Recently, a student of mine asked me: "Mosh, why on earth should we write code to test our code, when we can just run the application and see the result?".

Well, in the last lecture, you learned that with automated tests

you can test our application code on a frequent basis and in less time. But this is not the only benefit of automated testing. The most important benefit is that you can catch the bugs BEFORE deploying our application. And this is extremely important because it allows you to deploy your application with more confidence.

Have you been in a situation where you finished all the work, deployed your application, left the office thinking everything is working, and then got a call from your boss or an end-user, telling you that one of the major functions of the application is not working? Then you had to go back to the office and you thought that was a quick fix, but you ended up staying there till midnight!

That's why you should write tests: to reduce the number of defects or bugs that will go in the production… Now, note that I'm not saying that with automated tests you're going to release bug free software. That's not true! But you can certainly reduce the number of bugs and improve the quality of your software.

Another benefit of automated tests is that they allow you to refactor your code with confidence. Refactoring means changing the structure of your code, without changing its behavior. If you extract a few lines of a method into a separate private method, that's refactoring. If you rename a method, that's refactoring too. You're changing the structure of your code to make it cleaner and more maintainable, but you're not changing the functionality.

When you don't have automated tests, every time you refactor your code you have to manually test every part of the application that could be affected by your refactoring. And this is very painful because first of all, it's time consuming, and second, as your application grows, you may forget about the parts that need to be tested!

With automated tests, every time you refactor your code, you run your tests and make sure you didn't break anything that used to previously work. And finally, another benefit of writing tests is that it helps you focus more on the quality of the methods that you're writing. You make sure that every method works with different inputs under varying circumstances.

Next, I'm gonna talk about different types of tests.