Exercise 13 – Testing

Objective

To test a function with both doctest and unittest

Questions

1. Create a new file called vat\_calc.py. In this file, create a function called **vat\_calc()** which accepts **amt** and **rate** as arguments. The function should take the amount and the VAT rate, and return the amount of VAT to be paid.

Write a docstring which defines what the function does. Include a doctest with the values 100 and 20. These should return the value 20 (or 20.0 depending on the logic)

Create a main function which outputs the following:

print(f"20% of 100 = {vat\_Calc(100,20)}")

if \_\_name\_\_ == “\_\_main\_\_”:

import doctest

doctest.testmod()

main()

run the file to check that it has worked

Back in your function, write another test that uses 150.55 and 20 as the arguments. Add another line to your main function and retest. (the value should be 30.11)

1. Create a new file called testVat\_Calc.py. This file will be used to write the tests in an external file, which is far more likely in industry.

Import unittest first, then import your function from your first file.

Unit tests are class based, so create a new class called **TestVat\_Calc** which extends inherits from unittest.TestCase.

Create a new def called **test\_vat\_calc(self)** and assertEquals that vat\_Calc(100,20) is 20. Include a string to say what it should be at the end.

When you check if **name** == **main**, call unittest.main()

Run and check for output

**If time allows…**

Install pytest   
use this site [How to write and report assertions in tests — pytest documentation](https://docs.pytest.org/en/7.1.x/how-to/assert.html)  
to understand how to write code for pytest. Re-write your code to work with pytest

Solutions

Here are our versions of these exercises, remember that yours can be different to these, but still correct. If in doubt, ask your instructor.

def vat\_Calc(amt, rate):  
 """ Returns the amount of VAT to pay based on the amt and rate  
 >>> vat\_Calc(100, 20)  
 20.0  
  
 >>> vat\_Calc(150.55, 20)  
 30.11  
 """  
 return (amt / 100) \* rate  
  
  
def main():  
 print(f"20% of 100 = {vat\_Calc(100,20)}")  
 print(f"20% of 150.55 = {vat\_Calc(150.55, 20)}")  
 return None  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import doctest  
 doctest.testmod()  
 main()

2.

import unittest  
from vat\_calc import vat\_Calc  
  
class Testvat\_Calc(unittest.TestCase):  
 def test\_vat\_Calc(self):  
 self.assertEquals(vat\_Calc(100,20), 20, "Should be 20.0")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 unittest.main()

**If time allows…**

def vat\_calc(amt, rate):  
 return (amt / 100) \* rate  
  
  
def test\_vat\_calc():  
 assert vat\_calc(100,20) == 20.0

in the terminal focused on the folder with your files:

pytest vat\_calc.py