## What is Unit Testing?

## With unit testing, you can run independent tests on each function. You have to provide an input that’s supposed to produce a known output. And if the output that comes after testing matches the known output, then your testing is successful.

## When you test your codebase, you take a piece of code — typically a function — and verify it behaves correctly in a specific situation. Unit testing is a structured and automated way of doing this. As a result, the more tests you write, the bigger the benefit you receive. You will also have a greater level of confidence in your codebase as you continue to develop it.

## The core idea with unit testing is to test a function’s behavior when giving it a certain set of inputs. You call a function with certain parameters, and check you got the correct result.

## Unit testing is the first level of software testing performed before any other type of testing. Usually, writing unit tests is the responsibility of developers.

## Benefits of Testing

* the main benefit of unit testing is that it makes your project more agile.
* Whenever you add a new feature, you may have to change the old code. Changing a code that’s been tested already is costly and risky. However, if you run unit tests on the new code, you’ll be confident that the new code won’t break any old feature.
* unit testing helps you find bugs early in the development cycle. If you’re a developer, finding and fixing the bugs by yourself will improve the code quality and make the testing phase easier.

Popular Unit Testing Frameworks for JavaScript

1. [Unit.js](http://unitjs.com/): It is known as an open source assertion library running on browser and Node.js. It is extremely compatible with other JavaScript Unit Testing framework like Mocha, Karma, Jasmine, QUnit, Protractor, etc. Provides the full documented API of assertion list

2. [QUnit](https://qunitjs.com/): It is used for both client-side as well as server-side JavaScript Unit Testing. This Free JavaScript testing framework is used for jQuery projects. It follows Common JS unit testing Specification for unit testing in JavaScript. It supports the Node Long-term Support Schedule.

3. [Jasmine](https://jasmine.github.io/): Jasmine is the behavior-driven development framework to unit test JavaScript. It is used for testing both synchronous and asynchronous JavaScript Code. It does not require DOM and comes with the easy syntax that can be Written for any test.

4. [Karma](https://karma-runner.github.io/2.0/index.html): Karma is an open source productive testing environment. Easy workflow control Running on the command line. Offers the freedom to write the tests with Jasmine, Mocha, and QUnit. You can run the test on real devices with easy debugging.

5. [Mocha](https://mochajs.org/): Mocha runs on Node.js and in the browser. Mocha performs asynchronous Testing in a simpler way. Provides accuracy and flexibility in reporting. Provides tremendous support of rich features such as test-specific timeouts, JavaScript APIs etc.

6. [Jest](https://jestjs.io/): Jest is used by Facebook so far to test all of the JavaScript code. It provides the 'zero-configuration' testing experience. Supports independent and non-interrupting running test without any conflict. Do not require any other setup configuration and libraries.

7. [AVA](https://github.com/avajs): AVA is simple JavaScript Unit Testing Framework. Tests are being run in parallel and serially. Parallel tests run without interrupting each other. AVA Supports asynchronous testing as well. AVA uses subprocesses to run the unit test JavaScript.