@DisplayName Annotation on Test Methods

This lesson demonstrates how to provide custom display names to test methods.

@DisplayName on @Test Methods

JUnit 5 `@DisplayName` annotation is also used to provide a custom name for the `@Test` methods. Usually, by default, JUnit 5 reporting prints the test methods name in the IDE test report and also while executing test cases. We can use `@DisplayName` annotation to provide a custom name to `Test` methods, that makes it easier to read. Thus, it acts as a metadata for our test methods.

`@DisplayName` annotation can take up string having the following things:-

1. It can take spaces between words.
2. It can take special characters.
3. It can take emojis as well.

Let's look at a demo:-

```
package io.educative.junit5;

import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;

@DisplayName("Test class to demonstrate @Display name")
public class DisplayNameDemo {

    @Test
    public void testWithoutDisplayName() {
            System.out.println("test method got executed!!!");
    }

    @Test
    @DisplayName("Custom test name containing spaces")
    public void testWithDisplayNameContainingSpaces() {
```

```java
            System.out.println("test method got executed!!!");
    }


    @Test
    @DisplayName("Custom test name containing special characters - !@#$%^&*()_+~`⌐°□°) ⌐"
    public void testWithDisplayNameContainingSpecialCharacters() {
            System.out.println("test method got executed!!!");
    }

    @Test
    @DisplayName("Custom test name containing emojis - 🙀")
    public void testWithDisplayNameContainingEmoji() {
            System.out.println("test method got executed!!!");
    }
}
```
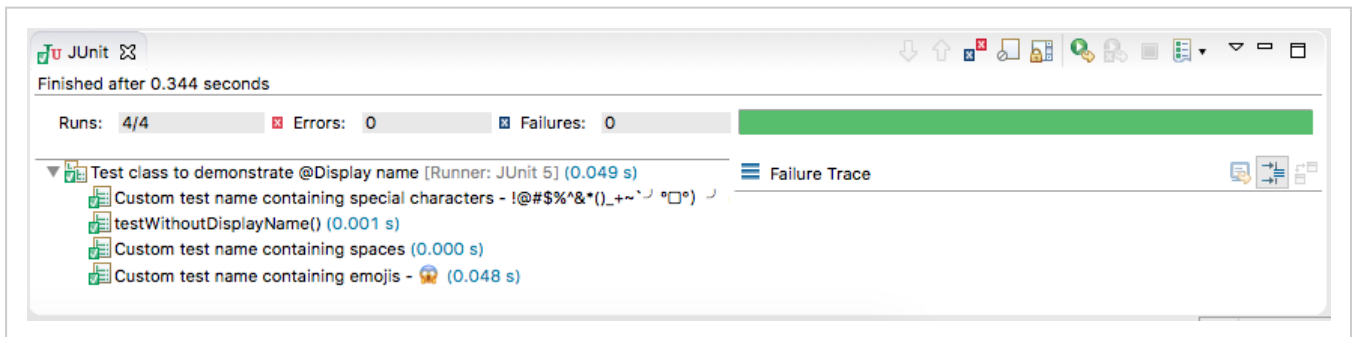

@DisplayName demonstration

Instead of printing actual test class and test names in the JUnit tab and reports, it prints string provided in `@DisplayName` annotation.

It actually provides a textual representation of what actually test class and `@Test` method do.

@DisplayName annotation

In the next chapter, we will discuss about Dependency Injection in JUnit 5.