

# Galen Framework

- Responsive Design Look and Feel Automation

- Deepshikha Singh
- Soumyajit Basu

# Why Galen Framework ?

- Layout Testing
- Responsive Design Testing
- Cross Browser Testing

# Pros and Cons with Galen

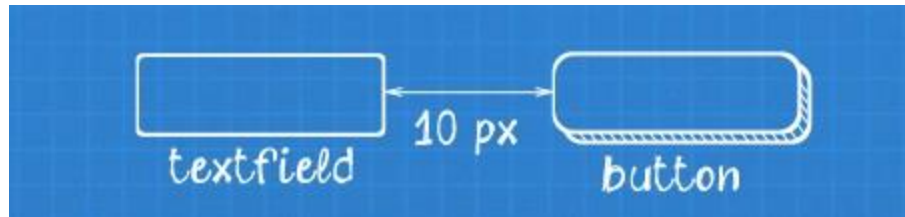
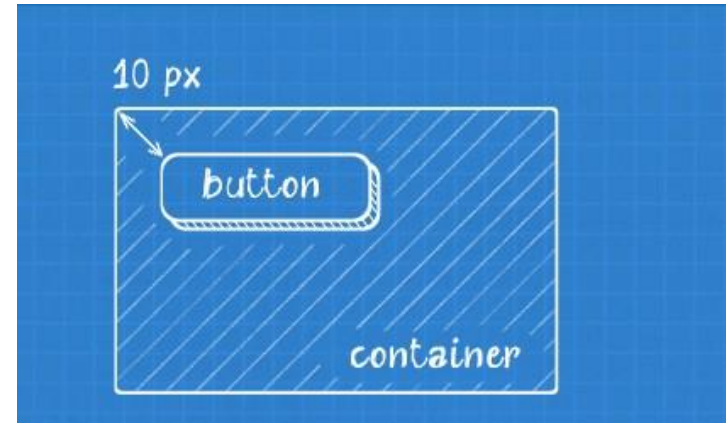
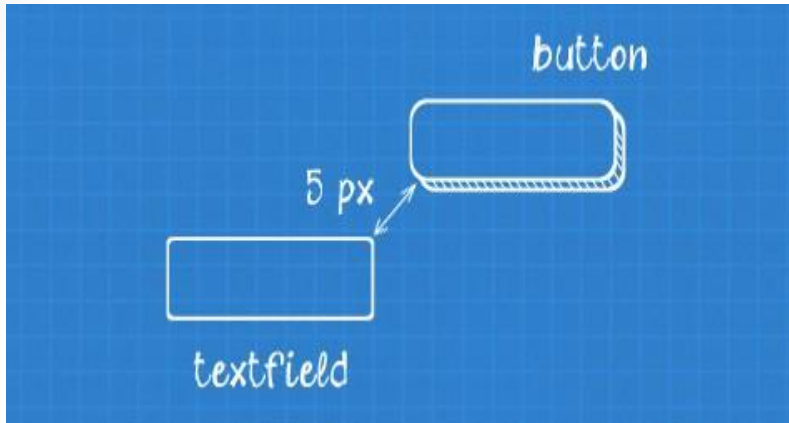
- **Pros:**

- Open Source
- Easy to install
- Error Reporting with screenshots for Pass & Fail
- Easy to Write/Read syntax
- Multiple Browsers support
- Supports Functional Testing

- **Cons:**

- Lack of Test Editor.
- Code maintenance is costly when there are lot of UI changes.

# Sample web page layout



# How it works?

- Describe any layout using special syntax – Spec Language.
- It uses Selenium
  - For interacting with elements on page and getting their locations and dimensions.
- Once it sees that something is wrong
  - Reports the error
  - Makes a screenshot and highlights the misbehaving element on it.
- How it Runs?
  - Open a page in browser
  - Resizes it to specified size
  - Test the layout according to user-defined specs

# Getting Started with Galen

- Define Galen Specs
  - Objects Definition
  - Object specs
- Run individual Spec files
- Inject JavaScript code
- Create Test suite
- Run Test suite

# Objects Definition

## @objects

Header      **id**          header

Logo-box    **css**          .logo-box

Search-bar  **xpath**      //input[@class='searchBar']

- Indentation is 1 to 8 spaces but not tab.
- xpath is not preferable unless css cannot be used.
- Syntax for Object definition depends on the Galen Framework version. ( Ref: <http://galenframework.com/docs/specslang-2.0-difference/>)

# Object Specs

=Test Global Header=

**@ on desktop**

**Header:**

height 40px

width 900px

centered horizontally inside screen

**@ on mobile, desktop**

**Logo-box:**

css color is “rgba(51, 51, 51, 1)”

css background-color is “rgba(120, 190, 32, 1)”

**Search-bar:**

below: header 0px



# Object Specs ( cont....)

- Object Definition
  - <http://galenframework.com/docs/reference-galen-spec-language-guide/#Objectdefinition>
- Multiple objects definition
  - <http://galenframework.com/docs/reference-galen-spec-language-guide/#Multipleobjectsdefinition>
- Object Groups
  - <http://galenframework.com/docs/reference-galen-spec-language-guide/#ObjectGroups>
- Many more:
  - Tagging and Sections
  - Variables
  - Conditional checks ( If, Else)
  - Loops ( For, ForEach)
  - Element locations relative to other elements
    - <http://galenframework.com/docs/reference-galen-spec-language-guide/#SpecsReference>

# Run Individual Spec file

```
galen check homepage.gspec  
  --url "http://example.com"  
  --size "640x480"  
  --javascript "some.js"  
  --include "mobile,all"  
  --exclude "toexclude"  
  --htmlreport "htmlreport-dir"  
  --testngreport "testng-report.xml"  
  --jsonreport "jsonreport-dir"  
  --junitreport "junit-report.xml"
```

## Example:

```
galen check specs/galen.spec --url "http://www.example.com"  
  --size "1366x768" --include "desktop" --htmlreport "reports"
```

# Test Suite

- Grouping Test Specs
- Create Global Variables
- Import Test Suites
- Run Specs on different browsers
- Run in Selenium Grid
- Parameterization
- Disabling tests

# Test Suite Syntax

`@@ set`

`domain http://www.example.com`

`@@table devices`

Device	Tags	Size
Mobile	Mobile	400x600
Tablet	Tablet	720x480
Desktop	Desktop	1364x800

`@@parameterized using devices`

`Home page on ${device} device`

`selenium chrome ${domain} ${size}`

`check specs/galen.spec --include "${tags}"`

# Run Test Suite

- `galen test mytest01.test`
  - `--htmlreport "htmlreport-dir"`
  - `--testngreport "report/testng.xml"`
  - `--jsonreport "jsonreport-dir"`
  - `--junitreport "junit-report.xml"`
  - `--parallel-tests 4`
  - `--filter "Home page on * device"`
- **Example:**
  - `galen test tests/galen.test --htmlreport "reports"`

# Reports

- Report for individual Test cases
- Screenshots for each Test case (Pass/Fail)
- Report for Test suite executed

## Galen Test Report

Tests

Groups

Test	▲ Passed	▲ Failed	▲ Warnings
<a href="#">Home page on Desktop device</a>	1	0	0
<a href="#">Home page on Mobile device</a>	1	0	0
<a href="#">Home page on Tablet device</a>	1	0	0

# Capabilities of Galen

- Testing relative location of elements in web page.
- Checking visible text.
- Integration with Selenium Grid.
- Inject JavaScript into code.
- Color scheme testing.
- Integration with Appium.
- Can be integrated with Browser Stack, Sauce labs.

# Additional Features

- Creating Page Dump
  - Creates information about all test objects on page with image samples.
- Image Comparison
- Java API
- JavaScript API
- Galen Extras
  - Advanced expressions which extends Galen specs language.



# Installation

- System Requirements:
  - Java 1.7 or higher.
  - Text Editor.
  - Drivers for Respective browsers.
- Galen Framework binary
  - <http://galenframework.com/download/>
- Add Galen folder to “PATH” environment variable.

# References

## Tutorials

<http://mindengine.net/>

## Documents

<http://galenframework.com/docs/all/>

## Sample Project

<http://galenframework.com/docs/tutorial-first-project/>

## Galen Spec Language

<http://galenframework.com/docs/reference-galen-spec-language-guide/>

Thank You