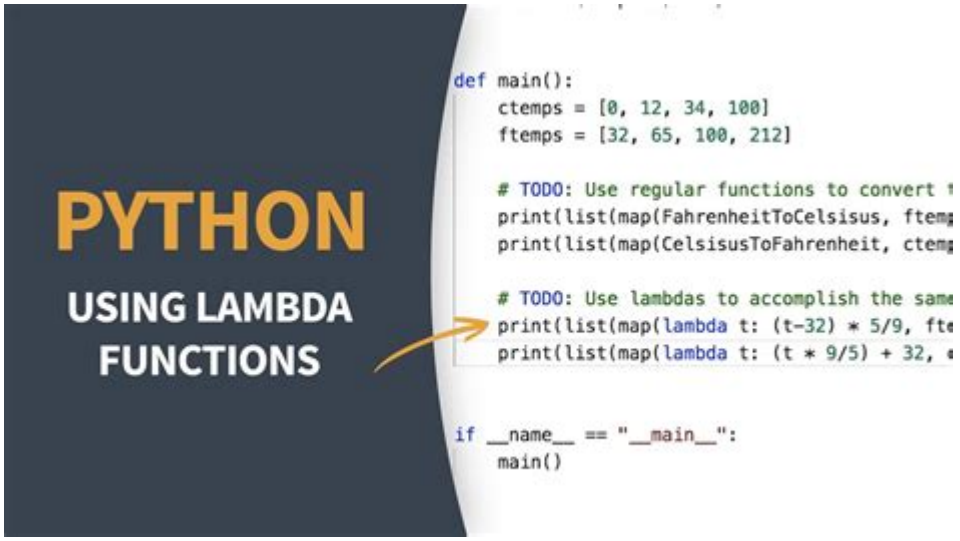
Python Lambda Function Practice Questions



Python lambda function practice questions are an essential part of mastering Python programming, especially for those looking to enhance their coding skills in functional programming. Lambda functions, also known as anonymous functions, provide a concise way to write simple functions without the need for a formal definition using the `def` keyword. They are particularly useful in scenarios where small functions are required for a short period, such as in higher-order functions like `map()`, `filter()`, and `reduce()`. In this article, we will explore various practice questions to solidify your understanding of Python lambda functions.

Understanding Lambda Functions in Python

Before diving into practice questions, it's important to understand what lambda functions are and how they work. A lambda function can take any number of arguments but can only have one expression. The syntax is as follows:

```python
lambda arguments: expression
```

The expression is evaluated and returned when the lambda function is called. Lambda functions are often used for short, throwaway functions, where defining a full function might seem excessive.

Basic Characteristics of Lambda Functions

1. Anonymous: They do not require a name.
2. Single Expression: They can only contain a single expression.

3. Return Value: The result of the expression is returned automatically.
4. Usage: Often used in functional programming constructs.

Practice Questions on Lambda Functions

Here are some practice questions that will help you understand and apply lambda functions effectively in Python.

Question 1: Simple Arithmetic Operations

Write a lambda function that takes two arguments and returns their sum, difference, product, and quotient. Test the lambda function with different input values.

Example Solution:

```python
operation = lambda x, y: (x + y, x - y, x y, x / y)
result = operation(10, 5)
print(result) Output: (15, 5, 50, 2.0)
```

Question 2: Filtering Even Numbers

Create a list of integers and use a lambda function to filter out the even numbers from the list. Use the `filter()` function for this purpose.

Example Solution:

```python
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))
print(even_numbers) Output: [2, 4, 6, 8, 10]
```

Question 3: Sorting a List of Tuples

Given a list of tuples representing (name, age), write a lambda function to sort the list based on age.

Example Solution:

```python
```

```
people = [("Alice", 30), ("Bob", 25), ("Charlie", 35)]
sorted_people = sorted(people, key=lambda x: x[1])
print(sorted_people) Output: [('Bob', 25), ('Alice', 30), ('Charlie', 35)]
```

Question 4: Mapping a Function to a List

Create a list of numbers and use a lambda function with `map()` to square each number in the list.

Example Solution:

```python
numbers = [1, 2, 3, 4, 5]
squared_numbers = list(map(lambda x: x 2, numbers))
print(squared_numbers) Output: [1, 4, 9, 16, 25]
```

Question 5: Finding Maximum in a List of Dictionaries

Given a list of dictionaries, where each dictionary contains a person's name and age, write a lambda function to find the dictionary of the oldest person.

Example Solution:

```python
people = [{"name": "Alice", "age": 30}, {"name": "Bob", "age": 25}, {"name": "Charlie", "age": 35}]
oldest_person = max(people, key=lambda x: x["age"])
print(oldest_person) Output: {'name': 'Charlie', 'age': 35}
```

Advanced Practice Questions

For those who want to challenge themselves further, here are some advanced practice questions involving lambda functions.

Question 6: Chaining Lambda Functions

Write two lambda functions: one that doubles a number and another that adds five to a number. Create a composition of these two functions to apply them in sequence.

Example Solution:

```python
double = lambda x: x 2
add_five = lambda x: x + 5

composed_function = lambda x: add_five(double(x))
result = composed_function(10)
print(result) Output: 25
```

Question 7: Extracting Specific Keys from a List of Dictionaries

Given a list of dictionaries that represent products with various attributes, write a lambda function to extract just the names of the products.

Example Solution:

```python
products = [{'name': 'Laptop', 'price': 1000}, {'name': 'Phone', 'price': 500}, {'name': 'Tablet', 'price': 300}]
product_names = list(map(lambda x: x['name'], products))
print(product_names) Output: ['Laptop', 'Phone', 'Tablet']
```

Question 8: Custom Sorting with Multiple Criteria

Given a list of tuples with (name, score), write a lambda function to sort the list first by score in descending order and then by name in alphabetical order.

Example Solution:

```python
scores = [("Alice", 85), ("Bob", 95), ("Charlie", 85)]
sorted_scores = sorted(scores, key=lambda x: (-x[1], x[0]))
print(sorted_scores) Output: [('Bob', 95), ('Alice', 85), ('Charlie', 85)]
```

Conclusion

Practicing with **Python lambda function practice questions** is a great way to enhance your programming skills and understand functional programming concepts in Python. The versatility of lambda functions makes them an

invaluable tool for any Python developer. By working through these questions, you can gain confidence in your ability to use lambda functions effectively in real-world applications. Whether you are a beginner or an experienced programmer, mastering lambda functions will undoubtedly improve your coding efficiency and style. Keep practicing and exploring more complex use cases to deepen your understanding!

Frequently Asked Questions

What is a Python lambda function?

A Python lambda function is an anonymous function defined using the 'lambda' keyword. It can take any number of arguments but can only have one expression.

How do you define a lambda function that adds two numbers?

You can define it as `add = lambda x, y: x + y`. This function can then be called with two arguments to return their sum.

Can you use lambda functions with the map() function?

Yes, lambda functions are commonly used with the map() function to apply a transformation to each item in an iterable. For example, `list(map(lambda x: x2, [1, 2, 3]))` returns `[1, 4, 9]`.

What is the difference between a lambda function and a regular function?

The main difference is that lambda functions are anonymous and can only consist of a single expression, while regular functions defined using 'def' can have multiple expressions and statements.

Can a lambda function return another lambda function?

Yes, a lambda function can return another lambda function. For example: `outer = lambda x: lambda y: x + y` returns a lambda that takes `y` and adds it to `x`.

How do you sort a list of tuples by the second element using a lambda function?

You can sort it using: `sorted(list_of_tuples, key=lambda x: x[1])`. This sorts the tuples based on their second element.

Is it possible to create a lambda function with no arguments?

Yes, you can create a lambda function with no arguments, like this: `no_arg = lambda: 'Hello, World!'`, which returns 'Hello, World!' when called.

Can you use lambda functions in conjunction with filter()?

Yes, lambda functions are often used with filter() to filter elements from an iterable. For example, `list(filter(lambda x: x > 0, [-1, 0, 1, 2]))` returns `[1, 2]`.

What is a common use case for lambda functions?

Common use cases include short, throwaway functions, especially in data manipulation tasks like sorting, filtering, or applying transformations in functional programming.

How do you create a list of squares using a lambda function and list comprehension?

You can create it as follows: `[lambda x: x2 for x in range(10)]`. However, to get actual values, you'd want to call the lambda inside the comprehension, like `[lambda x: x2(x) for x in range(10)]` with some adjustments.

Python Lambda Function Practice Questions

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 implementation. ...

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 6.7. …

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 print() …

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 a …

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

*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 6.7. ...

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 print() ...

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 a ...

python - What does ** (double star/asterisk) and * (star/asterisk) do ...
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 curated lambda function practice questions! Boost your skills and confidence in coding. Learn more and enhance your programming journey today!

Back to Home