

# @DisplayName Annotation on Test Methods

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

## We'll cover the following



- @DisplayName on @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.

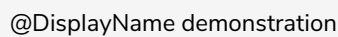
`@DisplayName` annotation can take up string having the following things:-

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:-

```
1 package io.educative.junit5;
2
3 import org.junit.jupiter.api.DisplayName;
4 import org.junit.jupiter.api.Test;
5
6 @DisplayName("Test class to demonstrate @Test methods")
7 public class DisplayNameDemo {
8
9     @Test
10     public void testWithoutDisplayName() {
11         System.out.println("testWithoutDisplayName");
12     }
13
14     @Test
15     @DisplayName("Custom test method with display name")
16     public void testWithDisplayName() {
17         System.out.println("testWithDisplayName");
18     }
19 }
```






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



@DisplayName annotation in JUnit 5

DEMO



@DisplayName

- With spaces
- With special characters
- With emojis



@DisplayName annotation

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