SEO Tech Developer

UnitTesting and Development Pipeline

What you will be able to do:

- Create documentation in markdown
- Apply a style guide to their code
- Setup and use an automated style checker
- Describe the 4 levels of testing
- Create a testing plan for given code
- Write a simple unit test

SEO Tech Developer

SEO Tech Developer

README Files

README.md

- It is common practice to include README files with code projects. Github automatically renders README.md files on repository pages.

- What would you include for someone trying to use your project?

SEO Tech Developer

README.md

- It is common practice to include README files with code projects. Github automatically renders README.md files on repository pages.

- Some things to include:
  - Requirements/dependencies
  - How to setup the project
  - Any usage instructions
  - Badges for workflows
  - Licensing
  - Contact information

SEO Tech Developer

Markdown

- Readme files are written in **markdown** which is a markup language

- Markup languages include:
  - HTML
  - MD
  - XML

- Have you used a markup language?

- How is it different from a programming language?

SEO Tech Developer

Markdown

- Readme files are written in **markdown** which is a markup language

- Markup languages include:
  - HTML
  - MD
  - XML

- **Markup languages format text** – bold, underline, insert images, etc.
  - Markup languages are not compiled and run like programming languages

SEO Tech Developer

Writing Markdown

- Headers:
  - # Title / H1
  - ## Subtitle / H2

- Empahsis:
  - **bold**
  - *italic*
  - > quote

- [Text of link](http://google.com)

- ![alt text](path/to/image_file.jpg)

- Bullet list:
  - Can use hyphen
  * Or asterisk

- Numbered list:
  1. This is one
  1. This is two
  1. Markdown handles counting
  1. To ease re-ordering!

- Emojis:
  :sparkles:
  :octocat:

SEO Tech Developer

SEO Tech Developer

Development Pipeline

Development Pipeline

- When code is pushed, typically it doesn't just go live –
  - What do you think might happen?

SEO Tech Developer

Development Pipeline

- When code is pushed, typically it doesn't just go live:
  - Automated Run of Static Analysis tools
    - Style Checker
    - Bug Checker
    - Security Checker
  - Automated Run of Tests
  - Code Review by another developer
  - Manual test by QA

SEO Tech Developer

Style Guide

- Python has a style guide called **PEP 8**
  - https://www.python.org/dev/peps/pep-0008/

- Why would a group of developers want to follow a style guide?

Style Guide

- Python has a style guide called **PEP 8**
  - https://www.python.org/dev/peps/pep-0008/

- Why would a group of developers want to follow a style guide?
  - Makes large code bases easier to read
  - Gives developers consistent experience (ex. maximum line length)
  - Agreed upon rules make code reviews faster/easier

- Compare the two examples below – which is easier to read?

```
# Wrong:
# operators sit far away from their operands
income = (gross_wages +
          taxable_interest +
          (dividends - qualified_dividends) -
          ira_deduction -
          student_loan_interest)
```

```
# Correct:
# easy to match operators with operands
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest)
```

SEO Tech Developer

Style Checker

- There is a static analysis tool to check that the style guide is followed
  - **pycodestyle -** https://pypi.org/project/pycodestyle/

- Running on terminal:
  - `pycodestyle --first file_name.py`

- Example output:
  - file_name.py: line#: character#: Style issue description
  - `calc.py:   11:  1:  W293 blank line contains whitespace`
  - `calc.py:   27:  5:  W292 no newline at end of file`

SEO Tech Developer

Automating Style Checker

- GitHub Actions
  - Automates tasks
  - Specified in YAML – Yet Another Markup Language
    - White space matters like python!
- Create `.github/workflows/style.yaml`
  - Name workflow
  - Install python and pycodestyle
  - Run pycodestyle

```yaml
name: Check Style
on: push

jobs:
  check-style:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - name: Setup python
        uses: actions/setup-python@v2
        with:
          python-version: 3.11.3

      - name: Install tools
        run: python -m pip install
             --upgrade pip pycodestyle

      - name: Check Style
        run: pycodestyle --first *.py
```

SEO Tech Developer

SEO Tech Developer

Testing

Put in chat any definitions you know already

- Acceptance Testing
- System Testing
- Integration Testing
- Unit Testing
- QA

SEO Tech Developer

Levels of Testing

- Acceptance Testing - testing done generally by client to ensure it meets their needs
- System Testing - testing the entire system end-to-end
- Integration Testing - test integrated components such as classes
- Unit Testing - testing individual components such as methods

When and Who Writes Tests?

- This depends a lot on the organization, but generally you (the developer) should write tests for any new code **before** committing it.

- There are some organizations who have dedicated test engineers who entire job is to create tests. This role can sometimes be called a QA or Quality Assurance engineer (though generally QAs have less experience than test engineers).

Testing Plans

- When designing the structure of code, you should consider both modularizing your code in a way that makes sense and is reusable, but also that each module has a discrete, testable function.

- As you are breaking your code into methods, think about (write in comments) a few test cases for each method.

- For example, if your method includes a conditional, your test cases should “cover” both branches

SEO Tech Developer

SEO Tech Developer

UnitTesting

UnitTest – Python Library

- Actual check or test is done with **assert** methods:
  - assertEqual(a, b)
  - assertNotEqual(a, b)
  - assertTrue(a)
  - assertFalse(a)
  - assertRaises(a)
  - assertAlmostEqual(a, b)
  - assertNotAlmostEqual(a, b)

UnitTest – Example

```
import unittest
from yourCodeFileName import function1, function2

class TestFileName(unittest.TestCase):

  def test_function1(self):
    self.assertEqual(function1(1), 0)

  def test_function2(self):
    self.assertEqual(function2(2,1), 3)
    self.assertEqual(function2(2.1, 1.2), 3.3)
```

SEO Tech Developer

python3 -m unittest testFileName.py

Automating UnitTesting

- GitHub Actions
  - Automates tasks
  - Specified in YAML – Yet Another Markup Language
    - White space matters like python!
- Create `.github/workflows/tests.yaml`
  - Name workflow
  - Install python and unittest
  - Run unittest

```yaml
name: Tests
on: push

jobs:
  check-style:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3

      - name: Setup python
        uses: actions/setup-python@v2
        with:
          python-version: 3.11.3

      - name: Install tools
        run: python -m pip install
             --upgrade pip pytest

      - name: Test with unittest
        run: python3 -m unittest test.py
```

SEO Tech Developer

Unit Testing Vocabulary

- A fake can refer to either a mock or a stub - any piece of code that is pretending to be fully implemented, production code.
  - A mock is a fake object that mimics an actual object
  - A stub can replace an object that isn't built yet
  - A stub will **never** fail a unit test, but a **mock** can.
    A stub could be replaced when the functionality is added.

- A spy is an observation point to check if code calls a component

- A dummy is an object that is passed around but never used

SEO Tech Developer

What questions do you have about…

- Create documentation in markdown
- Apply a style guide to their code
- Setup and use an automated style checker
- Describe the 4 levels of testing
- Create a testing plan for given code
- Write a simple unit test

SEO Tech Developer

SEO Tech Developer

Thank you!