



컴퓨팅사고와 코딩기초

chapter 9. 조건문 이해하기

목차

- 조건문의 개념
- 기본 조건문

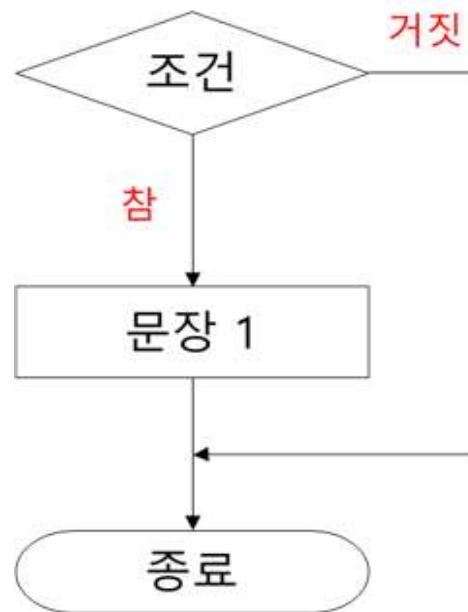


조건문 개념

- 조건문은 어떤 조건을 실행할 때 결과에 따라 서로 다른 명령을 수행하도록 제어하는 구조
- 결과가 참(true)이냐, 거짓(false)이냐에 따라 각각의 명령을 다르게 수행 할 수 있도록 구성되어 있음
- 조건문에서 가장 많이 사용되는 if문, if else문, 중첩 if문

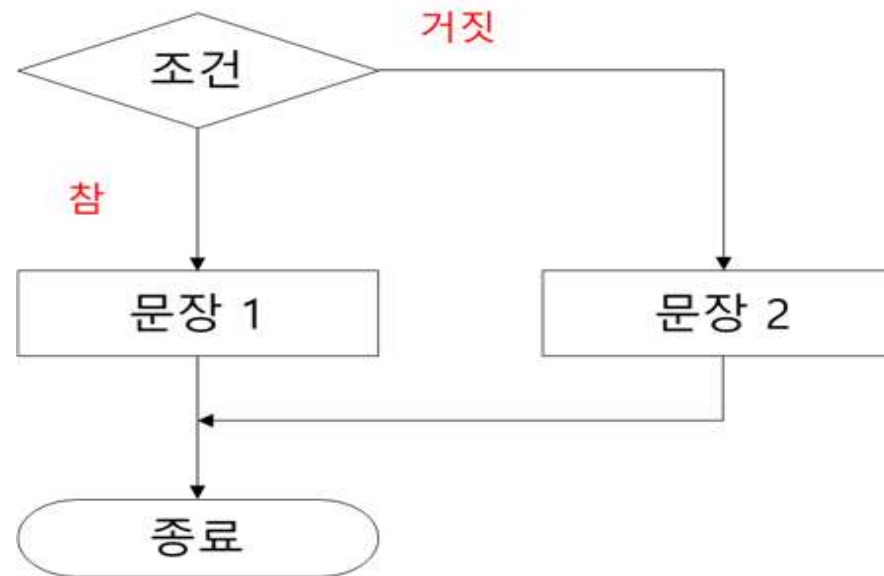
if 문

- 특정 조건이 참인 경우, 그에 해당하는 명령(문장1) 실행하고 종료
- 특정 조건이 거짓인 경우, 그에 해당하는 명령(문장1) 실행하지 않고 종료



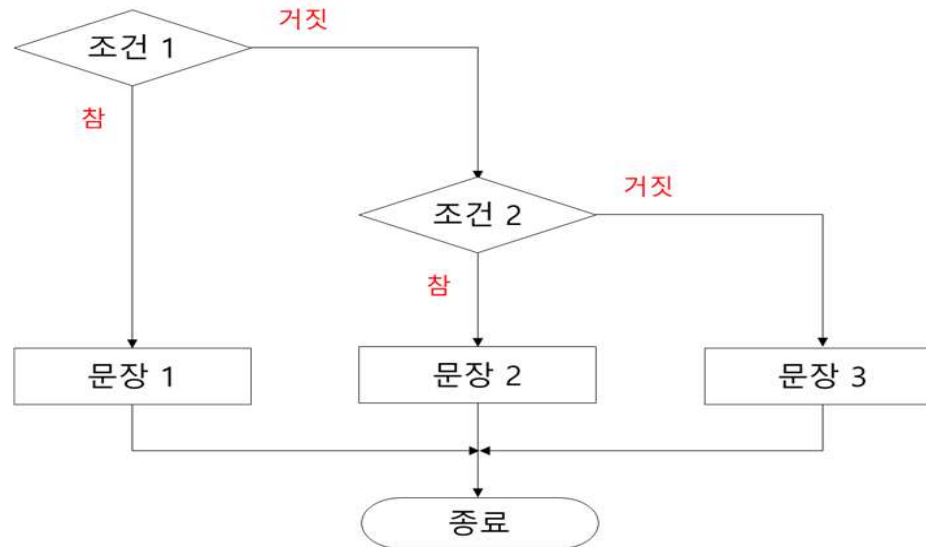
if ~ else 문

- 특정 조건이 참인 경우, 그에 해당하는 명령(문장1) 실행하고 종료
- 특정 조건이 거짓인 경우, 그에 해당하는 명령(문장2) 실행하고 종료



중첩 if 문

- 특정 조건(조건 1)이 참인 경우, 명령(문장 1) 실행하고 종료
- 만약, 특정 조건(조건 1)이 거짓인 경우, 두 번째 조건(조건2) 실행
- 두 번째 조건(조건 2)이 참인 경우, 명령(문장 2)를 실행하고 종료
- 만약, 특정 조건(조건 2)이 거짓인 경우에는 그에 해당하는 명령(문장 3)을 실행하고 종료



기본 조건문 - if 문

- 가장 기본적인 조건문
- 결과가 참인 경우에만 명령을 실행할 수 있도록 설정

```
if 조건식 : 콜론  
문장 1
```

탭(tab)키 또는 스페이스 4번

- 조건의 결과가 참이면 문장을 실행

기본 조건문 - if 문

- 변수 a의 값을 10으로 정한 후, 변수 a가 0보다 크면 print()문 실행

[a>0 경우]

```
a=10  
  
if a>0 :  
    print("a는 0보다 큰 수입니다.")  
|
```

<출력 결과>

```
>>> | a는 0보다 큰 수입니다.  
|
```


기본 조건문 - if 문

- 조건의 결과가 참일 때 여러 개의 문장 실행

```
if 조건식 :  
    문장 1  
    문장 2  
    문장 3
```

- 들여쓰기가 잘못 설정되면 결과가 다르게 나오기 때문에 반드시 주의

기본 조건문 - if 문

- 변수 a의 값을 100으로 정한 후 변수 a가 100과 같으므로 print()문이 순서대로 모두 실행

[a=100 경우]

```
a=100

if a==100 :
    print("a는 100입니다.")
    print("프로그램 끝.")
|
```

<출력 결과>

```
>>> a는 100입니다.
      프로그램 끝.
>>> |
```

기본 조건문 - if 문

- 들여쓰기가 다르게 설정된 경우, 결과는 다르게 실행되므로 주의

[a=100 경우]

```
a=100  
  
if a==100 :  
    print("a는 100입니다.")  
print("프로그램 끝.")  
|
```

<출력 결과>

```
>>> a는 100입니다.  
      프로그램 끝.  
>>> |
```

[a=0 경우]

```
a=0  
  
if a==100 :  
    print("a는 100입니다.")  
print("프로그램 끝.")  
|
```

<출력 결과>

```
>>> 프로그램 끝.  
>>> |
```

기본 조건문 - if 문

- input() 함수 이용하여 사용자가 데이터 직접 입력하여 조건문 작성

```
a=int(input("숫자를 입력하세요 : "))
```

정수형 데이터를 입력

```
a=float(input("숫자를 입력하세요 : "))
```

실수형 데이터를 입력

기본 조건문 – if ~ else 문

- 조건의 결과가 참이냐 거짓이냐에 따라 결과를 다르게 표현하고 싶을 때 사용할 수 있음

```
if 조건식 :  
    문장 1(결과가 참인 경우)  
else :  
    문장 2(결과가 거짓인 경우)
```

- 조건의 결과가 참이면 문장1을 실행, 거짓이면 문장 2 실행

기본 조건문 - if ~ else 문

- 변수 a의 값을 100으로 정한 후 변수 a가 60보다 크기 때문에 첫 번째 print()문만 실행

[a=100 경우]

```
a=100

if a>60 :
    print("합격입니다.")
else :
    print("불합격입니다.")
|
```

[출력 결과]

```
>>> 합격입니다.
>>> |
```

기본 조건문 – if ~ else 문

- 여러 개의 조건식 지정 → 논리 연산자(and, or 등) 사용

if 조건식 1 and 조건식 2 :

문장 1(조건식 1의 결과와 조건식 2의 결과가 모두 참인 경우)

else :

문장 2(조건식 1의 결과와 조건식 2의 결과 중 하나 이상 거짓인 경우)

- 조건식 1 결과와 조건식 2 결과가 모두 참 → 문장1 실행
- 조건식 1 결과와 조건식 2 결과 중 하나 이상이 거짓 → 문장 2 실행

기본 조건문 - if ~ else 문

- 변수 a의 값을 95, 변수 b의 값을 100으로 정한 후 두 변수의 값이 모두 90 이상이기 때문에 첫 번째 print()문만 실행

[a=95, b=100 경우]

```
a=95
b=100

if a>=90 and b>=90 :
    print("성공")
else :
    print("실패")
|
```

[출력 결과]

```
>>> 성공
>>> |
```


기본 조건문 – if ~ else 문

- random 모듈을 이용하여 임의의 숫자 선택

```
import random  
a=random.randint(1, 100)
```

- 변수 a의 값에 1부터 100까지의 숫자 중 임의의 숫자를 하나 선택하여 저장

기본 조건문 - 중첩 if 문

- 특정 조건이 하나인 경우 if 문 한 번 사용
- 특정 조건이 2개 이상인 경우에는 if 문 여러 번 반복하여 사용
 - 원하는 결과가 나올 때까지 다시 실행

```
if 조건식 1 :  
    if 조건식 2 :  
        문장 1  
    else :  
        문장 2  
else :  
    문장 3
```

- 조건식 1의 결과가 참이면 조건식2를 다시 실행하고, 거짓이면 문장 3을 실행
- 조건식 2가 다시 실행되어 그 결과가 참이면 문장 1을 실행, 거짓이면 문장 2 실행

기본 조건문 – if ~ elif ~ else 문

- if ~ else 문이 여러 번 반복되는 사용되면 코딩이 복잡
 - 복잡한 중첩 if 문(`else ~ if`)을 `elif` 문으로 간단하게 표현

```
a=85  
  
if a >= 90 :  
    print("A학점 입니다.")  
else :  
    if a >= 80 :  
        print("B학점 입니다.")  
    else :  
        if a >= 70 :  
            print("C학점 입니다.")  
        else :  
            if a >= 60 :  
                print("D학점 입니다.")  
            else :  
                print("F학점 입니다.")  
|
```



```
a=85  
  
if a >= 90 :  
    print("A학점 입니다.")  
elif a >= 80 :  
    print("B학점 입니다.")  
elif a >= 70 :  
    print("C학점 입니다.")  
elif a >= 60 :  
    print("D학점 입니다.")  
else :  
    print("F학점 입니다.")  
|
```