Control Flow

A Comprehensive Tutorial

Table of contents

01

if Statement

if statement

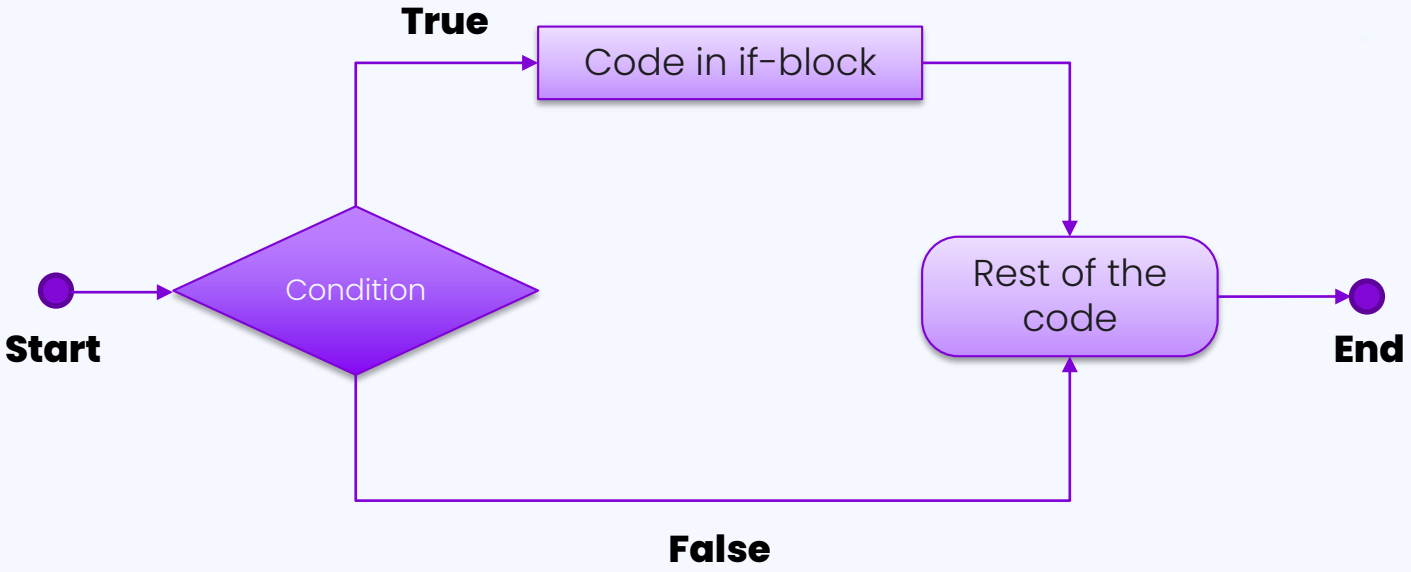
An **if statement** is used to execute a block of code only if a specified condition is **True**.

It is a fundamental control flow tool that allows for decision-making in programs.

Python

```python
if condition:
    # Code to execute if the condition is True
```

if statement - Flowchart



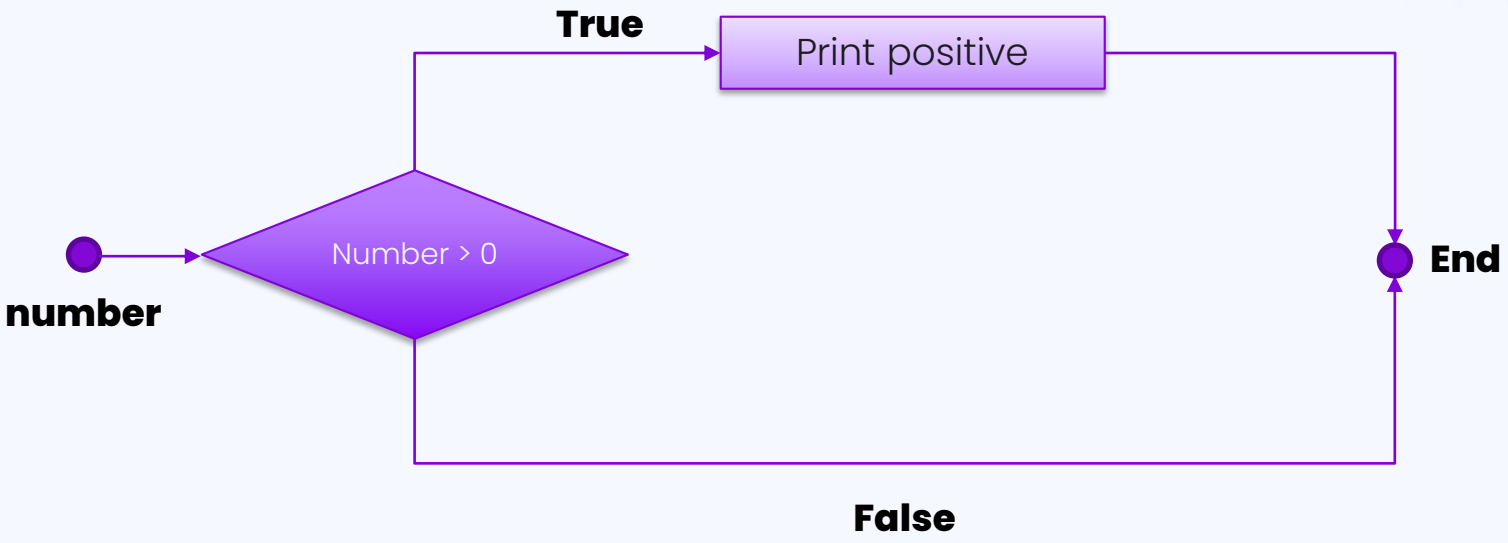
if statement - Example

Python

```python
number = int(input("Enter a number: "))

if number > 0:
    print("The number is positive.")
```

if statement - Flowchart



02

if-else Statement

if-else statement

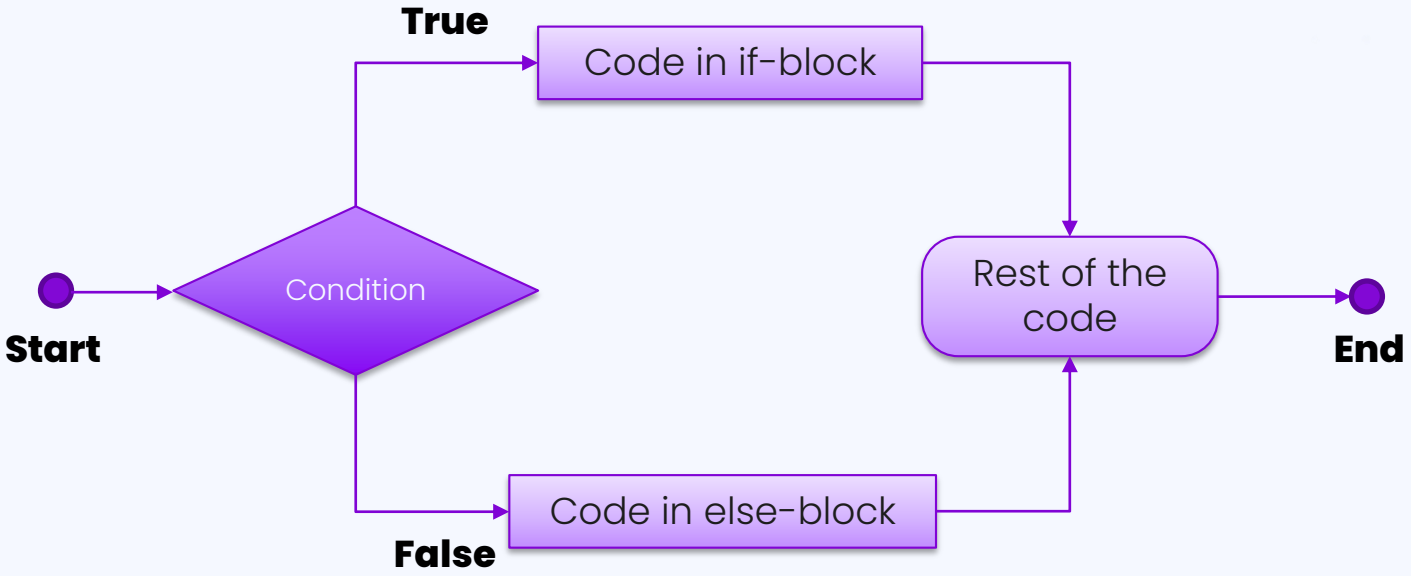
The **if-else statement** in Python is used for **conditional** execution of code.

It allows you to execute a block of code if a certain condition is **true**, and another block if the condition is **false**.

Python

```python
if condition:
    # Code block if condition is true
else:
    # Code block if condition is false
```

if-else - Flowchart



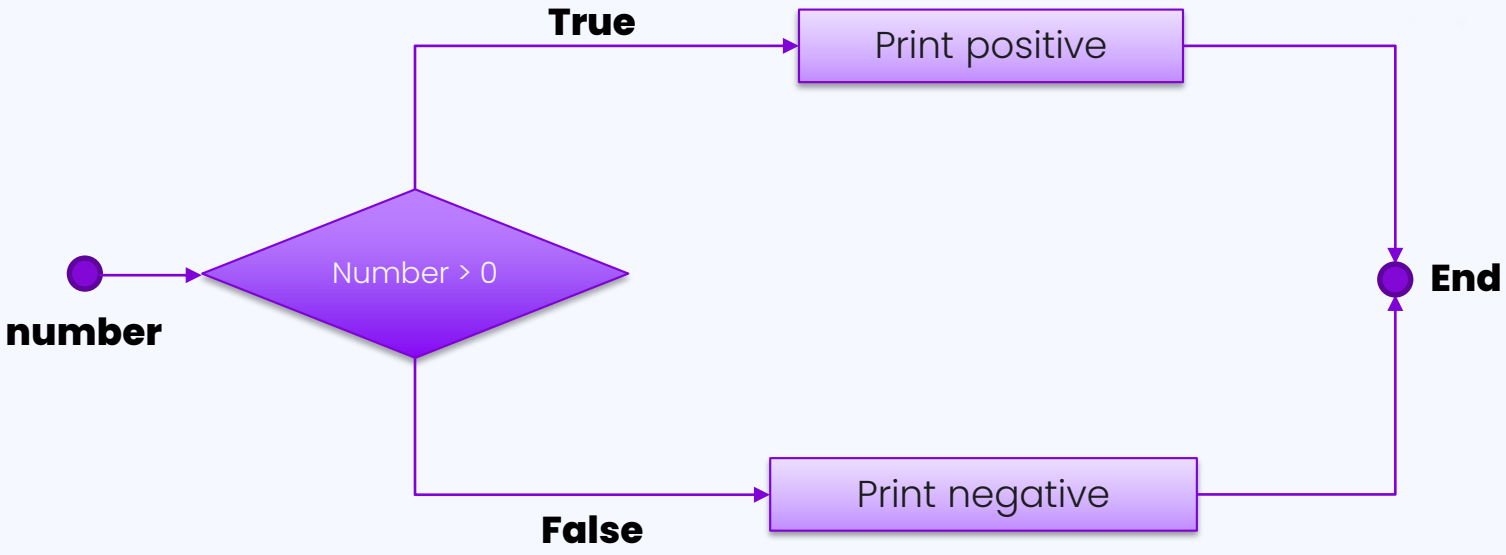
if-else - Example

Python

```python
number = int(input("Enter a number: "))

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

if-else - Flowchart



03

if-elif-else Statement

if-elif-else statement

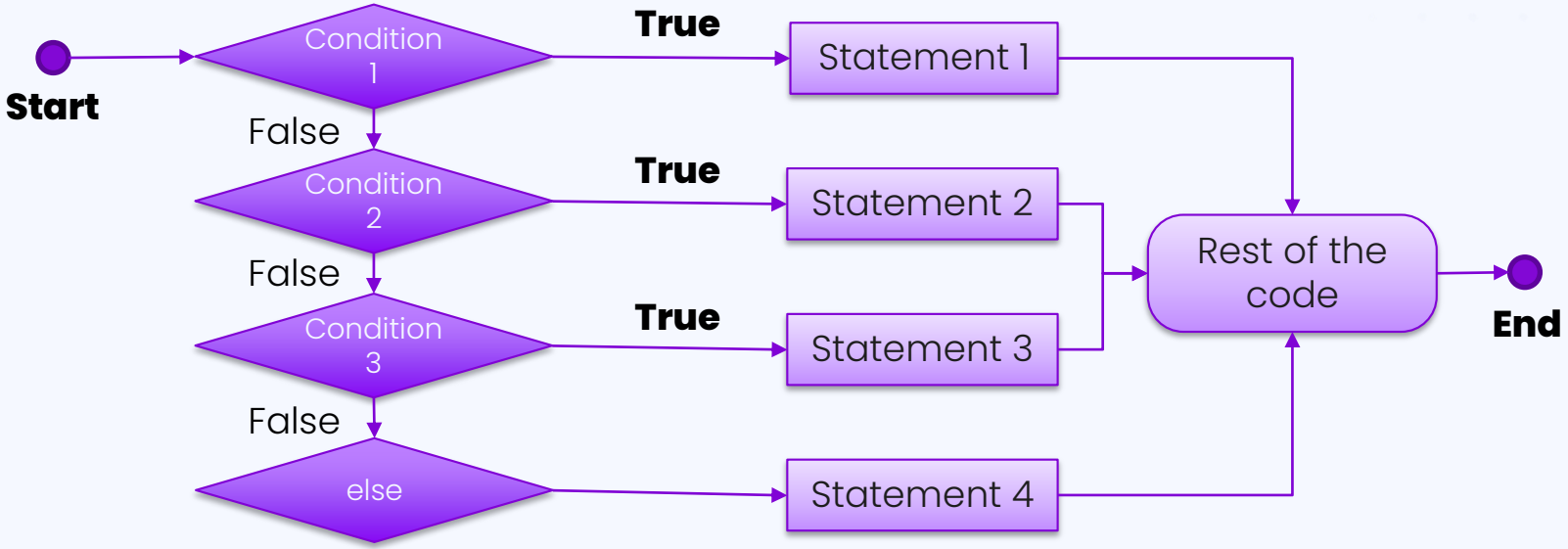
The **if-elif-else statement** in Python is a conditional control structure that allows you to execute a block of code among **multiple** options based on whether a condition is **true** or **false**.

It's used to control the flow of the program and make decisions depending on **different conditions.**

Python

```python
if condition1:
    # code block if condition1 is True
elif condition2:
    # code block if condition2 is True
elif condition3:
    # code block if condition3 is True
else:
    # code block if all conditions are False
```

if-elif-else - Flowchart



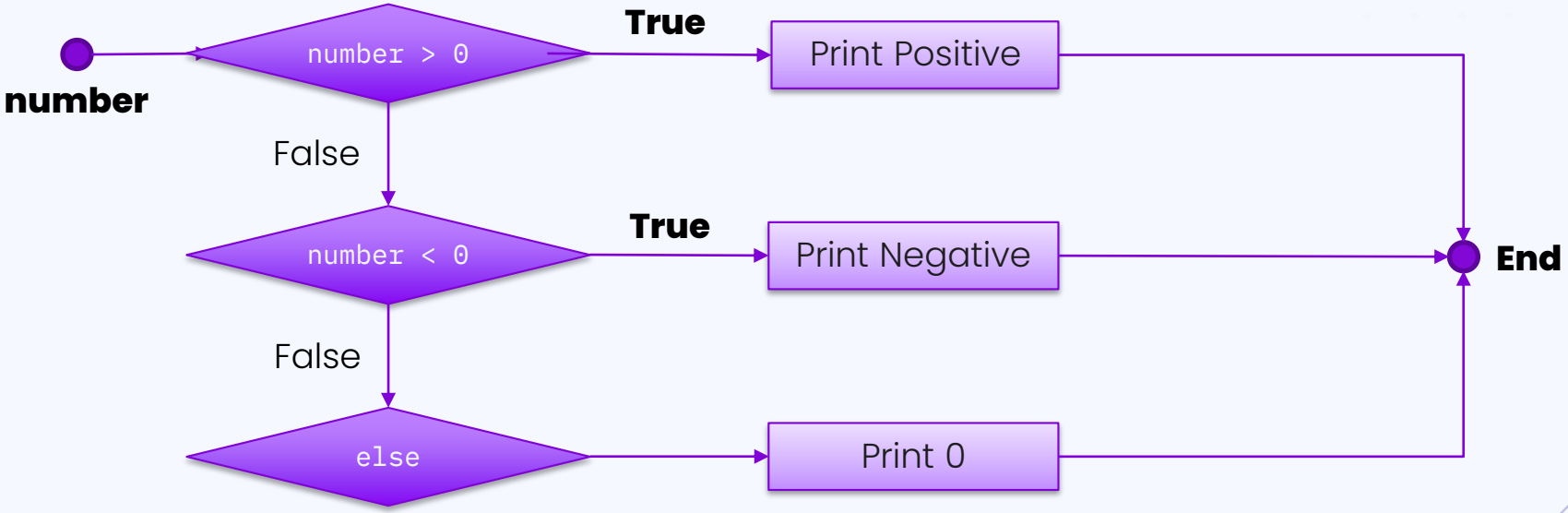
if-elif-else - Example

Python

```python
number = int(input("Enter a number: "))

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

if-elif-else - Flowchart



04

Nested if Statement

Nested If statement

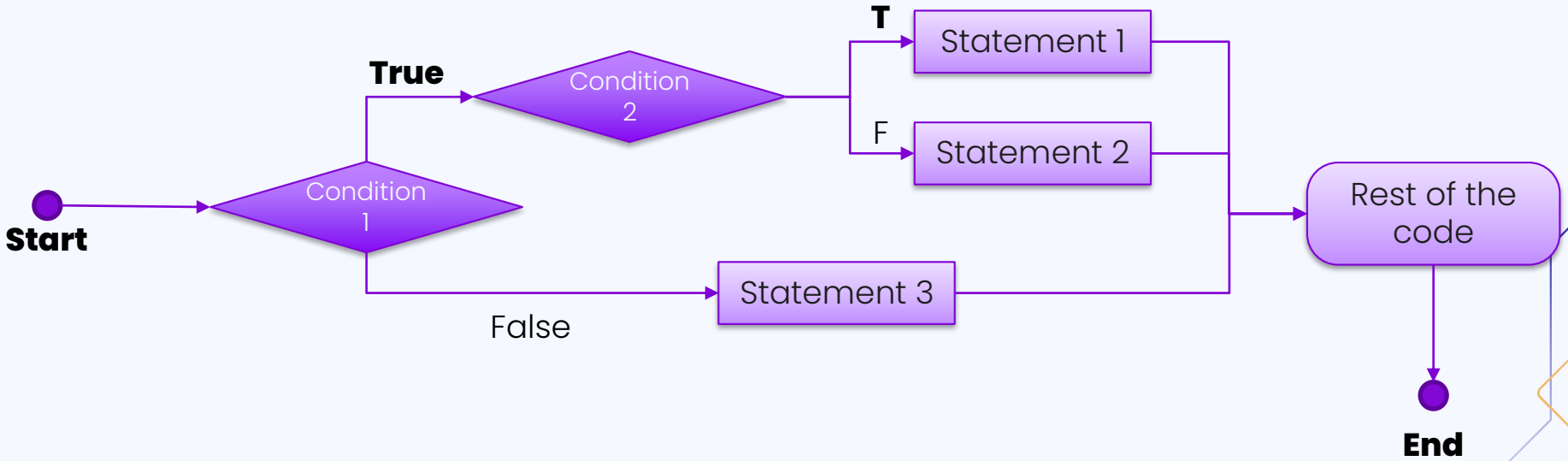
A **nested if** statement occurs when one **if** statement is placed inside another **if** or **else** block.

This allows for more complex decision-making structures, where the outcome of one condition leads to additional conditional checks.

Python

```
if condition1:
    # Condition 1 Code block
    if condition2:
        # Condition 2 Code block
    else:
        # Else 2 Code block
else:
    # Else 1 Code block
```

Nested if - Flowchart

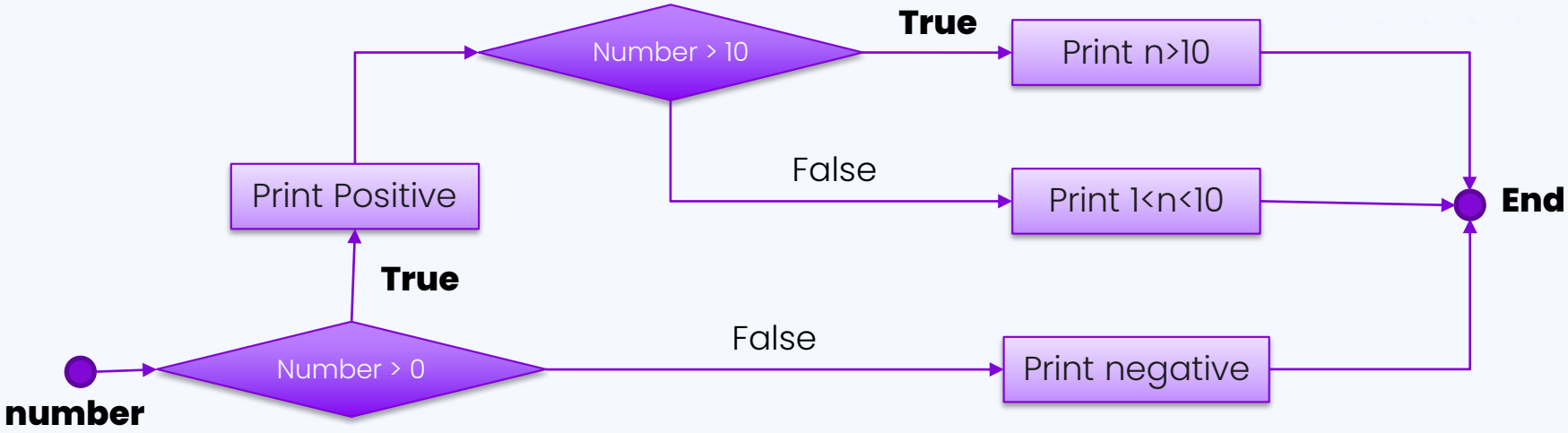


Nested if - Example

Python

```python
number = 15
if number > 0:
    print("The number is positive.")
    if number > 10:
        print("The number is greater than 10.")
    else:
        print("The number is between 1 and 10.")
else:
    print("The number is not positive.")
```

Nested if - Flowchart



05

Ternary Operator using if and nested if

Ternary Operator using if and nested if

The **ternary operator** in Python is a one-liner way of performing conditional operations, often referred to as the conditional expression.

It allows you to evaluate a **condition** and return one **value** if it's **true** and another value if it's **false**, all in a single line.

This is useful when you want to write concise, readable code for simple conditions.

Syntax:

```
Python
<expression_if_true> if <condition> else <expression_if_false>
```

Python

```python
# Direct approach without ternary operator
age = 18
if age >= 18:
    status = "Eligible"
else:
    status = "Not Eligible"
print(status)
```

Python

```python
# Using ternary operator
age = 18
status = "Eligible" if age >= 18 else "Not Eligible"
print(status)
```

Python

```python
# Direct approach without ternary operator
marks = 85

if marks >= 90:
    result = "Excellent"
elif marks >= 75:
    result = "Good"
else:
    result = "Average"
print(result)
```

Python

```python
 Using ternary operator
marks = 85
result = "Excellent" if marks >= 90 else "Good" if marks >= 75 else "Average"
print(result)
```

Knowledge Reinforcement

1

Question

What is the purpose of the **if statement** in Python?

Answer

A) To repeat code multiple times.

B) To execute a block of code only if a specified condition is true.

C) To organize the code into functions.

D) To terminate a loop.

1

Question

What is the purpose of the **if statement** in Python?

Answer

A) To repeat code multiple times.

B) To execute a block of code only if a specified condition is true.

C) To organize the code into functions.

D) To terminate a loop.

2

Question

What will be the output of the following code?

Python

```
number = 0
if number > 0:
    print("Positive")
else:
    print("Non-positive")
```

Answer

A) Positive

B) Non-positive

C) Zero

D) Error

2

Question

What will be the output of the following code?

Python

```
number = 0
if number > 0:
    print("Positive")
else:
    print("Non-positive")
```

Answer

A) Positive

B) Non-positive

C) Zero

D) Error

3

Question

A **nested if** statement allows **if-else statement** to be placed inside another **if** or **else** block.

Answer

A) True

B) False

3

Question

A **nested if** statement allows **if-else statement** to be placed inside another **if** or **else** block.

Answer

A) True

B) False

4

Question

Which statement best describes the **if-elif-else structure** in Python?

Answer

A) It allows all conditions to be checked and all true conditions are executed.

B) It executes all conditions regardless of whether they are true or false.

C) It allows multiple conditions to be checked, and only the first true condition is executed.

D) It is used to handle errors in code.

4

Question

Which statement best describes the **if-elif-else structure** in Python?

Answer

A) It allows all conditions to be checked and all true conditions are executed.

B) It executes all conditions regardless of whether they are true or false.

C) It allows multiple conditions to be checked, and only the first true condition is executed.

D) It is used to handle errors in code.

5

Question

What will be the output of the following code?

```python
number = 15
if number > 0:
    print("The number is positive.")
    if number > 10:
        print("The number is greater than 10.")
    else:
        print("The number is between 1 and 10.")
else:
    print("The number is not positive.")
```

Answer

A) The number is positive. The number is between 1 and 10.

B) The number is not positive.

C) The number is positive. The number is greater than 10.

D) Error

5

Question

What will be the output of the following code?

```python
number = 15
if number > 0:
    print("The number is positive.")
    if number > 10:
        print("The number is greater than 10.")
    else:
        print("The number is between 1 and 10.")
else:
    print("The number is not positive.")
```

Answer

A) The number is positive. The number is between 1 and 10.

B) The number is not positive.

C) The number is positive. The number is greater than 10.

D) Error

6

Question

The **ternary operator** in Python allows you to write an **if-else** condition in a single line.

Answer

A) True

B) False

6

Question

The **ternary operator** in Python allows you to write an **if-else** condition in a single line.

Answer

A) True

B) False

7

Question

What is the correct **syntax** for a ternary operator in Python?

Answer

A) `<expression_if_true> else <expression_if_false> if <condition>`

B) `<condition> ? <expression_if_true> : <expression_if_false>`

C) `<expression_if_true> if <condition> else <expression_if_false>`

D) `<if condition> <expression_if_true> else <expression_if_false>`

7

Question

What is the correct **syntax** for a ternary operator in Python?

Answer

A) `<expression_if_true> else <expression_if_false> if <condition>`

B) `<condition> ? <expression_if_true> : <expression_if_false>`

C) `<expression_if_true> if <condition> else <expression_if_false>`

D) `<if condition> <expression_if_true> else <expression_if_false>`

8

Question

What will be the output of the following code?

Python

```
age = 20
status = "Eligible" if age >= 18 else "Not Eligible"
print(status)
```

Answer

A) Eligible

B) Not Eligible

C) Error

D) None

8

Question

What will be the output of the following code?

Python

```
age = 20
status = "Eligible" if age >= 18 else "Not Eligible"
print(status)
```

Answer

A) Eligible

B) Not Eligible

C) Error

D) None

9

Question

Which of the following is **true** about the **if-else** statement in Python?

Answer

A) The else block is always mandatory.

B) The else block executes only if the if condition is false.

C) The else block executes only if the if condition is true.

D) The if statement cannot have more than one condition.

9

Question

Which of the following is **true** about the **if-else** statement in Python?

Answer

A) The else block is always mandatory.

B) The else block executes only if the if condition is false.

C) The else block executes only if the if condition is true.

D) The if statement cannot have more than one condition.

Practice Set

IF Condition based Questions - Part 1

Download Link in **Description**

WATCH

Level up your coding with each episode in this focused Python series.

Next Video!

Practice Set - 1 Solution