

S. No.

Date

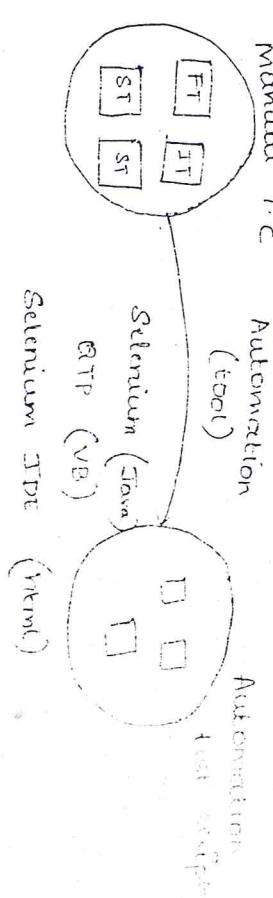
Title

Page No.

Teacher's
SignatureAutomation

The process of converting the manual test cases to test scripts with the help of tool & execution language

Bhima
Shankar
BS



Note

- * Automation is a process of converting manual test to test scripts using some tools like selenium language.
- * Two types of automation tool exist in market

- ① Functional automation tool
- ② Non-functional automation tool

Automation

Functional

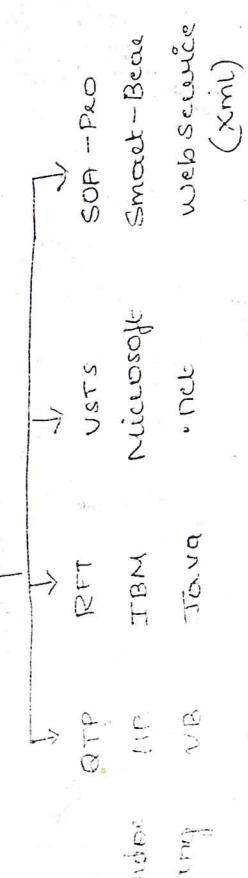
Non-Functional

- ↓
- ✓ Commercial tool
 - Open Source tool
 - (Available free in market)

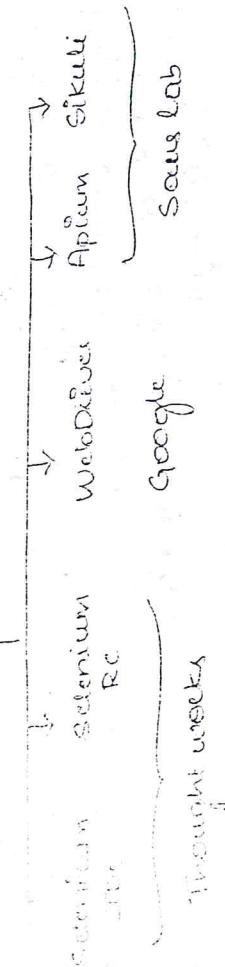
Management tool / source control

Performance testing
Checking the functionality of the application according
to the requirement

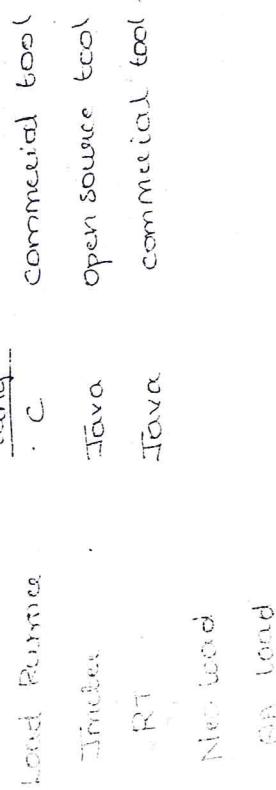
Commercial tool



Open Source tool



New Generation testing
Checking non functionality of the application like
Performance, compatibility, Usability, Lang.



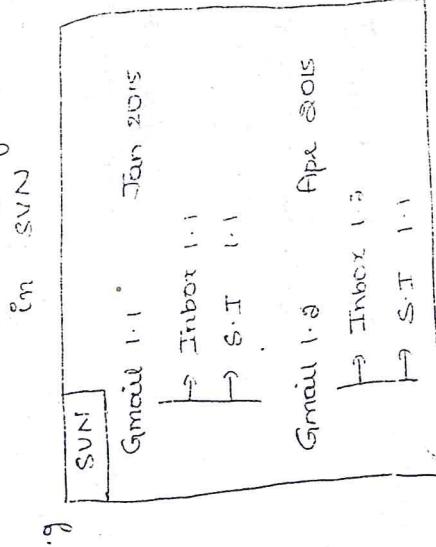
Configuration Management tool

SVN
Perforce
CVS
VSS

Functional Testing

- 1. NTE
- 2. Dev
- 3. PTE

NTE
BA collects user from customer & divides it into



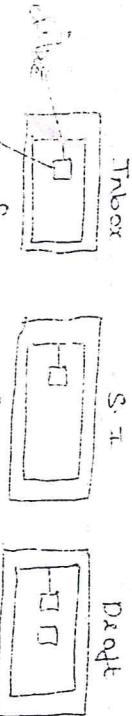
Gmail 1.1 → App 2.0
→ Inbox 1.1
→ S.I 1.1

- * In case of manual testing, SVN is used to maintain the requirement & specification & also we can keep track of version of the gen (hence called known as Version Control tool)

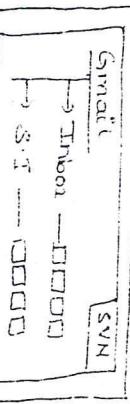
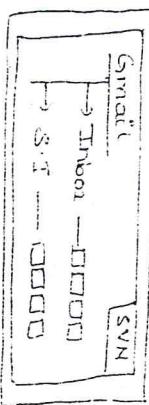
Developer

In case of Development, SVN is used to maintain the source code of the application in one place

SVN Review



Java
Class



- Manual Test steps test cases in Quality Center
- Configuration Test Script test scripts in SVN
- Test case Automation, SVN Server is used to maintain
- Test execution test script in one place
- SVN Sub version
- CVS - Concurrent Version System
- Visual source code
- Executable

filed → Application Life Management
(Advanced version etc)

Selenium Course Content

Selenium IDE

HTML

JavaScript

Object Identification

File base

Image frame

With Diver

→ Web Driver API

→ Web Element API

→ Select

→ WebDriver wait statement

→ Actions

→ Working with multiple web elements

→ Windows handling

→ Frame handling

→ Working with multiple browser

→ File attachment & download

Unit testing framework tool

→ testNG

→ Junit

Frame work implementation

→ hybrid driver frame work

db connection

Screen shots

SVN

Maven [build testing tool]

Selenium Grid

HTML

JavaScript

Object Identification

File base

Image frame

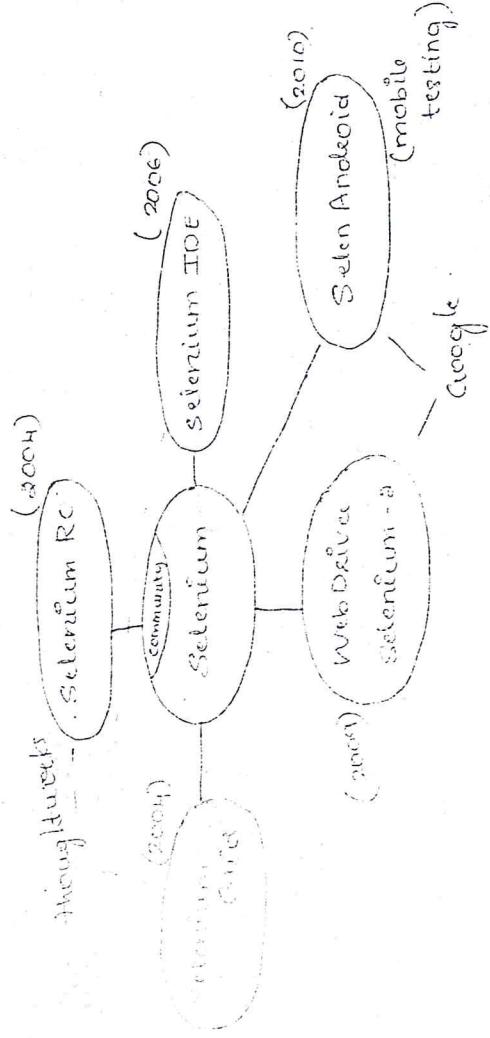
With Diver

→ Web Driver API

Selenium

Selenium is not a tool it is a community maintained as a open source community by Google & selenium works community

[<http://docs.seleniumhq.org>]



Google's thought works implemented these tool and they up-converted this tool in the <http://docs.seleniumhq.org> url.

Selenium supports only web Based Application [The Application which uses browser to open is called web based application] (<http://docs.seleniumhq.org>)

It supports stand alone application (<http://docs.seleniumhq.org>)

Selenium IDE is implemented as plug-in for chrome browser

Plug-in (extension) provides free for user
(This is the major drawback of Selenium IDE)

Selenium IDE

→ Installation steps for Selenium IDE

1. Google → search for download selenium IDE
2. click on first link , and navigate to selenium homepage (www.seleniumhq.org)
3. At the bottom of homepage , found the Selenium IDE adission and , click on the Allow
4. will get a popup & click on Allow
5. will get another popup & again click on Allow
6. click on Restart browser
→ To open Selenium IDE , goto tools in Firefox → tool (Firefox → tool)
choose Developer to visible tool if not visible

- * Selenium IDE is a UI based record and play back tool
- * whenever we open Selenium IDE , by default IDE in recording mode (i.e) IDE is ready to record the user action performed on web element only .

web elements → elements provided in the web page

so we can write a command wait for page loaded
because we can write a command click and
wait → by default it'll command waits
for 500ms.

so we can open source tool which is
available on plug-in for fire fox browser
this browser is used to view html source code for
any specific web element in UI. I select browser

so we can click on element selecting
so we can find link to firebug community
so we can click to firebox
and then click on this tab now
so open firebug, click on bug symbol

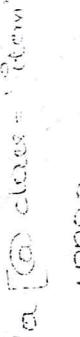
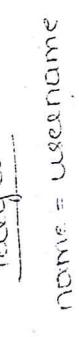
so we can click on bug symbol
so we can see html source code for an element
so we can inspect feature. 

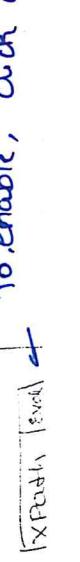
so we can implement user input in firebug
so we can also used to view html source code
so we can click on this link to firebug community
so we can click on firebox

so we can click on this link to firebox
so we can click on this link to firebox
so we can click on this link to firebox
so we can click on this link to firebox
so we can click on this link to firebox
so we can click on this link to firebox

so we can write a command wait for page loaded
because we can write a command click and
wait → by default it'll command waits
for 500ms.

e.g. Command → Target → Value → execute
name = username
name = pwd
id = loginButton

type → name = username
type → name = pwd
click → id = loginButton
10000
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
(so)

Target → Value → execute
name = username
name = pwd
id = loginButton
click →
10000
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 
Wait for Page To Load
click → 

to enable, click on finepath button in firebug [xpath] tool

Minimum → min & max xpath level

* Commands used used to perform specific task

web element

Target / locator

target
Image
button
checkbox
radio
element

* Element available in web application is known as

Web elements

Writing Selenium IDE test script, it duplicates user steps.

- (1) commands
- (2) Target or locator
- (3) Value

Commands

Web elements

commands

Select type	type / sendkeys	click / clickAndWait	Select-Selectee (based on id, name, class, value)	click	radio button	checkbox
Button						

→ To identify any web element, we need some path based on id, name, xpath, class, value etc.

→ As we find html source code using Right Click open html source code] we use most suitable exact location so we use following id

html source code

To write html source code (i.e.) `<p>`

Syntax : `//html tag [tag attribute = "value"]`

→ If we open a link and we have to click for some time [to synchronize] then use `sleep` command after actions.

Select command

but before we do

some action

Get text()

(2002 - 2010)

QTP Application

Commercial

Tool

Review

→

browsers

Support

IE
mozilla
chrome

→

long
support

→ VB script

→

OS
Support → windows

→

UI based (Record & Playback tool)

Execution is required because,

Application → web based App

Support

• supports all browsers

• supports all OS

• is open source.

Note

Selenium IDE uses HTML language to generate test scripts.

Selenium IDE is also known as HTML language.

Selenium IDE does not have capability to launch new browser and run the test (which can open only existing already opened browser).

While running key browser operation cannot be followed.

Keyboard operation will be succeeded whenever it is performed on visible elements.

If keyboard operation like press enter, etc., it is performed it doesn't succeeds.

* In all selenium tools, password encoding in the decoding feature is not available.

Batch execution / Suite execution

Collection of multiple test script is called suite. Batch, execute multiple test script in one go. Click is called batch execution.

Steps to create a suite

1. Create a new test
File → New Test Case

2. Save a test writer.html extension
3. Create a new suite.

File → New Test Suite

4. Add multiple test cases to test suite

TestCase	

TestCase	

→ Right click

↳ Add Test Case

↑ Function

{ Test Case }

5. Save test suite

File → Save As

Notes

- Start test suite
- Open existing test suite, when suite selected write multiple test script
- Open current active test (only one test script)

IDE provides a feature available in IDE which allows to verify the expected request with actual results

Check points

IDE provides two types of check points

- Actual check point
- Expected check point

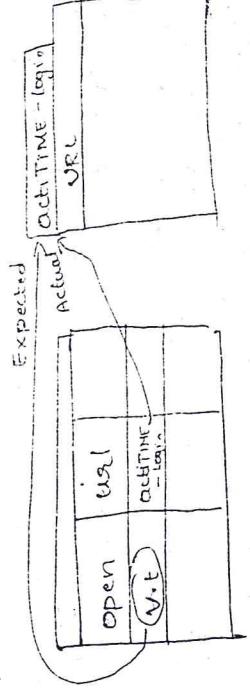
In actual check point we can succeed only after the expected request so we use check points for verification of expected result. These checkpoints have to be added manually & cannot be succeeded.

Title check point

There are two commands available in Title check point

- Verify title
- Set title

Action	Expected request
Enter URL & navigate to login page	Login page MBD
Login to FPP	Home page MBD
Logout	Login page



Verify title, If check step fails it will continue to next
execution of other & step

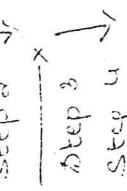
close title, if check step fails will stop here execution

Note

* Title check point is used to verify title of title with the UT title

case1: when verify check point fails

In this section IDE, we can succeed only after the expected request so we use check points for verification of expected result. These checkpoints have to be added manually & cannot be succeeded.



IDE generates a success message for the current step and continues execution with remaining test steps

Close : when assert check point fails

Step 1

Step 2

Step 3

Step 4

IDE generate error message for current step
and stop the current test script execution

Close : when assert check point fails in suite

Execution



IDE generates error message and stop the

current test script execution, and continue

execution with remaining test script

Text check point

Text check point is used to verify any visible
text available in UI. These are two common
examples in text checkpoint.

① Verify test checkpoint

② Assert test checkpoint

Using text checkpoint

It will Wait to Verify command

Navigate to entire html document. Verify the
expected answer remains the required

based on the locators capture entire page

shot

Command

Target

Result

open

/login.dio

actitime-login

Verify Title

type

Name = username

value

type

Name = pwd

value

click

css = input [type = "button"]

Verify Text

//span [class = 'sub - option-link']

Capture Entire Page

c:\Users\ASUS-1AB-381\Pictures\OCM3\Business

click And wait

css = input .logcatring

Disadvantage of Selenium IDE

* Selenium IDE works only with file based
* Selenium IDE cannot launch new browser and
run the test case

* conditional & loop statements can't be done
in a IDE

* move and keyboard operations can't be done

* using Selenium IDE connect customized application

application

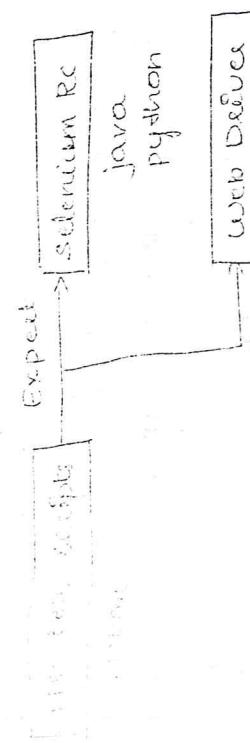
Note

* Ajax : The Application which contains dynamic
Dynamic objects is called Ajax

HTML, JavaScript, CSS, CSS Preprocessors

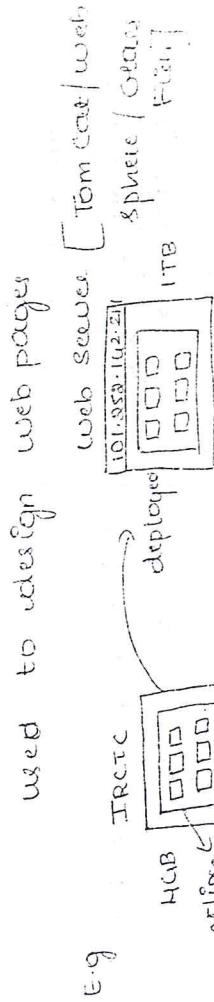
Advantage of Selenium IDE

- * Selenium IDE is UI based succeed and play back tool
- * Selenium + selenium IDE tool we can connect IDE to steps to selenium RC (ex) Web Driver

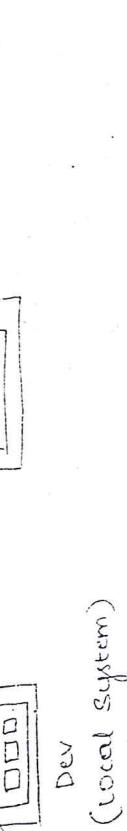


Advantage
1. easy to write and easy to understand

HTML → hyper text mark-up language



used to design web pages

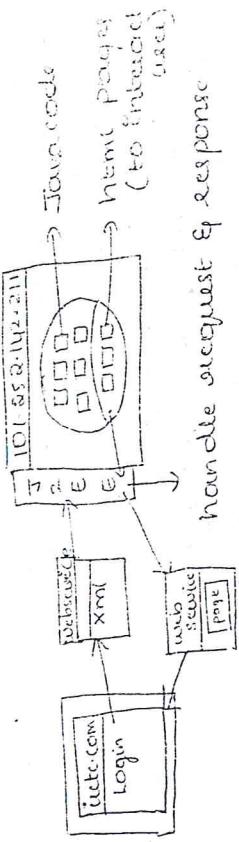


(local system)

C:\> ping lecture.com
Global URL

c:\> ping lecture.com

Dev



Web Server [Tomcat / web, Apache / GlassFish]
[192.168.1.102:8080]
deployed [Index]
[192.168.1.103]

- * HTML [hyper text mark up language] It is used to design web pages, web pages use intermixed HTML client and web service (ex) user and source code.
- * HTML is a collection of HTML tags, tags are always enclosed using single braces (< >)
 - e.g. of pre-defined HTML tags

< html >
< title >

<body>

<input>

<a>

<table>

<tr>

<td>

Each and every opened html tag should have

end html tag which is mandatory

e.g. <input> </input>

(ex)

<input />

<title>

<head>

<body>

<table>

<tr>

<td>

Ex:

```
<html>
  <head>
    <title> Amazon - Login </title>
  </head>
  <body>
    <h1> welcome to Amazon </h1>
    <label> userame : </label>
    <input type = 'text' id = 'user' name = 'username' placeholder = 'Enter your Email' />
    <label> password : </label>
    <input type = 'text' id = 'password' name = 'password' />
    <input type = 'button' value = 'Login' />
    <input type = 'button' value = 'Logout' name = 'Logout' />
  </body>
</html>
```

For login box , html tag used as input

Input → type = text [] } → based on what type of web elements

will suggest O . J.

Mandatory

Enter your Email

to execute body in code tag
the attribute placeholder

e.g.

<input type = 'text' id = 'user' name = 'username' placeholder = 'Enter your Email' />

Free header , use same html tag in

el element	input, type	name & value	additional attr	value
	attr	name	attr	value
input	type = text / password	name	value	custom value
				color
button	type = button / submit	name	value	" "
				name should be same for radio buttons
checkbox	input type = checkbox	name	value	" "
				name & value should be same for radio buttons
radio	input type = radio	name	value	" "
				name & value should be same for radio buttons
form	input type = file	name	value	" "
				name & value should be same for radio buttons
img	src = path of image	name	value	" "
				width height
select	option	name	value	" "
				multiple
multiple	option	name	value	" "
				multiple = true
table	tbody	name	value	" "
	tr			
tr	td			

checkbox

Eg :

<input type = 'checkbox' name = 'gender' value = 'male' />

Eg :

<input type = 'checkbox' name = 'gender' value = 'female' checked />

: male → checked

: Female

Male
 Female

→ same name

→ checked

To select any one of these option, go for single button.

click name

Male
 Female

X

Hence name should be same for radio button.

To select more than one option, use checkbox.

C
 C++
 Testing
 Java

edit box → type = 'password' means



The value entered will be

obscured

att is href = null

To design link, href tag used in 'a' & mandatory

This will be created but it will not be visible on the page so to provide visibility

text in html code will be like

 Flipkart

↓
back end attributes

for web element

visible
text for web
element

Image

Since html code is
image in same folder

If diff folder, path shld
be mentioned.

dropdown

<select name = 'sel' multiple = 'true' />

↓
it only creates a empty dropdown to

provide values use option tag



</select>

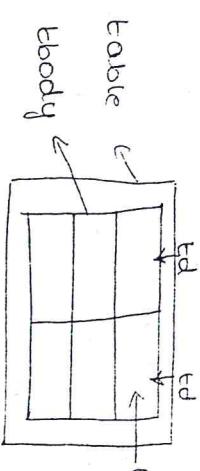
webtable

table → to create table

tbody → to body

tr → to create row

td → to create column



1	India
2	Australia

<option> Australia </option>
<option> Pakistan </option>

<select>

<option> India </option>
It only single
select & single
dropdown

<option> USA </option>

e.g

<select name = 'sel' multiple = 'true' />

<option> India </option>

<option> USA </option>

<option> Australia </option>

<option> Pakistan </option>

1	India
2	Australia

<select name = 'sel' />

<option> India </option>

It only single
select & single
dropdown

- 1. id
 - 2. name
 - 3. x-path
 - 4. CSS- selector
- * id locator is used to identify any web element using id back end attribute
-
- <td> <input type='checkbox' /> </td>
 <td> msg 3 </td>
- <td> <input type='checkbox' /> </td>
 <td> msg 3 </td>
- <td> <input type='checkbox' /> </td>
- SELECTORS IDENTIFICATION
- 1. id
 - 2. name
 - 3. x-path
 - 4. CSS- selector
- * name locator is used to identify any web element using name back end attribute written within <>
- * x-path is used to identify any web element using any backend attribute (id, class, text)
- * CSS- selector is used to identify the web element using @ class (@) or id attribute (@) → attribute
- NOTE
- * In real time application, most of the web element will never find id & name (@) unique attribute. In such cases we have to go for x-path (@) CSS- selector to find the object while writing x-path.
 - // symbolize → go to entire html source code
- [] → go to parent
 [] → provide web element backend attribute
- 1 → go to child
- <html>
 <body> Amazon- login </body>
 <title> Welcome to Amazon </title>
<input type='text' name='username' />
<input type='text' name='password' />
<input type='checkbox' />
</html>
- The element location in all selenium tools can be done by locators, because one written looking at the html source code.
- What is the location available in this

- ③ → attribute symbol
- → match any html tag.

therefore two types of x-path.

- absolute x-path

- relative x-path

Absolute x-path



```

<html>
  <body> Amazon - login </body>
</html>

```

- parent
- preceding

```

<div>
  <input type="text" name="username">
  <input type="password" name="password">
</div>

```

</body>

,

when both input fields changed then leads to failure.

```

<html>
  <body>
    <div>
      <input type="text" name="username">
      <input type="password" name="password">
    </div>
  </body>
</html>

```

Absolute x-path

Final result

```

// input[@name = 'username']
// input[@name = 'password']

```

When user we expect using web element using
the first approach always generates absolute

* Absolute x-path should not be used in selenium script because whenever object changes (ex) sequence changes, absolute x-path fails to identify the object.

eg if seq changes,



username
password

Absolute

```

<html>
  <body>
    <div>
      <input type="text" name="username">
      <input type="password" name="password">
    </div>
  </body>
</html>

```

Relative

x-path

wen absolute path is used for this selection

the IDE shows failure.

dev tag is used to create in next line.

wen absolute path is used for this selection the IDE shows failure.

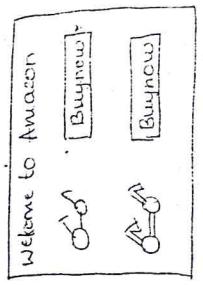
Relative x-path

Syntax : // newtag[@attr = 'value']
for username , // input[@name = 'usr']

value → X-path
attribute → attribute
radio → attribute
checkbox → attribute
multiple checkboxes

value → X-path
`//input[@name = 'username' and @type = 'text']`

Case 2:
When similar object present in UI
E.g.
Screenshot screenshot changes . evaluate X-path
never fails .



Screenshot X-path is written directly to the
HTML element using its back end attriute as
evaluated relative X-path .
Relative X-path never fail to identify the object,
since browser object location changes .

`<input type="checkbox" name="checkbox" value="checkbox" />`
Relative X-path always elements not able to identify with
the attribute , in such cases we can provide
multiple attribute using 'and' keyword .

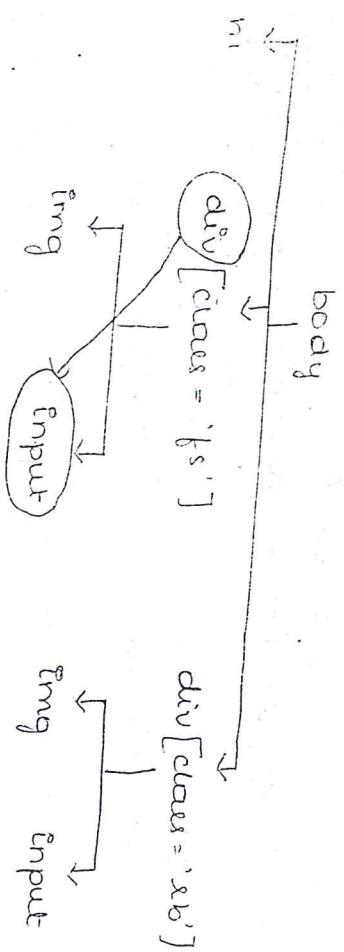
`<input type="text" name="username" />`
`<input type="text" name="password" />`
`<input type="checkbox" name="username" />`

Identify username , X-path
`//input[@name = 'username']` → a matching node

`</div>`
`</body>`
`</html>`

`<input type="button" value="button" />`

To identify buynow button.



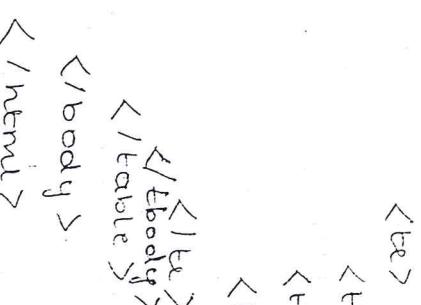
//div[@class = 'bs']//input[@value = 'buynow']

- * When we object not able to identify with multiple attributes in such cases take help of parent & grand parent html tag to identify those level elements.

Case 3:

working with web table

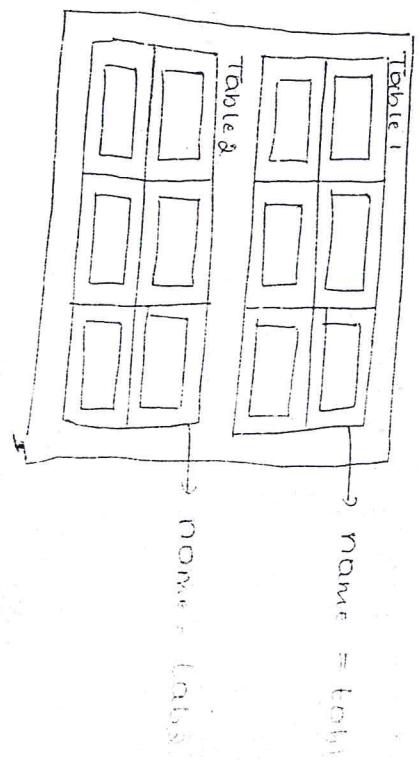
```
<html lang = "en">
<title> Amazon - login </title>
<body>
<table border = "1">
<tr> <td> <input type = "text" /> </td>
```



X-Path : //table//td[3]//input

case 4 :

working with multiple webtable



name = tabs

```
<html>
<title> Amazon - login </title>
<body>
```

```





```

Table 2
`<table name="table" border="1">`
 `<tr>`
 `<td> <input type="text"/> </td>`
 `<td> <input type="text"/> </td>`
 `<td> <input type="text"/> </td>`
 `<td> <input type="text"/> </td>`
 `</tr>`
 `<tr>`
 `<td> <input type="text"/> </td>`
 `<td> <input type="text"/> </td>`
 `<td> <input type="text"/> </td>`
 `<td> <input type="text"/> </td>`
 `</tr>`

tbody

```
table [name = tab1]
```

tbody

```

tbody
  <tr>
    <td>
      <td> <input type="text"/> </td>
      <td> <input type="text"/> </td>
      <td> <input type="text"/> </td>
    </td>
  </tr>
  <tr>
    <td>
      <td> <input type="text"/> </td>
      <td> <input type="text"/> </td>
      <td> <input type="text"/> </td>
    </td>
  </tr>

```

Xpath
`// table[@name = 'tab1'] / tbody / tr[1] / td[1]`
`input`

X-path functions

- ① text() { } X-path function
- ② normalize-space() { } X-path function
- ③ contains() { }
- ④ following-sibling:: { } X-path function
- ⑤ preceding-sibling:: { } X-path function
- ⑥ descendent() { }

```

<td> <input type="text"/> </td>
<td> <input type="text"/> </td>
<td> <input type="text"/> </td>
<td> <input type="text"/> </td>

```

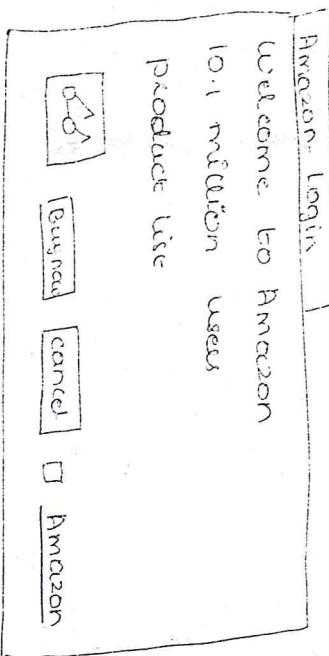
Table 1
`<table>`
 `<tr> <td> welcome to Amazon </td>`
 `<tr> <td> <input type="text" value="en" /> </td>`
 `<tr> <td> <input type="text" value="fr" /> </td>`
 `<tr> <td> <input type="text" value="de" /> </td>`
`</table>`

Table 1
`<h1> welcome to Amazon </h1>`
`<body>`
 `<h1> welcome to Amazon </h1>`

```

<hi class = "header" > product list </hi>
<img src = "image1.png" width = "100"
height = "100" />
<input type = "button" value = "buynow" />
<input type = "button" value = "cancel"
name = "username" />
<input type = "checkbox" name = "usename" />
<a href = "http://amazon.com" > Amazon </a>
</body>
</html>

```



when developer does not provide any link
end attribute return we select X-Path. This is used to identify web-elements.

Syntax :

`text() = 'Expected values'`
`// htmltag [text() = 'Expected values']`

<hi> Welcome to Amazon </hi>
<hi> 10.1 million user </hi>
<hi class = "header" > product list </hi>

Xpath

// hi [text() = '3 matching node
identifies all header with hi']

// hi [text() = 'Welcome to Amazon']
captures all the visibility text from
headless and then compares with
with expected value

// hi [text() = '10.1 million user']

 Amazon

↓
visible text

X-path : // a [text() = 'Amazon']

// menu tag [@ attr = value]

// a [@ class = 'header']

→ <hi class = "header" > product list </hi>
↑
new
back-end
value

new
value

visible text

new
attr

<hi> welcome to Amazon </hi>

→ // hi [text() = 'welcome to Amazon']

{ width: 10.1 million users }
 → `//input[@text = '10.1 million users']`

Problem: text cannot identify using text function

`<div class="header">Product list</div>`
→ `//div[1][@text = 'Product list']`
→ `//div[1][@class = 'header']`
Note: browser cannot ignore the space before e.g.
After this styling.

``
→ `//img[1][@alt = 'image1.png']`
HTML [@ width = '100'] * Path will fail to identify
width Since screen resolution may change

`<input type="button" value="buynow"/>`
→ `//input[@value = 'buynow']`
→ `//input[@value = 'buynow' and (@type = 'button')]`
→ `//input[@type = 'button' and (@value = 'buynow')]`
`<input type="button" value="cancel" name="username" />`

`<input type="checkbox" name="checkbox1"/>`
→ `//input[@type = 'checkbox']` → for future if user
→ `//input[@type = 'checkbox']` → for future if user
→ `//input[@type = 'checkbox' and (@name = 'checkbox')]`
Do not use mandatory attribute while identifying
Object [x-path]

`Amazon`
→ `//a[@text = 'Amazon']`
→ `//a[@href = 'http://amazon.com']`

NOTE

* Whenever object not able to identify using
web element back end browser. In such cases
we can go for text() function to identify the
object using web element visible text.
* text() function navigate to entire html document
and check for the expected value in UI, which
will turn true if complete string matches with
UI.

Disadvantage of text function

- * cannot identify the object using path of the text
& being (be) cannot identify dynamic text
- * cannot ignore the spaces before & after the
string

// html tag [normalize-space(text())@attribute] = Input value

<body>
<div>

Amazon

<h1 class='header'> Product List </h1>

Amazon

<h1>

<div>

Amazon

→ Xpath

// h1 [normalize-space(text()) = 'Product List']

Compare
Product list
Product list
Product list
Product list
Product list

tinus space
two before & after
new visible
text.

two before & after
new visible
text.

E.g.

Gmail → help

// a [normalize-space(text()) = 'need help']

E.g.: when writing attribute selector also have
given space when xpath will be

ActiTime → show button

// input [normalize-space(@value) = 'show']

<NOTE>

* normalize-space function can ignore the spaces
before & after the string before composing xpath
expected value

② value = " — show — "

Mining

Customer

Customer() function take 2 arguments

text() / @ attribute

Expected value

Syntax :

contains(text()) / @attribute, 'Expected value')

/html tag [contains(text()) / @attribute, 'Expected value')

Example :
Not condition user <hi>

dynamic text

Not
html[contains(text(), 'million user')]

html[contains(text(), 'lion user')]

contains function can tolerate piece of text being
(
) also it can tolerate the string by
leaving spaces before & after the string.

html[contains(text(), 'product list')]

Any previous application

Checking if customer carth name already existed when
selected msg displays → it is dynamic
**Customer class - "seesmsg" > customers "Airtel" already
exists. Please choose another customer name**

Notes

html[contains(text(), 'already exists')]

Span[@ class = 'exist msg' and text() = 'Customer
Name:']

Customer "IDFC" already exist
Customer Name:

span class = 'existmsg' > Customer "IDFC" already
exist
< span class = 'existmsg' > Customer name:

For ②,
x-path → //span[text() = 'Customer name']
//span[@ class = 'existmsg' and text() = 'Customer name']

For ①
x-path → //span[contains(text(), 'already exist')]

←
< html lang="en" >
< title > Amazon- Login </title >
< body >
< div class = "bs" >
< img src = "images.png" width = "100"
height = "100" />

< input type = 'button' value = '1000 Rs' />
</div>

< div class = "bs" >
< span class = 'existmsg' > Customer "Airtel" already
exists. Please choose another customer name


```
<img src = 'image3.png' width = '100'  
height = '100' />
```

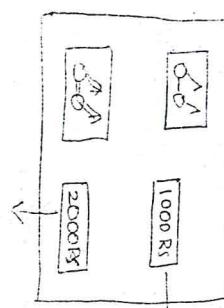
```
<input type = 'button' value = '8000 RS' />
```

```
</div>
```

```
</body>
```

```
</html>
```

It is dynamic.



→ x-path:

```
//div[@class = 'fs']//input  
[contains(@value, 'RS')]
```

xpath : //div[@class = 'fs']//input[contains(@value, 'RS')]

Ques element changing its characteristic is known as dynamic object.

Note

contains() function is used to identify the Ques element using part of the visible text (ex) present by the attribute value.

contains() function can automatically ignore whitespace before and after the string.

contains() function play major role while working on ajax application.

Basically contains() function is used to

dynamically object in UI

The object which is changing its characteristics dynamically at the run-time is called as dynamic object.

(H) following - sibling :: → scope resolution operator

Syntax :

/ following - sibling ::

```
// html tag / following - sibling :: html tag
```

```
<h1> Amazon - login </h1>
```

```
<body>
```

```
<div class = 'fs'>
```

```
<img src = 'image1.png' width = '100'  
height = '100' />
```

```
<input type = 'button' value = '500' />
```

```
</div>
```

```
<div class = 'fs'>
```

```
<img src = 'image3.png' width = '100'  
height = '100' />
```

```
<input type = 'checkbox' value = '50000' />
```

```
</div>
```

```
</body>
```

```
</html>
```

E.g. In class = "active" /

<input

K-block assignments
Original : Rs 101 - Rs 300

</div>

In sibling - siblings keyword need to identify next
element from current node.

Because web element changes its characteristics
complexity in such cases we take a help of
unique element to identify sibling.

Working - sibling

Whenever web element changes its characteristic
complete tag in else

for sibling - sibling keyword need to identify
previous immediate sibling from current node.

in application

E.g. [text() = 'Rs 101 - Rs 300'] / following-sibling
: <span[@class = 'count']>

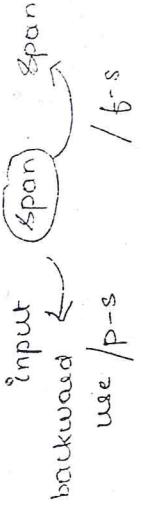
<span [text() = 'Rs 101 - Rs 300'] / preceding-sibling
: <input[@type = 'checkbox']>

<input type="checkbox"/>	msg1
<input type="checkbox"/>	msg2
<input type="checkbox"/>	msg3
<input type="checkbox"/>	msg4

<input checked="" type="checkbox"/>	Rs 101 - Rs 300
-------------------------------------	-----------------

checkbox

X-path :



backward use /p-s
use /b-s

//span [text() = 'Rs 101 - Rs 300'] / following-sibling :
span.

Working with Dynamic web table

Static table → never changes (Row & column does not change)

	abc	

X-path:

in html, index starts from 0

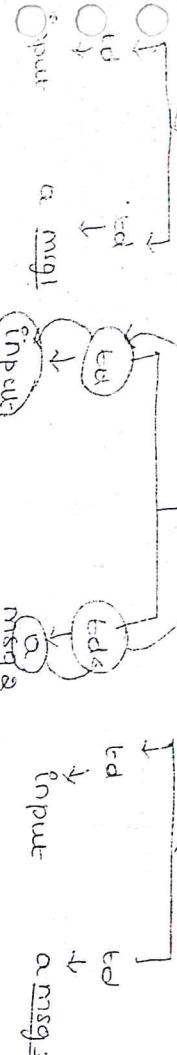
Dynamic webtable

This table gets changed due to
deletion (ex) deletion of some
rows (ex) columns due to their
we cannot identify it using
index (X-path)

	msg1	
	msg2	
	msg3	
	msg4	

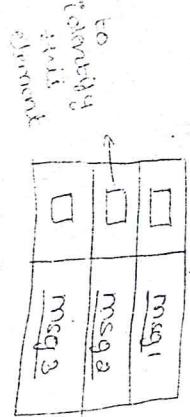
- i) msg9 is deleted, the row & column of msg9 will be changed Hence this will be dynamic.
- ii)

body



E.g.
AxisBank

unique



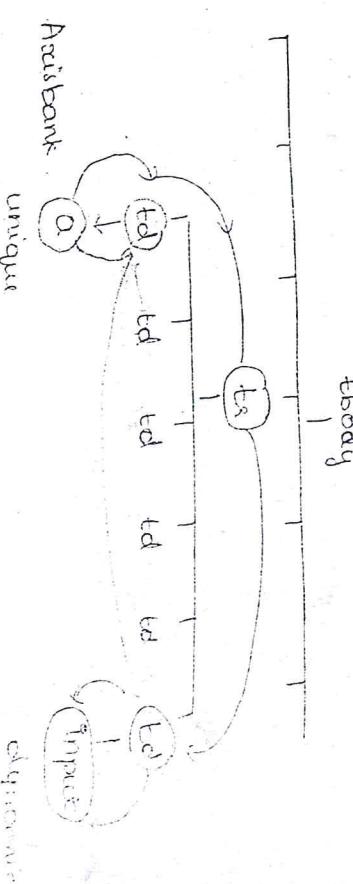
dynamic
unique

Since due to deletion &
deletion of some msg we
cannot identify that
element using index so we
use a unique element as
reference and identify the
dynamic element using

{ } → used to provide
backward attribute

↳ used to go to parent

// parent // td [td [td [text() = 'msg2']] / td [i]] input
↳ // td [a [text() = 'Axis Bank']] / following-sibling::td
↳ td [td [td [text() = 'Axis Bank']] / following-sibling::td]
↳ td [td [td [text() = 'Axis Bank']] / following-sibling::td]
↳ td [td [td [text() = 'Axis Bank']] / following-sibling::td]
↳ td [td [td [text() = 'Axis Bank']] / following-sibling::td]



dynamic
unique

// td [td [a [text() = 'AxisBank']]] / following-sibling::td
↳ td [td [td [text() = 'AxisBank']] / following-sibling::td]

// td [a [text() = 'AxisBank']] / following-sibling::td
↳ td [td [td [text() = 'AxisBank']] / following-sibling::td]

↳ steps to seek with dynamic webtable

Find a unique & dynamic webelement in
an web table.

Write down html structure for unique &
dynamic element.

Print & William present for both unique &

dynamic web elements

(H) write down the x-path for unique element
by taking reference of unique element in
identifying dynamic element

Remove till the common parent item both elements

⇒ How to write x-path from following?
Select absolute x-path for unique element
Select unique x-path for Dynamic element

Remove till the common parent item both elements

customer = customer[1].tbody[0].tr[6].td[0].td[6]

customer.click() / customer[1].tbody[0].tr[6].td[0].td[6].click()

remove two quotes in first row & then add a square bracket

// input[0][0].text('Asia bank') / input[0][0].input.

Question:

Q. How to click book
↳ take on mouse cursor on
book & media

click on new releases below the e-books

Point of
clicks → it happens for a
Reason

→ go to flipkart.com website
click on tourist train link available in tourism
dropdown

Book now for Manacaj express

QA Interview Question

Q. What is locator? what are available & which locator is preferred?

A. It's the measure of quality of correct checkpoints?
What are the advantages & disadvantages of Selenium IDE