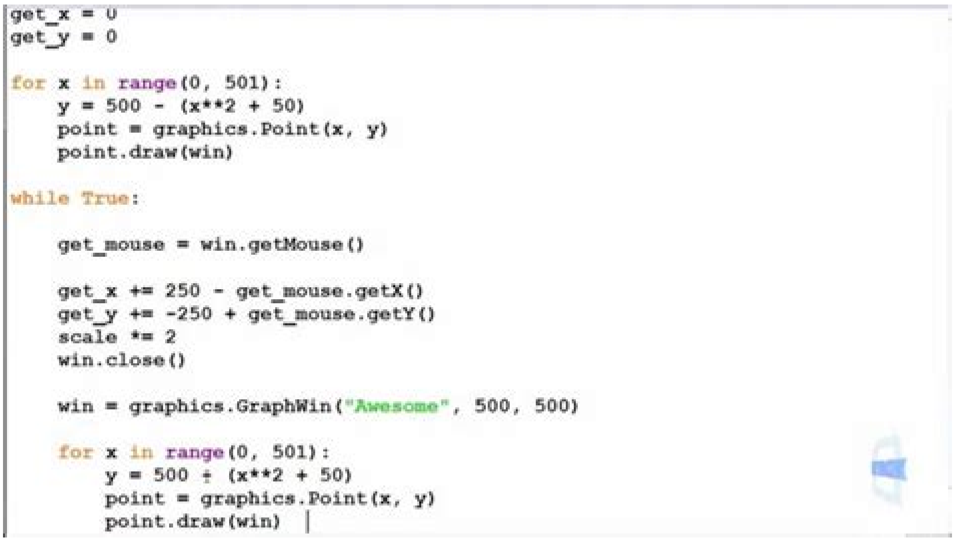
Python While Loop Exercises With Solutions



```python
get_x = 0
get_y = 0

for x in range(0, 501):
    y = 500 - (x**2 + 50)
    point = graphics.Point(x, y)
    point.draw(win)

while True:

    get_mouse = win.getMouse()

    get_x += 250 - get_mouse.getX()
    get_y += -250 + get_mouse.getY()
    scale *= 2
    win.close()

    win = graphics.GraphWin("Awesome", 500, 500)

    for x in range(0, 501):
        y = 500 ÷ (x**2 + 50)
        point = graphics.Point(x, y)
        point.draw(win)
```

Python while loop exercises with solutions are an essential part of mastering the Python programming language. While loops provide a means to repeat a block of code as long as a specified condition remains true. This ability to create iterative processes is fundamental in programming and can be applied in various scenarios, from simple calculations to complex data manipulation. This article will delve into several exercises that utilize while loops, complete with solutions and explanations to help reinforce the concepts.

Understanding the While Loop

Before diving into exercises, it's crucial to understand the structure and syntax of a while loop in Python. The basic syntax of a while loop is as follows:

```python
while condition:
code block to execute
```

The loop will continue to execute the code block as long as the condition evaluates to true. If the condition becomes false, the loop terminates, and the program continues with the next line of code following the loop.

Exercise 1: Basic Counting

Problem Statement

Write a program that counts from 1 to 10 and prints each number.

Solution

```python
count = 1
while count <= 10:
print(count)
count += 1
```

Explanation

In this exercise, we initialize a variable `count` to 1. The while loop checks if `count` is less than or equal to 10. If true, it prints the current value of `count` and then increments `count` by 1. The loop continues until `count` exceeds 10.

Exercise 2: Sum of Natural Numbers

Problem Statement

Write a program to calculate the sum of the first N natural numbers, where N is provided by the user.

Solution

```python
N = int(input("Enter a positive integer: "))
sum_n = 0
count = 1

while count <= N:
sum_n += count
count += 1

print("The sum of the first", N, "natural numbers is:", sum_n)
```

Explanation

This program prompts the user to enter a positive integer `N`. It initializes `sum_n` to 0 and `count` to 1. The while loop runs as long as `count` is less than or equal to `N`, adding `count` to `sum_n` during each iteration and then incrementing `count`. Finally, it prints the calculated sum.

Exercise 3: Factorial Calculation

Problem Statement

Write a program to calculate the factorial of a given number using a while loop.

Solution

```python
N = int(input("Enter a positive integer: "))
factorial = 1
count = 1

while count <= N:
factorial = count
count += 1

print("The factorial of", N, "is:", factorial)
```

Explanation

This program calculates the factorial of a number `N` entered by the user. It initializes `factorial` to 1. Within the while loop, the current `count` is multiplied into `factorial`, and then `count` is incremented. The loop continues until `count` exceeds `N`. The final factorial value is printed.

Exercise 4: Reverse a Number

Problem Statement

Write a program that reverses a given integer.

Solution

```python
number = int(input("Enter an integer: "))
reversed_number = 0

while number > 0:
digit = number % 10 Get last digit
reversed_number = (reversed_number 10) + digit Append digit
number //= 10 Remove last digit

print("Reversed number is:", reversed_number)
```

Explanation

In this exercise, the user inputs an integer, which is stored in `number`. The program initializes `reversed_number` to 0. The while loop runs as long as `number` is greater than 0. During each iteration, it extracts the last digit of `number` using the modulus operator and appends it to `reversed_number`. Finally, the last digit is removed from `number` using floor division. The loop terminates when all digits are reversed.

Exercise 5: Fibonacci Sequence

Problem Statement

Write a program to display the Fibonacci sequence up to the Nth term.

Solution

```python
N = int(input("Enter the number of terms: "))
a, b = 0, 1
count = 0

while count < N:
print(a)
a, b = b, a + b Update values
count += 1
```

Explanation

This program generates the Fibonacci sequence up to `N` terms. It initializes two variables, `a` and `b`, representing the first two terms of the sequence. The while loop runs until the count reaches `N`, printing the current term `a`. The values are updated in each iteration to proceed to the next term.

Exercise 6: Guessing Game

Problem Statement

Create a simple guessing game where the user has to guess a number between 1 and 100. The program should give hints if the guess is too high or too low.

Solution

```python
import random

number_to_guess = random.randint(1, 100)
guess = 0

while guess != number_to_guess:
guess = int(input("Guess a number between 1 and 100: "))
if guess < number_to_guess:
print("Too low! Try again.")
elif guess > number_to_guess:
print("Too high! Try again.")
else:
print("Congratulations! You've guessed the number.")
```

Explanation

In this guessing game, the program generates a random number between 1 and 100. The user is prompted to guess the number. The while loop continues until the user correctly guesses the number. Feedback is provided if the guess is too low or too high, and a congratulatory message is displayed upon success.

Exercise 7: Count Vowels in a String

Problem Statement

Write a program that counts the number of vowels in a given string.

Solution

```python
input_string = input("Enter a string: ")
vowels = "aeiouAEIOU"
count = 0
index = 0

while index < len(input_string):
if input_string[index] in vowels:
count += 1
index += 1

print("Number of vowels in the string:", count)
```

Explanation

This program counts the number of vowels in a user-provided string. It initializes a `count` variable to 0 and an `index` to traverse the string. The while loop iterates through each character of the string, checking if it is a vowel. If it is, `count` is incremented. Once the loop completes, the total number of vowels is printed.

Conclusion

In this article, we explored various Python while loop exercises with solutions, covering a range of topics from basic counting to more complex tasks like guessing games and string manipulation. While loops are a powerful tool in Python, enabling programmers to create dynamic and responsive applications. Practicing these exercises will help solidify your understanding of while loops and enhance your problem-solving skills in Python programming. Remember, the key to mastering programming is continuous practice and experimentation!

Frequently Asked Questions

What is a while loop in Python?

A while loop in Python is a control flow statement that allows code to be executed repeatedly based on a given condition. As long as the condition evaluates to True, the code inside the loop will continue to run.

How can I create a simple counting loop using a while loop?

You can create a simple counting loop like this:

```python
count = 0
while count < 5:
print(count)
count += 1
```

This will output numbers 0 to 4.

What is an infinite loop and how can it occur with a while loop?

An infinite loop occurs when the condition of a while loop always evaluates to True. For example, the code `while True:` will create an infinite loop unless there is a break statement inside to exit the loop.

How can I use a while loop to get user input until a specific condition is met?

You can use a while loop to continually prompt the user for input until they enter a specific value. For example:

```python
user_input = ''
while user_input != 'exit':
user_input = input('Enter something (type exit to stop): ')
```

What is the difference between a while loop and a for loop in Python?

A while loop continues to execute as long as its condition is True, while a for loop iterates over a sequence (like a list or a range). Use a while loop when the number of iterations is not known beforehand.

Can I use a break statement in a while loop?

Yes, you can use a break statement to exit a while loop prematurely. For example:

```python
count = 0
while True:
if count >= 5:
break
print(count)
count += 1
```
This will print numbers 0 to 4 and then exit.

How do I handle user input with validation using a while loop?

You can use a while loop to validate user input like this:

```python
age = -1
while age < 0:
age = int(input('Please enter your age (positive number): '))
```
This will keep prompting until a valid positive number is entered.

Find other PDF article:
https://soc.up.edu.ph/64-frame/pdf?dataid=odN96-7840&title=usmle-step-2-ck-practice-test.pdf

Python While Loop Exercises With Solutions

What does colon equal (:=) in Python mean? - Stack Overflow
Mar 21, 2023 · In Python this is simply =. To translate this pseudocode into Python you would need to know the data structures being referenced, and a bit more of the algorithm ...

What does asterisk * mean in Python? - Stack Overflow
What does asterisk * mean in Python? [duplicate] Asked 16 years, 7 months ago Modified 1 year, 6 months ago Viewed 319k times

What does the "at" (@) symbol do in Python? - Stack Overflow
Jun 17, 2011 · 96 What does the “at” (@) symbol do in Python? @ symbol is a syntactic sugar python provides to utilize decorator, to paraphrase the question, It's exactly about what does ...

Is there a "not equal" operator in Python? - Stack Overflow
Jun 16, 2012 · 1 You can use the != operator to check for inequality. Moreover in Python 2 there was <> operator which used to do the same thing, but it has been deprecated in Python 3.

Using or in if statement (Python) - Stack Overflow
Using or in if statement (Python) [duplicate] Asked 7 years, 6 months ago Modified 8 months ago Viewed 149k times

python - What is the purpose of the -m switch? - Stack Overflow
Python 2.4 adds the command line switch -m to allow modules to be located using the Python module namespace for execution as scripts. The motivating examples were standard library ...

What is Python's equivalent of && (logical-and) in an if-statement?
Mar 21, 2010 · There is no bitwise negation in Python (just the bitwise inverse operator ~ - but that is not equivalent to not). See also 6.6. Unary arithmetic and bitwise/binary operations and ...

syntax - What do >> and <
Apr 3, 2014 · 15 The other case involving print >>obj, "Hello World" is the "print chevron" syntax for the print statement in Python 2 (removed in Python 3, replaced by the file argument of the ...

python - Is there a difference between "==" and "is"? - Stack ...
Since is for comparing objects and since in Python 3+ every variable such as string interpret as an object, let's see what happened in above paragraphs. In python there is id function that shows ...

python - What does ** (double star/asterisk) and * (star/asterisk) ...
Aug 31, 2008 · A Python dict, semantically used for keyword argument passing, is arbitrarily ordered. However, in Python 3.6+, keyword arguments are guaranteed to remember insertion ...

What does colon equal (:=) in Python mean? - Stack Overflow
Mar 21, 2023 · In Python this is simply =. To translate this pseudocode into Python you would need to know the data structures being referenced, and a bit more of the algorithm ...

What does asterisk * mean in Python? - Stack Overflow
What does asterisk * mean in Python? [duplicate] Asked 16 years, 7 months ago Modified 1 year, 6 months ago Viewed 319k times

What does the "at" (@) symbol do in Python? - Stack Overflow
Jun 17, 2011 · 96 What does the “at” (@) symbol do in Python? @ symbol is a syntactic sugar python provides to utilize decorator, to paraphrase the question, It's exactly about what does ...

Is there a "not equal" operator in Python? - Stack Overflow
Jun 16, 2012 · 1 You can use the != operator to check for inequality. Moreover in Python 2 there was <> operator which used to do the same thing, but it has been deprecated in Python 3.

Using or in if statement (Python) - Stack Overflow
Using or in if statement (Python) [duplicate] Asked 7 years, 6 months ago Modified 8 months ago Viewed 149k times

python - What is the purpose of the -m switch? - Stack Overflow
Python 2.4 adds the command line switch -m to allow modules to be located using the Python module namespace for execution as scripts. The motivating examples were standard library ...

What is Python's equivalent of && (logical-and) in an if-statement?
Mar 21, 2010 · There is no bitwise negation in Python (just the bitwise inverse operator ~ - but that is not equivalent to not). See also 6.6. Unary arithmetic and bitwise/binary operations and ...

syntax - What do >> and <
Apr 3, 2014 · 15 The other case involving print >>obj, "Hello World" is the "print chevron" syntax for the print statement in Python 2 (removed in Python 3, replaced by the file argument of the ...

python - Is there a difference between "==" and "is"? - Stack ...

Since is for comparing objects and since in Python 3+ every variable such as string interpret as an object, let's see what happened in above paragraphs. In python there is id function that shows …

python - What does ** (double star/asterisk) and * (star/asterisk) ...

Aug 31, 2008 · A Python dict, semantically used for keyword argument passing, is arbitrarily ordered. However, in Python 3.6+, keyword arguments are guaranteed to remember insertion …

Master Python with our engaging while loop exercises

Back to Home