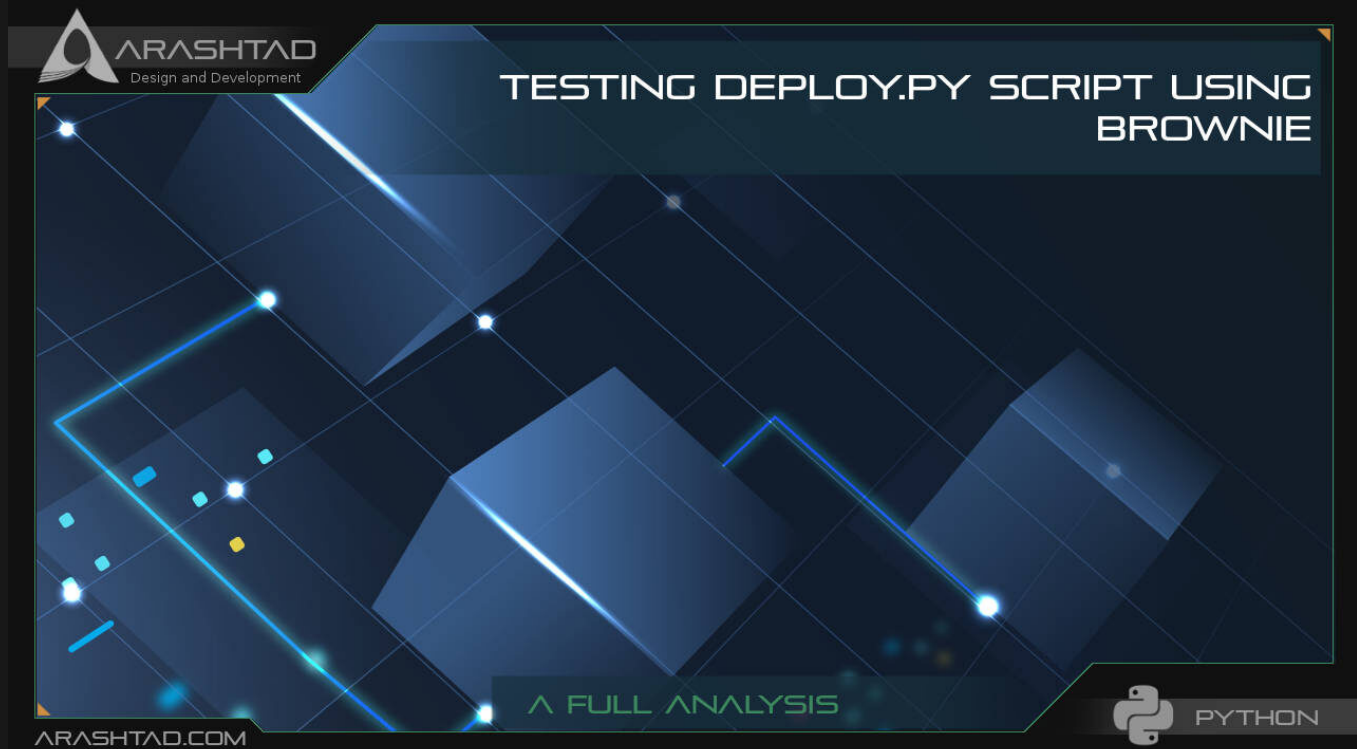


Testing Deploy.py Script Using Brownie: A Full Analysis

No comments



*In this post, we will complete **testing our deploy.py script using Brownie** in more detail so that we can see if there is any problem at all. And if there is, we can analyze it in more detail. This kind of detailed test and code analysis becomes handy, especially when the code is too long and complex that we cannot so easily understand where the unexpected result originates from.*

Detailed Testing Deploy.py Using Brownie

For more detailed testing, we can write in the terminal:

```
brownie test -k
```

Which in our case will be:

```
brownie test -k test_updating_storage
```

And the result will be:

```
Brownie v1.18.1 - Python development framework for
```



```
(Pdb) simple_storage.retrieve()>>0
```

By understanding the difference between the expected number and the retrieved one, we will be able to debug the code.

```
(Pdb) simple_storage
```

The better way to fully understand the problems and be able to find it, is to type in the terminal:

```
brownie test -s          Brownie v1.18.1 - Python development framework for
Ethereum===== test
session starts =====
platform linux -- Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
-- /home/mohamad/.local/pipx/venvs/eth-brownie/bin/python cachedir:
.pytest_cache hypothesis profile 'brownie-verbose' -> verbosity=2,
deadline=None, max_examples=50, stateful_step_count=10,
report_multiple_bugs=False,
database=DirectoryBasedExampleDatabase(PosixPath('/home/mohamad/.brownie/hypo
rootdir: /home/mohamad/Desktop/solidity/Solidity and web3
codes/brownie_simple_storage plugins: eth-brownie-1.18.1, web3-5.27.0,
xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 2
itemsLaunching 'ganache-cli --chain.vmErrorsOnRPCResponse true --
server.port 8545 --miner.blockGasLimit 12000000 --wallet.totalAccounts
10 --hardfork istanbul --wallet.mnemonic
brownie'...tests/test_simple_storage.py::test_deploy FAILED
tests/test_simple_storage.py::test_updating_storage
PASSED=====
FAILURES =====
_____ test_deploy
_____ def
test_deploy(): account = accounts[0] simple_storage =
SimpleStorage.deploy({"from": account}) starting_value =
simple_storage.retrieve() expected = 38 > assert starting_value ==
expected E assert 0 == 38 E +0 E -38tests/test_simple_storage.py:10:
AssertionError =====
short test summary info
===== FAILED
tests/test_simple_storage.py::test_deploy - assert 0 == 38
===== 1 failed, 1 passed
in 1.94s ===== Terminating
local RPC client...
```

As you can see in the result, the code and its variables are fully analyzed.

Fixing the Code

Now if we fix our code and change the expected in the first function to 0:

```
from brownie import SimpleStorage, accounts
def test_deploy():
```

```
account = accounts[0]
simple_storage = SimpleStorage.deploy({"from": account})
starting_value = simple_storage.retrieve()
expected = 0
assert starting_value == expected

def test_updating_storage():

    account = accounts[0]
    simple_storage = SimpleStorage.deploy({"from": account})
    expected = 38
    simple_storage.store(expected, {"from": account})
    assert expected == simple_storage.retrieve()
```

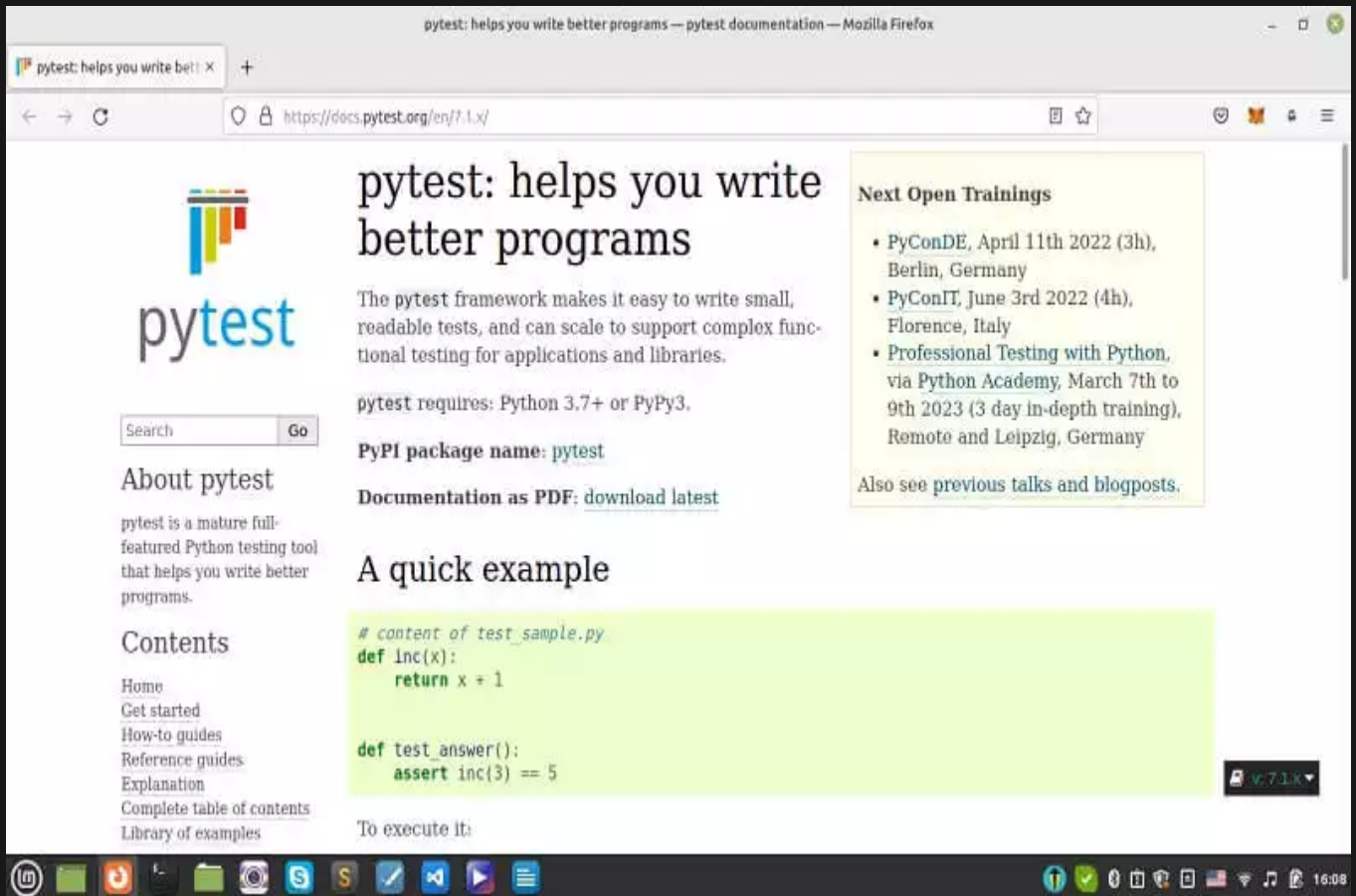
And test our deployment again:

```
brownie test -s
```

We will see:

```
rownie v1.18.1 - Python development framework for
Ethereum===== test
session starts =====
platform linux -- Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
-- /home/mohamad/.local/pipx/venvs/eth-brownie/bin/python cachedir:
.pytest_cache hypothesis profile 'brownie-verbose' -> verbosity=2,
deadline=None, max_examples=50, stateful_step_count=10,
report_multiple_bugs=False,
database=DirectoryBasedExampleDatabase(PosixPath('/home/mohamad/.brownie/hypo
rootdir: /home/mohamad/Desktop/solidity/Solidity and web3
codes/brownie_simple_storage plugins: eth-brownie-1.18.1, web3-5.27.0,
xdist-1.34.0, forked-1.4.0, hypothesis-6.27.3 collected 2
itemsLaunching 'ganache-cli --chain.vmErrorsOnRPCResponse true --
server.port 8545 --miner.blockGasLimit 12000000 --wallet.totalAccounts
10 --hardfork istanbul --wallet.mnemonic
brownie'...tests/test_simple_storage.py::test_deploy PASSED
tests/test_simple_storage.py::test_updating_storage
PASSED===== 2 passed
in 2.08s =====
Terminating local RPC client...
```

Brownie test uses the functions in the Pytest module and you can use more functions than mentioned here to be able to [have all different kinds of analysis](#) on your smart contract deployment.



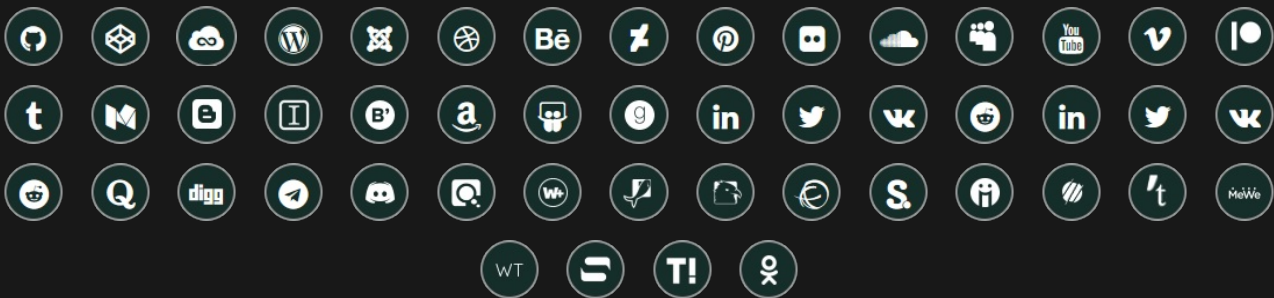
What We Achieved through Testing Deploy.py Using Brownie

In this article, we've completed testing the deployments of the smart contracts (deploy.py script) using Brownie in more detail, so that we can see if there is any problem at all and if there is, we can analyze it in more detail. This kind of detailed test and code analysis becomes handy, especially when the code is too long and complex that we cannot so easily understand where the unexpected result originates from.

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)