

Most Useful Function

A Little Bit of Unit Testing

February 13, 2025

```
assertEquals(expected, actual)
```

how it works

if expected and actual are the same
test passes

else
test fails

- ▶ function defined for several different types
- ▶ expected, actual must be same type
- ▶ typical actual: call function to test

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

Example

```
assertEquals(expected, actual)
```

```
void testSum() {  
    assertEquals(10, LameMath.sum(2,8));  
    assertEquals(-10, LameMath.sum(-2,-8));  
    assertEquals(-6, LameMath.sum(2,-8));  
    assertEquals(0, LameMath.sum(0,0));  
}
```

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

For True/False Values

```
assertTrue(some boolean condition)  
assertFalse(some boolean condition)
```

Example

```
void testIsEven() {  
    assertTrue(LameMath.isEven(2));  
    assertTrue(LameMath.isEven(42));  
    assertTrue(LameMath.isEven(-16));  
    assertFalse(LameMath.isEven(-1));  
    assertFalse(LameMath.isEven(11));  
}
```

◀ ▶ ⏪ ⏩ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ ⏻ ⏼ ⏽ ⏾ ⏿ 🔍 ↻

JUnit in Eclipse

To Add JUnit to Project

- ▶ Right click on a class (CTRL-click on a Mac)
- ▶ File → New → JUnit Test Case
- ▶ Be sure “New JUnit Jupiter test” is selected
- ▶ Click “Next”
- ▶ Check off functions you want tested, then click “Finish”

To Run the Tests

- ▶ Right click on the test class (CTRL-click on a Mac)
- ▶ Run → Run as → JUnit Test

