



# Definition

**UNIT TESTING** is a level of software testing where individual units/ components of a software are tested

A unit is the smallest testable part of any software

A unit may be an individual program, function, procedure, etc

# Agenda

Traditional testing strategies

Why unit test?

Assertions

Chai and Mocha

Frameworks for Python and Java

Guidelines

# Traditional testing strategies

Print statements

Use of debugger

Debugger statements

Test scripts

- Say “NO” to these traditional strategies now! Follow unit testing for development purpose.

# Why Unit test?

Faster debugging and defect identification

Future cost saving

Better development

Helpful in regression

Detection of problems early in the development cycle

Unit testing increases confidence in changing/maintaining code

Development is faster

Code is more reliable

# Assertions

Assertions are basis for unit tests

Assertions are boolean functions that are used to test behavior

For NodeJs [Chai](#) is one of the assertion library

Chai and Mocha is used for unit testing in Node Js applications

# Chai and Mocha

Mocha does not have a built in assertion library. Chai is used with mocha as an assertion library

Chai comes with three different assertion flavors. It has the should style, the expect style, and the assert style.

Commands to install:

```
npm install chai  
npm install mocha
```

Command to run the test case:

```
mocha testFilename
```

You can follow the [video](#) and get started with chai and mocha and create you first unit test case.

Also you can refer to my blog [here](#)

# Chai and Mocha: Code coverage

To check the code coverage:

Library that can be used with mocha: **Istanbul**

[Istanbul](#) is a code coverage tool that computes statement, line, function and branch coverage

Command to [install](#) : `npm i nyc`

To run: `npm coverage`

```
{
  "scripts": {
    "test": "mocha",
    "coverage": "nyc npm run test"
  }
}
```



# Other Node Js libraries for unit testing

For Node Js framework more libraries are available except Chai and Mocha which are Jasmine (whose assertions methods are similar to as Chai), Nock(for external http requests), Sinon, Mock-require etc.

You can refer the different types of libraries [here](#) and understand why and where they are used!

# Python and Java

For Python Unittest module has a similar flavor as major unit testing frameworks in other languages.

Other Python testing frameworks are Robot, PyTest, DocTest, Nose2 and Testify

Watch the [video](#) and get started with unittest module!

For java TestNG is easy framework to write customized tests. Other frameworks are Junit, Mockito, EasyMock, AssertJ, Hamcrest ect. Check [here](#) to learn more about them

Watch the [video](#) and get started with TestNG!

# Guidelines

Write small and fast tests

Isolated and non interactive

Fix failing tests immediately

Keep testing at unit level

Keep tests independent

# Guidelines

Use naming conventions

Think white box

Cover boundary cases

Use random generators

Use explicit asserts

# Assignment

Create unit test cases for Node Js, Python and Java assignments given to you by your trainers!

Thank You