# Bootcamp Session One

## Key Concepts

### Testing Basics

### Automation of Functional Regression Testing at the User Interface

This is the primary focus of the Awetest toolset.

### Script Writing is Coding

#### Hiding the Complexity

#### Trade offs

#### Useful Skills

##### Knowledge of the Application Under Test

##### Knowledge of HTML and DOM structures

##### Experience with Browser Developer Kits

## Awetestlib

Please note: This session is covering Windows only. Setup for Mac will be covered in another session. Except for installation and where things are stored, everything is the same.

### Installation & Setup

#### Ruby 1.8.7

##### Windows

<http://rubyforge.org/frs/download.php/76524/rubyinstaller-1.8.7-p371.exe>

##### Mac

Ruby 1.8.7 comes installed in Mac OS

#### Ruby Devkit

##### Windows

<https://github.com/downloads/oneclick/rubyinstaller/DevKit-tdm-32-4.5.2-20111229-1559-sfx.exe>

Instructions: <https://github.com/oneclick/rubyinstaller/wiki/Development-Kit>

##### Mac

Start here if Xcode is not installed already on your Mac <https://developer.apple.com/xcode/>

#### Setup script

awetestlib regression\_setup automated\_testing

### Cucumber under Awetestlib

#### Setup script

cd automated\_testing  
awetestlib cucumber\_setup bootcamp\_cucumber

## Getting Started with Cucumber

### Key Concepts

#### Easy to understand

Written in natural language. Restricted, of course, but structured to be readable and readily understood.

#### No conditional execution of steps

The first step to fail causes remaining any remaining steps to be skipped.

This makes building Cucumber scripts for deeper, more complex testing scenarios more difficult.

On the other hand it makes translation of many ‘routine’ regression tests into automated scripts much easier, both to write and to map to existing test case documentation.

#### Easy to write new tests once step libraries at critical mass

Building and maintaining the step libraries does require at least one person with some technical bent to learn Watir/Watir Webdriver and Ruby. But the depth of knowledge of those is minimal.

More important to this role is knowledge of the application under test and the structure/api of the platform, i.e., browser, mobile native and hybrid applications. A working knowledge of the browser developer tools is also important

### Installation & Setup

With Awetestlib, Cucumber is already installed and ready to use.

Awetestlib provides a core set of predefined step definitions to help get things started.

### Cucumber Landscape

#### ‘Feature’ file (a.k.a, cucumber script)

#### Step Definitions

<https://github.com/cucumber/cucumber/wiki/Step-Definitions>

##### Basic Structure

Consult the link above for details and examples.

We’ll point these components out in the samples that follow.

###### Preposition/Adverb

Any one of Given, When, Then, And, or But. They are completely interchangeable but must be capitalized.

###### Pattern

A pattern, usually written as a Regular Expression, that uniquely identifies the steps and the variable values that it uses.

###### Proc (Ruby process)

Almost always using the do…end form.

##### Custom Steps

##### Ruby libraries

#### Support files

##### env.rb

#### Awetest Predefined Steps

#### Data repository spreadsheet

#### Folders and Files

##### features

Your scripts (\*.feature) go here along with any data repository spreadsheets.

###### step definitions

Custom step definitions files and predefined\_step.rb live here.

Any additional Ruby library files go here as well.

###### Support

env.rb is located here

### A First Cucumber Script

We’ll call it ‘bootcamp\_0.feature’

First notice that in the directory created above by cucumber\_setup, the step\_definitions directory contains predefined\_steps.rb. We’ll build this first script using just the step definitions included there.

### Bootcamp\_1.feature

### Bootcamp\_2.feature

### Bootcamp\_3.feature