Meaning of Testing

In [ ]:
```
Meaning of Testing --> Testing is a process by which we can validate,verify and check our applications/softwares.
                   --> it is basically used to check how much your software is upto mark.
```

Types of Testing

In [ ]:
```
Types of Testing:

    Unit Testing         --> It focuses on the smallest unit of software design. In this, we test an individual unit or group of
                             interrelated units.
    Integration Testing --> The objective is to take unit-tested components and build a program structure that has been
                             dictated by design.
                        --> Integration testing is testing in which a group of components is combined to produce output.
    System Testing       --> System testing is testing conducted on a complete integrated system to evaluate the system's
                             compliance with its specified requirements.
    Smoke Testing        --> Smoke Testing is a software testing method that determines whether the employed build is
                             stable or not.
    Acceptance Testing  --> Acceptance Testing is the last phase of software testing performed after System Testing and
                             before making the system available for actual use.
```

assert keyword

In [ ]:
```
assert keyword --> Assert is a keyword that is used for debugging the code.
               --> Assert will check the condition if condition is true then assert will return True else assert will
                   return an Assertion Error
               --> You can write a message with assertion error as well if the assert will return False
```

Example

In [5]:
```
x="Aman"
assert x=="Aman1","x is not Aman"
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
Input In [5], in <cell line: 2>()
      1 x="Aman"
----> 2 assert x=="Aman1","x is not Aman"

AssertionError: x is not Aman
```

In [1]:
```
x="Aman"
assert x=="Aman"
```

Framework Present in Python for Testing

In [ ]:
```
Frameworks that are present in Python:
    1.Unittest
    2.pytest
    3.Doctest
    4.Testify
    ...
    ...
    ...
    ...
```

About Pytest

In [ ]:
```
--> pytest is framework in python that is used to write our own test cases based on certain inputs or criteria using python
    programming language.
```

Why pytest

In [ ]:
```
Why pytest :
    --> very easy to start beacuse it has very simple syntax
    --> Skip Test
    --> Open source
    --> Automatically detetct tests
```

Important Note

In [ ]:
```
--> You cannot directly use pytest for using pytest you need to use either teriminal or command prompt
--> Command --> pytest file_name.py
```

Example -1

In [ ]:
```
def add(x,y):
    return x+y
def product(x,y):
    return x*y

def test_add():
    assert add(7,3)==10
    assert add(9)==10
    assert add(5)==7

def test_product():
    assert product(5,5)==25
    assert product(5)==25
    assert product(7)==35
```

Example -2

In [10]:
```
def factorial(n):
    if n<0:
        return "Negative Number"
    elif n==0:
        return 1
    elif n==1:
        return 1
    else:
        fact=1
        for i in range(2,n+1):
            fact=fact*i
        return fact
def test_factorial():
    assert factorial(0)==1
    assert factorial(1)==1
    assert factorial(5)==120
    assert factorial(-9) == "Negative Number"
```

Command Prompt Output

In [ ]:
```
(base) C:\Users\praty>pytest pythontest.py
============================= test session starts =============================
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\praty
plugins: anyio-3.5.0
collected 2 items

pythontest.py .F

================================== FAILURES ===================================
_____________________________ test_product ______________________________

    def test_product():
        assert product(0,1) ==0
>       assert product (10,8) ==90
E       assert 80 == 90
E         +  where 80 = product(10, 8)

pythontest.py:13: AssertionError
=========================== short test summary info ===========================
FAILED pythontest.py::test_product - assert 80 == 90
========================= 1 failed, 1 passed in 0.22s =========================

(base) C:\Users\praty>pytest pythontest.py
============================= test session starts =============================
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\praty
plugins: anyio-3.5.0
collected 2 items

pythontest.py ..

============================== 2 passed in 0.04s ==============================

(base) C:\Users\praty>pytest pythontest.py
============================= test session starts =============================
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\praty
plugins: anyio-3.5.0
collected 1 item

pythontest.py E

==================================== ERRORS ====================================
_____________________ ERROR at setup of test_factorial _____________________
file C:\Users\praty\pythontest.py, line 12
  def test_factorial(x):
E       fixture 'x' not found
>       available fixtures: anyio_backend, anyio_backend_name, anyio_backend_options, cache, capfd, capfdbinary, caplog, capsys, capsysbinary, docte
>       use 'pytest --fixtures [testpath]' for help on them.

C:\Users\praty\pythontest.py:12
=========================== short test summary info ===========================
ERROR pythontest.py::test_factorial
=============================== 1 error in 0.08s ===============================

(base) C:\Users\praty>pytest pythontest.py
============================= test session starts =============================
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\praty
plugins: anyio-3.5.0
collected 1 item

pythontest.py F

================================== FAILURES ===================================
_____________________________ test_factorial ______________________________

    def test_factorial():
>       assert factorial(-1) =="Negative factorial is not possible"
E       AssertionError: assert 1 == 'Negative factorial is not possible'
E         +  where 1 = factorial(-1)

pythontest.py:13: AssertionError
=========================== short test summary info ===========================
FAILED pythontest.py::test_factorial - AssertionError: assert 1 == 'Negative factorial is not possible'
============================== 1 failed in 0.22s ==============================

(base) C:\Users\praty>pytest pythontest.py
============================= test session starts =============================
platform win32 -- Python 3.9.12, pytest-7.1.1, pluggy-1.0.0
rootdir: C:\Users\praty
plugins: anyio-3.5.0
collected 1 item

pythontest.py .

============================== 1 passed in 0.03s ==============================

(base) C:\Users\praty>
```

pdb (Python Debugger)

In [ ]:
```
pdb --> It is also a module of python that will help you to debug your.py file. Internally it  makes(basic debugger functions)
        and cmd. pdb is stand for Python debugger.
    --> Command for start debugger --> python -m pdb pdbdemo.py
```

Example

In [ ]:
```
import pdb
def add(x,y):
    return x+y
x=int(input("Enter a Number :"))
y=int(input("Enter a Number :"))
z=add(x,y)
print(z)
```

On Command Prompt:

In [ ]:
```
(base) C:\Users\praty>python pythonpdb.py
Enter a Number :10
Enter a Number :20
30

(base) C:\Users\praty>python  -m pdb pythonpdb.py
> c:\users\praty\pythonpdb.py(1)<module>()
-> import pdb
(Pdb) help

Documented commands (type help <topic>):
========================================
EOF     c          d          h         list      q        rv       undisplay
a       cl         debug      help      ll        quit     s        unt
alias   clear      disable    ignore    longlist  r        source   until
args    commands   display    interact  n         restart  step     up
b       condition  down       j         next      return   tbreak   w
break   cont       enable     jump      p         retval   u        whatis
bt      continue   exit       l         pp        run      unalias  where

Miscellaneous help topics:
==========================
exec  pdb

(Pdb) n
> c:\users\praty\pythonpdb.py(2)<module>()
-> def add(x,y):
(Pdb) n
> c:\users\praty\pythonpdb.py(4)<module>()
-> x=int(input("Enter a Number :"))
(Pdb) n
Enter a Number :10
> c:\users\praty\pythonpdb.py(5)<module>()
-> y=int(input("Enter a Number :"))
(Pdb) n
Enter a Number :20
> c:\users\praty\pythonpdb.py(6)<module>()
-> z=add(x,y)
(Pdb) n
> c:\users\praty\pythonpdb.py(7)<module>()
-> print(z)
(Pdb) n
30
--Return--
> c:\users\praty\pythonpdb.py(7)<module>()->None
-> print(z)
(Pdb) n
--Return--
> <string>(1)<module>()->None
(Pdb) exit
```