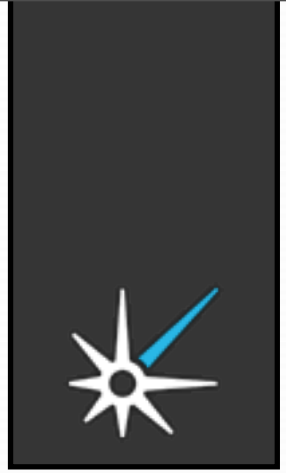
Acceptance Testing

How CSlim and FitNesse Can Help You Test Your Embedded System

Doug Bradbury
Software Craftsman, 8th Light

Tutorial Environment

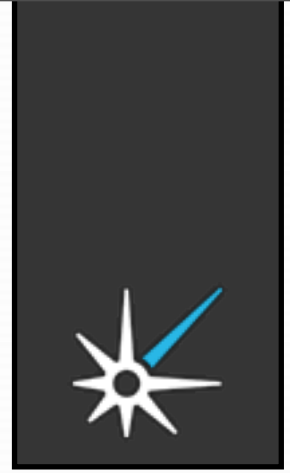


```
git clone git://github.com/dougbradbury/c_learning.git
cd c_learning
./bootstrap.sh
```

or with a live CD:

```
cp -R cslim_agile_package c_clearning
cd c_learning
git pull
./bootstrap.sh
```

Overview



Talk w/ exercises: Acceptance Tests

Tutorial: Writing Acceptance tests

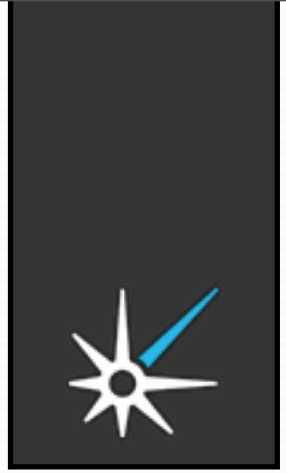
Tutorial: Fitnesse

Tutorial: CSlim

Talk: Embedded Systems Integration

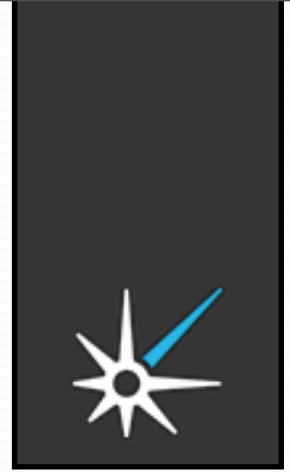
Bonus Topics

Introductions



Who are you?
Where do you work?
What experience do you have with ...
- embedded systems?
- acceptance testing?
- FitNesse and Slim?

Objectives



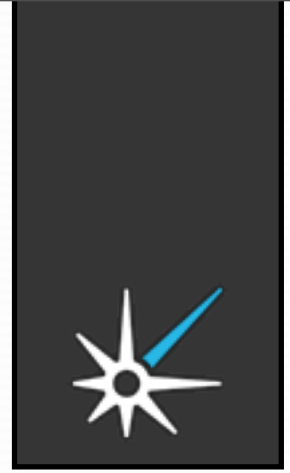
As a result of this course you will be able to:

Understand the purposes of acceptance testing;

Use acceptance tests to define and negotiate scope on embedded systems projects;

Integrate a CSlim Server into your embedded systems;

Objectives (cont)



As a result of this course you will be able to:

Add CSlim fixtures to your embedded system;

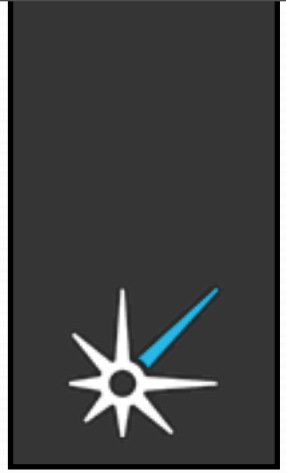
Write Fitnesse tests to drive the execution of CSlim fixtures;

Write and maintain suites of tests in a responsible manner.

Points on a star

How many points does this star have?

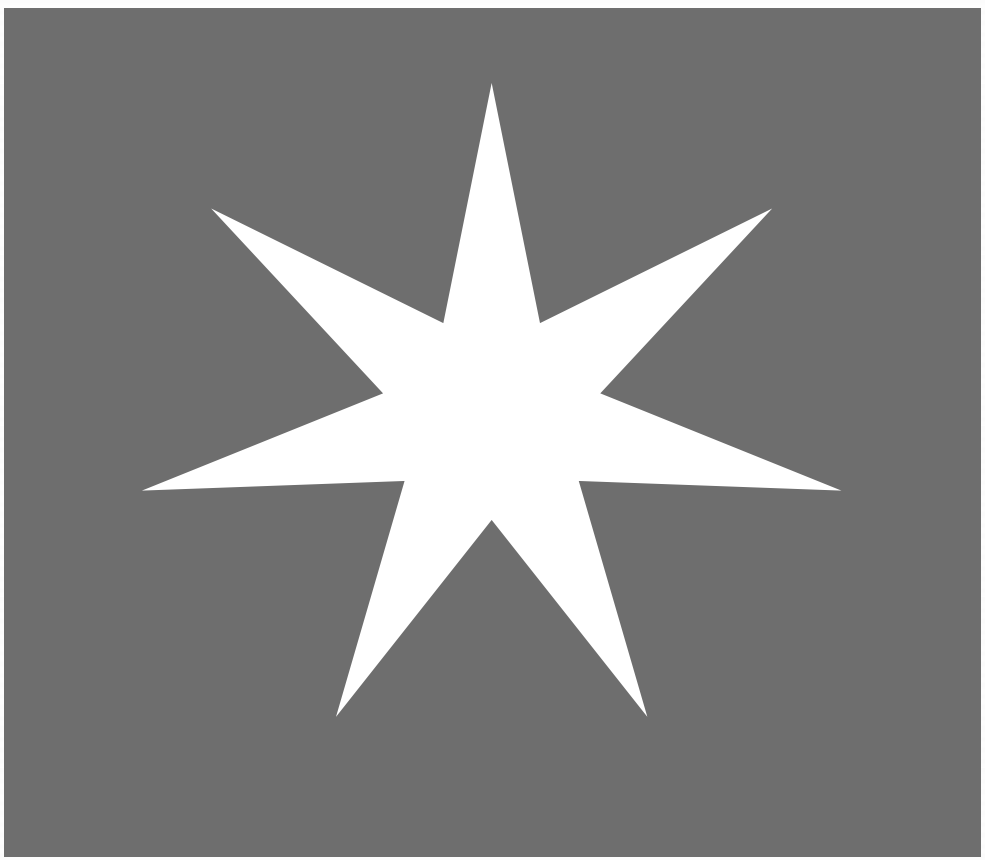
Star Point Specification



Points on a star are counted by the number of **exterior** points.

Points on a star

How many points does this star have?



By Example

3 5 9

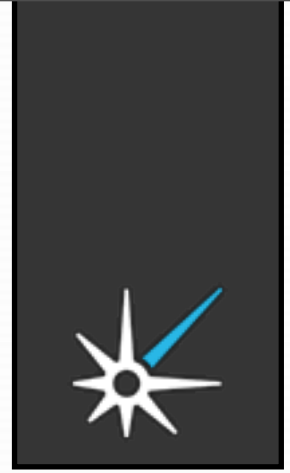
Points on a star

Now, how many points does this star have?

Robo-draw

Pick a partner ...

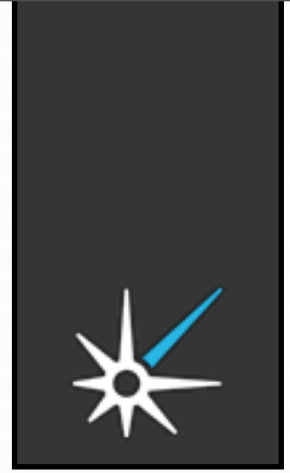
Acceptance Testing



Collaboratively producing examples of what a piece of software is supposed to do

Unit Tests

help you build the **code right**.



Acceptance Tests

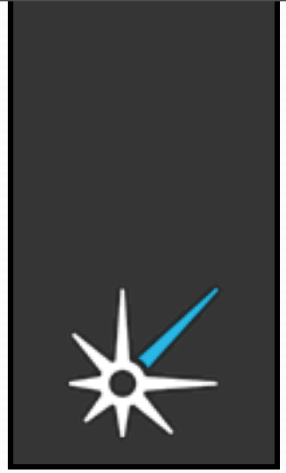
help you build the **right code**.

Acceptance Tests

A collaborative derivation of scope

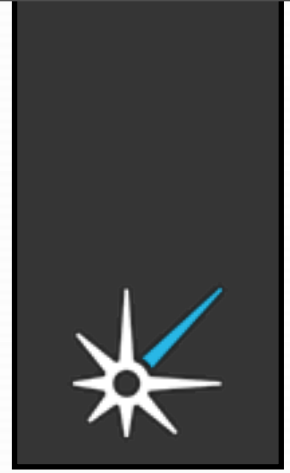
Acceptance Test

Living Documentation



Acceptance Test

A medium for communication

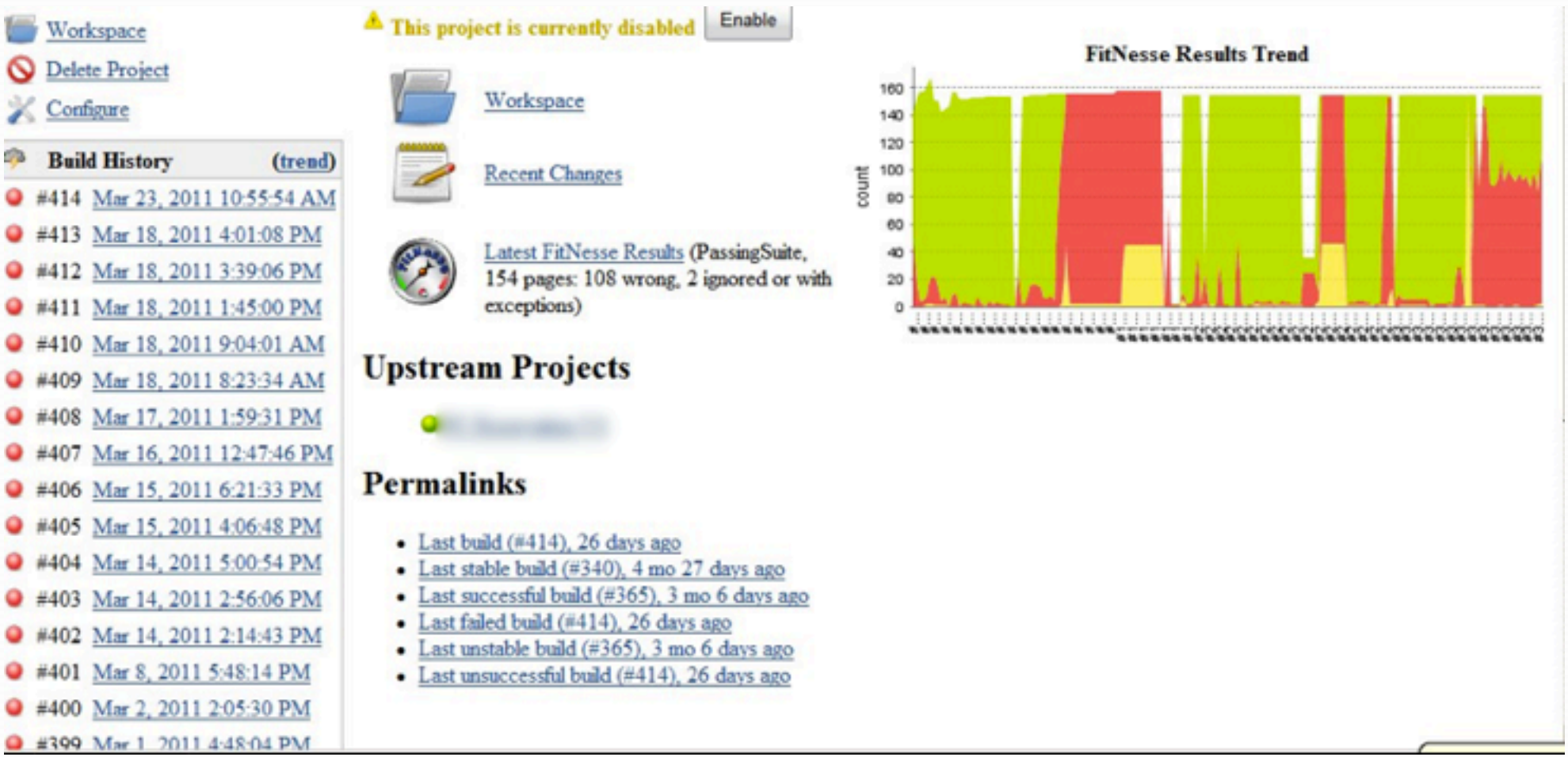


Acceptance Tests

Acceptance Tests

Fail

Don’t test the entire system

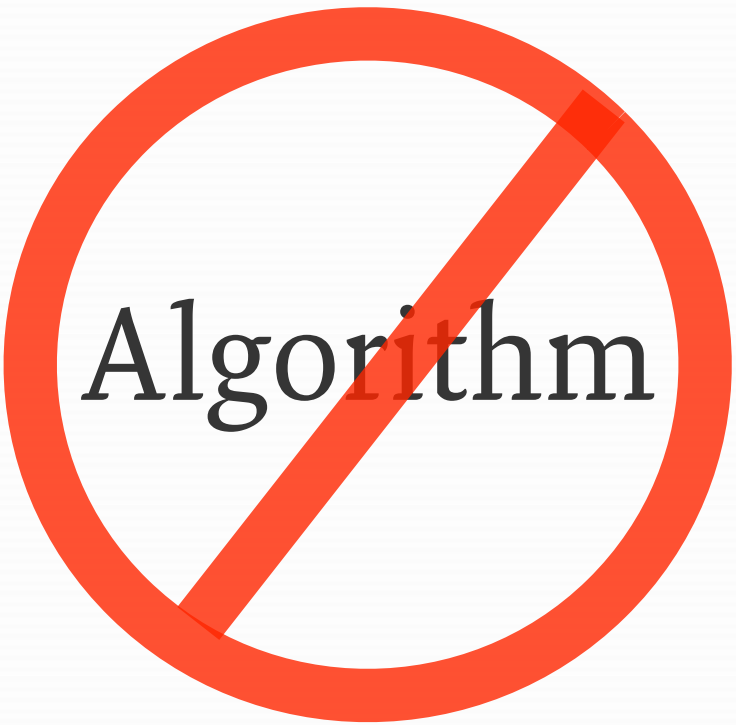


Acceptance Tests

Acceptance Tests

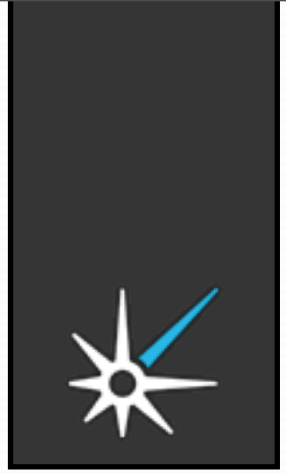
Acceptance Tests

Acceptance Tests



Acceptance Tests

Acceptance Tests



Collaboratively produced examples

A definition of scope

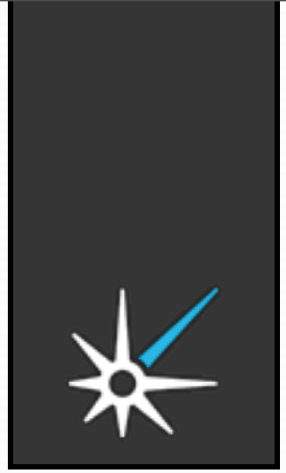
Living documentation

A medium for communication

Why you really need ATs

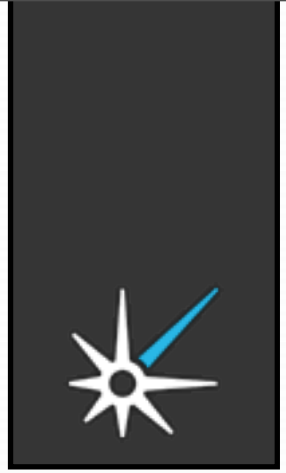
Твой софт - говно!

Workflow Tests



Given, When, Then
Preconditions, Action, Results

Workflow example



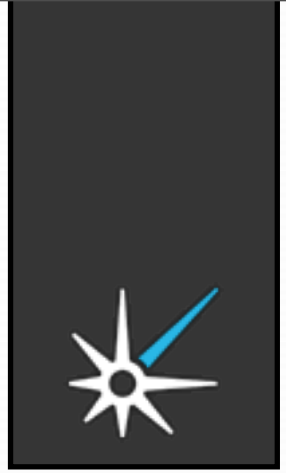
Given a new car traveling at 30 mph
When I fully apply the breaks
Then the car should stop within 35 yards

An example, by example

A treadmill controller

walkncode.com

Treadmill Requirements



It controls the speed of the treadmill.

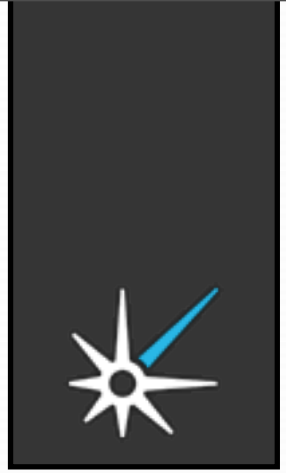
It increments and decrements the speed.

The max speed is configurable.

It keeps track of total distance walked.

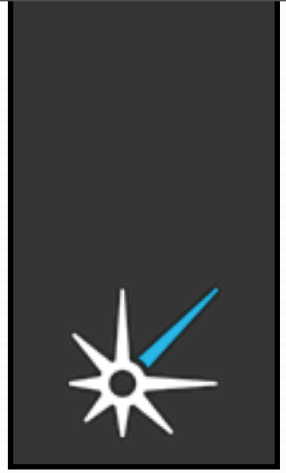
It keeps track of calories burned.

Ambiguities



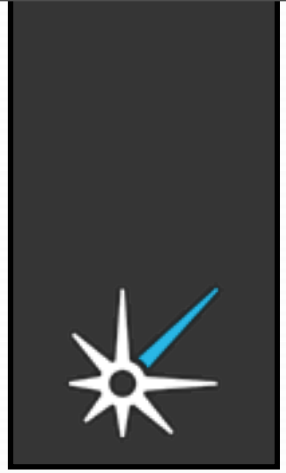
mph, kph, fps?
inc dec amounts?
range for max speed?
distance in miles, km?
formula for tracking calories?

Treadmill by example



Given treadmill speed is 1.0 mph
When I increment the speed
Then the speed should be 1.1 mph

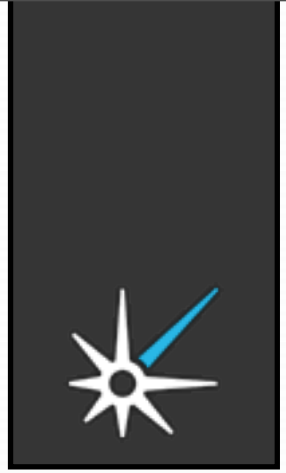
Your Turn



Write an example for the **decrement speed** scenario.

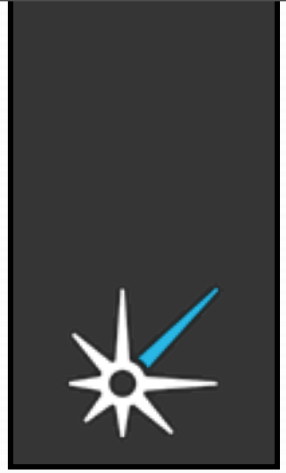
Write an example for the **max speed** scenario

Decrement Speed



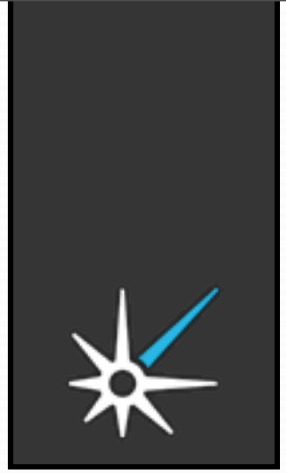
Given treadmill speed is 1.0 mph
When I decrement the speed
Then the speed should be 0.9 mph

Maximum Speed



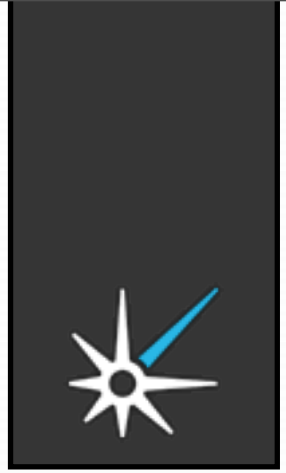
Given treadmill speed is 3.0 mph
Given max speed is 3.0 mph
When I increment the speed
Then the speed should be 3.0 mph

Calculation Tests



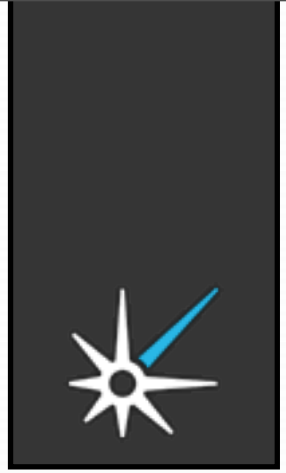
A series of inputs and outputs
Usually captured in table form

Division Test



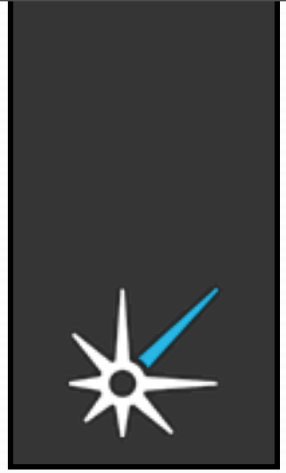
numerator	denominator	quotient?
10	5	2
39	3	13
5	2	2.5

Calories Burned Test



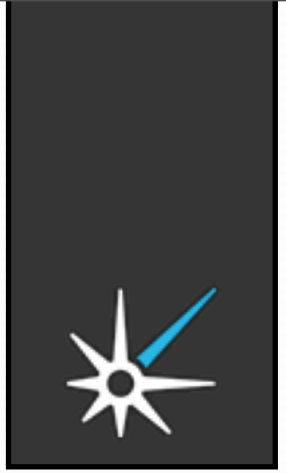
speed	time	calories?
1.0	60	70
2.0	60	150
3.0	60	220

Your Turn



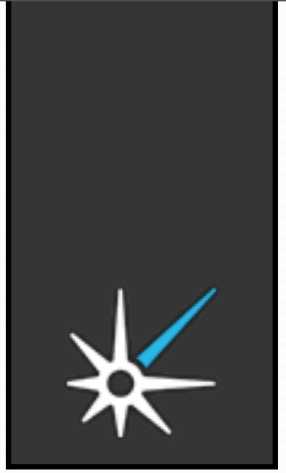
Write a calculation example for distance traveled.

Distance Travelled



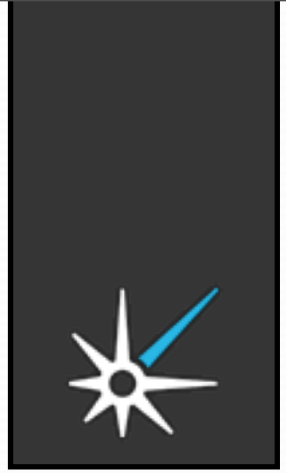
speed	time	distance?
1.0	30	0.5
2.0	30	1.5
1.5	60	3.0

Cumulative Distance

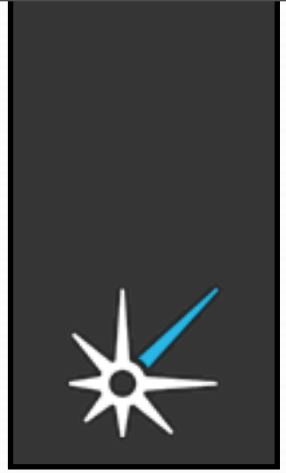


speed	time	distance?
1.0	30	0.5
2.0	30	1.0
1.5	60	1.5

When Do you stop?



Break

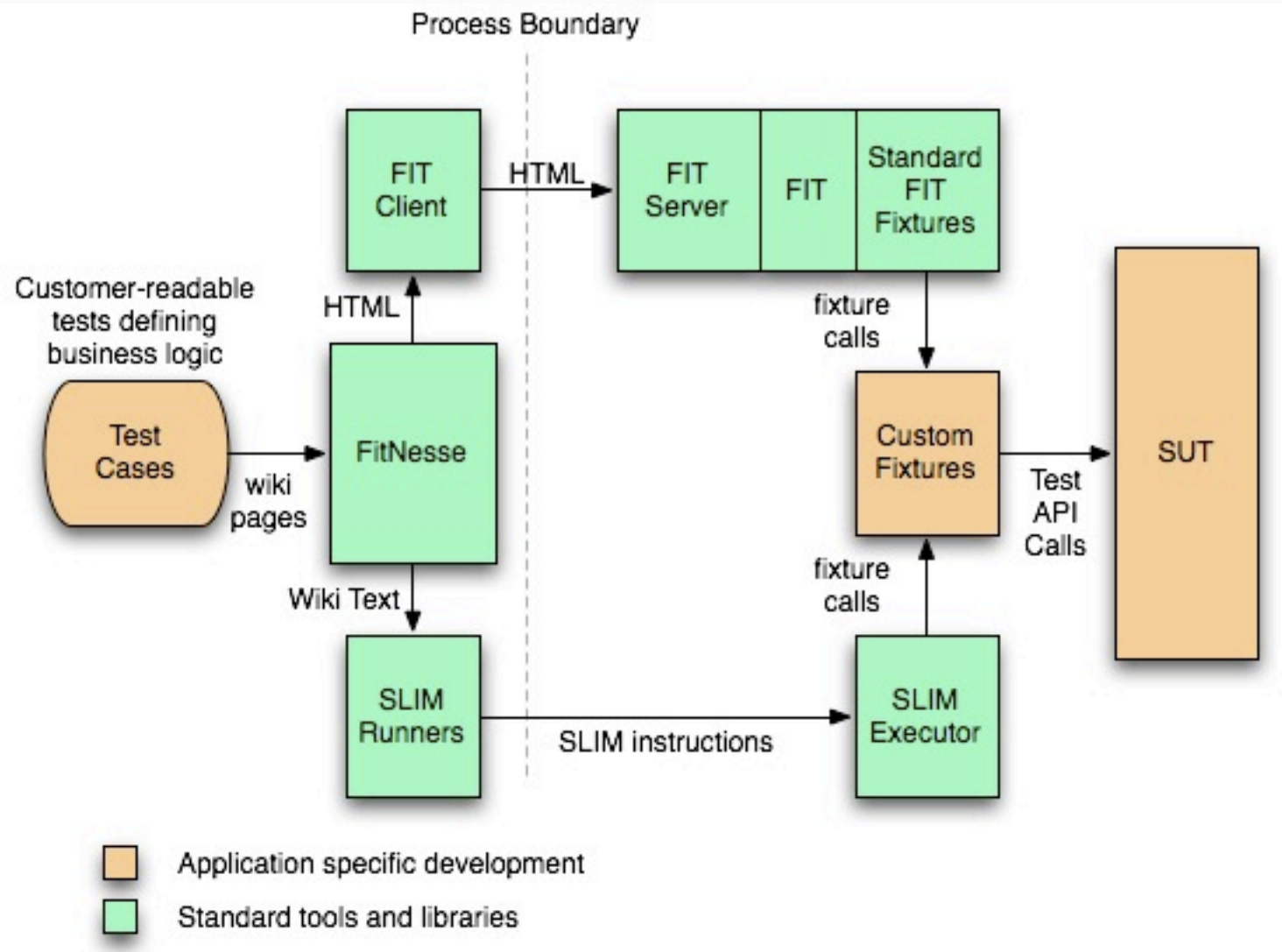


```
git clone git://github.com/dougbradbury/c_learning.git
cd c_learning
./bootstrap.sh
```

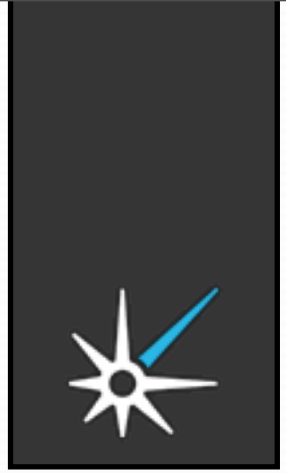
or with a live CD:

```
cp -R cslim_agile_package c_clearning
cd c_learning
git pull
./bootstrap.sh
```

FitNesse Ecosystem



Starting FitNesse

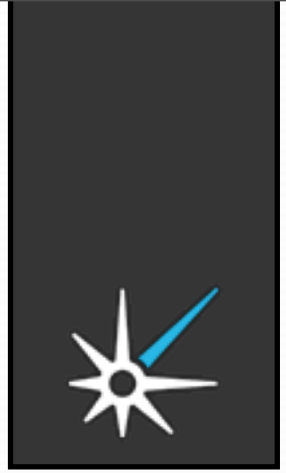


java -jar fitnesse.jar -p 8080 -e 0

Options

- Port: -p 8080
- No History: -e 0

FitNesse Tutorial



Follow along!

It's a wiki


FitNesse
Edit
Properties
Where Used
FrontPage [add child]
WELCOME TO FITNESSE!

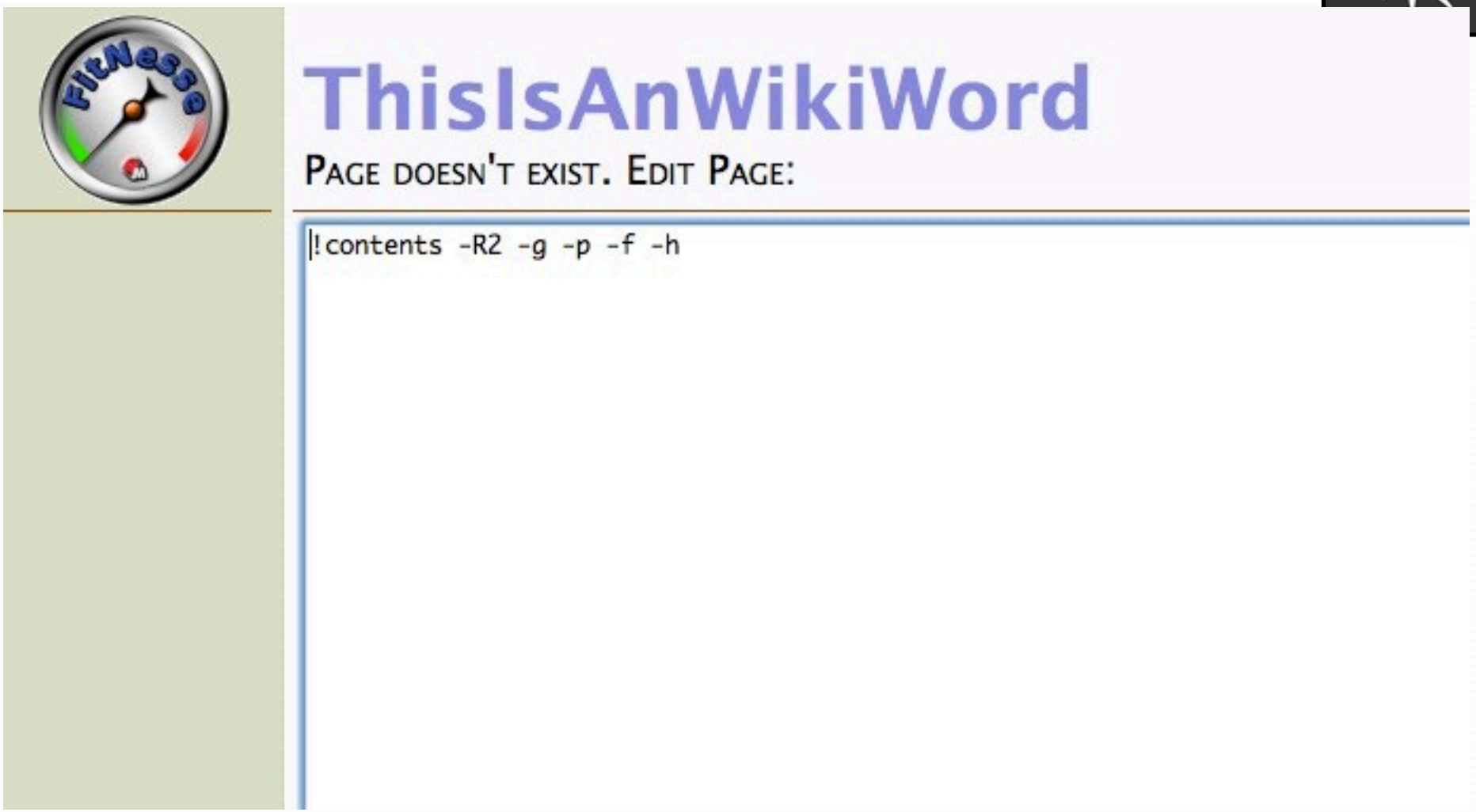
Editing a page

Try typing a WikiWord (at least 2 capital letters) - Then save

Creating a new page

Click on the '?' to create that new page

Editing a new page

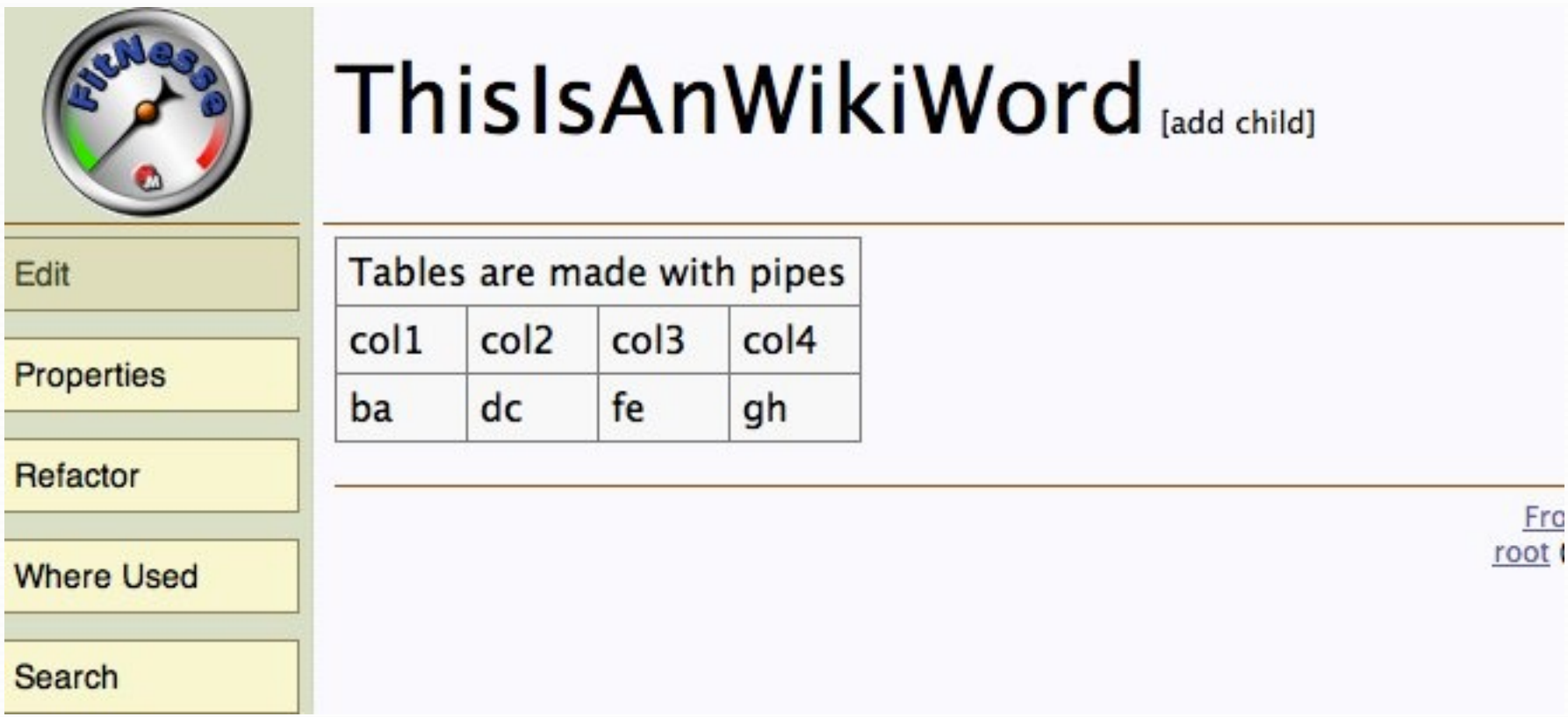


Tables

Tables in FitNesse are defined with pipes '|'

Tables

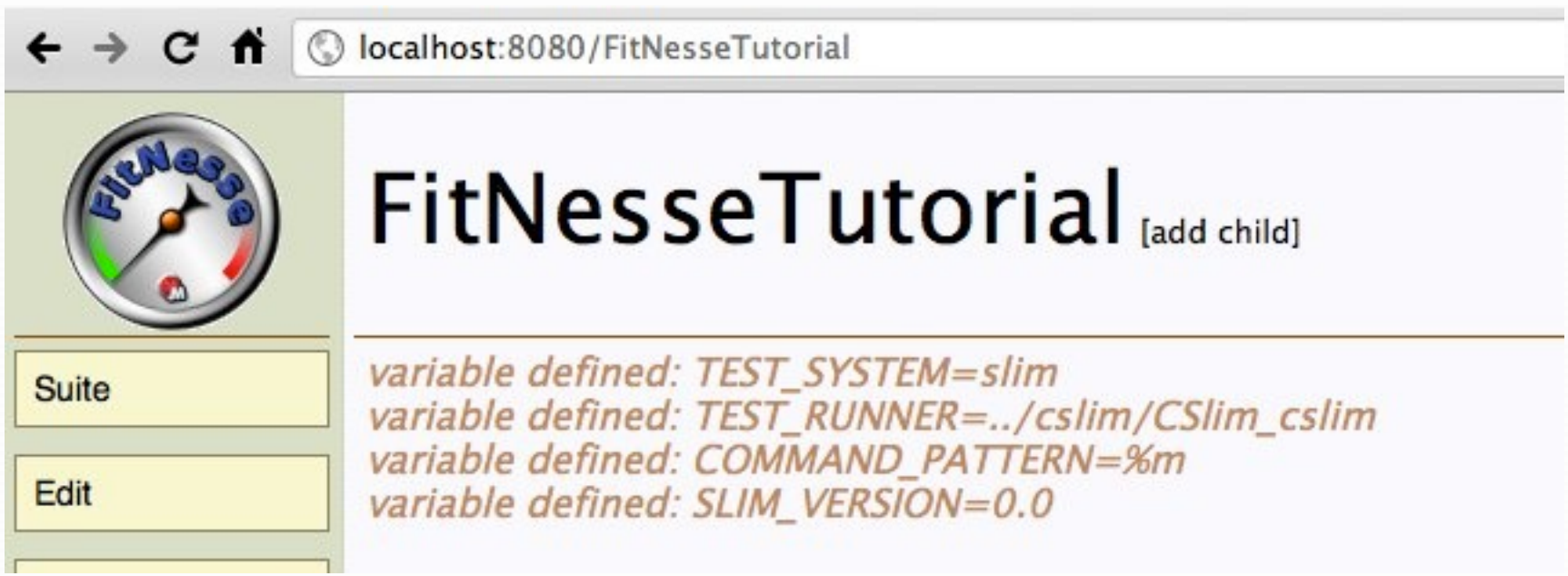
Save your table



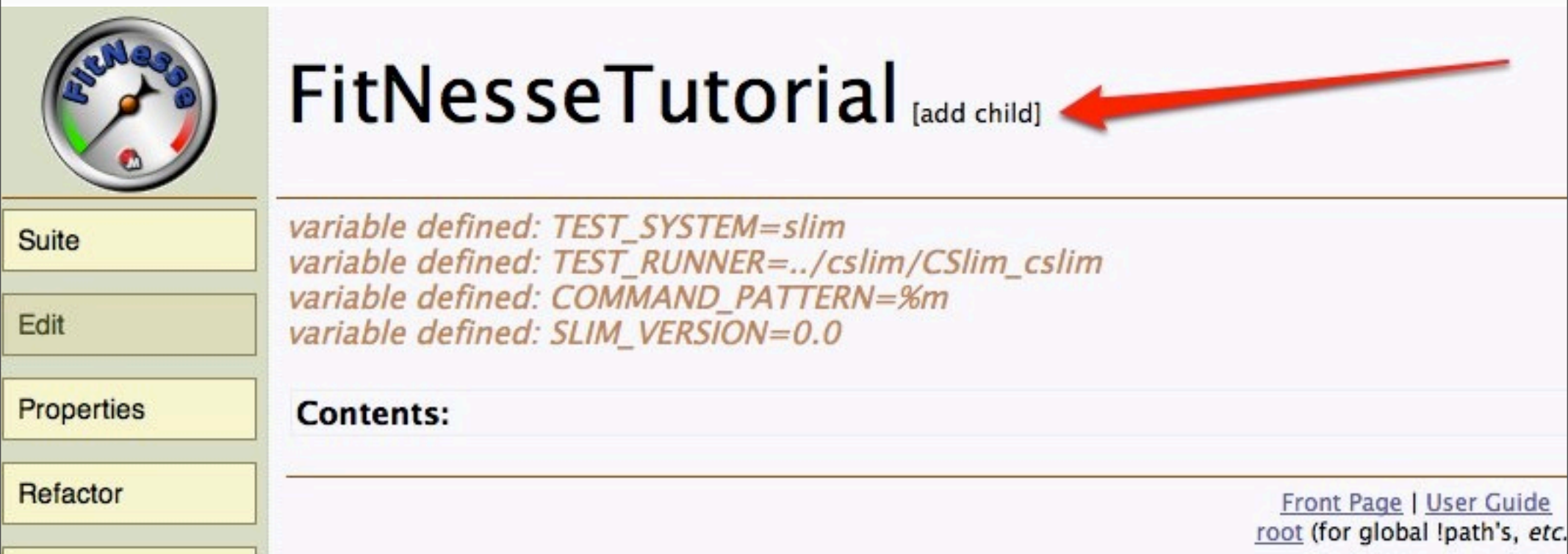
User Guide

Learn more about the wiki markup language

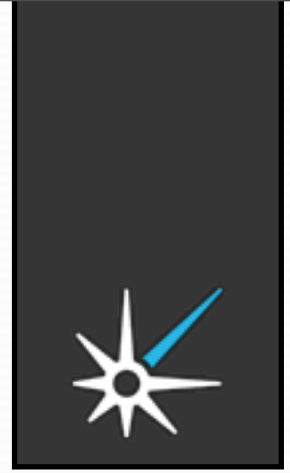
Let's write some examples



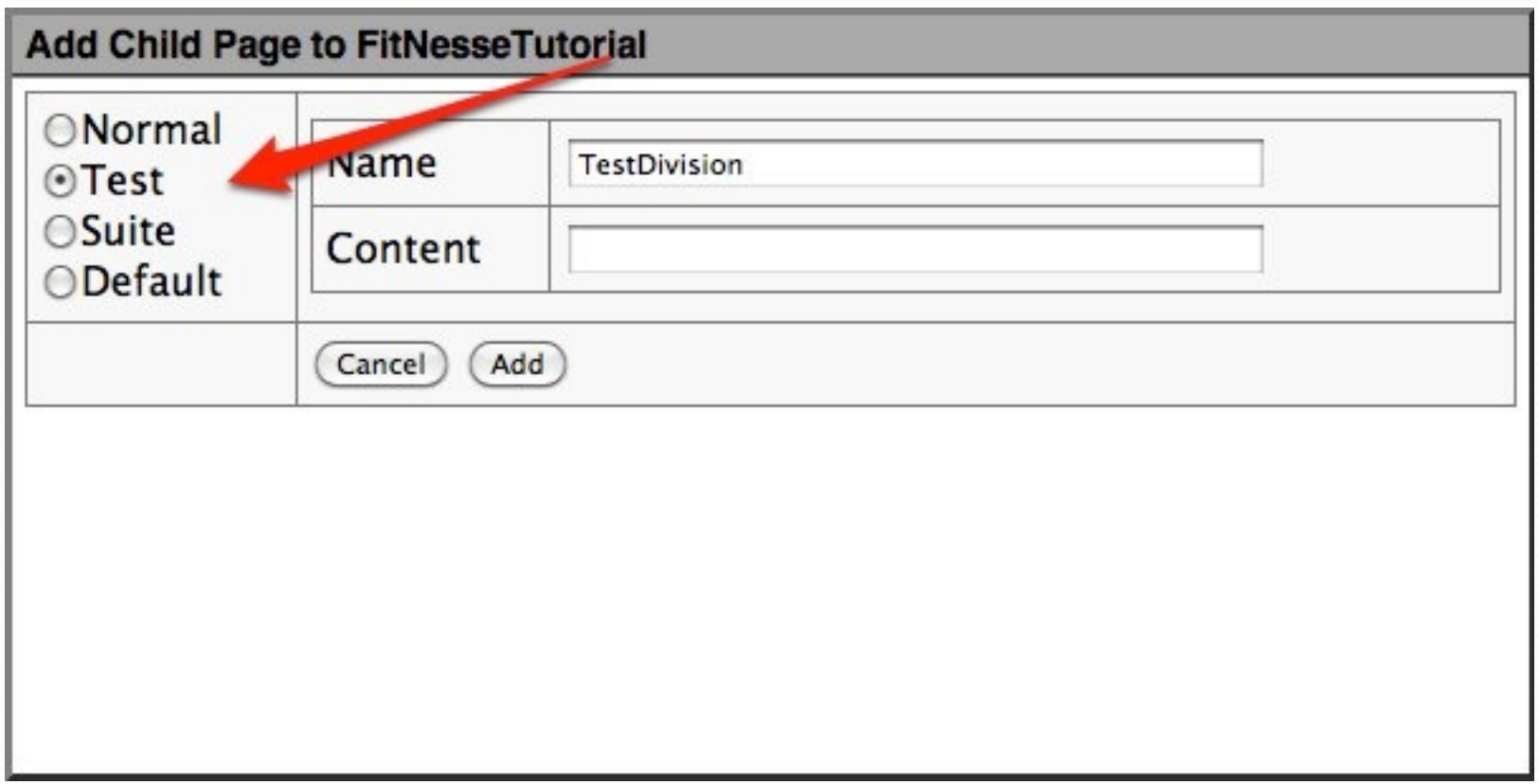
Another way to add pages



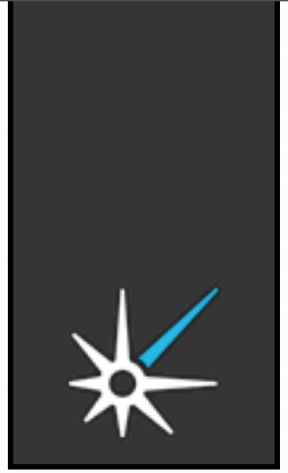
Three types of pages

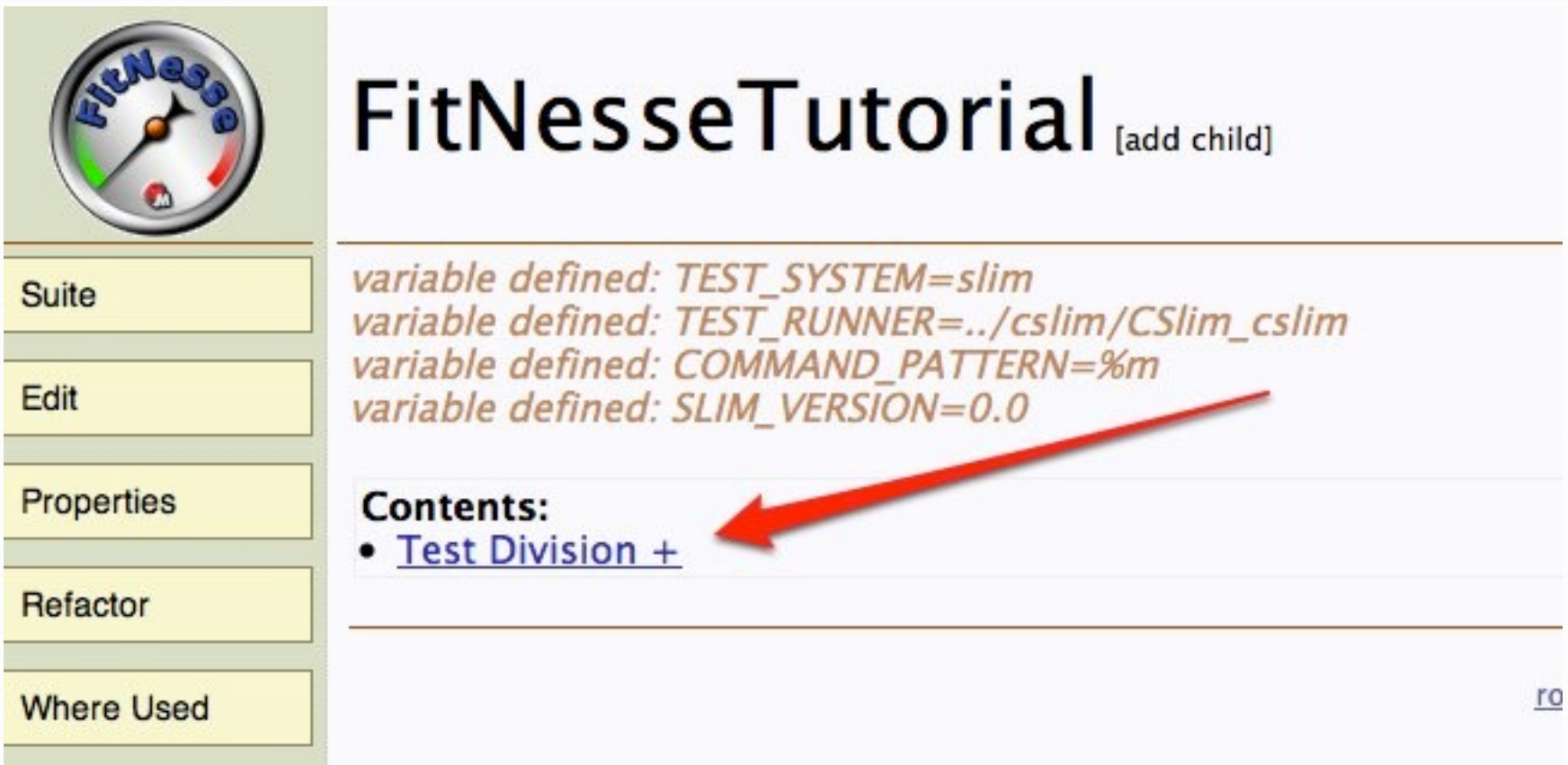


Normal - Contain only text and links
Test - Executable pages
Suite - a SubWiki of tests (or other suites)



Let's view the new test





Test Pages

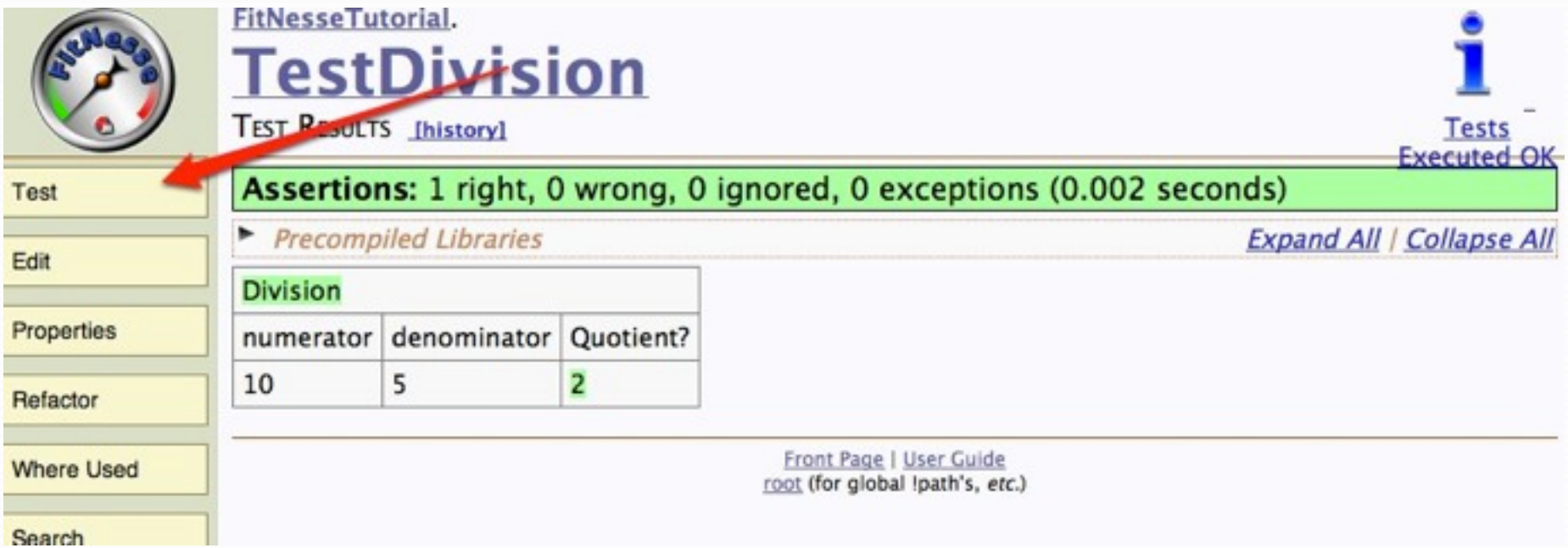
Behold, the test button

Calculation Test

Edit this test and create a decision table

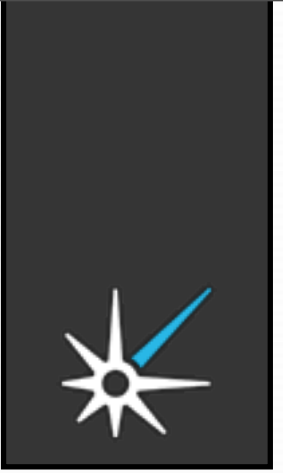
Push the button!

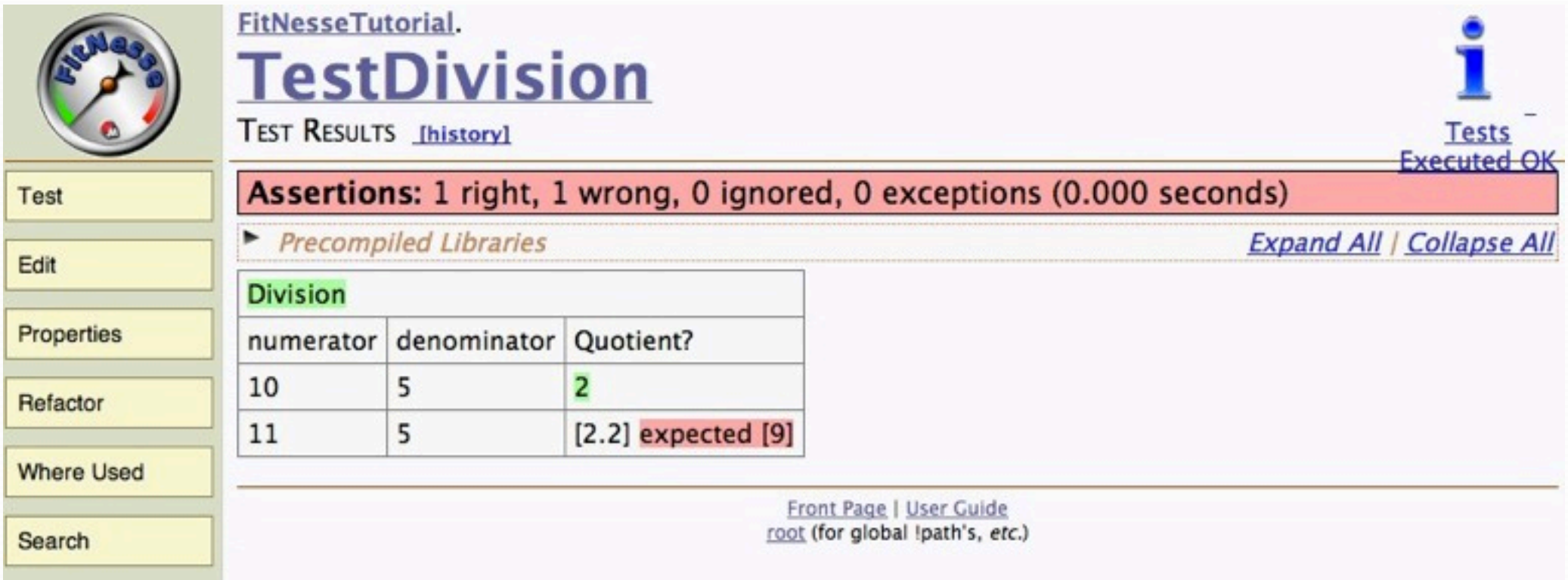
Green means passed



Try some other values

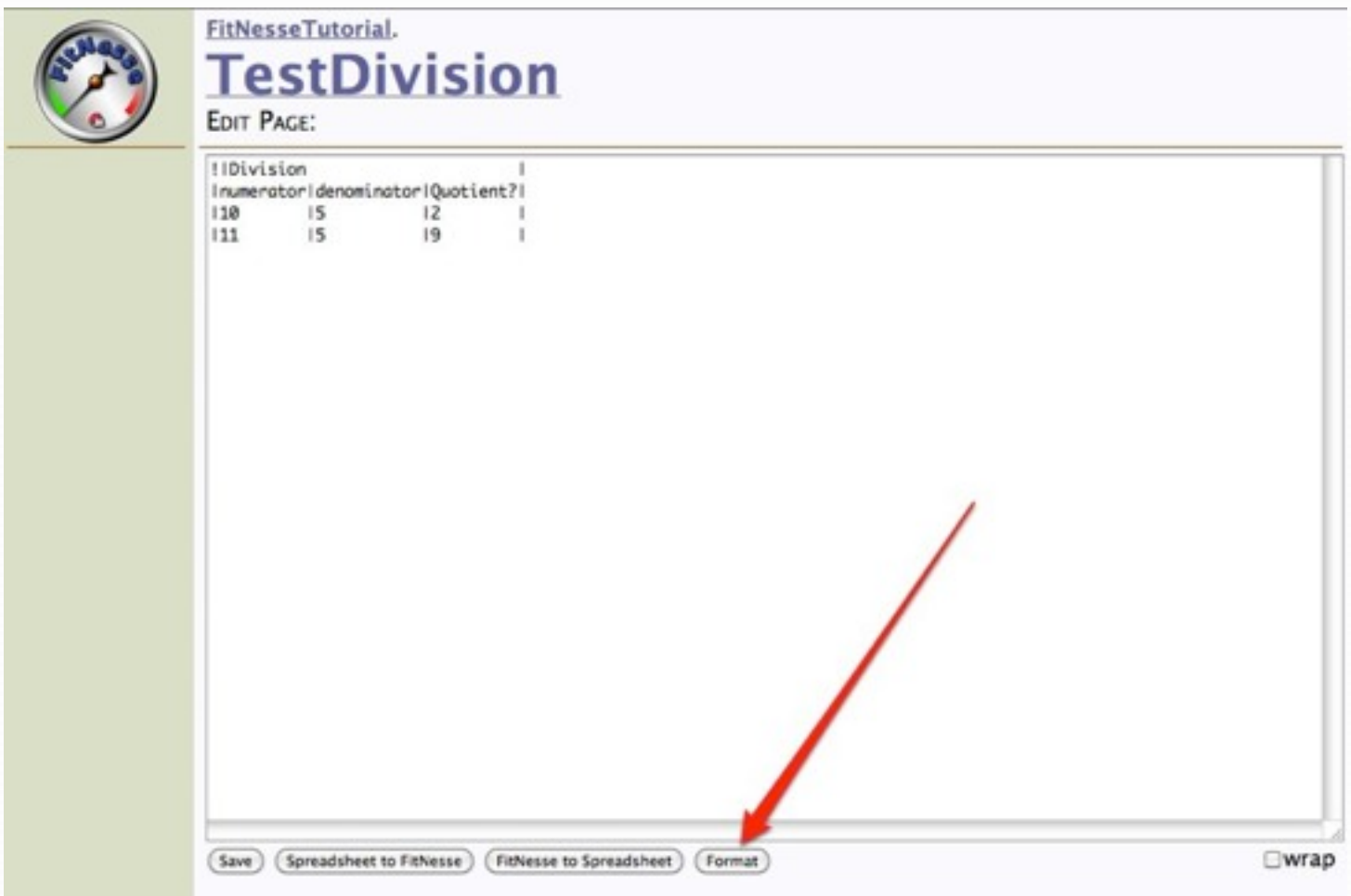
Red means failure





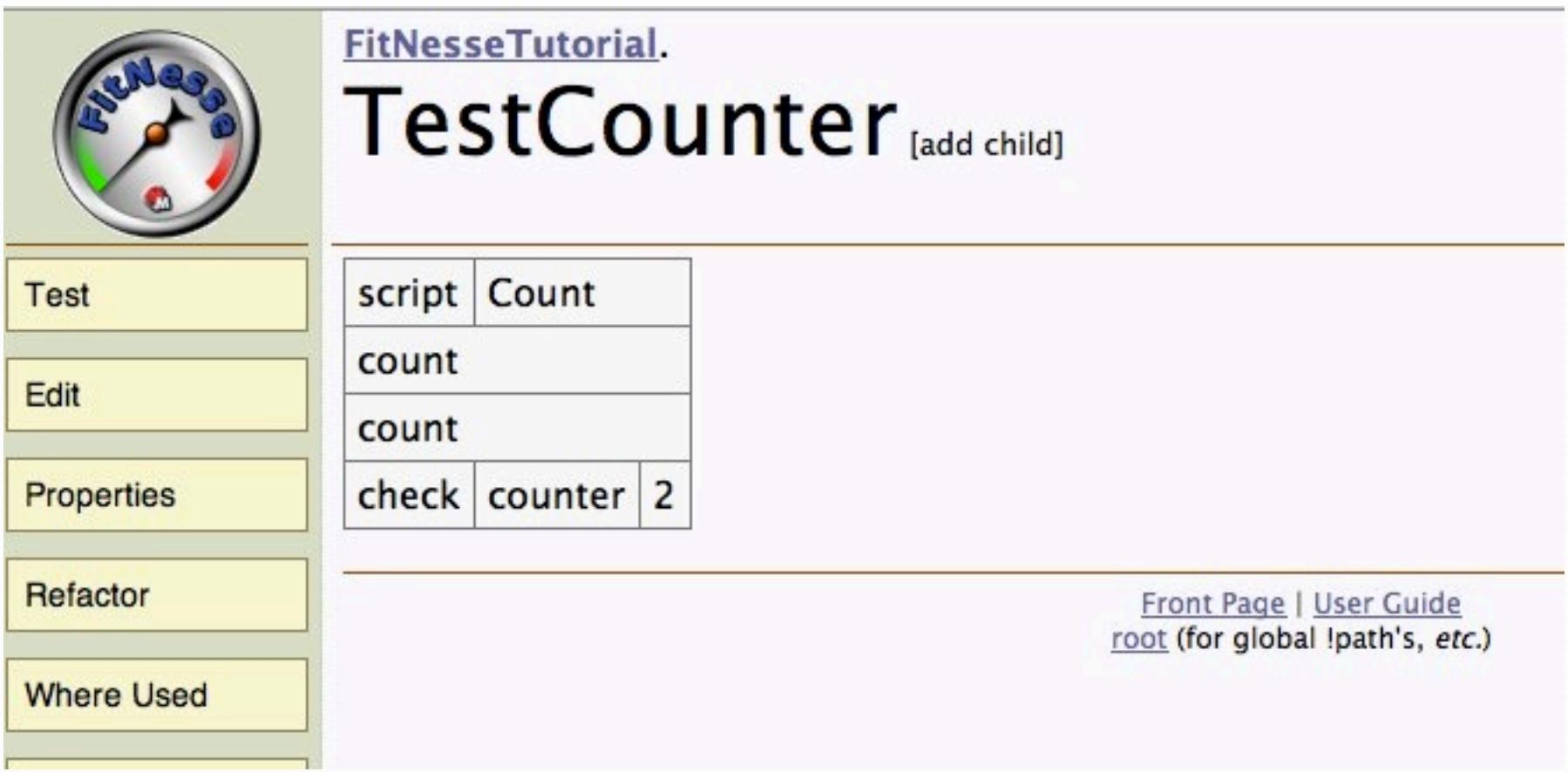
Line up your columns

Try the “format” button

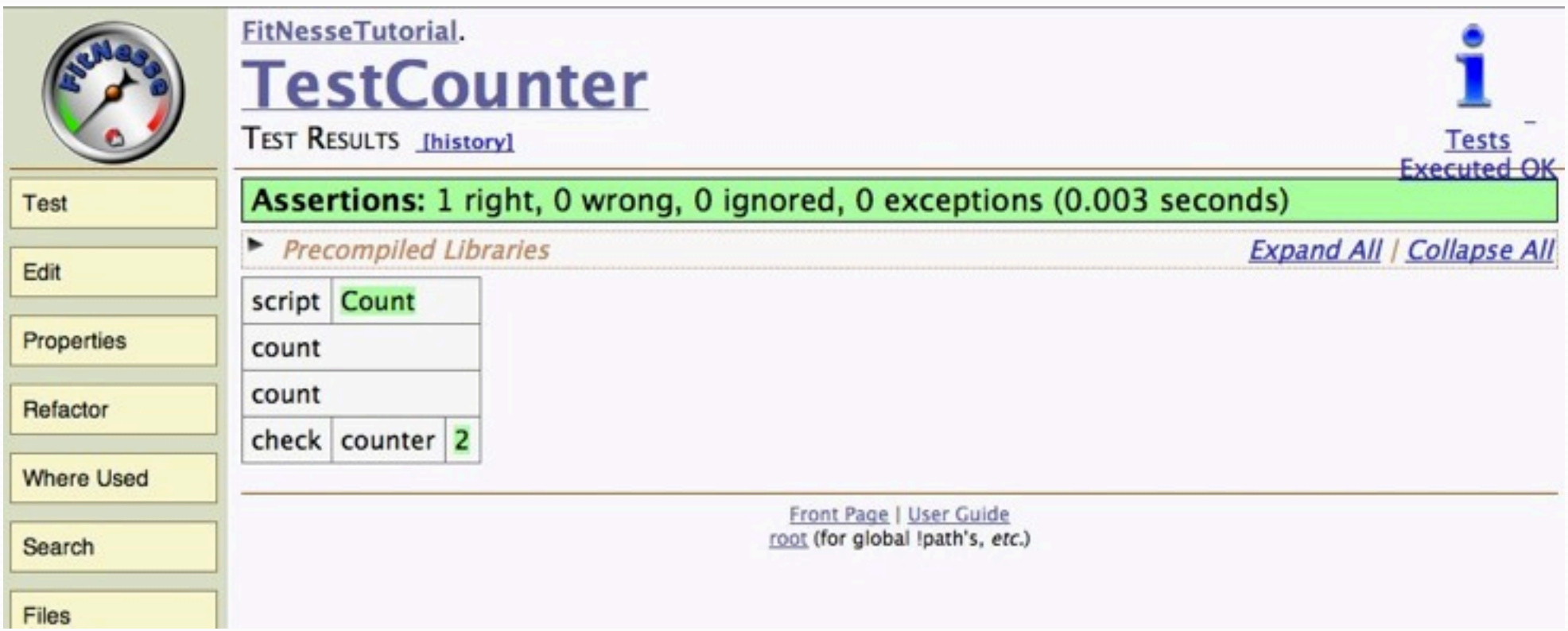


Workflow Test

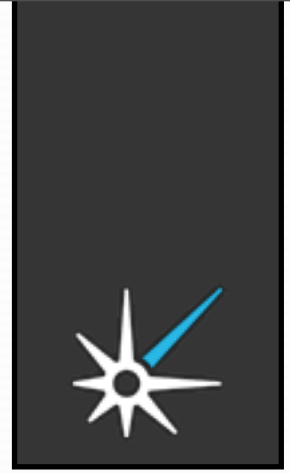
Create a new test with a script table

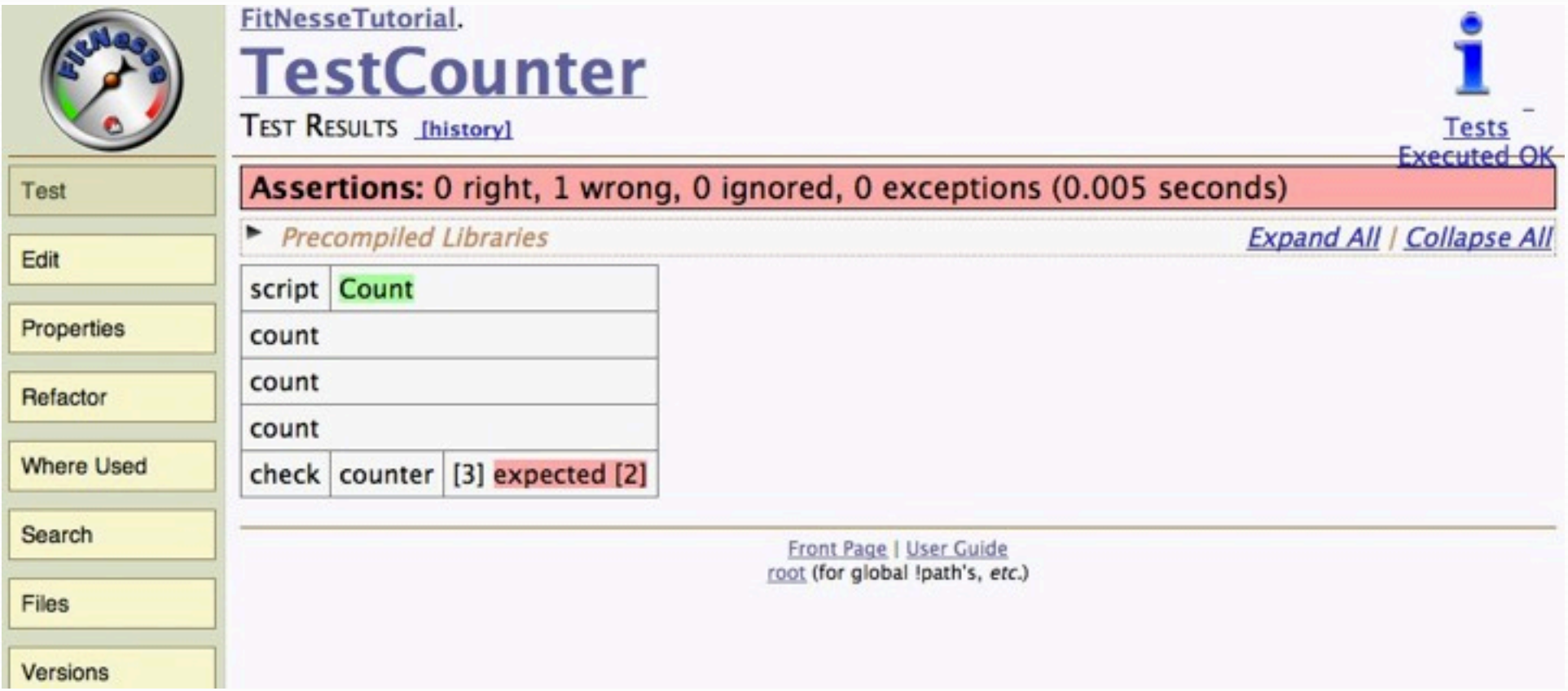


Workflow Test Run

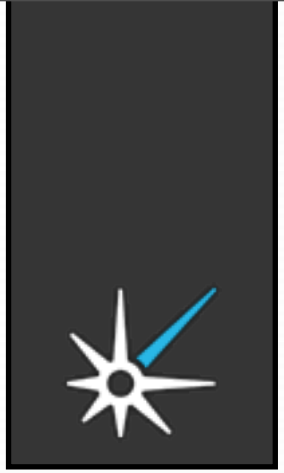


Workflow failure





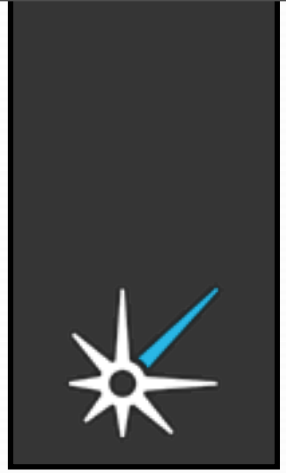
Exercise



Create a TreadmillSuite

Create the following tests

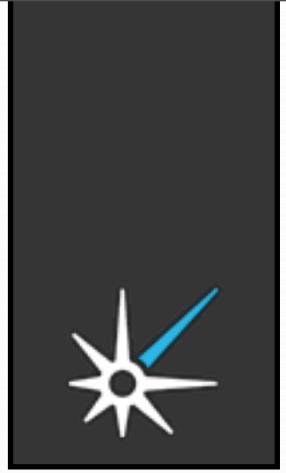
- TestIncrementTreadmill (workflow)
- TestDecrementTreadmill (workflow)
- TestMaxSpeed (workflow)
- TestCumulativeDistance (calculation)
- TestTotalDistance (calculation)



TreadmillControlSuite.

TestIncrementSpeed [add child]

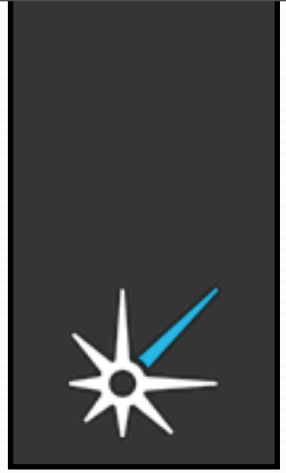
script	Treadmill	
given treadmill speed	1.0	
increment speed		
check	target speed	1.1



TreadmillControlSuite.

TestDecrementSpeed [add child]

script	Treadmill	
given treadmill speed	1.0	
decrement speed		
check	target speed	0.9

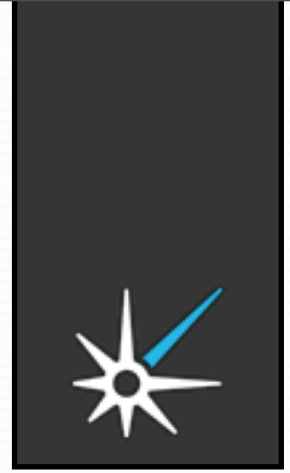


TreadmillControlSuite.

TestMaxSpeed [add child]

script	Treadmill	
given max speed	3.0	
given treadmill speed	3.0	
increment speed		
check	target speed	3.0

Front Page | User Guide
root (for global !path's, *etc.*)

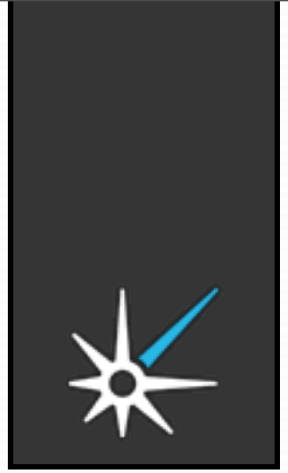


TreadmillControlSuite.

TestCumulativeDistance [add child]

Treadmill Cumulative Distance		
speed	time	distance?
1.0	30	0.5
2.0	30	1.5
1.5	60	3.0

Front Page | User Guide
root (for global !path's, *etc.*)



TreadmillControlSuite.

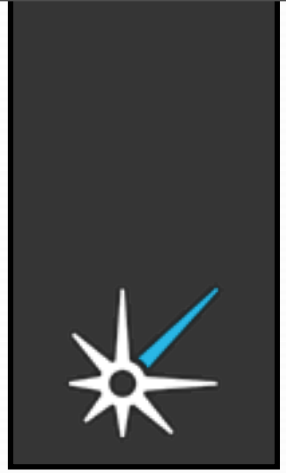
TestTotalDistance [add child]

Treadmill Distance		
speed	time	distance?
1.0	30	0.5
2.0	30	1.0
1.5	60	1.5

Front Page | User Guide
root (for global !path's, *etc.*)

It's Fixture Time

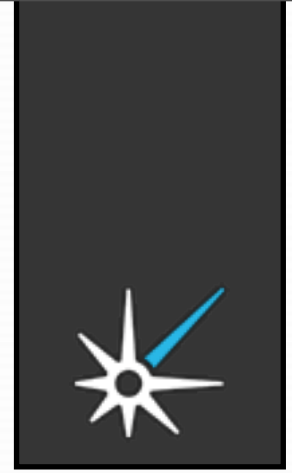
cslim/fixtures/Fixtures.c



```
1  #include "Fixtures.h"
2
3  SLIM_FIXTURES
4    SLIM_FIXTURE(Division)
5    SLIM_FIXTURE(Count)
6    SLIM_FIXTURE(EmployeePayRecordsRow)
7    SLIM_FIXTURE(ExceptionsExample)
8    SLIM_FIXTURE(Multiplication)
9  SLIM_END
10
11
```

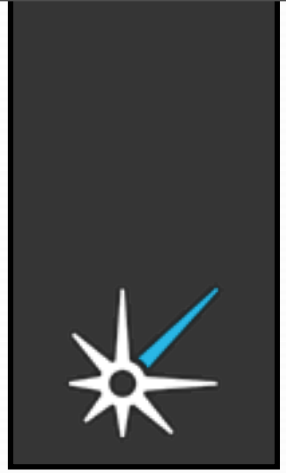
It's Fixture Time

cslim/fixtures/DecisionTableExample.c



```
64 SLIM_CREATE_FIXTURE(Division)
65   SLIM_FUNCTION(setNumerator)
66   SLIM_FUNCTION(setDenominator)
67   SLIM_FUNCTION(Quotient)
68   SLIM_FUNCTION(execute)
69   SLIM_FUNCTION(reset)
70   SLIM_FUNCTION(table)
71 SLIM_END
```

Fixture “Objects”

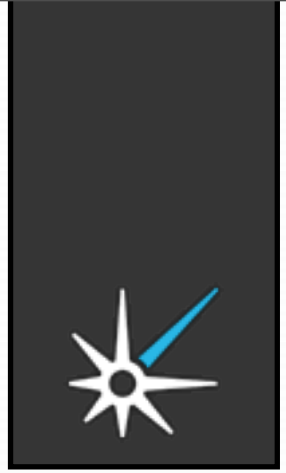


cslim/fixtures/DecisionTableExample.c

```
6
7  typedef struct Division
8  {
9    float numerator;
10   float denominator;
11   char result[32];
12 } Division;
13
14 void* Division_Create(StatementExecutor* errorHandler, SlimList* args)
15 {
16   Division* self = (Division*)malloc(sizeof(Division));
17   memset(self, 0, sizeof(Division));
18   return self;
19 }
20
21 void Division_Destroy(void* void_self)
22 {
23   free(void_self);
24 }
25
```

Fixture functions

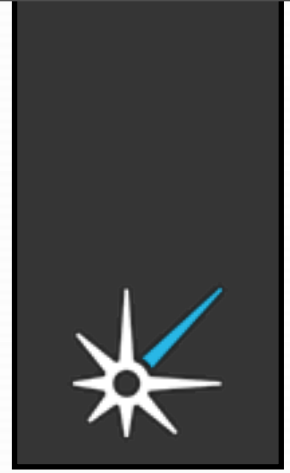
cslim/fixtures/DecisionTableExample.c



```
26 static char* setNumerator(void* void_self, SlimList* args) {
27   Division* self = (Division*)void_self;
28   self->numerator = atof(SlimList_GetStringAt(args, 0));
29   return "";
30 }
```

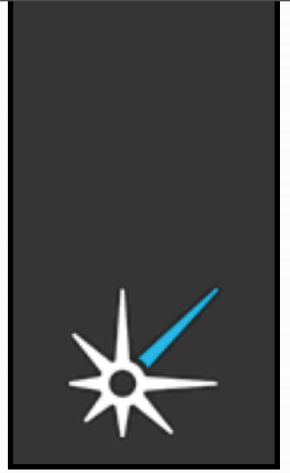
SlimList

cslim/include/CSlim/SlimList.h



```
16 SlimList * SlimList_GetListAt(SlimList* self, int index);
17 char * SlimList_GetStringAt(SlimList* self, int index);
18 double SlimList_GetDoubleAt(SlimList* self, int index);
19 SlimList* SlimList_GetHashAt(SlimList* self, int index);
```

Returning a Value

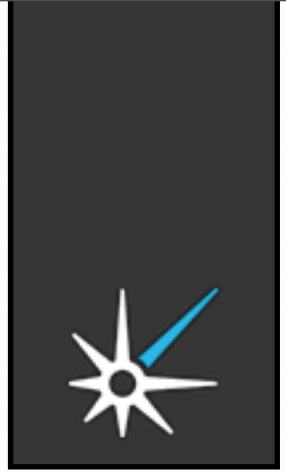


cslim/fixtures/DecisionTableExample.c

```
39
40 static char* Quotient(void* void_self, SlimList* args) {
41   Division* self = (Division*)void_self;
42   float quotient = self->numerator / self->denominator;
43   snprintf(self->result, 32, "%g", quotient);
44   return self->result;
45 }
46
```

All return values are strings.

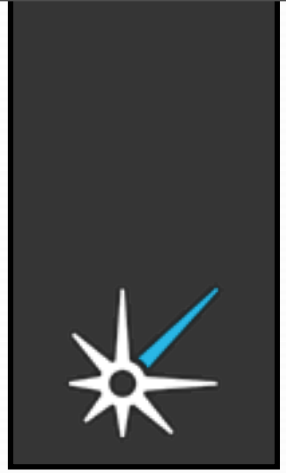
Extra functions



Decision tables have a few optional functions.

```
static char* execute(void* void_self, SlimList *args) {
  return "";
}

static char* reset(void* void_self, SlimList *args) {
  return "";
}
```

Order of Execution



all setters
execute()
all getters
reset()

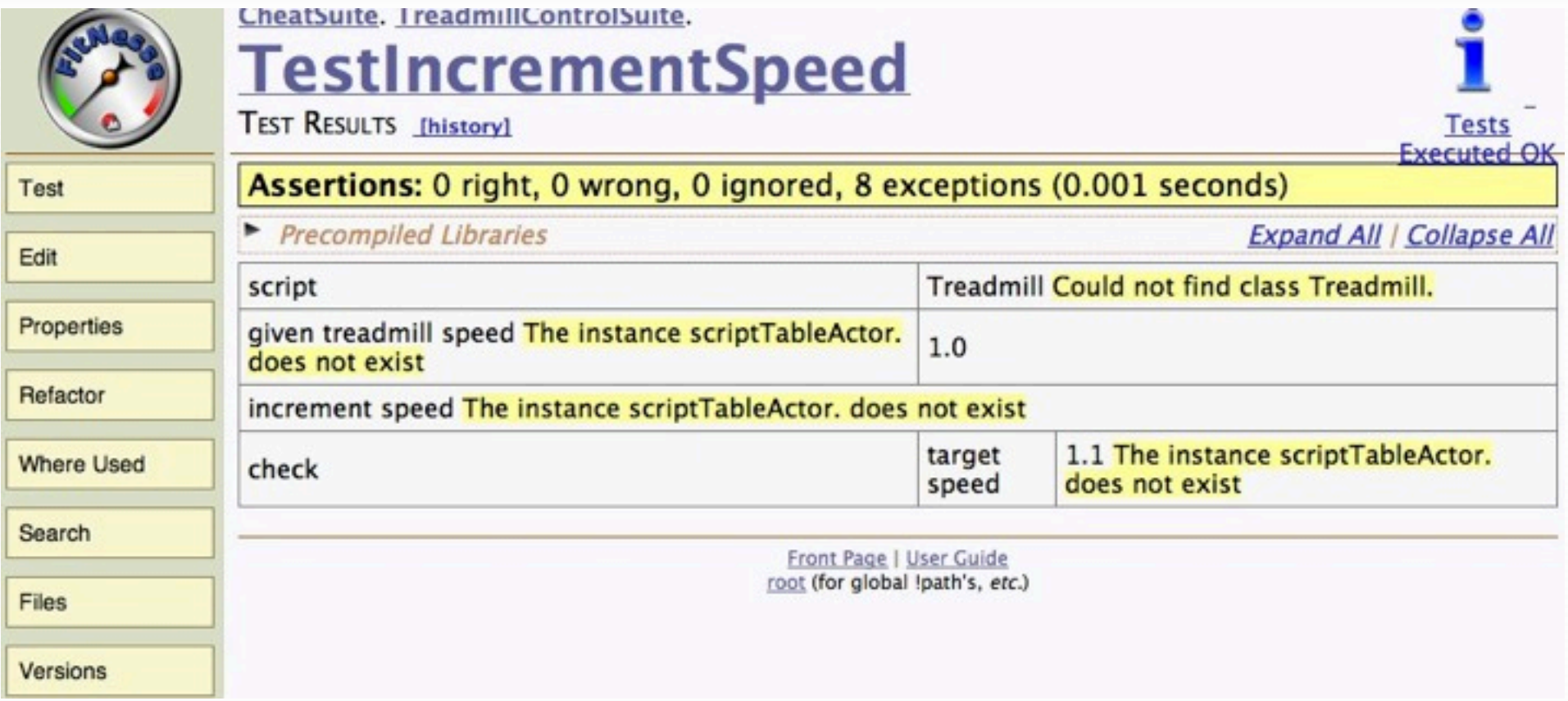
Division		
numerator	denominator	Quotient?
10	5	2
11	5	9

Let's build a fixture

CheatSuite.TreadmillControlSuite

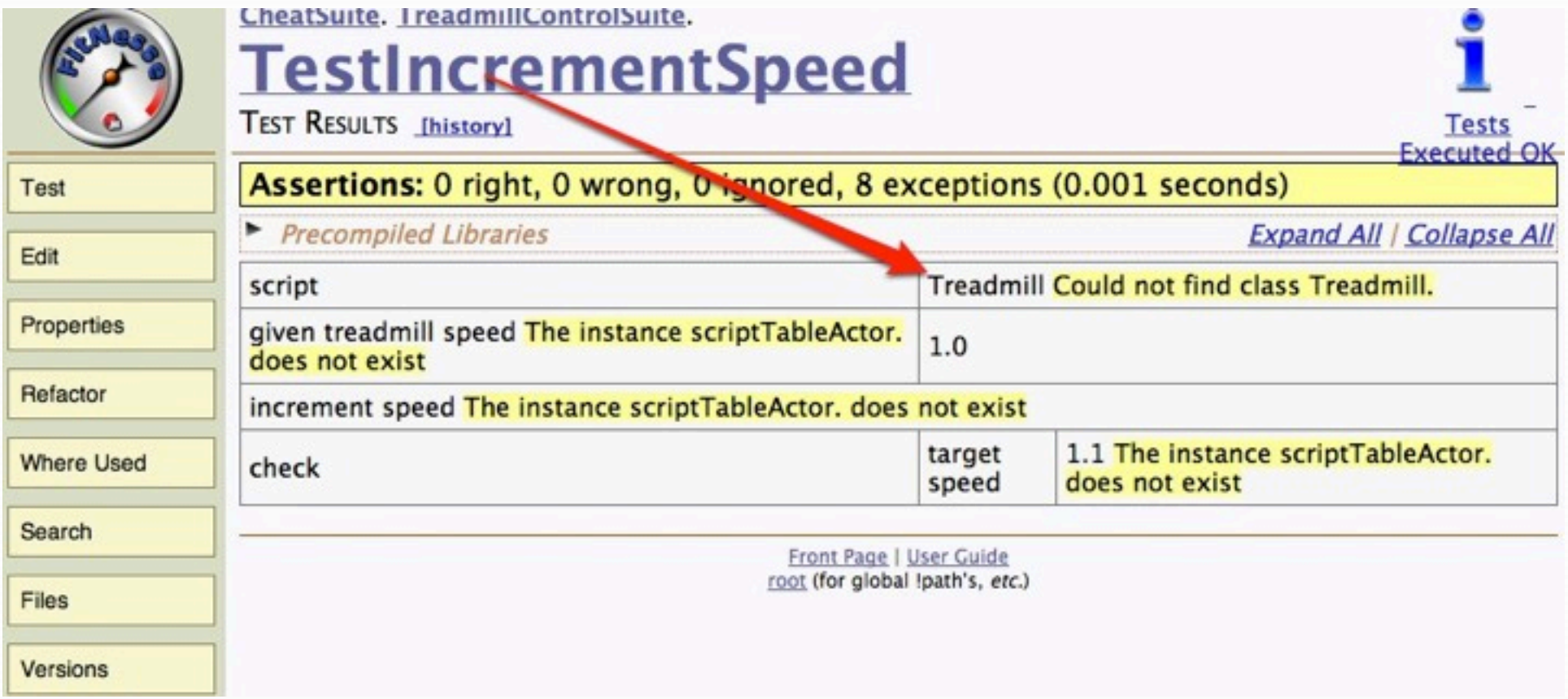
TestIncrementSpeed

Exceptions are yellow

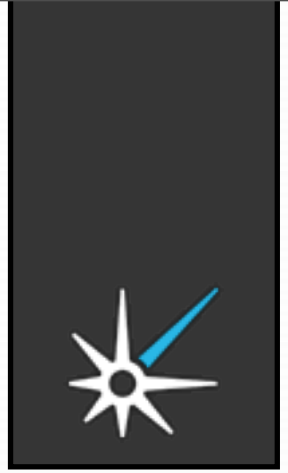


TestIncrementSpeed

Let's create a fixture called "Treadmill"



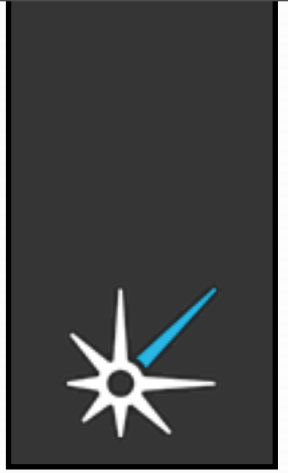
Create a Fixture



cd treadmill/fixtures/
cp FixtureTemplate.c Treadmill.c
sed -i '' s/ExampleFixture/Treadmill/g Treadmill.c

```
#include <stdlib.h>
#include <memory.h>
#include <stdio.h>
#include "Fixtures.h"
#include "SlimList.h"

typedef struct Treadmill
{
  char result[32];
} Treadmill;

void* Treadmill_Create(StatementExecutor* errorHandler, SlimList* args)
{
  Treadmill* self = (Treadmill*)malloc(sizeof(Treadmill));
  memset(self, 0, sizeof(Treadmill));
  return self;
}

void Treadmill_Destroy(void* void_self)
{
  Treadmill* self = (Treadmill*)void_self;
  free(self);
}

static char* exampleMethod(void* void_self, SlimList* args)
{
  Treadmill* self = (Treadmill*)void_self;
  return "";
}

SLIM_CREATE_FIXTURE(Treadmill)
  SLIM_FUNCTION(exampleMethod)
SLIM_END
```

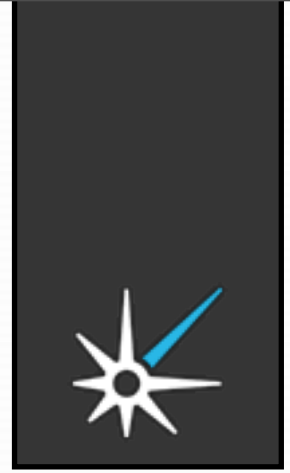
Register the Fixture



fixtures/FixtureMain.c

```
#include "Fixtures.h"

SLIM_FIXTURES
    SLIM_FIXTURE(Treadmill)
SLIM_END
```

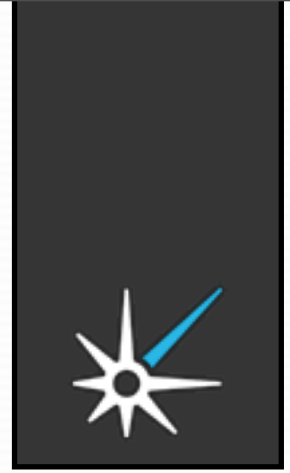
ReBuild the ‘at’ target



%> make at

Run the Test again

Fixture was found!



CheatSuite. TreadmillControlSuite.

TestIncrementSpeed

TEST RESULTS [history]

Tests Executed OK

Assertions: 0 right, 0 wrong, 0 ignored, 6 exceptions (0.005 seconds)

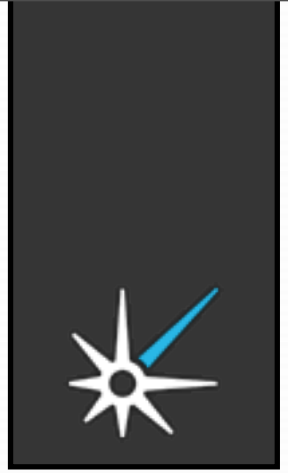
▶ *Precompiled Libraries* — Expand All | Collapse All

script	Treadmill	
given treadmill speed Method givenTreadmillSpeed[1] not found in Treadmill.	1.0	
increment speed Method incrementSpeed[0] not found in Treadmill.		
check	target speed	1.1 Method targetSpeed[0] not found in Treadmill.

Front Page | User Guide
root (for global !path's, *etc.*)

The First Method

givenTreadmillSpeed



CheatSuite. TreadmillControlSuite.

TestIncrementSpeed

TEST RESULTS [history]

Tests Executed OK

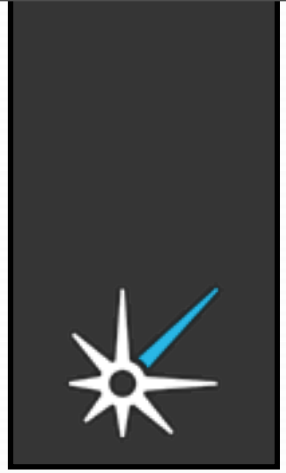
Assertions: 0 right, 0 wrong, 0 ignored, 6 exceptions (0.005 seconds)

▶ *Precompiled Libraries* — Expand All | Collapse All

script		Treadmill
given treadmill speed Method not found in Treadmill.	givenTreadmillSpeed[1]	1.0
increment speed Method incrementSpeed[0] not found in Treadmill.		
check	target speed	1.1 Method targetSpeed[0] not found in Treadmill.

Front Page | User Guide
root (for global !path's, *etc.*)

The First Method

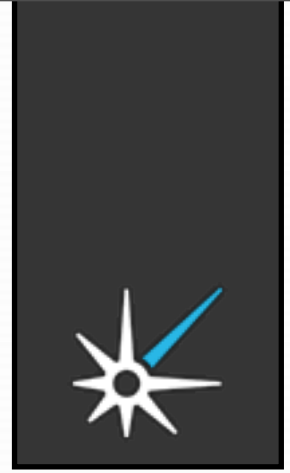


treadmill/fixtures/Treadmill.c

```
static char* givenTreadmillSpeed(void* void_self, SlimList* args)
{
  Treadmill* self = (Treadmill*)void_self;
  return "";
}

SLIM_CREATE_FIXTURE(Treadmill)
  SLIM_FUNCTION(givenTreadmillSpeed)
SLIM_END
```

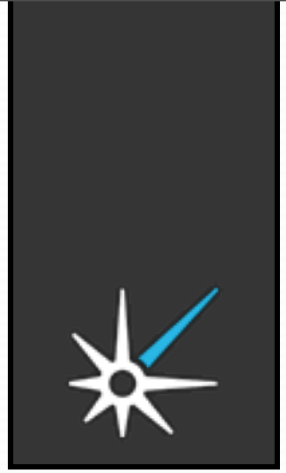
IncrementSpeed

Implement Increment Speed on your own.



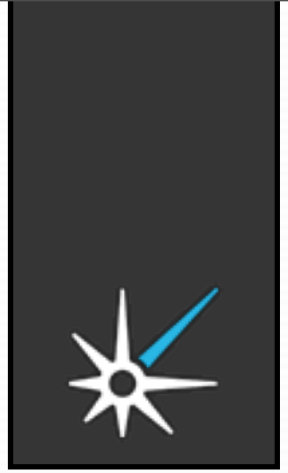
targetSpeed

Return a value to make the test fail.



```
static char* targetSpeed(void* void_self, SlimList* args)
{
  Treadmill* self = (Treadmill*)void_self;
  return "0.0";
}
```

targetSpeed Failure



CheatSuite. TreadmillControlSuite.

TestIncrementSpeed

TEST RESULTS [history]

Tests Executed OK

Assertions: 0 right, 1 wrong, 0 ignored, 0 exceptions (0.003 seconds)

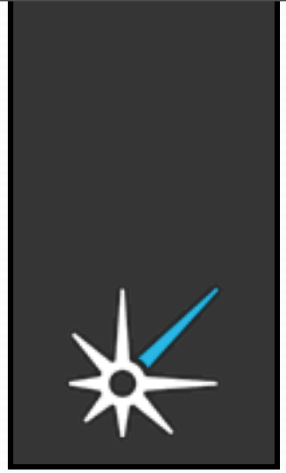
▸ *Precompiled Libraries* — *Expand All* | *Collapse All*

script	Treadmill	
given treadmill speed	1.0	
increment speed		
check	target speed	[0.0] expected [1.1]

Front Page | User Guide
root (for global !path's, *etc.*)

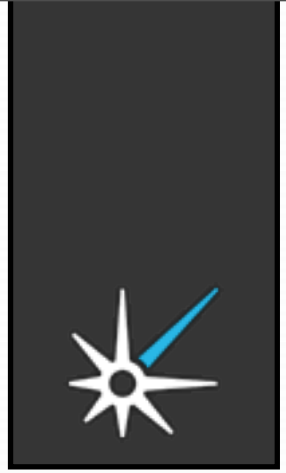
Treadmill Api

include/treadmill/Api.h



```
10 typedef struct ApiStruct * Api;
11
12 Api Api_Create(void);
13 void Api_Destroy(Api);
14 double Api_GetTargetSpeed(Api);
15 void Api_SetTargetSpeed(Api, double);
16 void Api_IncrementTargetSpeed(Api);
17 void Api_DecrementTargetSpeed(Api);
18 void Api_SetMaximumSpeed(Api, double);
19 double Api_DistanceTravelled(Api);
20 void Api_Reset(Api);
```

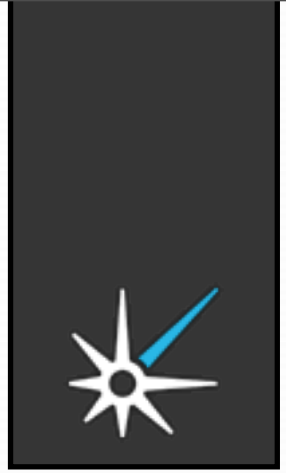
Declare an instance



treadmill/fixtures/Treadmill.c

```
 8 typedef struct Treadmill
 9 {
10   char result[32];
11   Api api;
12 } Treadmill;
```

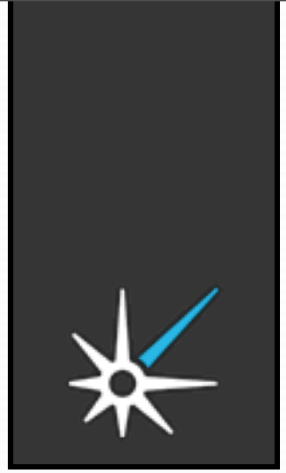
Create & Destroy



treadmill/fixtures/Treadmill.c

```
14 void* Treadmill_Create(StatementExecutor* errorHandler, SlimList* args)
15 {
16   Treadmill* self = (Treadmill*)malloc(sizeof(Treadmill));
17   memset(self, 0, sizeof(Treadmill));
18   self->api = Api_Create();
19   return self;
20 }
21
22 void Treadmill_Destroy(void* void_self)
23 {
24   Treadmill* self = (Treadmill*)void_self;
25   Api_Destroy(self->api);
26   free(self);
27 }
```

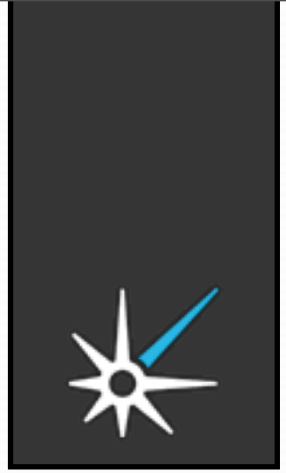
Reading from a SlimList



treadmill/fixtures/Treadmill.c

```
29 static char* givenTreadmillSpeed(void* void_self, SlimList* args)
30 {
31   Treadmill* self = (Treadmill*)void_self;
32   double speed = SlimList_GetDoubleAt(args, 0);
33   Api_SetTargetSpeed(self->api, speed);
34   return "";
35 }
36
```

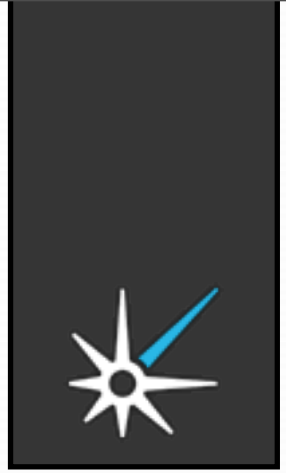
No Parameters



treadmill/fixtures/Treadmill.c

```
36
37 static char* incrementSpeed(void* void_self, SlimList* args)
38 {
39   Treadmill* self = (Treadmill*)void_self;
40   Api_IncrementTargetSpeed(self->api);
41   return "";
42 }
43
```

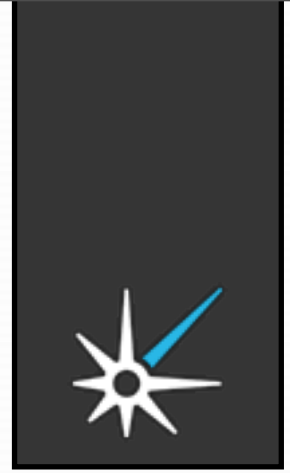
Result



treadmill/fixtures/Treadmill.c

```
44 static char* targetSpeed(void* void_self, SlimList* args)
45 {
46   Treadmill* self = (Treadmill*)void_self;
47   snprintf(self->result, 32, "%.1f", Api_GetTargetSpeed(self->api));
48   return self->result;
49 }
50
```

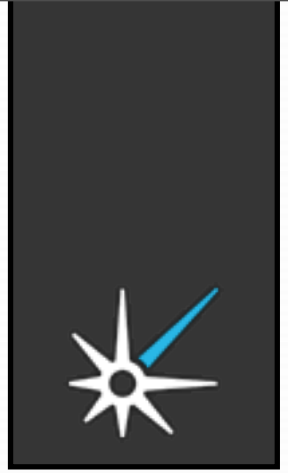
ReBuild the ‘at’ target



%> make at

Woot!

Green is good



CheatSuite. TreadmillControlSuite.

TestIncrementSpeed

TEST RESULTS [history]

Tests Executed OK

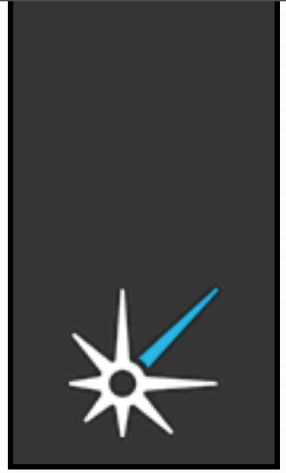
Assertions: 1 right, 0 wrong, 0 ignored, 0 exceptions (0.001 seconds)

▸ *Precompiled Libraries* — *Expand All* | *Collapse All*

script	Treadmill	
given treadmill speed	1.0	
increment speed		
check	target speed	1.1

Front Page | User Guide
root (for global !path's, *etc.*)

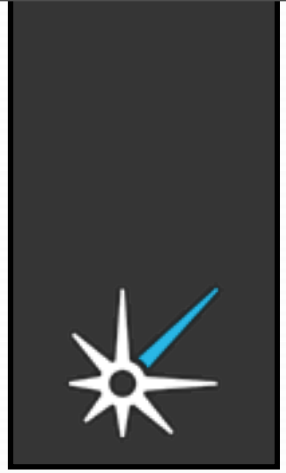
Exercise



Add fixture methods to Treadmill.c to support:

TestDecrementSpeed
TestMaxSpeed

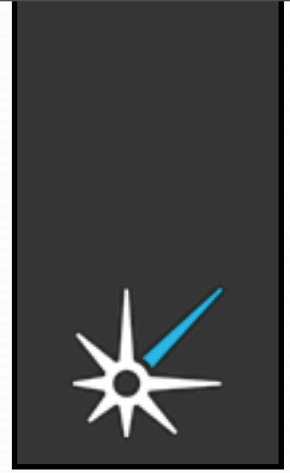
Catch up



If you need to catch up with the group

cp cheat_fixtures/Treadmill.c fixtures

Calculation Test



CheatSuite. TreadmillControlSuite.

TestCumulativeDistance

TEST RESULTS [history]

Tests Executed OK

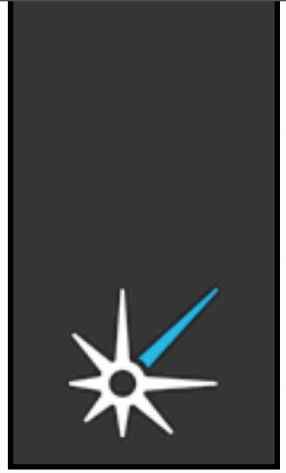
Assertions: 0 right, 0 wrong, 0 ignored, 29 exceptions (0.002 seconds)

▸ *Precompiled Libraries* — *Expand All* | *Collapse All*

Treadmill Cumulative Distance Could not find class TreadmillCumulativeDistance.		
speed	time	distance?
1.0 The instance decisionTable_0. does not exist	30 The instance decisionTable_0. does not exist	0.5 The instance decisionTable_0. does not exist
2.0 The instance decisionTable_0. does not exist	30 The instance decisionTable_0. does not exist	1.5 The instance decisionTable_0. does not exist
1.5 The instance decisionTable_0. does not exist	60 The instance decisionTable_0. does not exist	3.0 The instance decisionTable_0. does not exist

Front Page | User Guide
root (for global !path's, *etc.*)

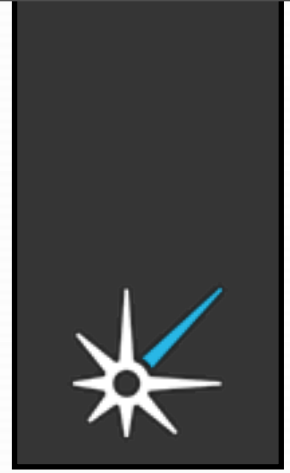
Create a new fixture



cd treadmill/fixtures/
cp FixtureTemplate.c TreadmillCumulativeDistance.c
sed -i '' s/ExampleFixture/TreadmillCumulativeDistance/g TreadmillCumulativeDistance.c

```
35 #include "Fixtures.h"
36
37 SLIM_FIXTURES
38   SLIM_FIXTURE(TreadmillCumulativeDistance)
39   SLIM_FIXTURE(Treadmill)
40 SLIM_END
41
```

Decision table naming



speed => setSpeed
time => setTime
distance? => distance

CheatSuite. TreadmillControlSuite.

TestCumulativeDistance

TEST RESULTS [history]

Tests Executed OK

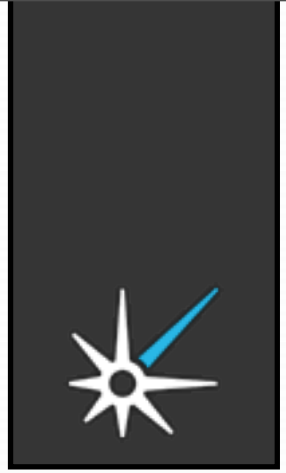
Assertions: 0 right, 0 wrong, 0 ignored, 18 exceptions (0.002 seconds)

▸ *Precompiled Libraries* — *Expand All* | *Collapse All*

Treadmill Cumulative Distance		
speed	time	distance?
1.0 Method setSpeed[1] not found in TreadmillCumulativeDistance.	30 Method setTime[1] not found in TreadmillCumulativeDistance.	0.5 Method distance[0] not found in TreadmillCumulativeDistance.
2.0 Method setSpeed[1] not found in TreadmillCumulativeDistance.	30 Method setTime[1] not found in TreadmillCumulativeDistance.	1.5 Method distance[0] not found in TreadmillCumulativeDistance.
1.5 Method setSpeed[1] not found in TreadmillCumulativeDistance.	60 Method setTime[1] not found in TreadmillCumulativeDistance.	3.0 Method distance[0] not found in TreadmillCumulativeDistance.

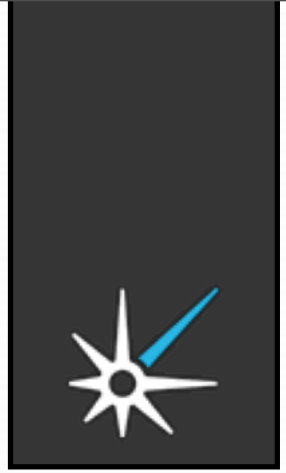
Front Page | User Guide

Make it red.



Implement stubs for:
setTime
setSpeed
distance

Does anybody really know what time it is?

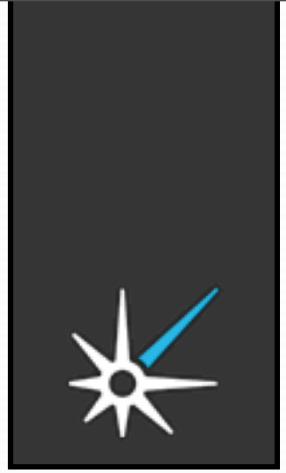


We will use a link-time seam to mock out uptime.

include/hardware/Uptime.h

```
11 void Uptime_Create(void);
12 void Uptime_Destroy(void);
13 long Uptime_MilliSeconds(void);
```

FakeUptime



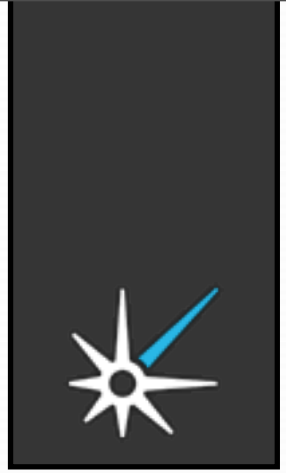
mocks/hardware/FakeUptime.h

```
11 #include "Uptime.h"
12 extern long uptimeMillis;
```

mocks/hardware/FakeUptime.c

```
 1 #include "Uptime.h"
 2 int uptimeMillis;
 3
 4 void Uptime_Create(void)
 5 {
 6   uptimeMillis = 0;
 7 }
 8
 9 long Uptime_MilliSeconds(void)
10 {
11   return uptimeMillis;
12 }
13
```

Back to our fixture



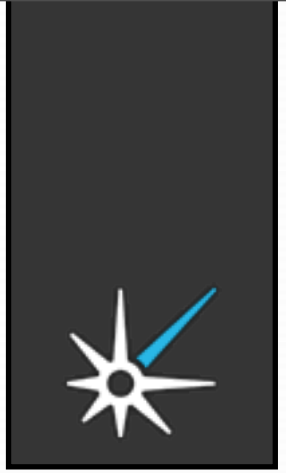
Use the uptimeMillis extern to set the current time

fixtures/TreadmillCumulativeDistance.c

```
#include "hardware/FakeUptime.h"
```

```
static char* execute(void* void_self, SlimList *args) {
  TreadmillCumulativeDistance* self = (TreadmillCumulativeDistance*)void_sel
  Api_SetTargetSpeed(self->api, self->speed);
  uptimeMillis += self->time;
  return "";
}
```

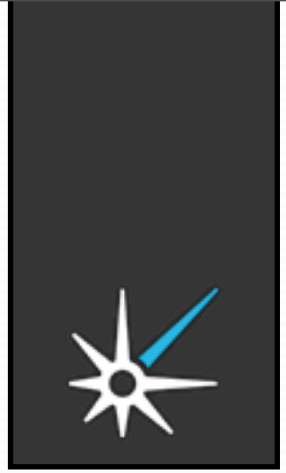
Exercise



Finish the TreadmillCumulativeDistance fixture.
Extra credit: TreadmillDistance fixture.
(hint: you'll need to use the reset function)

Cumulative Distance

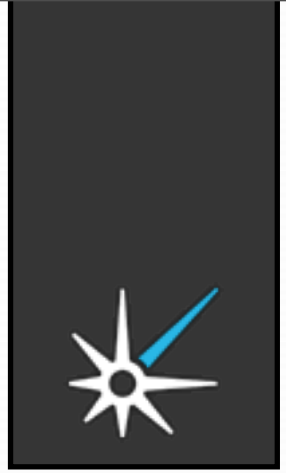
fixtures/TreadmillCumulativeDistance.c



```
 9 typedef struct TreadmillCumulativeDistance
10 {
11   char result[32];
12   Api api;
13   double speed;
14   double time;
15 } TreadmillCumulativeDistance;
16
```

Cumulative Distance

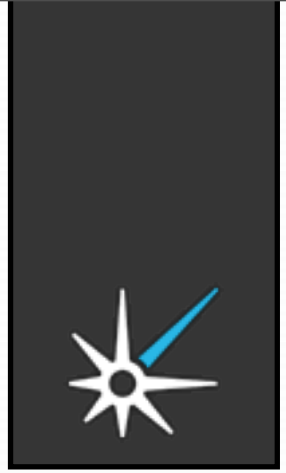
fixtures/TreadmillCumulativeDistance.c



```
39 static char* setSpeed(void* void_self, SlimList *args) {
40   TreadmillCumulativeDistance* self = (TreadmillCumulativeDistance*)void_se
41   self->speed = SlimList_GetDoubleAt(args, 0);
42   return "";
43 }
44
45 static char* setTime(void* void_self, SlimList *args) {
46   TreadmillCumulativeDistance* self = (TreadmillCumulativeDistance*)void_se
47   double minutes = SlimList_GetDoubleAt(args, 0);
48   self->time = minutes*60*1000;
49   return "";
50 }
```

Cumulative Distance

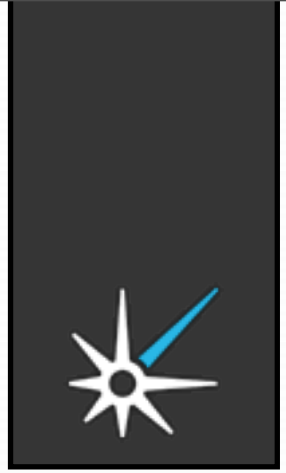
fixtures/TreadmillCumulativeDistance.c



```
31
32 static char* execute(void* void_self, SlimList *args) {
33   TreadmillCumulativeDistance* self = (TreadmillCumulativeDistance*)void_se
34   Api_SetTargetSpeed(self->api, self->speed);
35   uptimeMillis += self->time;
36   return "";
37 }
38
```

Cumulative Distance

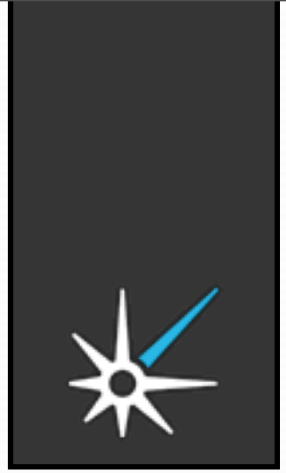
fixtures/TreadmillCumulativeDistance.c



```
52 static char* distance(void* void_self, SlimList *args) {
53   TreadmillCumulativeDistance* self = (TreadmillCumulativeDistance*)void_se
54   double d = Api_DistanceTravelled(self->api);
55   snprintf(self->result, 32, "%0.1f", d);
56   return self->result;
57 }
```

Total Distance

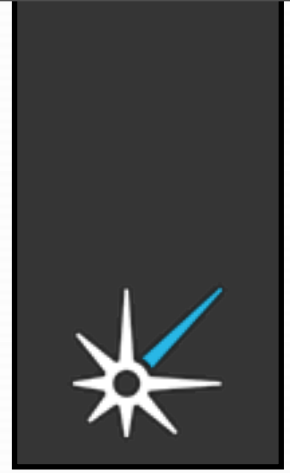
fixtures/TreadmillDistance.c



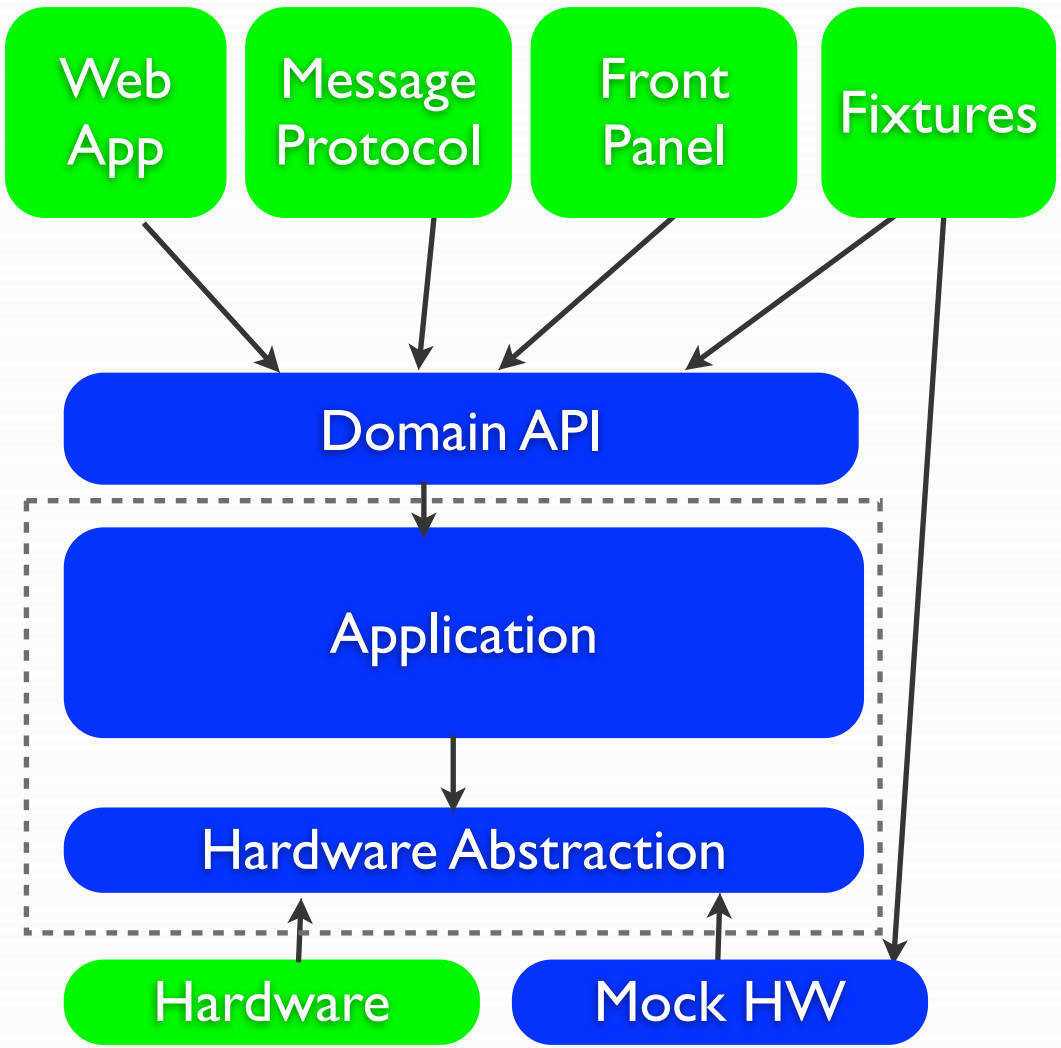
```
30 static char* reset(void* void_self, SlimList *args) {
31   TreadmillDistance* self = (TreadmillDistance*)void_self;
32   Api_Reset(self->api);
33   return "";
34 }
35
```

Break

Take 5.



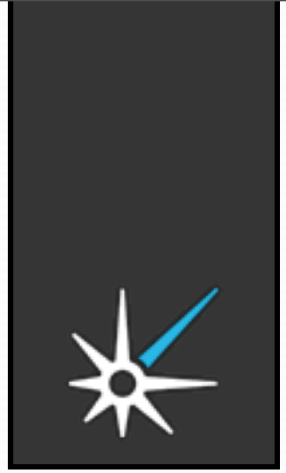
System Architecture



Production Build

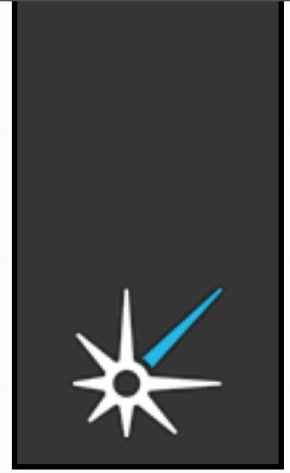
AT build

Hardware Abstraction



Remember, acceptance tests are not about testing the hardware.

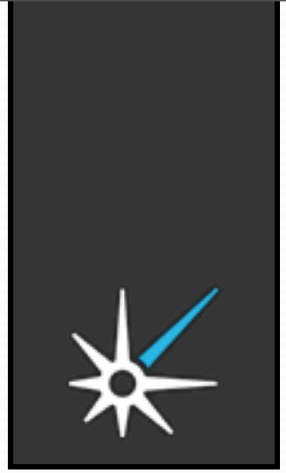
Hardware Abstraction



include/hardware/Pwm.h

```
 4 void Pwm_Create(void);
 5 void Pwm_Destroy(void);
 6
 7 void Pwm_Start();
 8 void Pwm_Stop();
 9 void Pwm_SetDutyCycle(double percent);
10 void Pwm_SetPeriod(int microseconds);
```

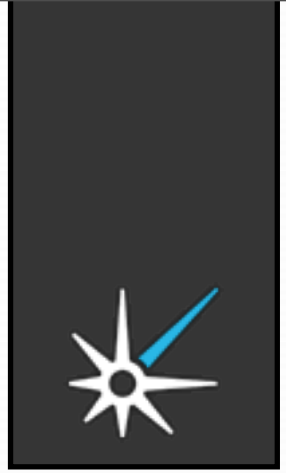
Hardware Abstraction



mocks/hardware/FakePwm.h

```
4 #include "Pwm.h"
5
6 extern int FakePwm_isRunning;
7 extern int FakePwm_period;
8 extern double FakePwm_dutyCycle;
```

Hardware Abstraction



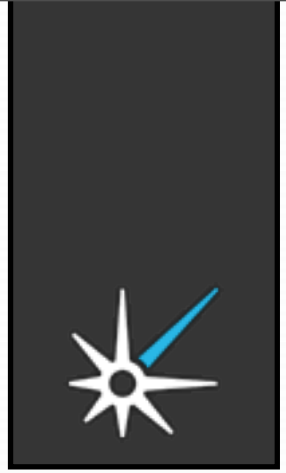
mocks/hardware/FakePwm.c

```
 3 int FakePwm_isRunning = 0;
 4 double FakePwm_dutyCycle = 0.0;
 5 int FakePwm_period = 0;
 6
11 void Pwm_Start(void) { FakePwm_isRunning = 1; }
13 void Pwm_Stop(void) { FakePwm_isRunning = 0; }
15 void Pwm_SetDutyCycle(double percent)
                                  { FakePwm_dutyCycle = percent; }
17 void Pwm_SetPeriod(int us) { FakePwm_period = us; }
```

UI abstraction

The UI uses the same API as the tests

UI Abstraction



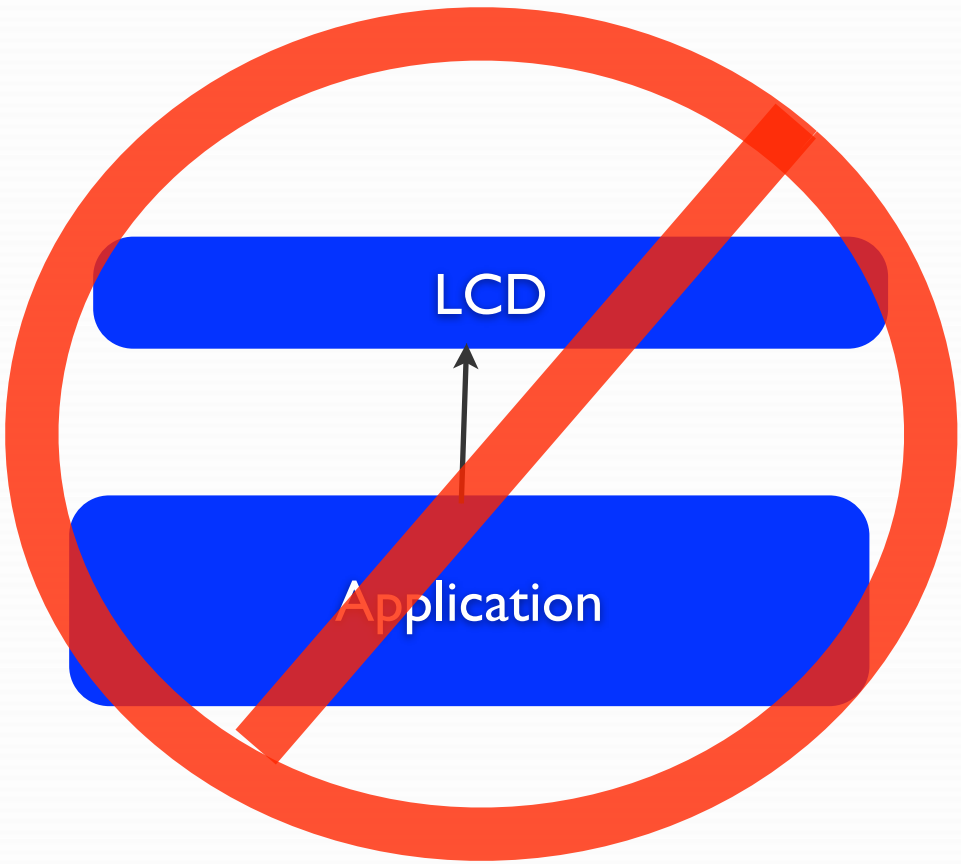
Detect button press …

Api_IncrementTargetSpeed();

….

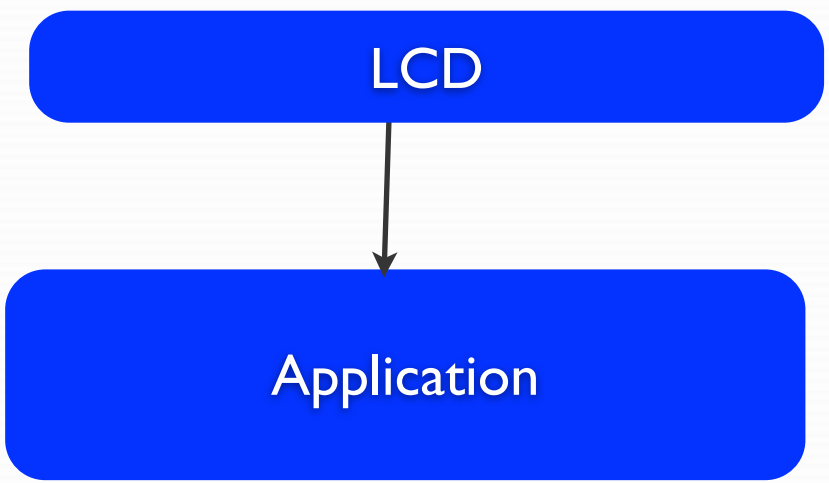
UI Abstraction

What about Displays?

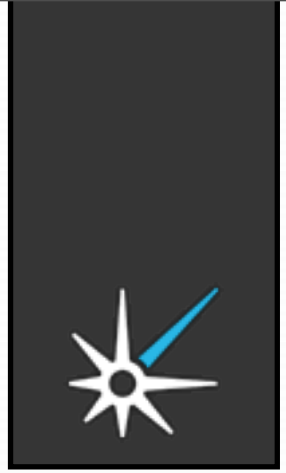


UI Abstraction

Turn the dependency around with an observer



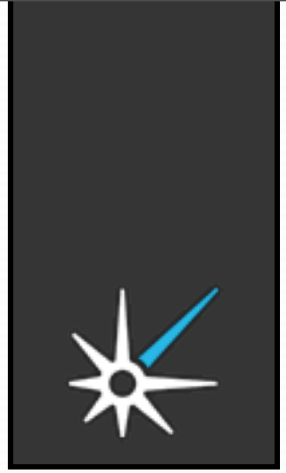
UI Abstraction



Observer Pattern

```
void onSpeedChange(double speed) {
/*update speed display */
}

Api_registerSpeedObserver(&onSpeedChange);
```

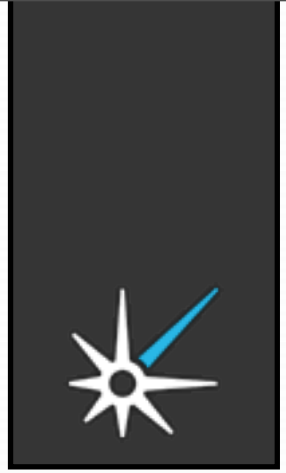
Messaging



Uses the same API as everyone else.

Test message parsing independently.

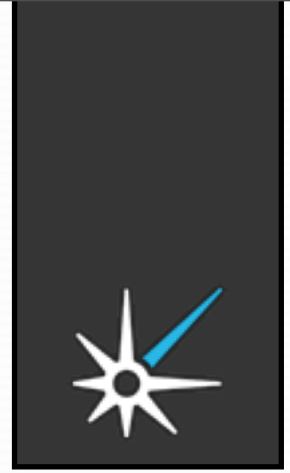
Timers



Can’t wait on real time

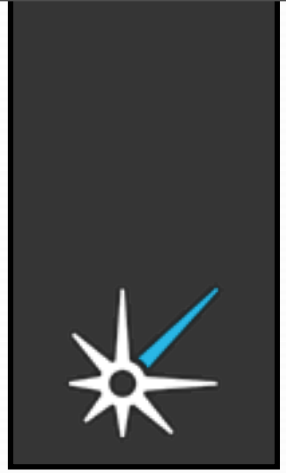
Timer also hit the same API

Interrupts



Generate events in system that can be simulated

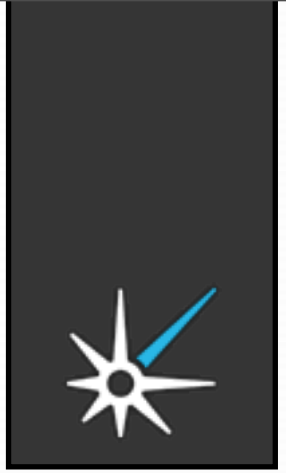
Then what's the point?



Acceptance tests are not entire system tests.

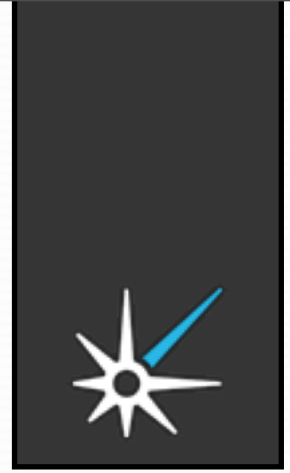
They are a software collaboration medium.

Running on target



Treadmill Cumulative Distance			
speed	time	distance?	time?
1.0	30	[0.0] expected [0.5]	30528

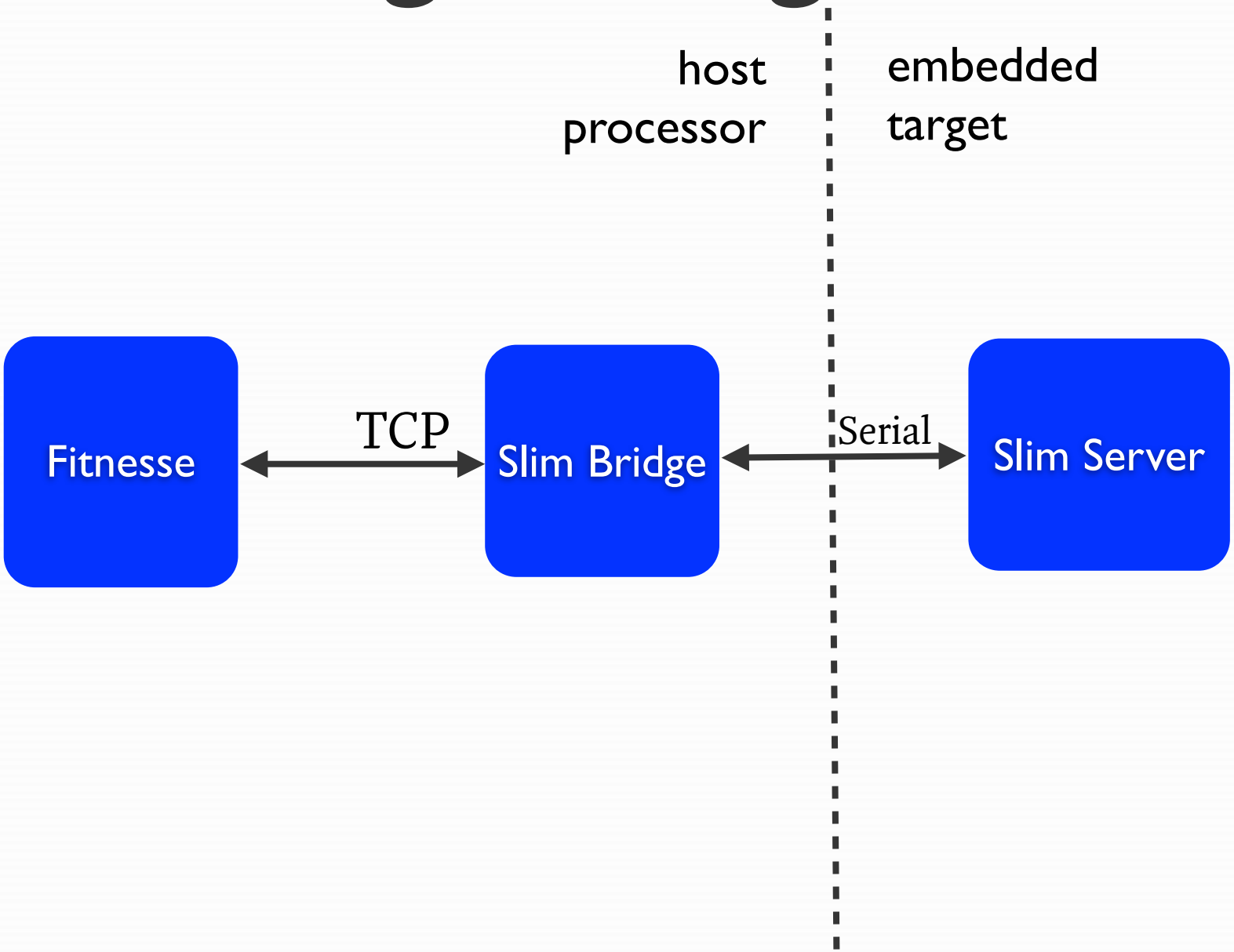
Serial Bridge



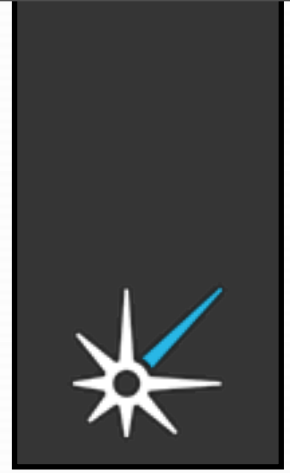
What about running on a target without an ethernet stack?

https://github.com/dougbradbury/slim_bridge

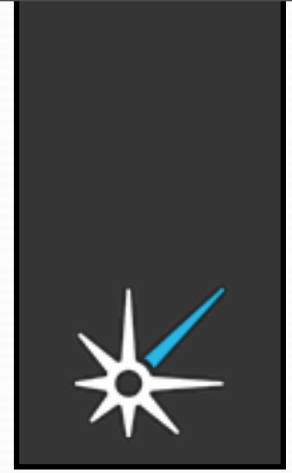
Running on target



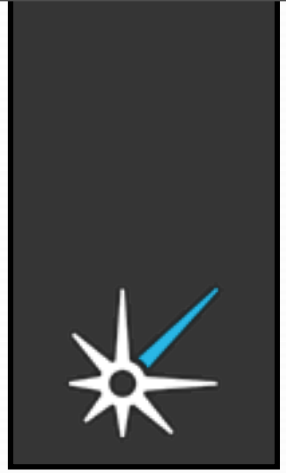
On Target Demo



Porting CSlim



Porting CSlim



Implement a TCP socket Server

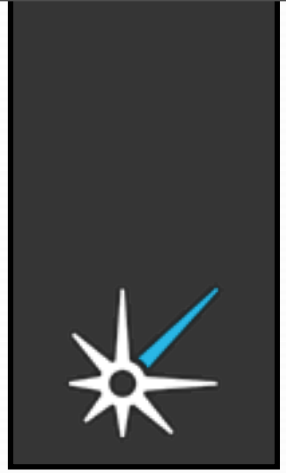
only needs to handle one connection at a time

calls Slim_HandleConnection with pointers to send / recv functions

```
5
6  typedef int(*com_func_t)(void * handle, char * msg, int length);
7  typedef char * (*handler_func_t)(void *, char *);
8
```

```
10 int Slim_HandleConnection(Slim* self, void* comLink, com_func_t send, com_func_t recv);
```

You can help



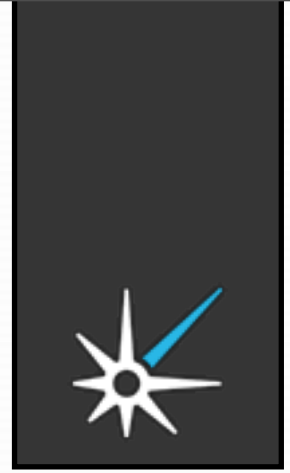
Still some features missing (sut, libraries)

Contribute your port back to the project

Help reduce memory usage

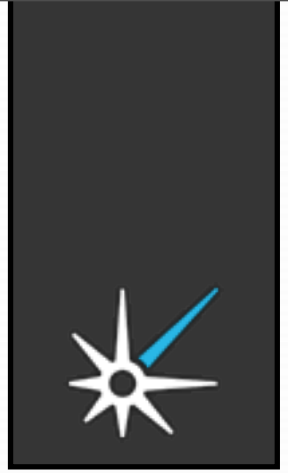
Slim(mer)?

Bonus topics



Multi-parameter script table functions
Returning Lists
Query tables

Multiple Parameters

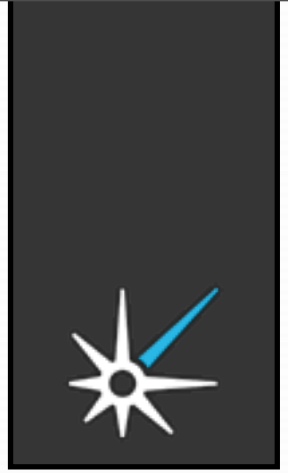


Function name alternates with parameters

script	Treadmill		
given treadmill speed	1.0	and incline	5.0
increment speed			
check	target speed	1.1	

Multiple Parameters

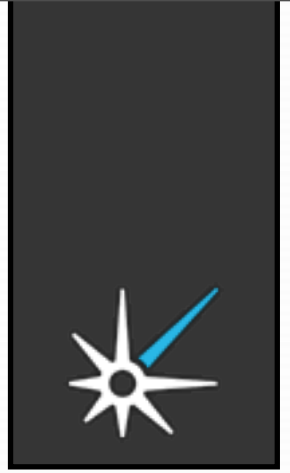
Function name alternates with parameters



script	Treadmill		
given treadmill speed	1.0	and incline	5.0
increment speed			
check	target speed	1.1	

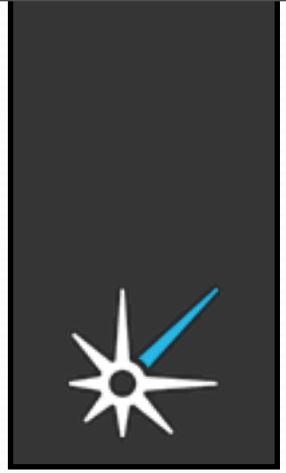
Multiple Parameters

Function name alternates with parameters



pt			Treadmill	
n treadmill speed Metho	givenTreadmillSpeedAndIncline[2]	ot found in Treadmill.	1.0	and
ement speed				
tk			target speed	[0.1

Returning Lists



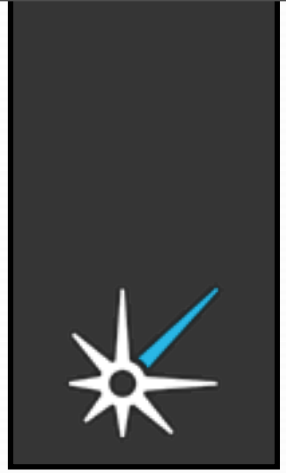
cslim/fixtures/QueryTableExample

```
31  SlimList* id = SlimList_Create();
32  SlimList_AddString(id, "id");
33  SlimList_AddString(id, "1");
34
```

. . .

```
47  self->result = SlimList_Serialize(records);
48
```

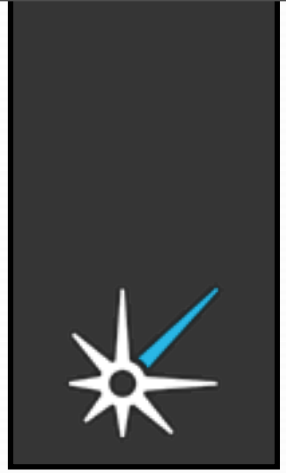
Query Tables



Rows of Records
Order doesn't matter

Query: EmployeePayRecordsRow	
id	pay
1	1000
[2] missing	1050

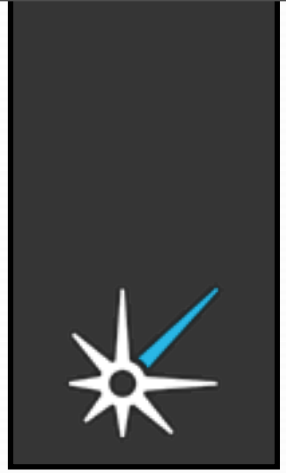
Query Tables



Return a List of records
A record is a list or key, value pairs
A key, value pair is a two element list

```
(
    ((id, 1), (pay, 1000))
    ((id 2), (pay, 1500))
)
```

Query Tables

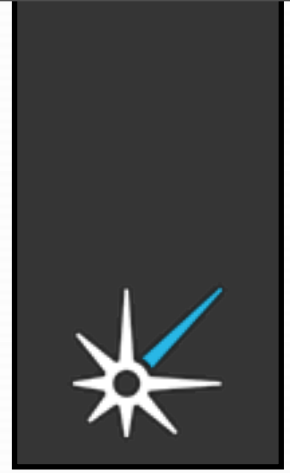


The list must be serialized and returned from the query function.

cslim/fixtures/QueryTableExample.c

```
47  self->result = SlimList_Serialize(records);
48
```

Recommended Reading



Specification by Example - Gojko Adzic

Test Driven Development for Embedded C - James Grenning

Fit - For Developing Software - Mugridge / Cunningham

FitNesse Users Guide