

# JBehave

## Create and Run a Scenario

- Three files needed:
  - Scenario text file with given-when-then format
  - Java/file class to extend Scenario file
  - Java steps file for the method shells

### Sample scenario text file (make\_pancakes\_scenario)

Scenario: Making pancakes in a skillet  
Given some pancake batter  
And a hot frying pan  
When I pour the batter into the frying pan  
And wait for 2 minutes or until the sides are dry  
And flip the pancake over  
And wait for 1 minute  
Then I get a cooked pancake

### Sample Java file/class (MakePancakesScenario.java)

```
package com.cse.simple.stories;  
  
import org.jbehave.scenario.Scenario;  
  
public class MakePancakesScenario extends Scenario {  
    public MakePancakesScenario() {  
        super (new PancakeCookingSteps() );  
    }  
}
```

## Sample Java steps file(1) (PancakeCookingSteps.java)

```
package com.cse.simple.steps;

import org.jbehave.core.annotations.Given;
import org.jbehave.core.annotations.Then;
import org.jbehave.core.annotations.When;

public class PancakeCookingSteps {
    @Given("some pancake batter")
    public void howMuchBatter() {
        // add code for amount of batter
    }

    @Given("hot frying pan")
    public void heatFryingPan() {
        // add code to determine heat
    }

    @When("I pour the batter into the frying pan")
    public void startCooking() {
        // add code to start cooking
    }

    // continued on next slide
}
```

## Sample Java steps file(2) (PancakeCookingSteps.java)

```
// continued from previous slide

@When("wait for 2 minutes or until the sides are dry")
public void cookFirstSide() {
    // add code to cook first side
}

@When("flip the pancake")
public void flipPancake() {
    // add code to flip pancake
}

@When("wait for 1 minute")
public void cookSecondSide() {
    // add code to cook second side
}

@Then("I get a cooked pancake")
public void putPancakeOnPlate() {
    // add code to move pancake from frying pan to plate
}
}
```