

5.2 Conditional Statements (if, else, elif)

- The `if` statement in Python is a conditional statement that allows you to execute a block of code only if a certain condition is met.
- Make decisions using `if`, `elif`, and `else` statements.

The general **syntax** of the if statement is as follows:

```
if condition:  
    # code to execute if condition is True
```

The `condition` can be any logical expression. If the condition is evaluated to `true`, the block of `statements` is executed. Otherwise, the block of `statements` is skipped.

Here is a simple example of an if statement in PYTHON:

Example #5.1:

```
x = 10  
  
if x > 5:  
    print('x is greater than 5.')
```

This code will print the message `x is greater than 5.` to the console.

You can also use `elif` statements to check for multiple conditions. The general **syntax** of the **elif statement** is as follows:

```
if condition1:  
    # code to execute if condition1 is True  
elif condition2:  
    # code to execute if condition2 is True
```

If the `condition` for the if statement is evaluated to `false`, the python interpreter will check the `condition` for the first elif statement. If the condition for the elif statement is evaluated to `true`, the corresponding block of `statements` is executed. Otherwise, the python interpreter will check the `condition` for the next elif statement, and so on.

Here is an example of an if statement with an elif statement:

```
x = 3

if x > 5:
    print('x is greater than 5.')
elif x < 5:
    print('x is less than 5.')
```

This code will print the message "x is less than 5." to the console.

You can also use an else statement to check for all other conditions. The general syntax of the else statement is as follows:

```
if condition1:
    # code to execute if condition1 is True
elif condition2:
    # code to execute if condition2 is True
else:
    # code to execute if none of the conditions are True
```

If all of the conditions for the if and elseif statements are evaluated to **false**, the block of **statements** in the **else** statement is executed.

Here is an example of an if statement with an **elif** statement and an **else** statement:

```
x = 2

if x > 5:
    print('x is greater than 5.')
elif x == 5:
    print('x is equal to 5.')
else:
    print('x is less than 5.')
```

Output: This code will print the message "x is less than 5." to the console.

Example #5.2: Basic If-Else

```
number = 10

if number > 0:
    print("The number is positive")
else:
    print("The number is not positive")
```

Explanation: This code checks if the variable **number** is greater than 0. If true, it prints "The number is positive", otherwise it prints "The number is not positive".

Example #5.3: Checking Even or Odd

[related video:](#)

```
number = 7

if number % 2 == 0:
    print("The number is even")
else:
    print("The number is odd")
```

Explanation: This code checks if the variable `number` is even or odd. If `number % 2` equals 0, it is even; otherwise, it is odd.

Example #5.4: Age Group Classification

```
age = 25

if age < 18:
    print("Minor")
else:
    print("Adult")
```

Explanation: This code classifies a person as a "Minor" if their age is less than 18, and as an "Adult" otherwise.

Example #5.5: Grade Assignment

[related video](#)

```
score = 85

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

Explanation: This code assigns a grade based on the `score`. It uses multiple `elif` statements to check for different score ranges.

Example #5.6: Voting Eligibility [\[video\]](#)

```
age = 17

if age >= 18:
    print("You are eligible to vote")
else:
    print("You are not eligible to vote")
```

Explanation: This code checks if a person is eligible to vote based on their age. If `age` is 18 or more, it prints "You are eligible to vote"; otherwise, it prints "You are not eligible to vote".

Example #5.7: Nested If-Else

```
number = -5

if number >= 0:
    if number == 0:
        print("The number is zero")
    else:
        print("The number is positive")
else:
    print("The number is negative")
```

Explanation: This code uses nested `if-else` statements to check if the number is zero, positive, or negative.

Example #5.8: Temperature Check

```
temperature = 30

if temperature > 30:
    print("It's a hot day")
else:
    print("It's not a hot day")
```

Explanation: This code checks if the temperature is greater than 30. If true, it prints "It's a hot day"; otherwise, it prints "It's not a hot day".

Example #5.9: Checking String Length

```
string = "Hello, World!"

if len(string) > 10:
    print("The string is long")
else:
    print("The string is short")
```

Explanation: This code checks if the length of the `string` is greater than 10. If true, it prints "The string is long"; otherwise, it prints "The string is short".

Task #5.1: Password Check

Objective: Write a Python program that checks if the provided password matches the correct password and prints an appropriate message.

Instructions:

1. Define a variable `password` and set it to a string value.
2. Create an if-else condition to compare the `password` with the correct password `"password123"`.
3. If the password matches `"password123"`, print `"Access granted"`.
4. If the password does not match, print `"Access denied"`.

Example Input/Output:

Input 1:

```
password = "password123"
```

Output 1:

```
Access granted
```

Input 2:

```
password = "wrongpassword"
```

Output 2:

```
Access denied
```

Explanation:

- The program compares the variable `password` with the correct password `"password123"`.
- In **Example 1**, since the password is correct, the output is `"Access granted"`.
- In **Example 2**, since the password is incorrect, the output is `"Access denied"`.

Task #5.2: Compare Two Numbers in Python

Write a Python program that compares two numbers `a` and `b`. If `a` is greater than `b`, the program should print the message `"a is greater than b"`. Otherwise, it should print `"a is not greater than b"`.

Input:

- The program sets two variables `a` and `b` to specific values (15 and 20 in this case).
- The comparison is then made based on these values.

Expected Output:

- The program will print one of the following outputs depending on the values of `a` and `b`:

1. If `a` is greater than `b`:

Output:

"a is greater than b"

2. If `a` is less than or equal to `b`:

Output:

"a is not greater than b"

Example:**1. Input:**

`a = 15, b = 20`

Output:

"a is not greater than b"

2. Input:

`a = 25, b = 20`

Output:

"a is greater than b"

Explanation: This code checks the relationship between `a` and `b`. If `a` is greater than `b`, it prints a message saying so; otherwise, it prints a different message.

Task #5.3: Determine if a Number is Negative, Zero, or Positive

Write a Python program that checks whether a number `x` is negative, zero, or positive. The program should print:

- "Negative" if `x` is less than 0,
- "Zero" if `x` is equal to 0, and
- "Positive" if `x` is greater than 0.

Input:

- The program sets a variable `x` to a specific value (10 in this case).

Expected Output:

- The program will print one of the following outputs based on the value of `x`:

1. If `x` is less than 0:

Output:

"Negative"

2. If `x` is equal to 0:

Output:

"Zero"

3. If x is greater than 0:

Output:

"Positive"

Example:

1. **Input:**

$x = 10$

Output:

"Positive"

2. **Input:**

$x = -5$

Output:

"Negative"

3. **Input:**

$x = 0$

Output:

"Zero"

Explanation: This code checks whether the value of x is less than 0, equal to 0, or greater than 0, and prints the corresponding message based on the comparison.

- [Video: Use Walrus Operator with if-else \(New\)](#)
- [Video: How to Write Single-Line Code Instead of If-Else Statements](#)
- [Python Quiz -IF](#)

Task #5.4: Compare Two Input Numbers

Write a program that takes two numbers as input and prints whether the first number is greater than, less than, or equal to the second number.

Input:

```
Enter the first number: 10
Enter the second number: 5
```

Sample Output:

```
10 is greater than 5
```

Another Sample Input:

```
Enter the first number: 4
Enter the second number: 4
```

Output:

```
4 is equal to 4
```

Task #5.5: Age Checker

Write a simple program that asks for the user's age and checks if the user is older than 18.

Sample Input:

```
Enter your age: 20
```

Sample Output:

```
You are older than 18.
```

Another Sample Input:

```
Enter your age: 16
```

Output:

```
You are younger than 18.
```

Task #5.6: Divisibility Check

- Write a program that checks if a given number is divisible by both 2 and 3.

Input:

- A single integer, *n*.

Output:

- Print "Yes" if the number is divisible by both 2 and 3.

- Print "No" otherwise.

Example:

- **Input:**

6

- **Output:**

Yes

- **Input:**

10

- **Output:**

No

Task #5.7: Voting Eligibility Check

- Create a program that takes the age and country of a person as input and checks if they are eligible to vote in that country (consider the voting age as 18 and the country as "USA").

Input:

- An integer `age`, representing the person's age.
- A string `country`, representing the person's country.

Output:

- Print "Eligible to vote" if the person is 18 years or older and from the USA.
- Print "Not eligible to vote" otherwise.

Example:

- **Input:**

`age = 20`

`country = USA`

- **Output:**

`Eligible to vote`

- **Input:**

`age = 17`

`country = USA`

- **Output:**

`Not eligible to vote`

- **Input:**

`age = 25`

`country = Canada`

- **Output:**

`Not eligible to vote`

