# MASTER THE PYTHON INTERVIEW

Explain **unit testing** in Python

part 1

*by Nicolas Georges*
www.master-the-interview.online

Explain **unit testing** in Python

# why **unit testing?**

- **find bug** early
- **repeatable** process
- allow **change** without the fear of breaking something
- allow to assess your **understanding** of the **expected** behaviour
- allow **other developers** to change your code (or you in the future!)
- provide **metrics**
- **document** behaviour of the source code

Explain **unit testing** in Python

# **unittesting** in Python

```
import unittest
class MyTests(unittest.TestCase):
    def test_my_function(self):
        assertion statements…
```

# MASTER THE PYTHON INTERVIEW

Explain **unit testing** in Python

part 2

*by Nicolas Georges*
www.master-the-interview.online

Explain **unit testing** in Python

# TDD: Test-driven development

- write the test **before** writing the implementation
    1. add a test
    2. run all tests and confirm that only the new test fails
    3. implement the minimum to make this new test pass
    4. run all tests
        1. if successful, add a new test
        2. if not, refactor the implementation

Explain **unit testing** in Python

advanced

# **unittesting** in Python

- setUp(), tearDown()
- other test runners (Nose)