

@DisplayName Annotation on Test Class

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

WE'LL COVER THE FOLLOWING ^

- @DisplayName on Test Class

@DisplayName on Test Class

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

`@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:-

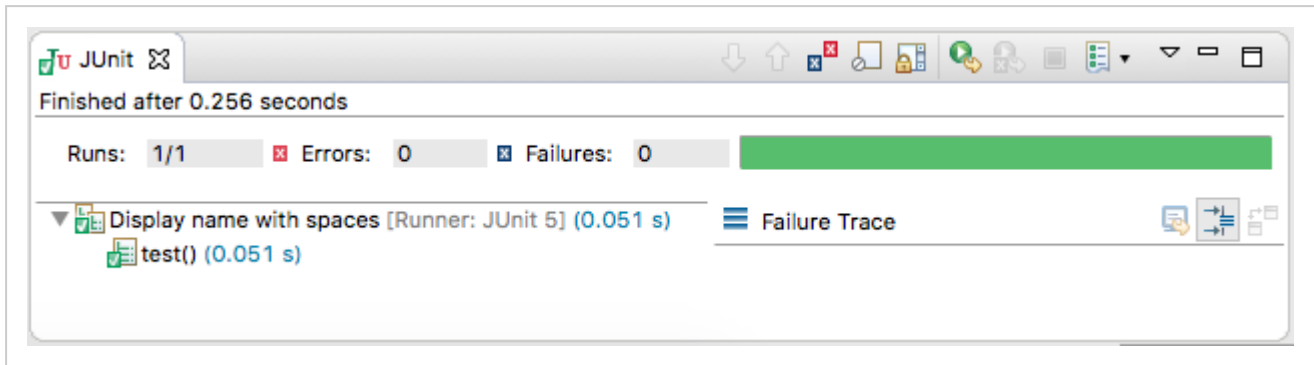
```
package io.educative.junit5;

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

@DisplayName("Display name with spaces")
public class DisplayNameWithSpaces {

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





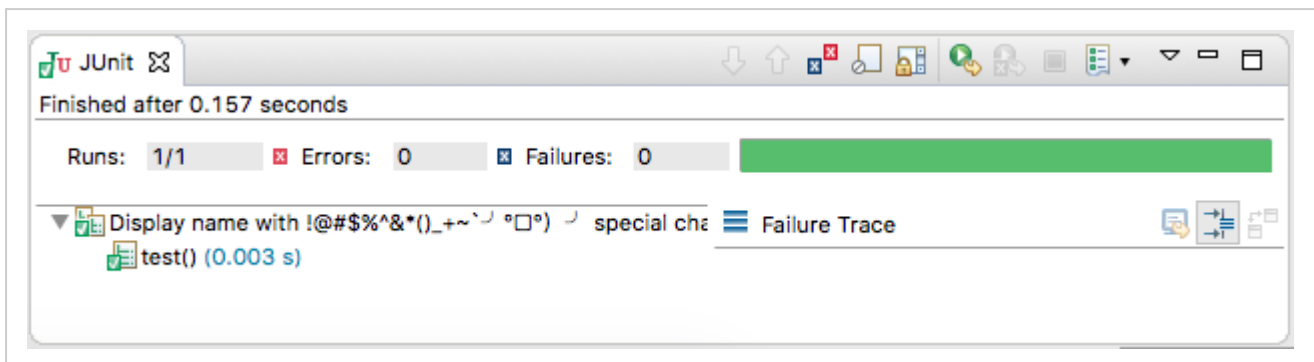
@DisplayName with spaces

```
package io.educative.junit5;

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

@DisplayName("Display name with !@#$%^&*()_+~`'°□°) ♪ special character")
public class DisplayNameWithSpecialCharacters {

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



@DisplayName with special characters

```
package io.educative.junit5;

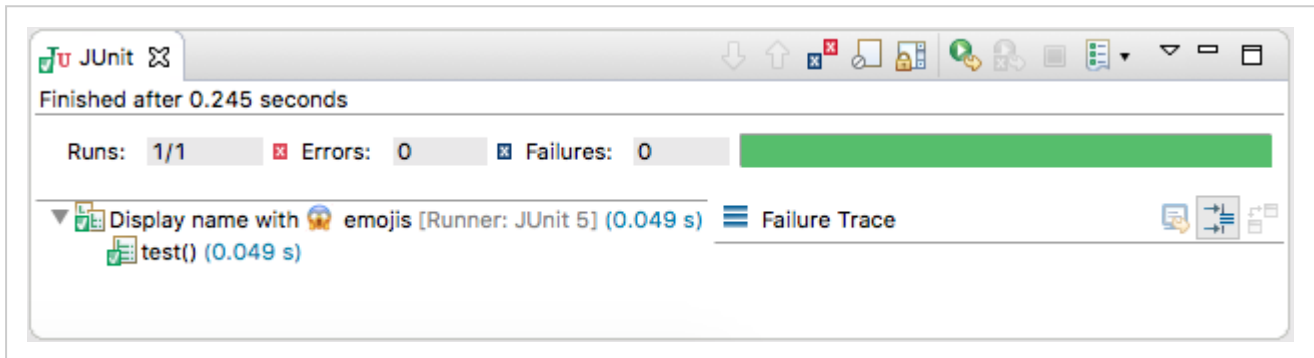
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;

@DisplayName("Display name with 🙋 emojis")
public class DisplayNameWithEmojis {

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



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



@DisplayName with Emojis

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

It actually provides a textual representation of what actually test class does.

In the next lesson, we will discuss about `@DisplayName` on test methods.