Python QA

20 April 2023   11:14

Q1. What is Python and why is it popular?

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Reason for Popularity :

1. Open Source & Community-driven : OSI-approved open source license
2. Simplicity : easily readable, clear and concise syntax, small learning curve and hence lowers product's maintenance cost.
3. Versatility : Scripting, Web development, Scientific computing, Machine learning, Data analysis etc.
4. Vast collection of ready-made libraries : develop complex applications with minimal effort
5. Light weight in nature : Go to for serverless applications & Raspberry Pi.

Q2. What are the differences between Python 2 and Python 3 ?

Feature	Python 2.x	Python 3.x
Print function syntax	print "hello"	print("hello")
Raw input function	raw_input()	input()
Division operators	7/2 == 3, 7//2 == 3.5	7/2 == 3.5, 7//2 == 3
Exception handling syntax	except Exception, e:	except Exception as e:
xrange function	returns an iterator	removed
range function	returns a list	returns an iterator
Comparison of different types	No error	TypeError
Division by zero	No exception	ZeroDivisionError
Unicode literals	u"hello"	"hello"
String type handling	ASCII by default	Unicode by default

Q3. What is the difference between a tuple and a list in Python?

Properties	Tuple	List
Mutability	Immutable	Mutable
Syntax	Created with ()	Created with []
Memory	Efficient	Less Efficient
Reading performance	Faster	Slower
Writing performance	Slower	Faster

Q4. How do you create a dictionary in Python?

```python
emptyDictionary = {}  # {} or dict()
fibonacci = {0: 0, 1: 1, 2: 2, 3: 3, 4: 5, 5: 8}
cars = dict(brand = "Ford", model = "Mustang", year = 1964) # mixed type
```

Q5. What is a function in Python and how do you define one?

A function is a block of code which only runs and returns a value when it is called by the function name with its parameter values.
it can be defined using 'def' keyword followed by function name with parameters enclosed by parathesis.

Syntax :
```python
def <FunctionName>(param1, param2, *params):
    pass
```

Q6. What is object-oriented programming (OOP) and how does it relate to Python ?

OOP is based on the idea of modeling real-world objects and relationships, rather than just writing code that executes a set of instructions.
This allows for more modular, reusable, and maintainable code, easier collaboration for building large-scale software systems.

Key Pillars / Principles of OOP : Encapsulation, Inheritance, Polymorphism, Abstraction

Python not only supports key pillars of OOP but also provides advanced OOP features such as multiple inheritance, abstract classes, and decorators.
In Python, everything is an object, including data types, functions, and modules.

Like any OOP language, Python allows defining your own classes, which are used to create objects with properties and behaviors.

Q7. How do you handle exceptions in Python?

```python
def OpenFile():
    try:
        file = open("Appraisal.txt", "r")
    except FileNotFoundError as e:
        print("File not found", e)
    finally:
        file.close()
```

Q8. How do you read and write files in Python?

```python
class FileAccessExample:

    def ReadFile():
        with open("C:\MyFile.txt", "r") as file:
            data = file.read()
            print(data)

    def WriteFile():
        try:
            with open("C:\MyFile.txt", "w") as file:
                file.write('Hello World')
        finally:
            file.close

    WriteFile()
    ReadFile()
```
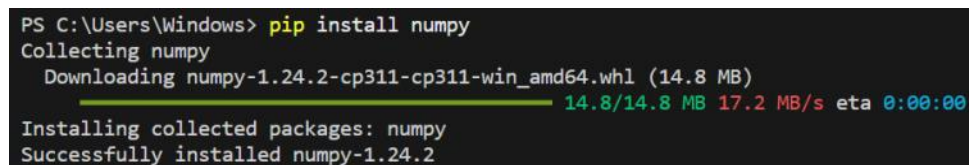
Q9. How do you install and use external packages in Python?

Code to install package using command prompt or VsCode terminal or PowerShell :

```
pip install package_name
```

```
PS C:\Users\Windows> pip install numpy
Collecting numpy
  Downloading numpy-1.24.2-cp311-cp311-win_amd64.whl (14.8 MB)
                                    14.8/14.8 MB 17.2 MB/s eta 0:00:00
Installing collected packages: numpy
Successfully installed numpy-1.24.2
```

Q10. How do you use the "if" statement in Python to perform conditional execution?

Syntax :

```python
if <condition1>:
    pass
elif <condition2>:
    pass
else:
    pass
```

Example :

```python
x = 5

if x > 10:
    print("x > 10")
elif x > 5:
    print("10 > x > 5")
else:
    print("x <= 5")
```