Control Flow Statements in Python

Control flow statements are used to control the order in which instructions are executed in a program. Python supports several control flow statements.

Conditional Statements

1. if Statement

The `if` statement is used to execute a block of code only if a specified condition is true.

```python
# Example of if statement
age = 18
if age >= 18:
    print("You are an adult.")
```

2. if...else Statement

The `if...else` statement is used to execute one block of code if a condition is true and another block if the condition is false.

```python
# Example of if...else statement
age = 16
if age >= 18:
    print("You are an adult.")
else:
    print("You are a minor.")
```

3. if...elif...else Statement

The `if...elif...else` statement is used to check multiple conditions.

```python
# Example of if...elif...else statement
age = 20
if age < 13:
    print("You are a child.")
elif 13 <= age < 18:
    print("You are a teenager.")
else:
    print("You are an adult.")
```

Looping Statements

1. for Loop

The `for` loop is used to iterate over a sequence (such as a list, tuple, string, or range).

```python
# Example of for loop
for i in range(5):
    print(i)
```

2. while Loop

The `while` loop is used to execute a block of code as long as a specified condition is true.

```python
# Example of while loop
count = 0
while count < 5:
    print(count)
    count += 1
```

3. Nested Loops

Loops can be nested inside other loops.

```python
# Example of nested loops
for i in range(3):
    for j in range(3):
        print(f"i={i}, j={j}")
```

Loop Control Statements

1. break Statement

The `break` statement is used to exit a loop prematurely.

```python
# Example of break statement
for i in range(5):
    if i == 3:
        break
    print(i)
```

2. continue Statement

The `continue` statement is used to skip the current iteration of a loop and move to the next iteration.

```python
# Example of continue statement
for i in range(5):
    if i == 3:
        continue
    print(i)
```

3. pass Statement

The `pass` statement is a null operation. It is used as a placeholder for future code.

```python
# Example of pass statement
for i in range(5):
    if i == 3:
        pass
    print(i)
```

Exception Handling

1. try...except Statement

The `try...except` statement is used to handle exceptions (errors) in a program.

```python
# Example of try...except statement
try:
    result = 10 / 0
except ZeroDivisionError:
    print("Cannot divide by zero!")
```

2. try...except...finally Statement

The `finally` block is used to execute code regardless of whether an exception occurs or not.

```python
# Example of try...except...finally statement
try:
    result = 10 / 2
except ZeroDivisionError:
    print("Cannot divide by zero!")
finally:
    print("This will always be executed.")
```

3. try...except...else Statement

The `else` block is executed if no exceptions are raised in the `try` block.

```python
# Example of try...except...else statement
try:
    result = 10 / 2
except ZeroDivisionError:
    print("Cannot divide by zero!")
else:
    print("Division successful.")
```

Stay Updated

Be sure to   this repository to stay updated with new examples and enhancements!

License

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with   by Panagiotis Moschos (https://github.com/pmoschos)