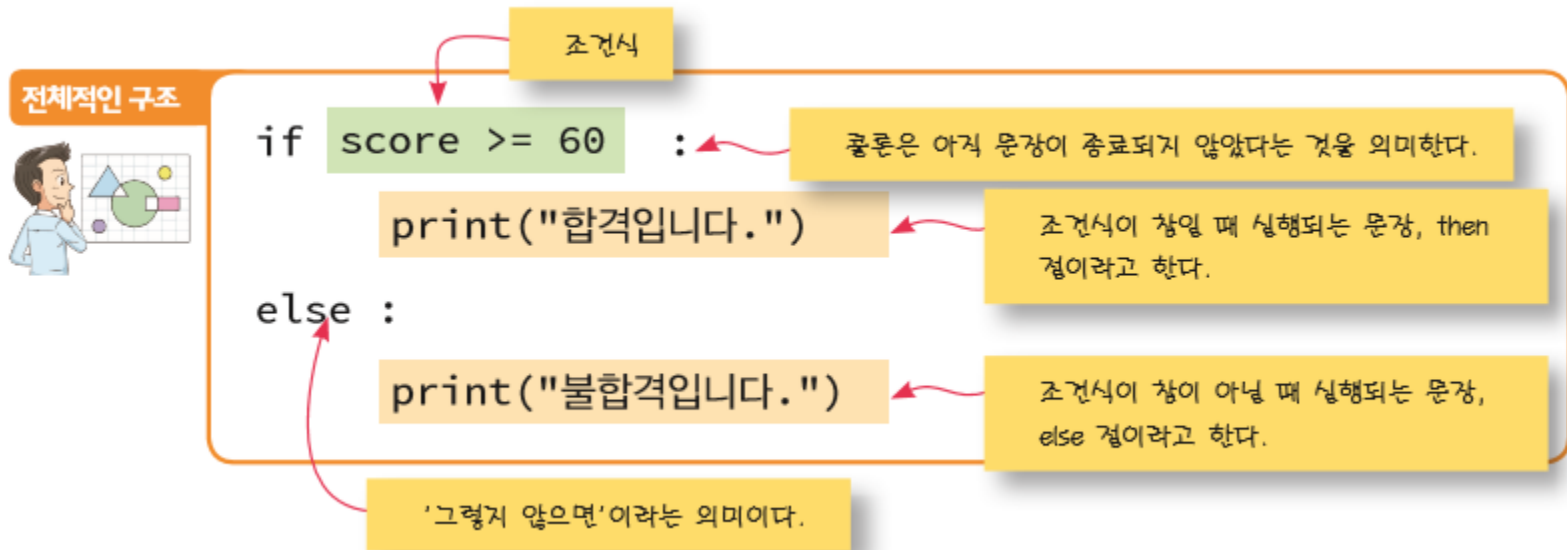
2장 조건 선택

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



관계 연산자

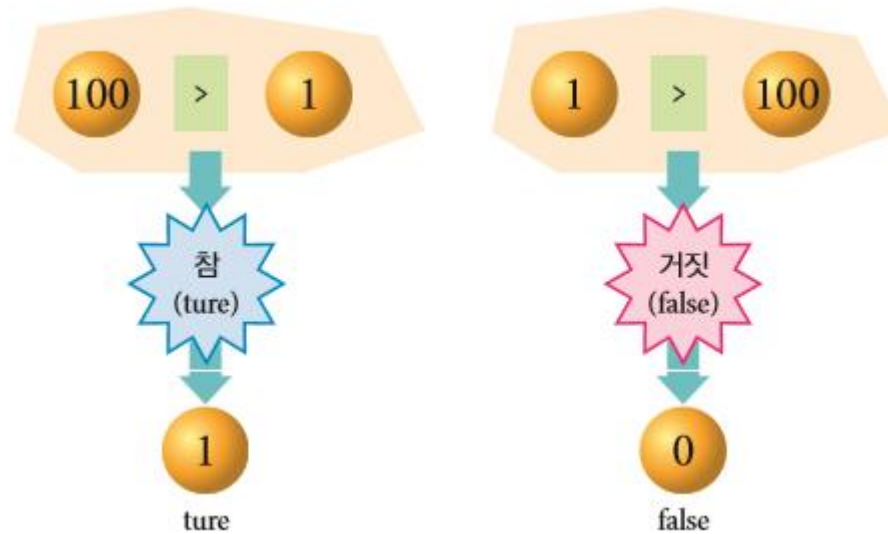
- 관계 연산자(relational operator)는 두 개의 피연산자를 비교하는데 사용된다. 예를 들면 "변수 x가 변수 y보다 큰지"를 따지는데 사용된다.

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!

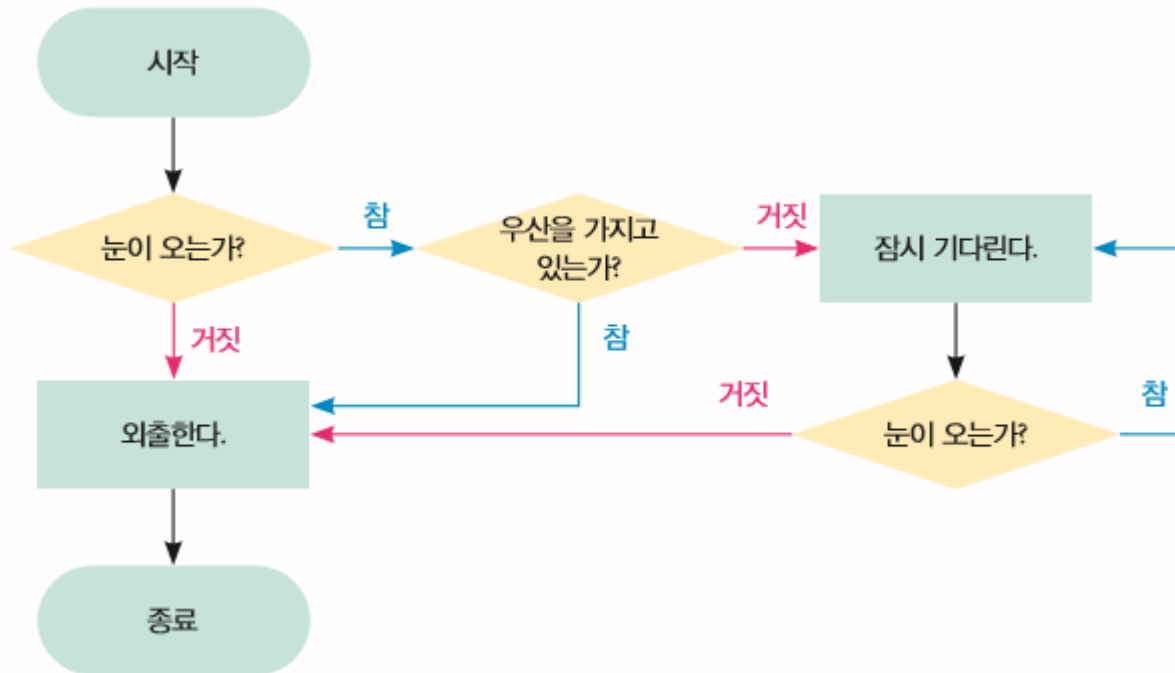
스테이크스테이크스테이크스테이크스테이크스테이크스테이크스테이크
스테이크스테이크

LAB : 순서도 그리기

- 비가 올 때 어떻게 대처할 것인지를 순서도로 그려보라. 비가 오지 않으면 외출한다. 비가 오면 우산을 가지고 있는지 검사한다. 우산을 가지고 있다면 외출한다. 우산을 가지고 있지 않다면 무한정 비가 그칠 때까지 기다린다.

기호	설명
	시작/종료
	도형들을 연결한다.
	입력과 출력
	처리
	판단

SOLUTION



블록

- 만약 조건이 참인 경우에 여러 개의 문장이 실행되어야 한다면 어떻게 하여야 하는가?

전체적인 구조



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

```
    print("합격입니다.")
```

```
    print("장학금도 받을 수 있습니다.")
```

블록: 여러 문장들을 묶은 것이다.

블록은 들여쓰기로 구분

```
if sales > 1000 :  
    discount = sales*0.1  
    print(discount, "할인되었음!")  
else:  
    if sales > 500 :  
        discount = sales*0.05  
        print(discount, "할인되었음!")  
    else:  
        print("할인은 없습니다!")
```

들여쓰기 수준 0 1 2

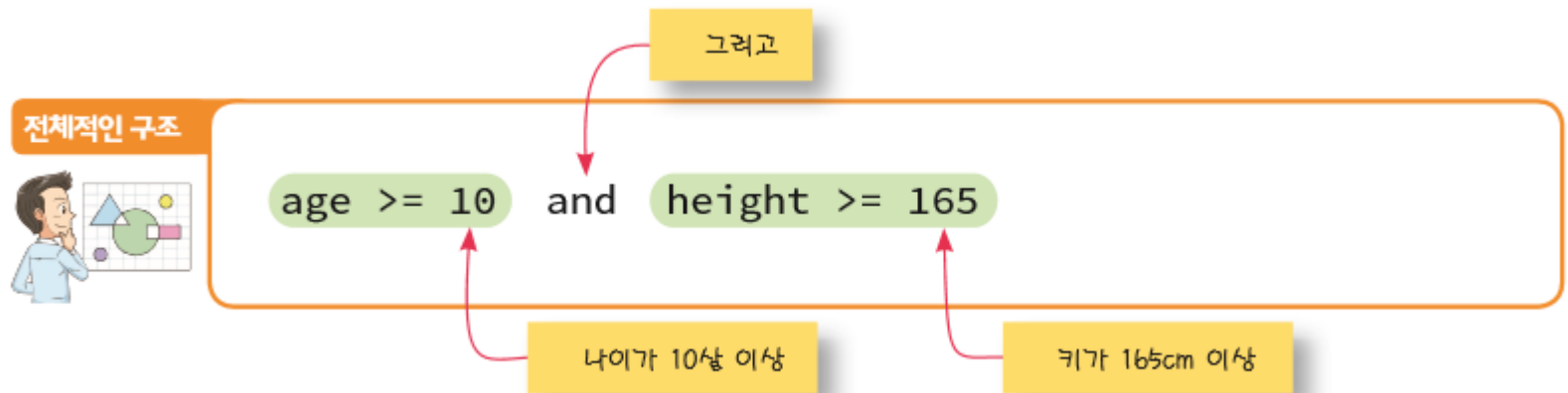
논리 연산자

- 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.")
```

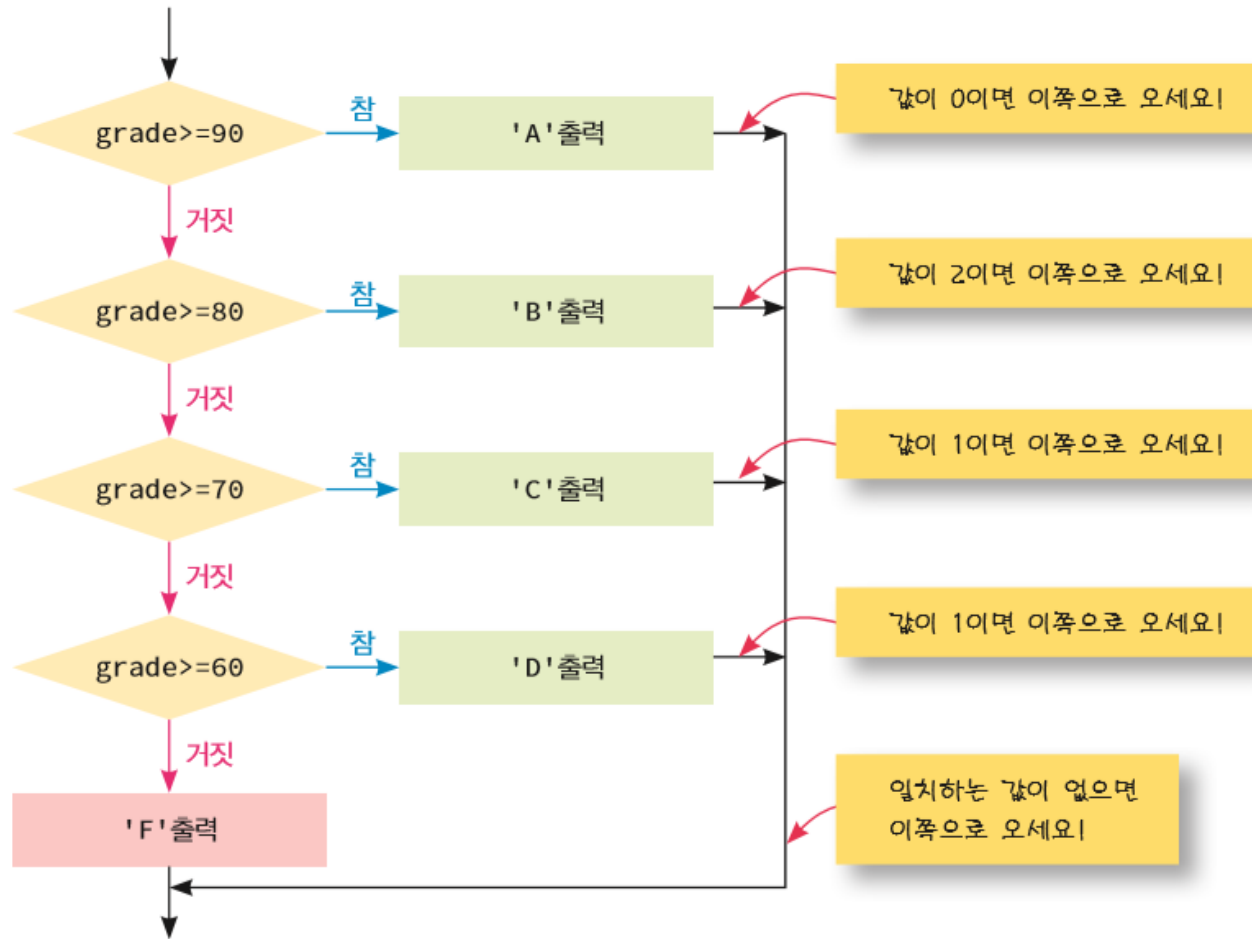
놀이 기구를 탈 수 있습니다.

연속적인 IF-ELSE 문

종종 우리는 조건에 따라서 다중으로 분기되는 결정을 내려야 하는 경우가 있다.

학생들의 성적을 받아서 학점을 출력하는 프로그램을 작성하여 실행하여 보자. 성적이 90점 이상이면 A학점, 80점 이상이고 90점 미만이면 B학점, 70점 이상이고 80점 미만이면 C학점과 같이 결정하는 것이다.





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

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
```

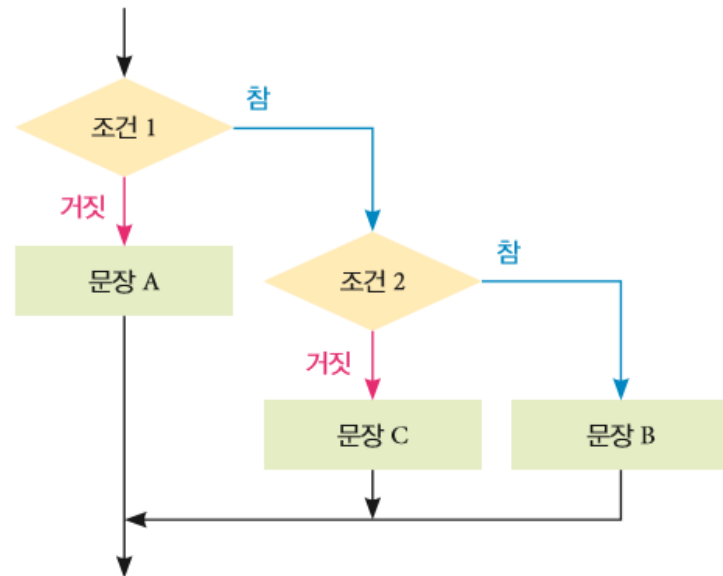
```
else:
```

```
    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.")
```

문자열과 숫자

숫자 100과 문자열 "100"은 어떤 차이가 있을까? 유사해보이지만 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.

```
>>>
```