Testing

Agenda

- End-to-End vs Unit Testing
- Unit testing
  - Python unittest example
  - Mocking
- End-to-End Testing:Selenium

End-to-End vs Unit Testing

E2E Testing	Unit Testing
Test the complete integrated system	Test the smallest testable part of your code e.g: function,class
Written by testers	Written by developers
Takes the entire system into consideration	Easier to detect and fix errors

Automated Tests

- Test execution is handled by a tool rather than a human
- Advantages:
  - Faster test execution
  - Higher testing coverage

Unit Testing

Example:
Python unittest

Automated Unit Tests

- Script to check unit behaves as expected
- Pass different inputs to the unit and assert that the output is similar to the expectation

Example

Function behaves as follows:

- Return True if the two strings start with the same letter
- Return False if the two strings start with different letters
- Raise a TypeError if one of the strings is None

Example

```
class TestStartWithSameLetterFn(TestCase):
    def test_different_first_letter(self):
        self.assertFalse(student.start_with_same_letter("hello", "world"))

    def test_same_first_letter(self):
        self.assertTrue(student.start_with_same_letter("hello", "hi"))

    def test_none_parameter(self):
        with self.assertRaises(TypeError):
            student.start_with_same_letter("hello", None)
```

Example

- Unit test should not depend on each other
- Set up test variables before executing each test function

```
class TestStudentClass(TestCase):
    def setUp(self):
        self.test_object = student.Student('Ahmed', ["Software Engineering", "Compilers"])

    def test_add_course(self):
        self.test_object.add_course("Arch")
        self.assertIn('Arch', self.test_object.get_courses())

    def test_get_credit_hours(self):
        self.assertEqual(self.test_object.get_credit_hours(), 4)
```

Mocking

Mocking

- Unit depends on a service provided by module that is not yet implemented
- Test the unit by creating an object that mocks the interface and behaviour of the missing service
- Mock object returns fake(hard coded) data that have the same format as the original

Example

Refer to the example in the “Mocking” folder

End-to-End Testing: Selenium

Selenium

- Web testing automation framework
- Provides playback tool(Selenium IDE) and browser automation driver(Selenium WebDriver)
- Supports many languages and many browsers

Selenium:Example

- Webdriver contains classes that control different browsers
- get : directs the browser to url
- We can interact with any element in the page by searching for the element using its name,tag,id and other attributes
- Element interaction includes entering text,clicking on the element,..

```
driver=webdriver.Chrome()
driver.get("http://google.com")

search_bar=driver.find_element_by_name("q")
search_bar.send_keys('"info"')
search_bar.send_keys(Keys.ENTER)
```

Selenium:Example

- If an action will cause a page or dynamic content to load we can wait a few seconds using WebDriverWait to ensure that the content is loaded

```
search_bar.send_keys(Keys.ENTER)

webdriver.support.wait.WebDriverWait(driver,1) #wait for request and r

results=driver.find_element_by_id("center_col")
assert results.text.find("info") #check that the results contian the s
```

Thank You