

They support behavior-driven development, emphasizing human-readable specifications of behavior of code with accompanying test cases.

JVM's -ea and -da flags.

- `assert (Boolean)`

- `assert (Boolean, String)` (explanation to be included in `AssertionError`.)

It typically mirrors the source tree's directory hierarchy.

ScalaTest's

```
assert(this === that)
expect(this) {
  that
}
```

```
intercept[IllegalArgumentException] {  
    //something that throws it  
}
```

```
extend org.scalatest.FunSuite
```

```
test("my cool test") {  
  //use assert() or exceptions  
}
```

Specs provides matches to be used in ScalaTest or JUnit.

ScalaTest's Spec uses

```
describe("some object") {  
  it("should do A") {  
    //test for A  
  }  
}
```


Property-based testing.

You give properties with implication operator \implies , and it checks the properties on the values it generates for you.

Use `ensuring(op: Any => Boolean)`

Raises `AssertionError` if predicate fails, and returns the argument correctly cast to result type of function otherwise.

```
ensuring(_.dollars >= 0)
```

Result of function will be inserted and returned if the predicate holds.