Bloomify

Your Quiz Performance Feedback

Mcq Quiz • Aug 16, 2025 6:54 AM

Syllabus: Python

Teacher: teacher

Intermediate Knowledge

Performance Overview

2

Max Level Reached

(Apply)

6

Correct Answers

12

Total Questions

50.0%

Accuracy

50.0%

Your Learning Path Recommendation

Intermediate knowledge achieved! Try building projects and practical applications. You have a good grasp of concepts and can apply them effectively.

Performance by Bloom's Taxonomy Level

Remember	3/3 (100.0%)
Understand	3/6 (50.0%)
Apply	0/3 (0.0%)

Detailed Question Analysis

Remember Level • Python Environment Setup

Which of the following is NOT typically considered an Integrated Development Environment (IDE) for Python?

Your Answer: Notepad++

Remember Level • Dictionaries

What is used to access the value associated with a specific key in a Python dictionary?

Your Answer: Key

Remember Level • Function Scope

A variable defined *inside* a function is considered what type of variable?

Your Answer: Local

Understand Level • String Methods

Which of the following BEST describes the purpose of string slicing in Python?

Your Answer:

String slicing allows you to extract a portion of a string, specified by start and end indices.

Understand Level • While and For Loops

What is the key difference in how `while` and `for` loops operate in Python?

Your Answer:

`for` loops are used for iterating over sequences, while `while` loops repeat based on a conditional statement.

Understand Level • File Handling

Imagine you're building a program to log user activity. Which file handling mode would be MOST appropriate to append new entries to an existing log file without overwriting previous data?

Your Answer:

'a' mode appends new data to the end of the file, preserving existing content.

Apply Level • Modules and Functions

You need to create a function that calculates the area of a circle given its radius, and then import and u this function from another file. Which code snippets correctly define and utilize the function across tw (main.py and circle_calc.py)?

Your Answer:

`# circle_calc.py\nfunction circle_area(radius):\n return 3.14159 * radius**2\n\n# main.py\nimport circle_calc\narea = circle_calc.circle_a

Correct Answer:

`# circle_calc.py\ndef circle_area(radius):\n return 3.14159 * radius**2\n\n# main.py\nimport circle_calc\narea = circle_calc.circle_area(5

Apply Level • Lists

You have a list of numbers, and you need to find the largest number in the list. Which code snippet correctly identifies the largest number?

Your Answer:

`numbers = [1, 5, 2, 8, 3]\nlargest = 0\nfor number in numbers:\n if number < largest:\n largest = number\nprint(largest)`

Correct Answer:

`numbers = [1, 5, 2, 8, 3]\nlargest = numbers[0]\nfor number in numbers:\n if number > largest:\n largest = number\nprint(largest)`

Apply Level • Basic Data Types and Operators

A program needs to check if a user-inputted number is even. Which code snippet correctly determines handling potential errors gracefully?

Your Answer: `num = int(input("Enter a number: "))\nif num / 2 == 0:\n print("Even")\nelse:\n print("Odd")`

Correct Answer:

`try:\n num = int(input("Enter a number: "))\n if num % 2 == 0:\n print("Even")\n else:\n print("Odd")\nexcept ValueError:\n print("In

---

Understand Level • Sets and Set Operations

Which of the following BEST explains the concept of set union in Python?

Your Answer:

Set union only includes elements from the first set specified.

Correct Answer:

Set union combines all unique elements from two or more sets into a new set.

Understand Level • Tuples

Which statement best describes the key difference between lists and tuples in Python?

Your Answer:

Tuples are mutable, allowing elements to be added or removed after creation, unlike lists.

Correct Answer:

Tuples are immutable, meaning their contents cannot be changed after creation, unlike lists.

Understand Level • Conditional Statements (if, elif, else)

Consider a scenario where you need to check if a number is positive, negative, or zero. Which approach BEST reflects the most efficient and Pythonic way to implement this using conditional statements?

Your Answer:

Using three separate `if` statements, one for each condition (positive, negative, zero), which may lead to redundant checks.

Correct Answer:

Using a single `if-elif-else` structure, checking for positivity in the `if`, negativity in the `elif`, and zero in the `else` block.

View All Feedback | Back to Dashboard