

# Conditional Statement-2: Making Two-Way Decisions and Applying Conditional Statements

## Topic 03 Key concept

### 1. Two-way decision using else

#### 1.1 if statement without else statement

- Let's consider a program that handles the following situation using the if statement we learned before.

If the time is before 12 o'clock, it prints "It's morning.", and if it is after 12 o'clock, it prints "It's afternoon."

### 1. Two-way decision using else

#### 1.1 if statement without else statement

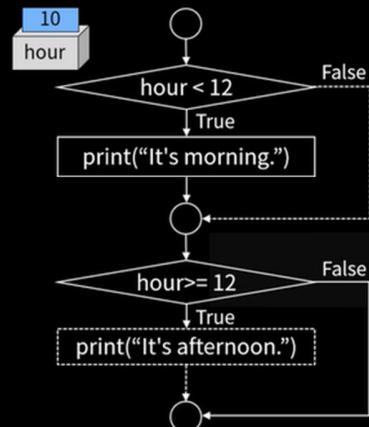
- If we only use the if statement, we can implement it using the following method with two if statements.
- The flowchart on the right represents the flow of this code.

```

1 hour = 10
2 if hour < 12 :
3     print("It's morning.")
4 if hour >= 12 :
5     print("It's afternoon.")

```

It's morning.



Prime  
AmB

→ > 2 < ←

## 1. Two-way decision using else

### 1.2 if-else statement

- The previous code using the if statement can be made with if-else statement as follows.

```

1 hour = 10
2
3 if hour < 12 :
4     print("It's morning.")
5 else :
6     print("It's afternoon.")

```

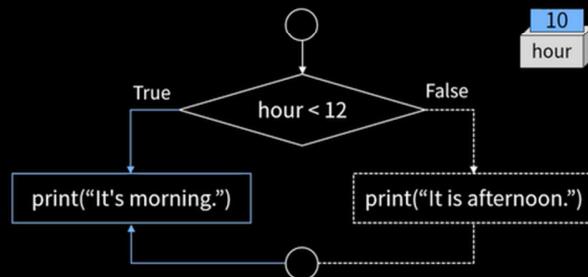
It's morning.

- In the case of being morning, it cannot be afternoon at the same time.  
This relationship is called an exclusive relationship.
- The if-else statement is a statement that can be used in such exclusive relationships, and it is similar to saying "If the condition is true, execute this, and if the condition is not true, execute that."

## 1. Two-way decision using else

### 1.2 if-else statement

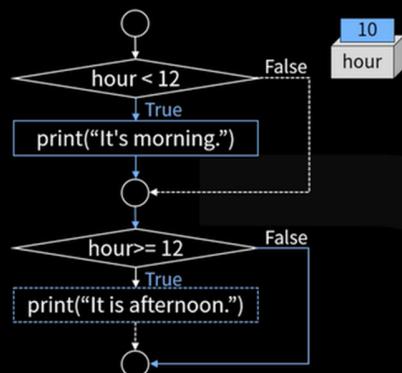
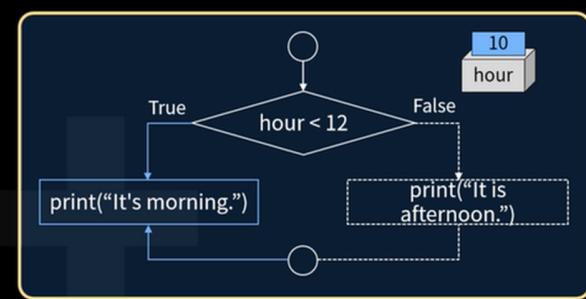
- The previous code using the if statement can be made with if-else statement as follows.
- If the exclusive relationship is expressed with if-else, the flowchart becomes more concise.



## 1. Two-way decision using else

### 1.3 Comparison of flowcharts between if-else and if

- Let's compare the flowchart of using if-else and the flowchart of using if twice once again.
- The one on the left is much clearer.



Prime  
AmB

-->3<--

## 2. Various examples of if-else statements

### 2.1 Pass or fail determination based on score

- In Python, if-else statement is provided for selection structure.  
If it is an exclusive relationship, you can use the if-else statement.
- Let's assume that if the score is 90 or above, it is considered a pass and receives a scholarship. On the other hand, if it is below 90, it should be treated as a fail.

```
1 score = 99
2 if score > 90 :
3     print('You passed.')
4     print('Get a scholarship, too.')
5 else :
6     print('You failed.'
```

You passed.  
Get a scholarship, too.

## 2. Various examples of if-else statements

### 2.2 Odd and even number determination

- Let's determine odd/even numbers.

```
1 num = 10
2 if num % 2 == 0 :
3     print(num, 'is an even number.')
4 else :
5     print(num, 'is an odd number.'
```

10 is an even number.

## 2. Various examples of if-else statements

### 2.2 Odd and even number determination

- Let's determine odd/even numbers.

```
1 num = 10
2 if num % 2 == 0 :
3     print(num, 'is an even number.')
4 else :
5     print(num, 'is an odd number.'
```

10 is an even number.

• Line 2

- If the remainder of num divided by 2 is 0, Line 3 below is executed.

## 2. Various examples of if-else statements

### 2.2 Odd and even number determination

- Let's determine odd/even numbers.

```
1 num = 10
2 if num % 2 == 0 :
3     print(num, 'is an even number.')
4 else :
5     print(num, 'is an odd number.'
```

10 is an even number.

• Line 4

- Otherwise, Line 5 below else is executed.

## 2. Various examples of if-else statements

### 2.3 Numeric type determination

- We learned that the isdigit method of a string returns True or False indicating whether the string consists only of numeric characters from 0 to 9.
- Let's create an if statement that checks if it is possible before converting the data type to int.

```
1 x = input("Enter the first integer: ")
2 print('Input data type: ', type(x))
```

Enter the first integer: 100  
Input data type: <class 'str'>

## 2. Various examples of if-else statements

### 2.3 Numeric type determination

```
1 x = input("Enter the first integer: ")
2 print('Input data type: ', type(x))
```

Enter the first integer: 100  
Input data type: <class 'str'>

```
1 if x.isdigit() == True :
2     x = int(x)
3     print('Input data type : ', type(x))
4 else :
5     print('x is not a numeric character.'
```

Input data type : <class 'int'>

Variable x returned True by the isdigit method. It means that the variable x consists only of numbers.

Therefore, it is possible to convert the data type to int.

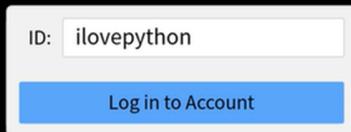
AmB — Prime

— → 5 < —

## 2. Various examples of if-else statements

### 2.4 Program to check if the ID matches

- Write a program that takes an ID from the user and prints whether it matches the stored ID ‘ilovepython’.



## 2. Various examples of if-else statements

### 2.4 Program to check if the ID matches

- Problem-solving code using if-else statement.

```
1 id = "ilovepython"
2 s = input("Enter your ID: ")
3 if s == id :
4     print("Welcome.")
5 else :
6     print("No ID found.")
```

Enter your ID: ilovepython  
Welcome.

- Use the `input` function to receive a string. Then check if the input string is the same as ‘ilovepython’ using the `==` operator. If the result of the comparison is True, print ‘Welcome.’

## 2. Various examples of if-else statements

### 2.5 Creating a coin flipping game with the random function

- Let's write a code for a coin flipping game. Generating random numbers will simulate flipping a coin.
- We have learned the `random` module, a built-in module in Python used for random number generation. After executing `import random`, calling the `random.randrange(2)` function can generate either 0 or 1 randomly.



Prinme  
AmB

-->6<--

## 2. Various examples of if-else statements

### 2.5 Creating a coin flipping game with the random function

- Let's use the 'randrange' function of the random module to generate 0 or 1 randomly.
- If the value is 0, print "Front". Otherwise, print "Back" as shown in the code below.

```
1 import random
2
3 print("Let's start the coin toss game.")
4 coin = random.randrange(2)
5 if coin == 0 :
6     print("Front")
7 else :
8     print("Back")
9 print("Game over")
```

Line 4

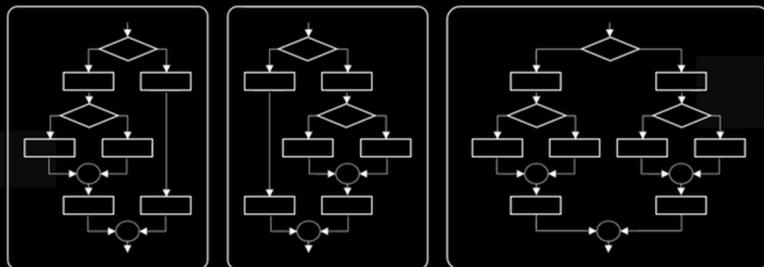
- Calling randrange(2) will return 0 or 1 through the random number generator.

Let's start the coin toss game.  
Front  
Game over

## 3. Nested conditional statements

### 3.1 What is nested conditional statement?

- Sometimes, another if statement can be placed inside an if statement, and this is called the nested conditional statement.
- In situations where specific conditions result in True or False, there may be a need to check further detailed conditions. In such cases, nested conditional statements are used.



## 3. Nested conditional statements

### 3.2 Putting if-else statements inside if-else statement

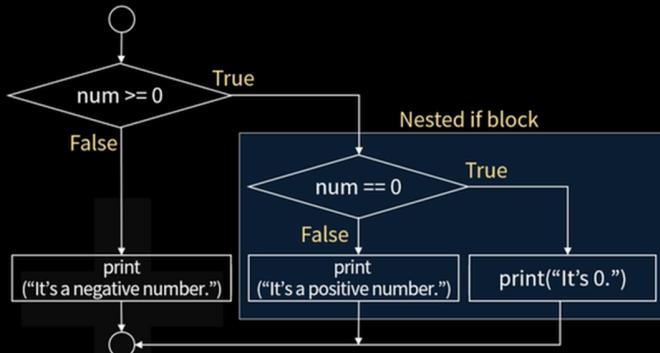
- Let's consider a somewhat complex case where we put if-else statement inside if-else statement.
- Through coding, we should be able to handle various situations.
- Let's assume we want to determine whether the integer num is positive, zero, or negative and print the result on the screen.

Prime  
AmB

-->7<--

## 3. Nested conditional statements

### 3.2 Putting if-else statements inside if-else statement



- If the condition `num >= 0` evaluates to True, we need to perform an additional check of the condition `num == 0`. Depending on the result, we print either 'It's 0.' or 'It's a positive number.'

## 3. Nested conditional statements

### 3.2 Putting if-else statements inside if-else statement

- Let's receive an integer from the user and determine whether it is positive, zero, or negative and print the result on the screen.
- We can solve the problem using nested conditional statements, where we put if-else inside if-else.

```

1 num = int(input("Enter an integer: "))
2 if num >= 0 :
3     if num == 0 :
4         print("It's 0.")
5     else :
6         print("It's a positive number.")
7 else :
8     print("It's a negative number.")
  
```

Enter an integer: -20  
It's a negative number.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's use another if-else statement inside a block composed of if-else statements. In this code, we want to determine whether the number is even or odd only when it is positive, and print the result.

```

1 num = 100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.')
  
```

100 is not a negative number.  
100 is an even number.

Prinme  
AnB

-->8<--

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

```
1 num = 100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.'
```

100 is not a negative number.  
100 is an even number.

• Line 2

- Since num is greater than 0, the block directly below it is not executed, and the else statement in Line 4 is executed.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

```
1 num = 100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.'
```

100 is not a negative number.  
100 is an even number.

• Line 5

- After printing 'is not a negative number' using the print function,

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

```
1 num = 100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.'
```

100 is not a negative number.  
100 is an even number.

• Line 6

- we check if the remainder of num divided by 2 is 0.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

```
1 num = 100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.'
```

100 is not a negative number.  
100 is an even number.

#### Line 8

- Since  $100 \% 2$  equals 0, the block of the else statement is skipped, and the program ends.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```
1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.'
```

-100 is a negative number.

- In this code, if-else appears twice, and the outer if-else statement is referred to as the outer if-else, while the inner if-else statement is referred to as the inner if-else. Let's take a closer look.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```
1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.'
```

-100 is a negative number.

- The if-else conditional statements in Lines 2 and 4 are the outer if-else statement.
- The if statement in Line 2 is executed only when the value of num is less than 0, i.e., when it is negative.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```
1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.')

-100 is a negative number.
```

- The else statement in Line 4 is executed only when the value of num is not negative.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```
1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.')

-100 is a negative number.
```

- Lines 6-9 are the inner if-else statement within the else statement.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```
1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.')

-100 is a negative number.
```

- The if statement in Line 6 is executed only when the variable num divided by 2 has no remainder, i.e., when it is even.

Prinme  
AmB

-->11<--

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```

1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.')

```

-100 is a negative number.

- Let's look at Line 8. This else block is executed when num is odd.

## 4. Various Examples of Nested Conditional Statements

### 4.1 Using nesting to determine even or odd

- Let's examine the behavior of the code when num is -100.

```

1 num = -100
2 if num < 0 :
3     print(num, 'is a negative number.')
4 else :
5     print(num, 'is not a negative number.')
6     if num % 2 == 0 :
7         print(num, 'is an even number.')
8     else :
9         print(num, 'is an odd number.')

```

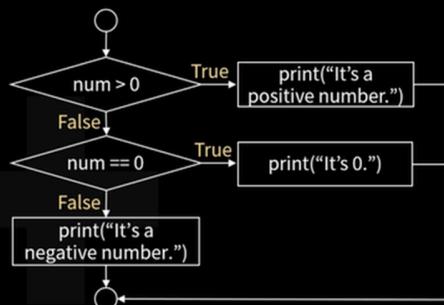
-100 is a negative number.

- Therefore, when the value of the variable num is changed to -100, it executes as follows.

## 5. Making Multi-directional Decisions Using elif

### 5.1 Checking multiple conditions sequentially

- Let's explore how to check additional conditions when the initial condition is false.
- Consider a situation where we receive an integer num from the user and based on its value, we need to print "It's a positive number.", "It's 0.", or "It's a negative number."

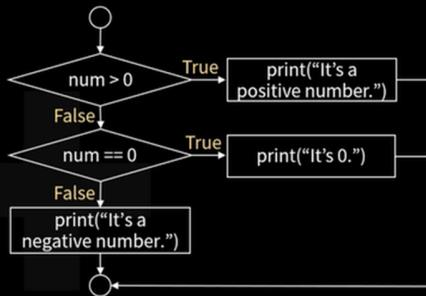


- In this flowchart, if the first if condition evaluates to True, the corresponding print statement is executed, and the program exits the conditional statement.
- If not(False), the second condition is checked, and the decision is made based on this condition.

## 5. Making Multi-directional Decisions Using elif

### 5.1 Checking multiple conditions sequentially

- Let's explore how to check additional conditions when the initial condition is false.
- Consider a situation where we receive an integer num from the user and based on its value, we need to print "It's a positive number.", "It's 0.", or "It's a negative number."



- In such cases, we can use if-elif-else statements to implement the solution.
- elif is a combination of 'else' and 'if'.

## 5. Making Multi-directional Decisions Using elif

### 5.2 Code example to print whether an integer is positive, zero, or negative

- Let's create a conditional statement using if-elif-else to determine whether the input integer is 'positive', 'zero' or 'negative'.

```
1 num = int(input("Enter an integer: "))
2
3 if num > 0 :
4     print("It's a positive number.")
5 elif num == 0 :
6     print("It's zero(0).")
7 else :
8     print("It's a negative number.")
```

```
Enter an integer: 0
It's zero(0).
```

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

```
#####
It's David's fruit shop.
1: Apple( Price : 5,000 won)
2: Grape( Price : 6,000 won)
3: Melon( Price : 8,000 won)
4: Orange( Price : 2,000 won)
#####
Enter the number of the item (between 1 and 4) >> 1
Enter the number of items (between 1 and 10) >> 2
The fruit you chose is : Apple
The price of the selected fruit is : 5000
The number you selected is : 2
The total amount is 10000 won.
Please put the money in(ex: 15000) >> 12000
Got 12000 won. The change is 2000 won.
```

- Let's assume David's fruit store offers four types of fruits. We will receive the fruit menu number and quantity as input.

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

```
#####
It's David's fruit shop.
1: Apple( Price : 5,000 won)
2: Grape( Price : 6,000 won)
3: Melon( Price : 8,000 won)
4: Orange( Price : 2,000 won)
#####
Enter the number of the item (between 1 and 4) >> 1
Enter the number of items (between 1 and 10) >> 2
The fruit you chose is : Apple
The price of the selected fruit is : 5000
The number you selected is : 2
The total amount is 10000 won.
Please put the money in(ex: 15000) >>> 12000
Got 12000 won. The change is 2000 won.
```

- Let's assume David's fruit store offers four types of fruits. We will receive the fruit menu number and quantity as input.
- We inform the user about the total amount for the selected fruit and prompt them to insert money.

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

```
#####
It's David's fruit shop.
1: Apple( Price : 5,000 won)
2: Grape( Price : 6,000 won)
3: Melon( Price : 8,000 won)
4: Orange( Price : 2,000 won)
#####
Enter the number of the item (between 1 and 4) >> 1
Enter the number of items (between 1 and 10) >> 2
The fruit you chose is : Apple
The price of the selected fruit is : 5000
The number you selected is : 2
The total amount is 10000 won.
Please put the money in(ex: 15000) >>> 12000
Got 12000 won. The change is 2000 won.
```

- Once the money is inserted, we deliver the goods and provide the change.
- If the money is insufficient, we send a message stating that there is not enough money.

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

- We display the menu as follows and receive input from the user.

```
1 print('#' * 50)
2 print("It's David's fruit shop.")
3 print('1: Apple( Price : 5,000 won)')
4 print('2: Grape( Price : 6,000 won)')
5 print('3: Melon( Price : 8,000 won)')
6 print('4: Orange( Price : 2,000 won)')
7 print('#' * 50)
8
9 total_price = 0
10 order = int(input('Enter the number of the item (between 1 and 4) >> '))
11 count = int(input('Enter the number of items (between 1 and 10) >> '))
```

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

```
13 if order == 1 :  
14     fruit = 'Apple'  
15     price = 5000  
16 elif order == 2 :  
17     fruit = 'Grape'  
18     price = 6000  
19 elif order == 3 :  
20     fruit = 'Melon'  
21     price = 8000  
22 else :  
23     fruit = 'Orange'  
24     price = 2000  
25  
26 print('The fruit you chose is :', fruit)  
27 print('The price of the selected fruit is : ', price)  
28 print('The number you selected is : ', count)  
29 print('The total amount is', price * count, 'won.')
```

- We specify the selected fruit and its price and calculate the total amount.

If the money is insufficient, we print 'Not enough money,' otherwise, we display the change.

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

- We specify the selected fruit and its price and calculate the total amount. If the money is insufficient, we print 'Not enough money,' otherwise, we display the change.

```
30  
31 money = int(input('Please put the money in(ex: 15000) >>> '))  
32 if money < price * count :  
33     print('Not enough money')  
34 else :  
35     change = money - price * count  
36     print('Got', money, 'won. The change is', change, 'won.')
```

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

- Key Code Explanation

```
1 if order == 1 :  
2     fruit = 'Apple'  
3     price = 5000  
4 elif order == 2 :  
5     fruit = 'Grape'  
6     price = 6000  
7 elif order == 3 :  
8     fruit = 'Melon'  
9     price = 8000  
10 else :  
11     fruit = 'Orange'  
12     price = 2000
```

- If the order number is 1, the fruit is 'apple' and the price is 5000. Otherwise, it goes through another if-elif statement for condition checking.

## 6. Various Examples of if-elif-else Statements

### 6.1 Automatic menu ordering system

- Key Code Explanation
  - Using if-elif-else statement makes the code more readable.

```
1 if order == 1 :  
2     fruit = 'Apple'  
3     price = 5000  
4 elif order == 2 :  
5     fruit = 'Grape'  
6     price = 6000  
7 elif order == 3 :  
8     fruit = 'Melon'  
9     price = 8000  
10 else :  
11     fruit = 'Orange'  
12     price = 2000
```

## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

- Implement a login feature using the username and password.
- When checking the password, write a program that produces the following output. The stored password is “mypass1234”.
- If both the username and password are correct, print “Welcome.” If not, print either “No ID found.” or “The password is wrong.”



## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

```
1 my_id = "ilovepython"  
2 password = "mypass1234"  
3 string1 = input("Enter your ID: ")  
4 string2 = input("Enter your password: ")  
5 if string1 == my_id and string2 == password :  
6     print("Welcome.")  
7 elif string1 != my_id :  
8     print("No ID found.")  
9 else :  
10    print("The password is wrong.")
```

#### Line 3~4

- Receive the username and password from the user.

```
Enter your ID: ilovepython  
Enter your password: mypass1234  
Welcome.
```

## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

```
1 my_id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter your ID: ")
4 string2 = input("Enter your password: ")
5 if string1 == my_id and string2 == password :
6     print("Welcome.")
7 elif string1 != my_id :
8     print("No ID found.")
9 else :
10    print("The password is wrong.")
```

Enter your ID: ilovepython  
Enter your password: mypass1234  
Welcome.

#### Line 5

- Compare the user's input for the username and password with my\_id and password. Since the "and" operator is used, the following block will not be executed if any of them are incorrect.

## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

```
1 my_id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter your ID: ")
4 string2 = input("Enter your password: ")
5 if string1 == my_id and string2 == password :
6     print("Welcome.")
7 elif string1 != my_id :
8     print("No ID found.")
9 else :
10    print("The password is wrong.")
```

Enter your ID: ilovepython  
Enter your password: mypass1234  
Welcome.

#### Line 6

- If the condition is True, print "Welcome."

## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

```
1 my_id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter your ID: ")
4 string2 = input("Enter your password: ")
5 if string1 == my_id and string2 == password :
6     print("Welcome.")
7 elif string1 != my_id :
8     print("No ID found.")
9 else :
10    print("The password is wrong.")
```

Enter your ID: ilovepython  
Enter your password: mypass1234  
Welcome.

#### Line 7

- If the if statement on Line 5 is false, compare my\_id with string1 (user input) using the elif statement.

## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

```
1 my_id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter your ID: ")
4 string2 = input("Enter your password: ")
5 if string1 == my_id and string2 == password :
6     print("Welcome.")
7 elif string1 != my_id :
8     print("No ID found.") → Line 8
9 else :
10    print("The password is wrong.")
```

Line 8  
- If they are different, print "No ID found."

Enter your ID: ilovepython  
Enter your password: mypass1234  
Welcome.

## 6. Various Examples of if-elif-else Statements

### 6.2 Program to check if the ID and password match

```
1 my_id = "ilovepython"
2 password = "mypass1234"
3 string1 = input("Enter your ID: ")
4 string2 = input("Enter your password: ")
5 if string1 == my_id and string2 == password :
6     print("Welcome.")
7 elif string1 != my_id :
8     print("No ID found.") → Line 10
9 else :
10    print("The password is wrong.")
```

Line 10  
- If both the if statement on Line 5 and the elif statement on Line 7 are false, the print statement in the final else block is executed.

Enter your ID: ilovepython  
Enter your password: mypass1234  
Welcome.

## 7. Comparison in conditional statements

### 7.1 GPA calculator

- Let's assume a grade calculator problem where we divide the grades according to the score range using multiple if statements. How can we print the grades based on the score range shown in the table below?

Score	Grade
90 and above	A
Below 90 and above 80	B
Below 80 and above 70	C
Below 70 and above 60	D
Below 60	F

Prinme  
AmB

-->18<--

## 7. Comparison in conditional statements

### 7.1 GPA calculator

- Let's assume a grade calculator problem where we divide the grades according to the score range using multiple if statements. How can we print the grades based on the score range shown in the table below?



After solving the problem using if statement, if-else statement, if-elif-else statement, and nested conditional statement, let's compare the code.

Below 80 and above 70

B  
C

## 7. Comparison in conditional statements

### 7.2 GPA calculator example A: multiple if statements

- This problem can be solved by applying multiple if statements with the condition of the "and" logical operator.

```
1 score = int(input('Enter your score : '))
2 if score >= 90 :
3     grade = 'A'
4 if score < 90 and score >= 80 :
5     grade = 'B'
6 if score < 80 and score >= 70 :
7     grade = "C"
8 if score < 70 and score >= 60 :
9     grade = 'D'
10 if score < 60 :
11     grade = 'F'
12 print('Your grade :', grade)
```

Enter your score : 88
Your grade : B

## 7. Comparison in conditional statements

### 7.2 GPA calculator example A: multiple if statements

- This problem can be solved by applying multiple if statements with the condition of the "and" logical operator.

```
1 score = int(input('Enter your score : '))
2 if score >= 90 :
3     grade = 'A'
4 if score < 90 and score >= 80 :
5     grade = 'B'
6 if score < 80 and score >= 70 :
7     grade = "C"
8 if score < 70 and score >= 60 :
9     grade = 'D'
10 if score < 60 :
11     grade = 'F'
12 print('Your grade :', grade)
```

Enter your score : 88
Your grade : B

- Grade B for a score of 88 is correctly printed.
- However, this code is complex and hard to read. It's difficult to spot errors in each condition.
- Understanding the meaning of each if statement requires careful examination, increasing the possibility of errors.
- To solve this, let's try using if-else statement as shown in the next code.

Prinme  
AmB

-->19<--

## 7. Comparison in conditional statements

### 7.3 GPA calculator example B: nested if-else statement

- This problem can be solved by applying nested conditional statement to if-else statement.

```
1 score = int(input('Enter your grade : '))
2 if score >= 90 :
3     grade = 'A'
4 else :
5     if score >= 80 :
6         grade = 'B'
7     else :
8         if score >= 70 :
9             grade = 'C'
10        else :
11            if score >= 60 :
12                grade = 'D'
13            else :
14                grade = 'F'
15 print('Your grade : ', grade)
```

Enter your grade : 88  
Your grade : B

- Grade B for a score of 88 is correctly printed.
- The possibility of errors is slightly reduced compared to before, but if-else statements can only represent two conditions, so the readability is still poor, and there are too many indentations.
- What is a simpler method than this when there are multiple conditions?

## 7. Comparison in conditional statements

### 7.4 GPA calculator example C: if-elif-else statement

- Let's compare the code using if-else statement and if-elif-else statement.

#### Code using if-else

```
If score >= 90 :
    statement
else :
    if score >= 80 :
        statement
    else :
        statement
```

#### Code using if-elif

```
If score >= 90 :
    statement
else score >= 80 :
    statement
else :
    statement
```



- The left and right codes have the same role.
- The code on the right has fewer indentations and fewer lines, making it easier to understand the code.

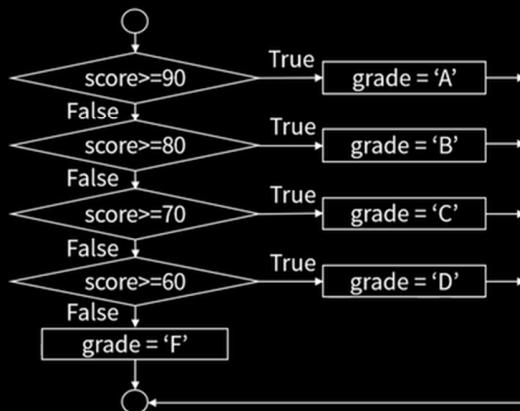
Prime  
AnB

-->20<--

## 7. Comparison in conditional statements

### 7.4 GPA calculator example C: if-elif-else statement

- The execution flow of if-elif-else statement is as follows.



## 7. Comparison in conditional statements

### 7.4 GPA calculator example C: if-elif-else statement

- Now let's write the code using if-elif-else statement.

```
1 score = int(input('Enter your grade : '))
2 if score >= 90 :
3     grade = 'A'
4 elif score >= 80 :
5     grade = 'B'
6 elif score >= 70 :
7     grade = 'C'
8 elif score >= 60 :
9     grade = 'D'
10 else:
11     grade = 'F'
12 print('Your grade : ', grade)
```

```
Enter your grade : 88
Your grade : B
```

- Grade B for a score of 88 is correctly printed.
- Compared to the grade calculators A and B, using if-elif-else statements reduces the possibility of errors.
- It is also easier to understand because the conditions are clear.

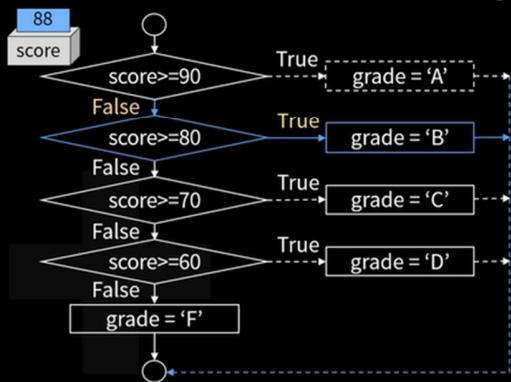
Prinme  
AmB

-->21<--

## 7. Comparison in conditional statements

### 7.4 GPA calculator example C: if-elif-else statement

- When score is 88, it follows the following flow.



- First, the program executes the if  $score \geq 90$  statement to check if the score is greater than or equal to 90.
- The result of the comparison is False, so it executes the elif  $score \geq 80$  statement to check if the score is greater than or equal to 80.
- Now the result is True, so it executes the grade = 'B' statement and skips the execution of the elif-else statements below.
- Therefore, the program ultimately prints "Your grade : B."

## 7. Comparison in conditional statements

### 7.4 GPA calculator example C: if-elif-else statement

- We have written the code using various conditional statements based on the problem.
- Effort is needed to apply efficient conditional statements in various situations to match the problem requirements.

if statement

if-else statement

if-elif-else statement

Nested conditional statement