TestLeaf
Always Ahead

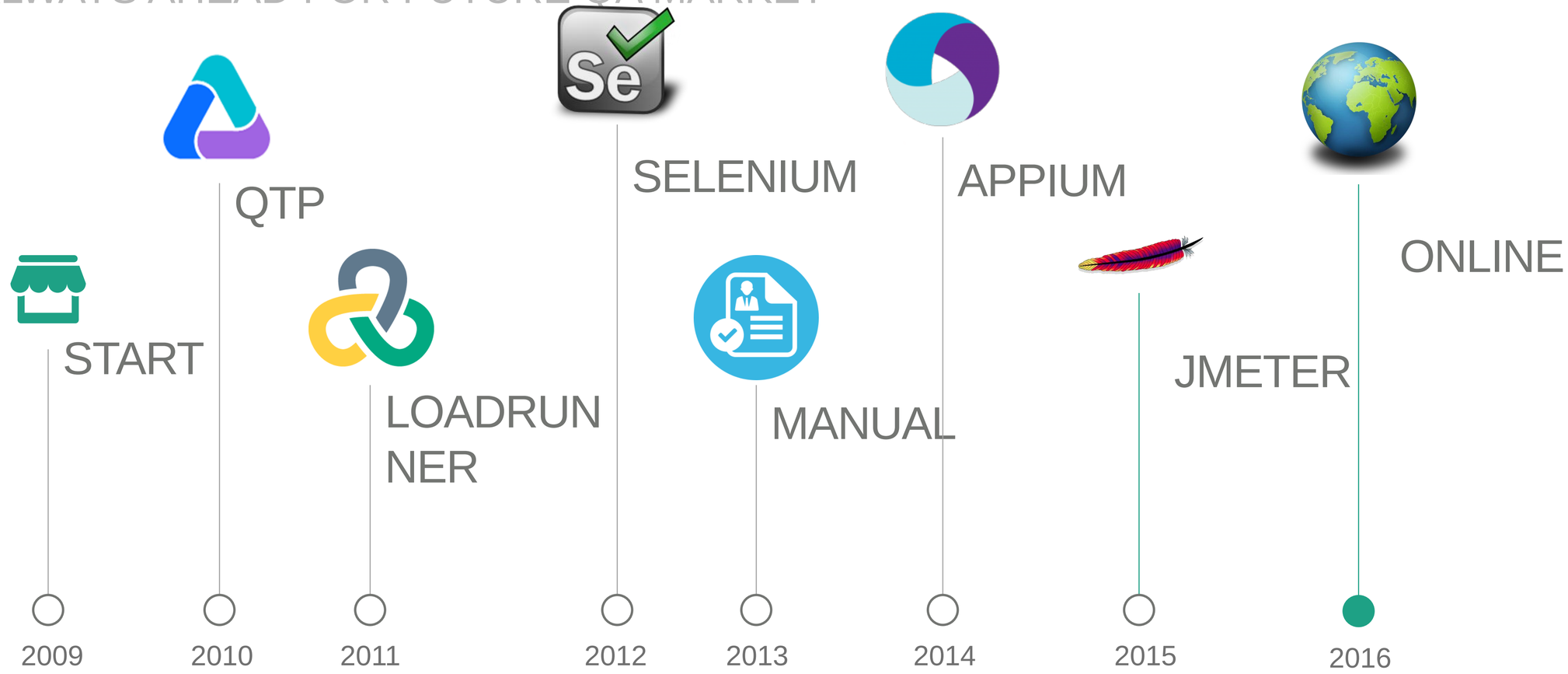
TestLeaf

Always Ahead

ABOUT US

OUR HISTORY FROM THE BEGINNING

ALWAYS AHEAD FOR FUTURE QA MARKET



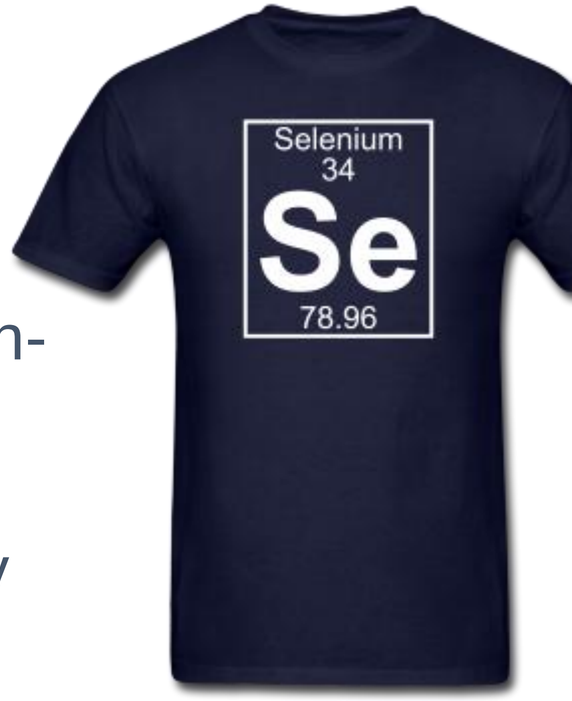
TestLeaf
Always Ahead

Selenium

Dictionary meaning -

the chemical element of atomic number 34, a grey crystalline non-metal with semiconducting properties

Increased dietary selenium intakes reduce the effects of mercury toxicity



What is Selenium Means?

TestLeaf
Always Ahead

SELENIUM PROJECTS

SELENIUM WEBDRIVER

The API that can drive a browser natively either locally or on remote machines.

SELENIUM IDE

Firefox add-on that makes it easy to record and playback tests in Firefox 2+.

SELENIUM GRID

The API that allows tests on many servers at the same time, cutting down on the time.

SELENIUM RC

Client/server system that allows you to control web browsers locally or on other computers, using almost any programming language [Deprecated in Selenium 3.X onwards)

TestLeaf
Always Ahead

Se History

Selenium History

- 2004 - Jason Huggins was testing an internal application at Thoughtworks.

- That library eventually became Selenium Core, which underlies all the functionality of Selenium Remote Control (RC) and Selenium IDE.

- In 2006, Simon Stewart started work on a project – WebDriver. Google had long been a heavy user of Selenium, but testers had to work around the limitations of the product.

- Simon Stewart, the creator of WebDriver, in a joint email to the WebDriver and Selenium community on August 6, 2009.

  "Why the projects merged?
  Partly because webdriver addresses some shortcomings in selenium (by being able to bypass the JS sandbox, for example. And we've got a gorgeous API), partly because selenium addresses some shortcomings in webdriver (such as supporting a broader range of browsers) and partly because the main selenium contributors and I felt that it was the best way to offer users the best possible framework."`

TestLeaf
Always Ahead

Selenium Team

- **Simon Stewart**: Original WebDriver developer and leading the Selenium 2 effort.
- **Julian Harty**: Dabbled with WebDriver since 2007 mainly finding ways to make the code real and useful by testing it, and by documenting it. Currently working at eBay to find ways to make software testing more efficient and effective.
- **Jari Bakken**: Has been working on WebDriver since late 2009, developed and now maintaining all things Ruby.
- **David Burns**: Has been working with Selenium 1 for about 4 years and with WebDriver since the beginning of 2010 and now maintaining the .NET and Python bindings.
- **Anthony Long**: Has been working with Selenium since 2008, and is currently working to improve the Selenium Python bindings.
- **Jim Evans**: Started working with the WebDriver and Selenium since the end of 2009, working mostly on the .NET bindings.

TestLeaf
Always Ahead

What We'll Cover Today

- Selenium 3.x – Heads Up
- Html tags and structure
- Begin WebDriver coding using Core Java
- Java basics

TestLeaf
Always Ahead

- Commercial (very costly)
- Supports Multi applications
- Works only with VBScript / JavaScript
- Can Integrate with ALM
- Does not script / work in MAC, Linux

- Open-source (no license cost)
- Only web based apps
- Several languages (API in Java)
- No integration with Test management tools
- Works with Windows, MAC, Linux

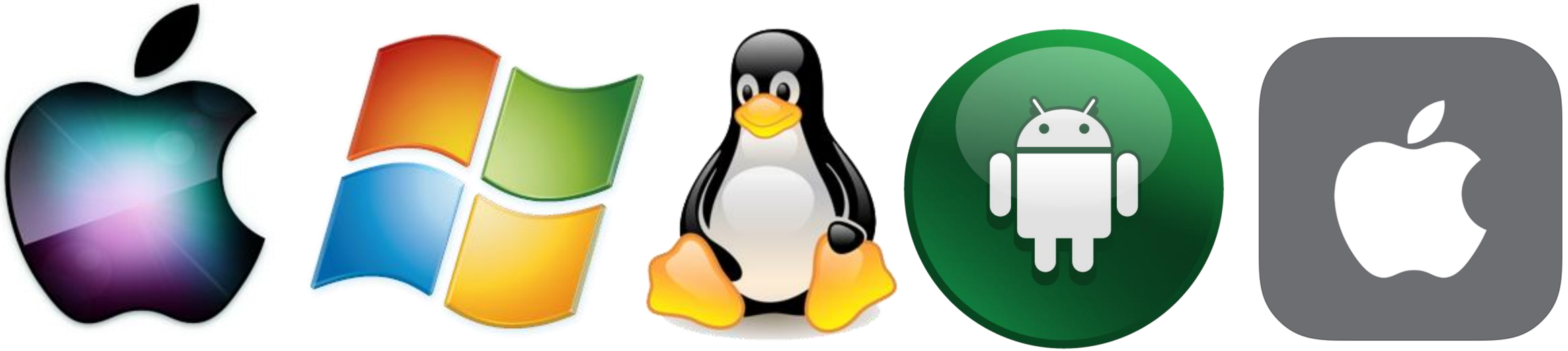
Tools Comparison

TestLeaf
Always Ahead

Browsers Supported

TestLeaf
Always Ahead

Platform/OS Supported



TestLeaf
Always Ahead

Languages Supported

TestLeaf
Always Ahead

TestLeaf

Always Ahead

HTML TAGS

Some of the important HTML Tags that we need to know

The most common used for Selenium Locators

Tag	Description
<html>	Defines the root of the web page
<a>	Defines a hyperlink
<button>	Defines a clickable button
	Defines a section in a document
<iframe>	Defines an inline frame
<img>	Defines an image
<input>	Defines an input control
<option>	Defines an option in a drop-down list
<select>	Defines a drop-down list
	Defines a section in a document
<table>	Defines a table

TestLeaf
Always Ahead

Locators

Locators	IDE	WebDriver	When do you use?
Id	id	id	*Always (Most preferred)*
Name	name	name	*If “id” does not exist*
Link	link	linkText	*If it is a link*
Tag name	X	tagName	*For collection of objects*
Class name	X	className	*When class is unique w/o spaces*
Xpath	xpath	xpath	*If none of the above, works*
Style sheet	css	cssSelector	*This is the last option you have!*

Firefox	Chrome	IE	Notes
IDE	Chrome Developer Tools	IE Developer Options	*Chrome tool is quite easy; If incase your app does not work other than IE, prefer to use Macro (but that is not a supported utility)*
FireBug	FireBug	Macro	
Inspect	Xpath Finder		

Locators & Tools

TestLeaf
Always Ahead

Locators	WebDriver	When you cannot use?
Id	id	Numbers
Name	name	Duplicate
Link	linkText	Duplicate
Tag name	tagName	Duplicate
Class name	className	White spaces, Duplicate
Xpath	xpath	Xpath can change
Style sheet	cssSelector	CSS can change

TestLeaf
Always Ahead

Locating by Id

This type of locator is more limited than the identifier locator type, but also more explicit. Use this when you know an element's id attribute.

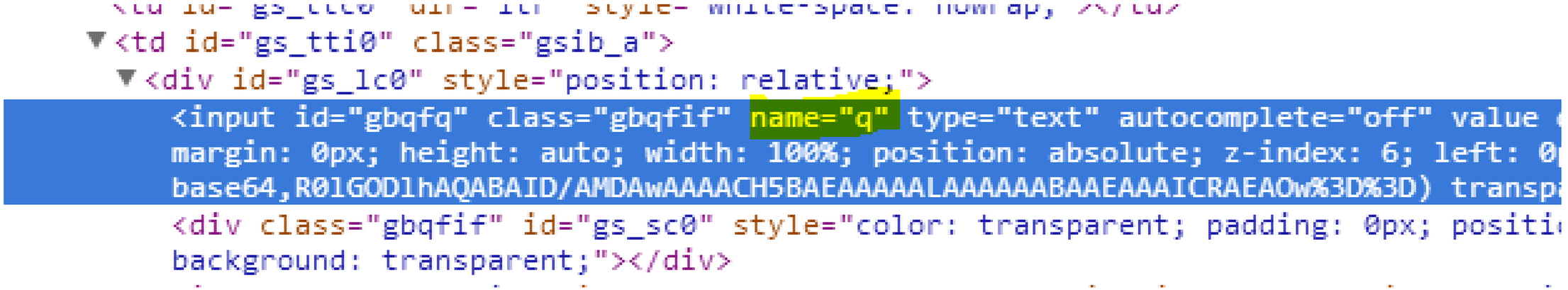
```
<td id="gs_ttc0" dir="ltr" style="white-space: nowrap;"></td>
<td id="gs_tti0" class="gsib_a">
<div id="gs_lc0" style="position: relative;">
<input id="gbqfq" class="gbqfif" name="q" type="text" autocomplete="off" value dir="
margin: 0px; height: auto; width: 100%; position: absolute; z-index: 6; left: 0px; c
base64,R0lGODlhAQABAID/AMDAwAAAACH5BAEAAAAALAAAAAABAAEAAAICRAEAOw%3D%3D) transparent
<div class="gbqfif" id="gs_sc0" style="color: transparent; padding: 0px; position: a
background: transparent;"></div>
```

If there are more than one Id with same value present, then the first match will be used

TestLeaf
Always Ahead

Locating by Name

The name locator type will locate the first element with a matching name attribute. If multiple elements have the same value for a name attribute, then you can use filters to further refine your location strategy.



If there are more than one same name present, then the first match will be used

TestLeaf
Always Ahead

Locating by Link Text

Method of locating a hyperlink in your web page by using the text of the link.

If two links with the same text are present, then the first match will be used

By Partial Link Text

Find the link element with partial matching visible text.

```
▶ <div class="gb_e gb_f gb_g gb_Ic">…</div>
▼ <div class="gb_e gb_f">
    <a class="gb_d" href="https://mail.google.com/mail/?tab=wm" data-pid="23" data-ved="0CAMQwi4oAQ">Gmail</a>
  </div>
▶ <div class="gb_e gb_f">…</div>
```

TestLeaf
Always Ahead

Locating by tag name

If none of the above locator helps you or if you want to find the collection of the (specific) objects in a given page, use “tagname”

```
<div class="gb_sc gb_f gb_Ic gb_Ac gb_h">
  <div class="gb_e gb_f gb_g gb_Ic">
    <a class="gb_d gb_f" href="https://plus.google.com/?gpsrc=ogpy0&tab=wX" data-pid="119" data-ved="0CAIQwi4oAA">+You</a>
  </div>
  <div class="gb_e gb_f">
    <a class="gb_d" href="https://mail.google.com/mail/?tab=wm" data-pid="23" data-ved="0CAMQwi4oAQ">Gmail</a>
  </div>
  <div class="gb_e gb_f">
    <a class="gb_d" href="https://www.google.co.uk/imghp?hl=en&tab=wi&ei=mLntU8ajA8aThQfYq4D4Dw&ved=0CAQQqi4oAg" data-pid="2">Images</a>
  </div>
```

Get All links in a page

TestLeaf
Always Ahead

Locating by class name

If none of the above locator helps you or if you want to find the collection of the (specific) objects in a given page, use “class name”

```
<div class="gb_sc gb_f gb_Ic gb_Ac gb_h">
  <div class="gb_e gb_f gb_g gb_Ic">
    <a class="gb_d gb_f" href="https://plus.google.com/?gpsrc=ogpy0&tab=
  </div>
  <div class="gb_e gb_f">
    <a class="gb_d" href="https://mail.google.com/mail/?tab=wm" data-pid
  </div>
  <div class="gb_e gb_f">
    <a class="gb_d" href="https://www.google.co.uk/imghp?hl=en&tab=wi&ei
  </div>
</div>
```

Get specific type of links

TestLeaf
Always Ahead

Question 1

If there are more than matching object for the identifier, locator does what?

1. Throws error saying that there are multiple matches
2. The last matching object is located
3. The first matching object is located
4. None is located with no error thrown

TestLeaf
Always Ahead

Question 2

What is the most preferred locator in this given list for text field?

1. CSS
2. id
3. name
4. xpath

TestLeaf
Always Ahead

Question 3

What is the least preferred locator in this given list for text field?

1. CSS
2. id
3. name
4. xpath

TestLeaf
Always Ahead

Question 4

If a portion of link changes dynamically, what is the best way to find it?

1. Linktext
2. tagName
3. Partiallinktext
4. className

TestLeaf
Always Ahead

CONTACT US

TestLeaf

Always Ahead

KEEP IN TOUCH WITH US

WE ARE HERE TO ASSIST YOU

Address

22, First Floor, College Road, Nanganallur
Chennai, India , 600061

Contact Info

Email: sakthivel@testleaf.com
Information: info@testleaf.com
Support: contact@testleaf.com

Telephone

Office Phone: +91 (44) 4554 4246
Sales: +91 (950) 000 6373
Support: +91 (984) 057 7905

THANKS FOR LISTENING