Python: Flow Control Statements

Connect with me: Youtube | LinkedIn | WhatsApp Channel | Web | Facebook | Twitter

- Download PDF
- To access the updated handouts, please click on the following link: https://yasirbhutta.github.io/python/docs/control-flow.html

Flow control statements

Flow control statements in Python determine the order in which your code is executed. They allow you to make decisions, repeat actions, and control the program's flow based on specific conditions.

1. Conditional Statements (if, else, elif)
2. Looping Statements (for, while)
3. Loop Control Statements (break, continue, pass)
4. Match Statement (Python 3.10+)

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 MATLAB:

Example #:

```
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:

Example #1:

```
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 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 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 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: Grade Assignment [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 6: 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 7: 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 8: 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 9: Password Check

```
password = "password123"

if password == "password123":
    print("Access granted")
else:
    print("Access denied")
```

Explanation: This code checks if the `password` is correct. If it matches "password123", it prints "Access granted"; otherwise, it prints "Access denied".

Example 10: Maximum of Two Numbers

```
a = 15
b = 20

if a > b:
    print("a is greater than b")
else:
    print("a is not greater than b")
```

Explanation: This code checks which number is greater between `a` and `b`. If `a` is greater than `b`, it prints "a is greater than b"; otherwise, it prints "a is not greater than b".

Example 11: 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".

Example #12: Using if-elif-else

```
x = 10

if x < 0:
```

```
    print("Negative")
elif x == 0:
    print("Zero")
else:
    print("Positive")
```

- Video: Use Walrus Operator with if-else (New)
- Video: How to Write Single-Line Code Instead of If-Else Statements

Example #:

- Python Quiz -IF

loops

- There are two ways to create loops in Python: with the for-loop and the while-loop.
- Repeat actions using `for` and `while` loops.

for loop

- A for loop in Python is a programming statement that repeats a block of code a fixed number of times.
- The for-loop is always used in combination with an iterable object[^1], like a list or a range.
- The Python for statement iterates over the members of a sequence in order, executing the block each time.
- Video: For loops in Python

Syntax:

```
for item in iterable:
    # code block
```

`iterable` is a sequence of elements such as a list, tuple, dictionary, set, or string. item is a variable that takes on the value of each element in the sequence, one at a time. The code block is executed once for each element in the sequence.

range() function:

- We can use the `range()` function as an iterable in a for loop in Python. The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.
- We can also specify the starting value and the increment value of the sequence using the range() function. For example, range(2, 10, 2) returns a sequence of numbers starting from 2, incrementing by 2, and ending at 8. read more ...

Example #: Print Numbers from 1 to 5

Question: Write a MATLAB program to print the numbers from 1 to 5, using a for loop.

Example #:

```
for i in range(6):
    print(i)
```

Example #: Printing "Building the future, one line at a time." 5 Times Using a for Loop

Question: Write a Python program to print the string "Building the future, one line at a time." 5 times, using a for loop.

Example #:

```
for i in range(5):
    print("Building the future, one line at a time.")
```

Example #: Sum of Numbers from 1 to N

Question: Write a python program to calculate the sum of the first N natural numbers using a for loop.

```
N = 10
sum = 0
for i in range(1,N+1):
    sum = sum + i
print(f"Sum = {sum}")
```

Example #: Print Even Numbers from 2 to 10

Question: Write a Python program to display the even numbers from 2 to 10, inclusive, using a for loop.

```
sum = 0
for i in range(2,11):
    if i%2 == 0:
        sum +=i
        print(i)
print(f"Sum of even numbers: {sum}")
```

Example #: Strings as an iterable

```
string = "python is versatile"
for x in string:
    print(x)
```

Example #: Lists as an iterable

```
collection = ['python', 5, 'd']
for x in collection:
    print(x)
```

Example #: Loop over Lists of lists

```
list_of_lists = [ [1, 2, 3], [4, 5, 6], [7, 8, 9]]
for list in list_of_lists:
    for x in list:
        print(x)
```

- A **nested loop** is a loop inside another loop. It is a powerful programming technique that can be used to solve a wide variety of problems.

Example #: Nested Loops - Multiplication Tables

```
for i in range(1, 11):
    print(f"Multiplication table of {i}")
    for j in range(1, 11):
        print('%d * %d = %d' % (i, j, i*j))
```

for loop examples:

- Video: String as an iterable
- Video: 6 Ways to use List in For loop in Python
- Video: Underscore to Ignore Values in for loop
- How to Print Multiplication Tables in Python
- String as an iterable using for loop
- Calculate the sum of the first N natural numbers

Further reading:

- For loop - Python wiki

while loop

- A while loop in python is a control flow statement that repeatedly executes a block of code until a specified condition is met.

Syntax:

```
while condition:
    # code block
```

Here, condition is a boolean expression that is evaluated before each iteration of the loop. If the condition is `True`, the code block is executed. The loop continues to execute as long as the condition remains True.

Example #: Print numbers from 1 to 10 using while loop

Question: Write a Python program to print the numbers from 1 to 10, using a while loop?

```
count = 1  # Start counting at 1
while count <= 10:  # Keep counting as long as we're less than or equal to 10
    print(count)  # Print the current number
    count += 1  # Add 1 to the count for the next round
```

Example #: Print "Hello, world!" 5 times using while loop

Question: Write a Python program to print the string "Hello, world!" 5 times, using a while loop?

```
i = 1
while i <= 10:
    print('Hello, world!')
    i += 1
```

Example #: Sum of numbers from 1 to 100 using while loop

Question: Write a MATLAB program to calculates the sum of the numbers from 1 to 100 using a while loop.

```
i = 1
sum = 0

while i <= 100:
    sum += i
    i += 1

print(f'Sum = {sum}');
```

Example #: Sum of even numbers from 2 to 20 using while loop

Question: Write a Python program to calculate the sum of the even numbers from 2 to 20 using a while loop.

```
sum = 0  # Initialize a variable to store the sum
number = 1

while number <= 20:
    if number%2 == 0:
      sum += number  # Add the current number to the sum
    number += 1
```

```
print(f'The sum of even numbers from 1 to 20 is: {sum}')
```

Example: Square of numbers less than 5 using while loop

Question : Write a program that prints the sum of the squares of all the numbers from 1 to 4, using a while loop.

```
i = 1
while i < 5:
    square = i ** 2
    print(f'Square of {i} is {square}')
    i += 1
```

Output:

```
Square of 1 is 1
Square of 2 is 4
Square of 3 is 9
Square of 4 is 16
```

Example #:

Question: Write a Python program to prompt the user to enter lines of text until the user enters a blank line. The program should then display the message "You entered a blank line.".

```
inputStr = 'Start';

while inputStr != "":
  inputStr = input("Enter a line of text:")

print('You entered a blank line.')
```

Example: Sum of given numbers till the number entered is zero

Question: Write a Python program to add all the numbers entered by the user until the user enters zero. The program should display the sum of the numbers.

```
sum = 0; # Initialize the sum

# Prompt the user to enter a number
number = int(input('Enter a number: ')) # int() Convert string input to integer

# While the number entered is not zero, add the number to the sum and prompt the
```

```
user to enter another number
while number != 0:
    sum += number
    number = int(input('Enter another number: ')) # int() Convert string input to
integer

# Display the sum of the numbers
print(f'The sum of the numbers is: {sum}')
```

Example : While loop from 1 to infinity, therefore running forever.

```
x = 1
while True:
    print("To infinity and beyond! We're getting close, on %d now!" % (x))
    x += 1
```

while loop examples:

- How to Count Digits in a Positive Integer using Python
- How to Count Occurrence of a Specific Digit in a Number using Python

See also:

- Video: Learn how to use while loops
- Video: Learn how to use INFINITE while loop
- Video: Python while loop

The Range() function

- Video: The range() Function
- Video: Use of range() in for loop

Loop Control Statements (break, continue, pass)

These statements modify the behavior of loops.

break: Terminates the loop entirely. **continue:** Skips the current iteration and moves to the next one. **pass:** Does nothing, often used as a placeholder.

`break`

Exits the loop prematurely.

```
for item in sequence:
    if some_condition:
        break  # exit the loop
```

`continue`

Skips the current iteration and proceeds to the next iteration of the loop.

```
for item in sequence:
    if some_condition:
        continue  # skip the rest of the code in this iteration
    # code to execute if some_condition is False
```

`pass`

A null statement, used as a placeholder.

```
if condition:
    pass  # do nothing
```

Example #:

```
for x in range(3):
    if x == 1:
        break
```

Example #:

```
for i in range(10):
    if i == 5:
        break  # exit the loop when i is 5
    if i % 2 == 0:
        continue  # skip even numbers
    print(i)  # print only odd numbers less than 5
```

- Video: How to Effectively Use Break and Continue Statements
- Video: Using Python break statement with a while loop

The else Clauses on Loops

Example #: for..else

```
for x in range(3):
    print(x)
else:
    print('Final x = %d' % (x))
```

Match Statement (Python 3.10+)

The match statement offers a more concise way to handle multiple comparison cases.

```
def get_fruit_color(fruit):
    match fruit:
        case "apple":
            return "red"
        case "banana":
            return "yellow"
        case _:
            return "unknown"
```

Python Challenge to test your knowledge: Quiz1 | Quiz2 | Quiz3 | Quiz4 | Quiz5

Key Terms

- for
- while
- range
- pass
- else
- match
- break
- continue
- f-string

True/False (Mark T for True and F for False)

Multiple Choice (Select the best answer)

if statement:

1. **Which of the following correctly represents the syntax of an if statement in Python?**
    - A) if condition { block of code }
    - B) if(condition) { block of code }
    - C) if condition: block of code
    - D) None
2. **What is the syntax for a simple if statement in Python?**
    - A) if x == 10:
    - B) for x in range(10):
    - C) while x < 10:
    - D) if (x == 10) then:
3. **What is the purpose of the else block in an if statement?**

A) To execute a code block when the if condition is True B) To execute a code block when the if condition is False C) To create an infinite loop D) To define a function

4. **What happens when none of the conditions in an if-elif-else chain are True, and there is no else block?**

A) The program raises an error. B) The program executes the first if block. C) The program executes the last elif block. D) The program does nothing and continues to the next statement.

5. **What will be the output of the following code?**

```
x = 10
y = 5
if x < y:
    print("x is greater than y")
```

A) "x is greater than y" B) "x is less than y" C) Nothing will be printed D) An error will occur

6. **What happens if none of the conditions in an if-elif chain are True, and there is no else block?**

A) The program raises an error B) The program executes the first if block C) The program does nothing and continues to the next statement D) The program executes the last elif block

7. **What will be the output of the following code?**

```
number = 10
if number % 2 == 0:
    print("Number is even")
else:
    print("Number is odd")
```

A) Number is odd B) Number is even C) Nothing D) Error

8. **Which of the following is the correct way to compare if two variables are equal in an if statement?** A) if x equals 5: B) if x == 5: C) if x = 5: D) if x := 5:

9. **How can you check if a value is NOT equal to 10 in an if statement?** A) if x != 10: B) if x <> 10: C) if x =! 10: D) if x not 10:

10. **What does the following code snippet do?**

```
if x > 0:
    print("Positive")
elif x < 0:
    print("Negative")
else:
    print("Zero")
```

A) Prints "Positive" if x is greater than 0, "Negative" if x is less than 0, and "Zero" if x is 0. B) Prints "Positive" if x is less than 0, "Negative" if x is greater than 0, and "Zero" if x is 0. C) Prints "Positive" if x is 0, "Negative" if x is greater than 0, and "Zero" if x is less than 0. D) Causes an error because the conditions are conflicting.

2. **What is the correct syntax for an if-else statement?**
    - A) if condition: code block
    - B) if condition: code block else: code block
    - C) if condition: code block elif condition: code block
    - D) if condition: code block else: code block elif condition: code block
3. **What happens if none of the conditions in an if-elif-else block are true?**
    - a) The program terminates
    - b) The else block is executed
    - c) An error occurs
    - d) The program continues without executing any block
4. **What will be the output of the following code?**

```
x = 10
if x > 5 and x < 15:
    print("x is between 5 and 15")
else:
    print("x is not between 5 and 15")
```

    - a) x is between 5 and 15
    - b) x is not between 5 and 15
    - c) Error
    - d) No output
5. **What will be the output of the following code?**

```
num = 7
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

    - a) Even
    - b) Odd
    - c) Error
    - d) No output
6. **What will be the output of the following code?**

```
score = 85
if score >= 90:
    print("A")
elif score >= 80:
    print("B")
else:
    print("C")
```

What is the output of the following code?

```
x = 5
if x > 10:
    print("Greater than 10")
elif x > 5:
    print("Greater than 5 but not greater than 10")
else:
    print("5 or less")
```

A) Greater than 10
B) Greater than 5 but not greater than 10
C) 5 or less
D) No output

Answer: C

Which of the following is true about the `elif` statement?

A) `elif` stands for "else if" and is used to check additional conditions after the `if` statement
B) `elif` is the same as `else`
C) `elif` must be followed by an `else` statement
D) You can have multiple `elif` statements in one block

Answer: A, D

What will be the output of the following code?

```
y = 20
if y < 20:
    print("Less than 20")
elif y == 20:
    print("Equal to 20")
else:
    print("Greater than 20")
```

A) Less than 20
B) Equal to 20

C) Greater than 20
D) None of the above

Answer: B

What will be the output of the following code snippet?

```
a = 10
b = 5
if a > b:
    if a > 0:
        print("a is positive")
    else:
        print("a is non-positive")
else:
    print("a is less than or equal to b")
```

A) a is positive
B) a is non-positive
C) a is less than or equal to b
D) No output

Answer: A

Which of the following statements is false about `if-elif-else` statements in Python?

A) `elif` statements are optional in an `if-elif-else` construct
B) An `if` statement can be followed by multiple `elif` statements
C) An `else` statement is mandatory in an `if-elif-else` construct
D) An `if` statement can exist without `elif` or `else`

Answer: C

What will be the output of the following code?

```
x = 0
if x:
    print("True")
else:
    print("False")
```

A) True
B) False
C) No output
D) Error

Answer: B

Truthy and Falsy Values in Python In Python, certain values are considered False when evaluated in a boolean context (like in an if statement). These values include:

- None
- False
- 0 (zero of any numeric type, including 0.0)
- Empty sequences and collections (e.g., '', (), [], {})
- Any object that implements **bool** or **len** to return False or 0, respectively

Which of the following expressions can be used in an `if` condition to check if a number `n` is between 1 and 100 inclusive?

A) `if n > 1 and n < 100:`
B) `if n >= 1 and n <= 100:`
C) `if 1 <= n <= 100:`
D) `if n == 1 or n == 100:`

Answer: B, C

What will be the output of the following code?

```
z = 7
if z < 5:
    print("Less than 5")
elif z == 10:
    print("Equal to 10")
elif z > 5:
    print("Greater than 5 but not 10")
else:
    print("Something else")
```

A) Less than 5
B) Equal to 10
C) Greater than 5 but not 10
D) Something else

Answer: C

What is the purpose of the `else` statement in an `if-elif-else` structure?

A) To check another condition if the previous ones were false
B) To execute a block of code if none of the previous conditions were true
C) To execute a block of code if the previous conditions were true
D) To end the `if-elif-else` structure

Answer: B

What will be the output of the following code?

```
x = 10
if x > 5:
    if x == 10:
        print("x is 10")
    else:
        print("x is greater than 5 but not 10")
else:
    print("x is 5 or less")
```

A) x is 10
B) x is greater than 5 but not 10
C) x is 5 or less
D) No output

Answer: A

Which of the following is correct syntax for an `if-elif-else` statement in Python?

A)

```
if x > y:
print("x is greater")
elif x == y:
print("x is equal to y")
else:
print("x is less")
```

B)

```
if x > y:
    print("x is greater")
elif x == y:
    print("x is equal to y")
else:
    print("x is less")
```

C)

```
if (x > y):
    print("x is greater")
elif (x == y):
    print("x is equal to y")
else:
    print("x is less")
```

D)

```
if x > y:
    print("x is greater")
elif x == y
    print("x is equal to y")
else
    print("x is less")
```

Answer: B, C

What will be the output of the following code?

```
a = 3
b = 4
if a > b:
    print("a is greater")
elif a == b:
    print("a and b are equal")
else:
    print("b is greater")
```

A) a is greater
B) a and b are equal
C) b is greater
D) No output

Answer: C

Which of the following statements will execute if `n = 7`?

```
if n < 5:
    print("n is less than 5")
elif n < 10:
    print("n is less than 10")
else:
    print("n is 10 or more")
```

A) n is less than 5
B) n is less than 10
C) n is 10 or more
D) None of the above

Answer: B

Question 5

Which keyword is used to check an additional condition if the initial `if` statement is `False`?

A) else
B) elif
C) elseif
D) if

Answer: B

What will be the output of the following code?

```
score = 85
if score >= 90:
    print("Grade: A")
elif score >= 80:
    print("Grade: B")
elif score >= 70:
    print("Grade: C")
else:
    print("Grade: F")
```

A) Grade: A
B) Grade: B
C) Grade: C
D) Grade: F

What is the result of the following code?

```
x = 0
if x > 0:
    print("Positive")
elif x == 0:
    print("Zero")
else:
    print("Negative")
```

A) Positive
B) Zero
C) Negative
D) No output

Answer: B

How many `elif` clauses can be included in an `if-elif-else` statement?

A) One
B) Two

C) Unlimited
D) None

Answer: C

What will be the output of the following code?

```
age = 20
if age < 18:
    print("Minor")
elif age >= 18 and age < 65:
    print("Adult")
else:
    print("Senior")
```

A) Minor
B) Adult
C) Senior
D) No output

Answer: B

for loop:

> What is the output of the following code?

```
fruits = ["Apple", "Banana", "Cherry"]
for i, fruit in enumerate(fruits):
    if i == 1:
        print(fruit)
```

1. ☐ Apple
2. ☐ Banana
3. ☐ Cherry
4. ☐ Error

related video: https://youtu.be/-FErgsl9njQ

> What is the correct syntax for a for loop in Python?

1. ☐ for (int i = 0; i < 10; i++):
2. ☐ for i in range(10):
3. ☐ for i = 0 to 9:
4. ☐ for i in 10:

What will be the output of the following code?

```
for i in range(5):
  print(i * 2)
```

a) 0 1 2 3 4 b) 2 4 6 8 10 c) 10 8 6 4 2 d) 0 2 4 6 8

How many times will the following loop execute?

for i in [1, 2, 3, 3]: print("Hello")

a) 3 b) 4 c) 1 d) Infinitely

What is the primary purpose of a for loop in Python? a. To define a function b. To iterate over a sequence c. To create a conditional statement d. To perform mathematical operations

In Python, what does the range() function do when used in a for loop? a. Generates a sequence of numbers b. Defines a list c. Calculates the average d. Determines the length of a string

How is the syntax for a for loop in Python? a. for x in range(10): b. while x < 10: c. loop for x in 10: d. for x = 0; x < 10; x++:

In a for loop, what is the role of the loop variable? a. It is used to define the loop b. It holds the result of the loop c. It is the counter for the loop iterations d. It is optional and can be omitted

Which of the following is the correct way to iterate over elements in a list using a for loop? a. for item in my_list: b. for i = 0; i < len(my_list): c. for element = my_list: d. for (i, item) in enumerate(my_list):

How can you iterate over both the index and elements of a list using a for loop? a. for i in my_list: b. for (i, element) in enumerate(my_list): c. for element in range(my_list): d. for index, element in my_list:

What is the output of the following code?

```
numbers = [1, 2, 3, 4, 5]
for x in numbers:
    print(x * 2)
```

a. 2 4 6 8 10 b. 1 2 3 4 5 c. 1 4 9 16 25 d. 2 4 8 16 32

while loop

What is the output of the following code?

```
count = 0
while count < 3:
    print(count)
    count += 1
```

(a) 0 1 2 (b) 0 1 (c) 1 2 3 (d) The code will run indefinitely.

What will happen if you try to modify the loop variable within the body of a while loop?

(a) The loop will continue as normal. (b) The loop will terminate immediately. (c) The loop may behave unexpectedly, depending on how the variable is modified. (d) The loop will always run indefinitely.

What is the correct syntax for a while loop in Python?

(a) while (condition): (b) while condition {} (c) while condition: (d) while (condition) {}

What is the output of the following code?

```
x = 10
while x > 0:
    print(x)
    x -= 2
```

(a) 10 8 6 4 2 (b) 9 7 5 3 1 (c) 10 9 8 7 6 (d) The code will run indefinitely.

What is the purpose of the else clause in a while loop?

(a) To execute a block of code if the loop condition is never true. (b) To execute a block of code if the loop completes without being terminated by a break statement. (c) To execute a block of code if the loop encounters an error. (d) The else clause cannot be used with a while loop.

What is the difference between pass and return statements in Python?3

A) pass does nothing and return exits the function B) pass exits the function and return does nothing C) pass and return both do nothing D) pass and return both exit the function Answer: A

What is the output of the following code snippet in Python?

i = 0 while i < 5: print(i) i += 1 else: print("Done")

> **A)** 0 1 2 3 4 Done **B)** 0 1 2 3 Done **C)** SyntaxError: invalid syntax **D)** IndentationError: expected an indented block

Watch this video for the answer: https://youtube.com/shorts/9Zw-LuNX9h0

How can you create an if-elif-else statement without using elif keyword in Python?2

A) by using nested if-else statements B) by using logical operators with if-else statements C) by using multiple if statements with indentation D) none of the above Answer: A

What is the difference between break and continue statements in Python?3

A) break terminates the loop and continue skips the current iteration B) break skips the current iteration and continue terminates the loop C) break and continue both terminate the loop D) break and continue both skip the current iteration Answer: A

What is the benefit of using control structures in programming languages?3

A) to make the code more modular and reusable B) to control the flow of execution based on certain parameters or conditions C) to implement various algorithms and logic in the code D) all of the above Answer: D

How can you create an if-else statement in one line in Python?2

A) by using a ternary operator with condition and values B) by using a lambda function with condition and values C) by using a colon after if block and before else block D) by using parentheses around if block and else block Answer: A

What is the output of the following code snippet in Python?

for i in range(5): if i == 3: continue print(i)

A) 0 1 2 3 **B)** 0 1 2 4 **C)** SyntaxError: invalid syntax **D)** IndentationError: expected an indented block

Watch this video for answer: https://youtube.com/shorts/x7CIxqoqccY

What is the disadvantage of using control structures in programming languages?3

A) to make the code more difficult to read and maintain B) to increase the complexity and redundancy of the code C) to decrease the efficiency and performance of the code D) none of the above Answer: D

How can you create an if-elif-else statement in Python?2

A) by using elif keyword after if block and before else block B) by using elif keyword after else block and before if block C) by using elif keyword before if block and after else block D) by using elif keyword before else block and after if block Answer: A

How can you create a nested loop in Python?1

A) by using a loop inside another loop with proper indentation B) by using a loop inside another loop with parentheses C) by using a nested keyword before the inner loop D) by using a colon after the outer loop and before the inner loop Answer: A

What is the output of the following code snippet in Python?

for i in range(5): if i == 3: break print(i)

A) 0 1 2 **B)** 0 1 2 3 **C)** 1 2 3 **D)** 0 1 2 3 4

Watch this video for answer: https://youtube.com/shorts/XNLL6j-P61A

Exercises

if statement

Certainly! Here are 30 exercises to help you practice using `if-elif-else` statements in Python:

1. **Check Positive, Negative, or Zero**

```
# Write a program to check if a number is positive, negative, or zero.
number = int(input("Enter a number: "))
```

2. **Even or Odd**

```
# Write a program to check if a number is even or odd.
number = int(input("Enter a number: "))
```

3. **Largest of Three Numbers**

```
# Write a program to find the largest of three numbers.
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
```

4. **Grade Assignment**

```
# Write a program to assign grades based on the score.
score = int(input("Enter your score: "))
```

5. **Leap Year Check**

```
# Write a program to check if a year is a leap year or not.
year = int(input("Enter a year: "))
```

6. **Age Group Classification**

```
# Write a program to classify age into child, teenager, adult, and senior.
age = int(input("Enter your age: "))
```

7. **Temperature Description**

```
# Write a program to describe the temperature (hot, warm, cold).
temperature = float(input("Enter the temperature: "))
```

8. **Voter Eligibility**

```
# Write a program to check if a person is eligible to vote.
age = int(input("Enter your age: "))
```

9. **Password Strength Checker**

```
# Write a program to check the strength of a password.
password = input("Enter your password: ")
```

10. **Simple Calculator**

```
# Write a simple calculator program using if-elif-else.
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
operation = input("Enter operation (+, -, *, /): ")
```

11. **BMI Calculator**

```
# Write a program to calculate BMI and categorize it (underweight, normal,
overweight, obese).
height = float(input("Enter your height in meters: "))
weight = float(input("Enter your weight in kilograms: "))
```

12. **Triangle Type Checker**

```
# Write a program to check the type of a triangle (equilateral, isosceles,
scalene).
a = float(input("Enter first side: "))
b = float(input("Enter second side: "))
c = float(input("Enter third side: "))
```

13. **Day of the Week**

```
# Write a program to print the day of the week based on a number (1 for
Monday, 7 for Sunday).
day_number = int(input("Enter a number (1-7): "))
```

14. **Letter Grade Assignment**

```
# Write a program to convert a numerical score into a letter grade.
score = int(input("Enter your score: "))
```

15. **Month Days**

```
# Write a program to print the number of days in a given month.
month = int(input("Enter the month number (1-12): "))
```

16. **Discount Calculator**

```
# Write a program to calculate the discount on a product based on the price.
price = float(input("Enter the price of the product: "))
```

17. **Character Checker**

```
# Write a program to check if a character is a vowel or consonant.
char = input("Enter a character: ").lower()
```

18. **Valid Triangle Checker**

```
# Write a program to check if three given lengths can form a triangle.
a = float(input("Enter first side: "))
b = float(input("Enter second side: "))
c = float(input("Enter third side: "))
```

19. **Multiples of Five**

```
# Write a program to check if a number is a multiple of five.
number = int(input("Enter a number: "))
```

20. **Alphabet Case Checker**

```
# Write a program to check if an alphabet is uppercase or lowercase.
char = input("Enter a character: ")
```

21. **Login System**

```
# Write a simple login system that checks username and password.
username = input("Enter username: ")
password = input("Enter password: ")
```

22. **Divisibility Check**

```
# Write a program to check if a number is divisible by 3 and 5.
number = int(input("Enter a number: "))
```

23. **Prime Number Checker**

```
# Write a program to check if a number is prime.
number = int(input("Enter a number: "))
```

24. **Digit or Alphabet Checker**

```
# Write a program to check if a character is a digit or an alphabet.
char = input("Enter a character: ")
```

25. **Fuel Efficiency**

```
# Write a program to categorize fuel efficiency (excellent, good, average,
poor).
mpg = float(input("Enter miles per gallon: "))
```

26. **Odd or Even List**

```
# Write a program to classify numbers in a list as odd or even.
numbers = [int(x) for x in input("Enter numbers separated by space:
").split()]
```

27. **Quadratic Equation Solver**

```
# Write a program to solve a quadratic equation and determine the nature of
the roots.
a = float(input("Enter coefficient a: "))
b = float(input("Enter coefficient b: "))
c = float(input("Enter coefficient c: "))
```

28. **User Age Validator**

```
# Write a program to validate the age input by the user (must be between 0
and 120).
age = int(input("Enter your age: "))
```

29. **Student Result System**

```
# Write a program to check if a student passed or failed based on marks in
three subjects.
marks1 = int(input("Enter marks in subject 1: "))
marks2 = int(input("Enter marks in subject 2: "))
marks3 = int(input("Enter marks in subject 3: "))
```

30. **Shopping Cart Total**

```
# Write a program to calculate the total cost of items in a shopping cart
and apply discount if applicable.
item1 = float(input("Enter price of item 1: "))
item2 = float(input("Enter price of item 2: "))
item3 = float(input("Enter price of item 3: "))
```

Review Questions

for loop:

- What is the difference between a for loop and a while loop in Python?
- Can you use a for loop to iterate over a string?

if statement:

- Can you have multiple elif blocks in an if-elif-else statement?
- What is the purpose of nesting if statements?
-

References and Bibliography

[^1]: In Python, an iterable object is an object that you can loop over using a "for" loop. It's any object that can return its elements one at a time.

Which of the following correctly fixes the syntax error in the code below?

```
if x == 10  # Missing colon
    print("x is 10")
```

A) Remove the comment. B) Add a colon after `if x == 10`. C) Add parentheses around `x == 10`. D) Indent the print statement correctly.

Which of the following will NOT cause a syntax error in Python?

A)

```
1st_variable = 10
```

B)

```
if x == 10:
print("x is 10")
```

C)

```
print("Hello World!"
```

D)

```
print "Hello World!)
```