reflection in Python

Last Updated : 17 Nov, 2020

Reflection refers to the ability for code to be able to examine attributes about objects that might be passed as parameters to a function. For example, if we write type(obj) then Python will return an object which represents the type of obj.

Using reflection, we can write one recursive reverse function that will work for strings, lists, and any other sequence that supports slicing and concatenation. If an obj is a reference to a string, then Python will return the str type object. Further, if we write str() we get a string which is the empty string. In other words, writing str() is the same thing as writing “”. Likewise, writing list() is the same thing as writing [].

```
# Python program to illustrate reflection
def reverse(sequence):
    sequence_type = type(sequence)
    empty_sequence = sequence_type()

    if sequence == empty_sequence:
        return empty_sequence

    rest = reverse(sequence[1:])
    first_sequence = sequence[0:1]

    # Combine the result
    final_result = rest + first_sequence

    return final_result

# Driver code
print(reverse([10, 20, 30, 40]))
print(reverse("GeeksForGeeks"))
```

Output:

```
[40, 30, 20, 10]
skeeGroFskeeG
```

Reflection-enabling functions

Reflection-enabling functions include type(), isinstance(), callable(), dir() and getattr().

1. **type and isinstance** – Refer here
2. **Callable() :**A callable means anything that can be called. For an object, determines

method. The callable() method returns True if the object passed appears callable. If not, it returns False.
Examples:

```
x = 5

def testFunction():
  print("Test")

y = testFunction

if (callable(x)):
    print("x is callable")
else:
    print("x is not callable")

if (callable(y)):
    print("y is callable")
else:
    print("y is not callable")
```

Output:

```
x is not callable
y is callable
```

callable when used in OOP

```
class Foo1:
  def __call__(self):
    print('Print Something')

print(callable(Foo1))
```

Output:

```
True
```

3. **Dir :** The dir() method tries to return a list of valid attributes of the object. The dir() tries to return a list of valid attributes of the object.
If the object has __dir__() method, the method will be called and must return the list of attributes.
If the object doesn't have __dir()__ method, this method tries to find information from the __dict__ attribute (if defined), and from type object. In this case, the list returned from dir() may not be complete.

```
number = [1,2,3]
print(dir(number))

characters = ["a", "b"]
print(dir(number))
```

Output:

```
>>> number = [1,2,3]
>>> print(dir(number))
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__f
ormat__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__ini
t__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex
__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subc
lasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', '
sort']
>>> number = ["a", "b"]
>>> print(dir(number))
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__f
ormat__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__ini
t__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex
__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subc
lasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', '
sort']
>>>
```

4. **Getattr :** The getattr() method returns the value of the named attribute of an object. If not found, it returns the default value provided to the function.The getattr method takes three parameters **object**, **name** and **default(optional).**

```
class Employee:
    salary = 25000
    company_name= "geeksforgeeks"

employee = Employee()
print('The salary is:', getattr(employee, "salary"))
print('The salary is:', employee.salary)
```

Output:

```
The salary is: 25000
The salary is: 25000
```

Reference links

2. docs_python
3. wikibooks

Next Article

Sets in Python

Similar Reads

Reflection refers to the ability for code to be able to examine attributes about objects that might be passed as parameters to a function. For example, if we write type(obj) then Python will return an object which represents the ty...

3 min read

Sets in Python

A Set in Python is used to store a collection of items with the following properties. No duplicate elements. If try to insert the same item again, it overwrites previous one.An unordered collection. When we access all items, they are...

9 min read

Self Type in Python

Self-type is a brand-new function added to Python 3.11. A method or function that returns an instance of the class it belongs to is defined by the self type. In situations where we wish to enforce that a method returns an instance of the...

2 min read

str() vs repr() in Python

In Python, the str() and repr() functions are used to obtain string representations of objects. While they may seem similar at first glance, there are some differences in how they behave. Both of the functions can be helpful in debuggin...

3 min read

set copy() in python

The copy() method returns a shallow copy of the set in python. If we use "=" to copy a set to another set, when we modify in the copied set, the changes are also reflected in the original set. So we have to create a shallow copy of the...

2 min read

Duck Typing in Python

Duck Typing is a type system used in dynamic languages. For example, Python, Perl, Ruby, PHP, Javascript, etc. where the type or the class of an object is less important than the method it defines. Using Duck Typing, we do not check...

2 min read

Python List Reverse()

The reverse() method is an inbuilt method in Python that reverses the order of elements in a list. This method modifies the original list and does not return a new list, which makes it an efficient way to perform the reversal without...

2 min read

Python Version History

Python, one of the most popular programming languages today, has a rich history of development and evolution. From its inception in the late 1980s to its current status as a versatile and powerful language, Python's version history...

5 min read

Taking input in Python

Developers often have a need to interact with users, either to get data or to provide some sort of result. Most programs today use a dialog box as a way of asking the user to provide some type of input. While Python provides us with two...

3 min read

Python String

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects