

Steph • Capture me of books available for

R₈₁ = R₁₀₀

Expected Result: 2 books should be available

Paste convolution while writing

Actions → used to perform some keyboard operation,
mouse click, move, dropdown etc
(all user - interfaced operation)

[factions]

- moveToElement()
 - back the mouse to place to click on the element
- sendKeys()
- contextClick()
- doubleClick()
- dragAndDrop()
 - so we use Actions class
- peekAlert()
 - very often we do not need to perform any actions
- clickAndHold()

```
act, move TO Element (who);  
    // perform actual actions class action.  
    act. perform();  
}  
}  
}
```

\$ public cities House Over

public static void main (String[] args)

1

// pre Condition : \$ before test

11 top Data

String expectedDelta = "2"

Step 1: find a webcrawler who needs to perform mouse over operations.

```

class : Range(). timeOut(). implicitlyWait(30,
TimeUnit : Seconds);

driver.get("http://Flipkart.com");

WebElement wb = driver.findElement(By.xpath("//span[text()='Books & media']"));

Actions act = new Actions(web);
act.moveToElement(wb).perform();

```

```
System.out.println("Element at " + index + " is " + books[i]);
available = "Pass");
else
    System.out.println("account + " = books are not
available = "FAIL");
classer.quit();
```

A link on "meusgelege" link
 \Rightarrow my ap = " /u[α]text(" = 'Book & epante');
 following - sibling : u[β] / α";
 previous . first Element (By.xpath("//p)). click();

1. capture num of books available for "Rulito"
 "Rg 100"

String Spt = " //span[text() = 'Rg .1 - Rg .100'] /
 following - sibling :: span" ,
 clickCount = clickElement(By.xpath
 "(Spt))

Each column equals (expected Data)

Factions → user Interception operation

Books & Media	drop down menu box
input	error

Keyboard Operations

For keyboard operations, use sendkeys() method in : Actions class and also use file ENUM. Keys class to specify which key to be pressed.

ENUM

keys

ENTER
ESCAPE
ALT
TAB
CONTROL

static final
method.

check() —> static method

E.g. Keyboard Operations

public class KeyboardOper

{
public static void main(String[] args)
{

WebDriver driver = new FirefoxDriver();
driver.get("http://www.pc/login.do");

driver.findElement(By.name("username")).

sendKeys("asimin");
driver.findElement(By.name("Pwd")).sendKeys

("manager");
Actions act = new Actions(driver);

act.sendKeys(Keys.ENTER).perform();
}

}

In C/C++, #define Max=5 (macros)
instead of 5, we can replace with name
Max.

In java, enum class keys

ENTER = Enter()
CONTROL = control()

short key (ex) user defined data type can be
created in enum class

Instead of writing the method names like shift
we can use a user defined keycodes

E.g for concurrent keyboard operation

like
ctrl + c
ctrl + v

Enum class

double click

right click

only double click

Keyboard Operation



public void KeyBoardOpen

public static void main (String [] args)

```

{
    WebDriver driver = new FirefoxDriver();
    driver.get ("http://deeps-pc/login/do");
    WebElement unob = driver.findElement(By.name
        ("username"));
    unob.sendKeys ("admin");
    Actions act = new Actions (driver);
    act.sendKeys (unob).perform();
    act.sendKeys (Keys.chord (Keys.CONTROL, "C"));
    act.sendKeys (Keys.TAB).perform();
    act.sendKeys (Keys.ENTER).perform();
}
  
```

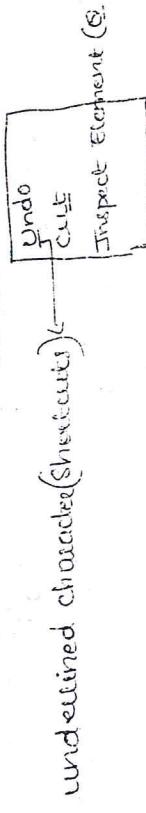
act.sendKeys (Keys.ENTER).perform();

act.sendKeys (Keys.chord (Keys.CONTROL, "V")).

sendKeys () ;

E.g. for right click operation

username [admin] → right click



undertined character(Selenium) → right click
 Action.click → undertined character(Selenium)
 undertined character(Selenium) → right click
 operation in Action.click
 public class SampleRightClick

public static void main (String [] args)

```

{
    WebDriver driver = new FirefoxDriver();
    driver.get ("http://deeps-pc/login/do");
    WebElement unob = driver.findElement(By.name
        ("username"));
    unob.sendKeys ("admin");
    Actions act = new Actions (driver);
    act.contextClick (unob).perform();
    act.sendKeys ("Q").perform();
}
  
```

Keyboard Operation
 character → act.sendKeys ("C")
 functional → act.sendKeys (Keys.ENTER)

concurrent

↳ character + functional

act. & sendkeys (Keys. chord (Keys. CONTROL, "C")).

chord

NOTE

- * Actions class is a webDriver class available in Interaction package which is used to perform user interaction/ interaction operations like mouse over, keyboard operation, single click, double click etc.

moveToElement()

- "moveToElement()" method is used to take mouse cursor middle of the webElement (mouse over - & next click used)

sendKeys()

- sendKeys() method is used to perform key board operation. There are 3 types of Keyboard Operation.

Character Keyboard Operations

e.g.

act. & sendkeys ("A") . perform();

↓
character string

(or)
Webdriver sendkeys

can also be used

② Functional Keyboard Operation

Eg act. & sendkeys (Keys. ENTER). perform()

functional
control

TAB

ARROW UP

ARROW DOWN

In order to perform functional keyboard operation take help of Keys enum class available in WebDriver jar.

③ Concurrent Keyboard Operation

In order to perform concurrent keyboard operation take help of chord method in Action class

Eg

act. & sendkeys (Keys. chord(Keys. CONTROL, "C"))

perform();

Context Click()

Context Click() method is used to perform

Right Click operation on web Element

e.g.

act. & sendkeys ("A") . perform();

It is not possible
to perform

moving image from desktop to app

on selection

Driver by click

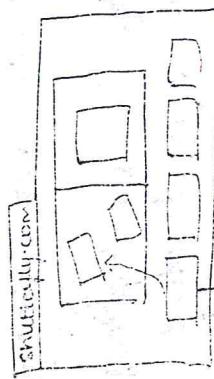


image1.png

```
distutils -> id .IE (By .xpath ("//div[-.--]"));
Same file -> id .IE (By .xpath ("//img[@id = 'image1']"));
png ] ) .
```

Action: click = new Action (driver);
click. dragAndDrop (webElement, destination). perform();

if we have scroll page, we can use
Space

Page up, page down
Arrow up, Arrow left

WebElement driver = new FirefoxDriver();
This gives control
of browser to
driver

Actions (driver),

perform()

- ↳ ① to perform actions.
- ↳ ② to select browser control to driver again

NOTE

Whenever we create object to actions -
it should mandatory pass WebDriver (or) webdriver
control as an argument because
sequences browser object to perform Keyboard or
mouse operation

Select class

Select

when html tag is
Select only we go
for this

→ selectByValue()

→ selectByVisibleText()

→ selectByIndex()

→ selectMultiple()

→ deselectAll()

→ deselectByVisibleText()

→ deselectByValue()

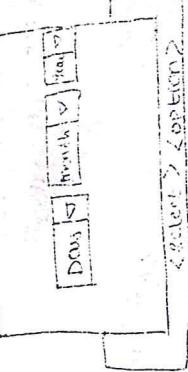
→ deselectByIndex()

→ getOptions(): List

→ getFirstSelectedOption()

→ getAllSelectedOptions()

Facebook login



state - driver be ('chrome')
Select (feedback)
↳ select by visible
text (Logout)

```
public class SampleWebListCode
```

```
{ public static void main (String [] args)
```

```
{ WebDriver driver = new FirefoxDriver();
```

```
driver.get ("http://facebook.com");
```

```
// Step 1 : find dropdown in UI ; findElement method.
```

```
WebElement sub = driver.findElement(By.id ("month"));
```

```
// Step 2 : create an object to select class as pass
```

```
// web element as object
```

```
Select sel = new Select (sub);
```

```
// perform operation on dropdown
```

```
sel.selectByVisibleText ("Dec");
```

```
WebElement sub1 = driver.findElement(By.id ("day"));
```

```
if (sel . isMultiple ())
```

```
// get the no of item available in list
```

```
int count = list . size ();
```

```
for (int i = 0; i < count; i++)
```

```
{
```

```
sel . selectByIndex (i);
```

```
Working with multi select dropdown
```



```
select (String str);
```

```
public class SampleMultiSel
```

```
{ public static void main (String [] args)
```

```
{ WebDriver driver = new FirefoxDriver();
```

```
driver.get ("file:///C:/Users/Deeplop/Desktop/JavaSel.html");
```

```
// find dropdown in UI
```

```
Select sel = new Select (driver.findElement (By.name ("sel")));
```

```
// get the options available on multi select dropdown
```

```
list <WebElement> list = sel . getOptions();
```

```
// if the dropdown is multiple select then the value
```

```
else (sel . isMultiple ())
```

```
// get the no of item available in list
```

```
int count = list . size ();
```

```
for (int i = 0; i < count; i++)
```

```
{
```

```
sel . selectByIndex (i);
```

```
else
```

```
sel . selectByValue ("");
```

```

    } // capture the text from web element
    string actual = $ele.get("value");
    if( actual.equals(expectedVal))
    {
        System.out.println("Expected value is present, hence this action
        // if expected value is present, select this action
        // come out of the loop by setting flag value
        // select By Visible Text (expectedVal);
        flag = true;
        break;
    }
}

private void Dynamic_Select_DID()
{
    Atomic void main (String [] args)
    {
        String expectedValue = "sql testing";
        System.out.println("Expected Value : "+expectedValue);
        System.out.println("Actual Value : "+actual);
        if(actual.equals(expectedValue))
        {
            System.out.println("Expected Value + Actual Value are same");
        }
        else
        {
            System.out.println("Expected Value + Actual Value are not same");
        }
    }
}
}

```

i = 1

i = 2

i = 3

i = 4

i = 5

i = 6

i = 7

i = 8

i = 9

i = 10

i = 11

i = 12

i = 13

i = 14

i = 15

i = 16

i = 17

i = 18

i = 19

i = 20

i = 21

i = 22

i = 23

i = 24

i = 25

i = 26

i = 27

i = 28

i = 29

i = 30

i = 31

i = 32

i = 33

i = 34

i = 35

i = 36

i = 37

i = 38

i = 39

i = 40

i = 41

i = 42

i = 43

i = 44

i = 45

i = 46

i = 47

i = 48

i = 49

i = 50

i = 51

i = 52

i = 53

i = 54

i = 55

i = 56

i = 57

i = 58

i = 59

i = 60

i = 61

i = 62

i = 63

i = 64

i = 65

i = 66

i = 67

i = 68

i = 69

i = 70

i = 71

i = 72

i = 73

i = 74

i = 75

i = 76

i = 77

i = 78

i = 79

i = 80

i = 81

i = 82

i = 83

i = 84

i = 85

i = 86

i = 87

i = 88

i = 89

i = 90

i = 91

i = 92

i = 93

i = 94

i = 95

i = 96

i = 97

i = 98

i = 99

i = 100

i = 101

i = 102

i = 103

i = 104

i = 105

i = 106

i = 107

i = 108

i = 109

i = 110

i = 111

i = 112

i = 113

i = 114

i = 115

i = 116

i = 117

i = 118

i = 119

i = 120

i = 121

i = 122

i = 123

i = 124

i = 125

i = 126

i = 127

i = 128

i = 129

i = 130

i = 131

i = 132

i = 133

i = 134

i = 135

i = 136

i = 137

i = 138

i = 139

i = 140

i = 141

i = 142

i = 143

i = 144

i = 145

i = 146

i = 147

i = 148

i = 149

i = 150

i = 151

i = 152

i = 153

i = 154

i = 155

i = 156

i = 157

i = 158

i = 159

i = 160

i = 161

i = 162

i = 163

i = 164

i = 165

i = 166

i = 167

i = 168

i = 169

i = 170

i = 171

i = 172

i = 173

i = 174

i = 175

i = 176

i = 177

i = 178

i = 179

i = 180

i = 181

i = 182

i = 183

i = 184

i = 185

i = 186

i = 187

i = 188

i = 189

i = 190

i = 191

i = 192

i = 193

i = 194

i = 195

i = 196

i = 197

i = 198

i = 199

i = 200

i = 201

i = 202

i = 203

i = 204

i = 205

i = 206

i = 207

i = 208

i = 209

i = 210

i = 211

i = 212

i = 213

i = 214

i = 215

i = 216

i = 217

i = 218

i = 219

i = 220

i = 221

i = 222

i = 223

i = 224

i = 225

i = 226

i = 227

i = 228

i = 229

i = 230

i = 231

i = 232

i = 233

i = 234

i = 235

i = 236

i = 237

i = 238

i = 239

i = 240

i = 241

i = 242

i = 243

i = 244

i = 245

i = 246

i = 247

i = 248

i = 249

i = 250

i = 251

i = 252

i = 253

i = 254

i = 255

i = 256

i = 257

i = 258

i = 259

i = 260

i = 261

i = 262

i = 263

i = 264

i = 265

i = 266

i = 267

i = 268

i = 269

i = 270

i = 271

i = 272

i = 273

i = 274

i = 275

i = 276

i = 277

i = 278

i = 279

i = 280

i = 281

i = 282

i = 283

i = 284

i = 285

i = 286

i = 287

i = 288

i = 289

i = 290

i = 291

i = 292

i = 293

i = 294

i = 295

i = 296

i = 297

i = 298

i = 299

i = 300

i = 301

i = 302

i = 303

i = 304

i = 305

i = 306

i = 307

i = 308

i = 309

i = 310

i = 311

i = 312

i = 313

i = 314

i = 315

i = 316

i = 317

i = 318

i = 319

i = 320

i = 321

i = 322

i = 323

i = 324

i = 325

i = 326

i = 327

i = 328

i = 329

i = 330

i = 331

i = 332

i = 333

i = 334

i = 335

i = 336

i = 337

i = 338

i = 339

i = 340

i = 341

i = 342

i = 343

i = 344

i = 345

i = 346

i = 347

i = 348

i = 349

i = 350

i = 351

i = 352

i = 353

i = 354

i = 355

i = 356

i = 357

i = 358

i = 359

i = 360

i = 361

i = 362

i = 363

i = 364

i = 365

i = 366

i = 367

i = 368

i = 369

i = 370

i = 371

i = 372

i = 373

i = 374

i = 375

i = 376

i = 377

i = 378

i = 379

i = 380

i = 381

i = 382

i = 383

i = 384

i = 385

i = 386

i = 387

i = 388

i = 389

i = 390

i = 391

i = 392

i = 393

i = 394

i = 395

i = 396

i = 397

i = 398

i = 399

i = 400

i = 401

i = 402

i = 403

i = 404

i = 405

i = 406

i = 407

i = 408

i = 409

i = 410

i = 411

i = 412

i = 413

i = 414

i = 415

i = 416

i = 417

i = 418

i = 419

i = 420

i = 421

i = 422

i = 423

i = 424

i = 425

i = 426

i = 427

i = 428

i = 429

i = 430

i = 431

i = 432

i = 433

i = 434

i = 435

i = 436

i = 437

i = 438

i = 439

i = 440

i = 441

i = 442

i = 443

i = 444

i = 445

i = 446

i = 447

i = 448

i = 449

i = 450

i = 451

i = 452

i = 453

i = 454

i = 455

i = 456

i = 457

i = 458

i = 459

i = 460

i = 461

i = 462

i = 463

i = 464

i = 465

i = 466

i = 467

i = 468

i = 469

i = 470

i = 471

i = 472

i = 473

i = 474

i = 475

i = 476

i = 477

i = 478

i = 479

i = 480

i = 481

i = 482

i = 483

i = 484

i = 485

i = 486

i = 487

i = 488

i = 489

i = 490

i = 491

i = 492

i = 493

i = 494

i = 495

i = 496

i = 497

i = 498

i = 499

i = 500

NOTE

In order to perform action on the drop down list

- we cannot use only web element interface because

inside we have elements interface we have sendkeys for typing & click for clicking web element but there is no Select method to select the option on the list.

So in order to select the value from the list we

- should take help of Select class

Select class is a web driver class available in selenium package, which is used to select the value from single Select list Box (ex) must select and Dynamic select list Box.

While use 3 ways use click & select this value from drop down

SelectByVisibleText()

Select class which based on visible text in UI

SelectByValue("Text")

SelectByIndex(index)

SelectByVisibleText("Text")

Select the value based on position of the option

(ex) When available in list box Index is always start from zero

SelectByValue("String")

Select the value based on value of the value

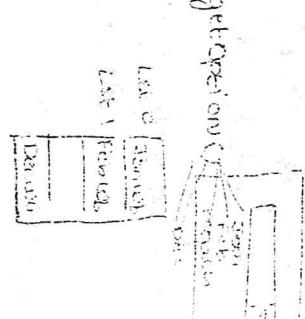
value from option html tag.

option value = "88" > LaxmiBank Listings

Get All Selected Options()

Get only already selected option weberwise from

the dropdown, which return



* These are four ways we can represent the value from drop down.

① deSelectAll()

In order to deselect all the values based on multiselect list box we can go for deselect first

② deSelectByIndex()

③ deSelectByValue()

④ deSelectByVisibleText()

* There are multiple ways we can get the value from the drop down

① getOptions()

Get all the option web element from dropdown like collection list



getOptions() returns

list of WebElement

list

Desh

Laxmi

Ranjan

Multiple Select

- ④ Used to check whether select box is multiple select (ex) Single Select Box we go with example () which returns boolean (true) if dropdown is multiple select

Note : Multiple Select

- ⑤ We cannot take a help of select class when we want to implement using select multiple tag.
- Ex. Google Search Edit box looks like a multiple select box which is not implemented by using select multiple

Handling

- ⑥ We are writing a code for dropdown list, dropdown component should be from selenium package (ie) import org.openqa.selenium.support.ui.
- vi. support

- ⑦ And here we can see API should be import from java.util.List

Multiple Question

- ⑧ Used to work with dropdown, list box or combo box
 - v. This operation is used when dropdown consists of multiple choices to select choices in form of pull down menu
 - v. It contains an argument of using select method
 - v. It works on mechanism on action

Window Handling

True case four types of window available in web
Obscured application

- ① Html window / New tab & new browser
- ② Alert Pop up → Java Script pop-up.
- ③ Invisible Pop ups / windows → html source code not found.
- ④ Browser on load Pop up

Alert pop up / windows

iret

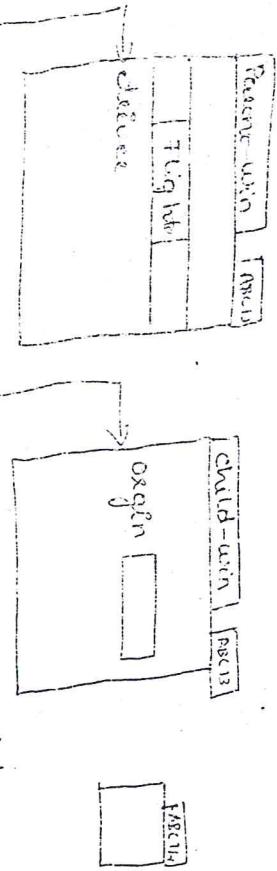
→ flights

//a[link() = 'flights']

if * [text() = 'flights'] → to check how many links

One link is present with flights present with flights so we can use linkText()

located to find webelement.

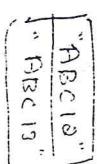


passing control to child win
eleme... .click(...); window('window Tab')

window Id

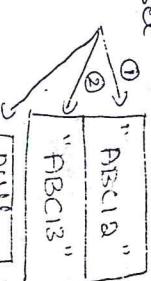
Set <String> set = driver.getWindowsHandle();

List



String parentID = list.get(0)
String child ID = list.get(0)

since it contains
duplicate values in
so use parent set



Iterate it = list.iterator()

String parentID = it.next()

String child ID = it.next()

List

Set

① Index based | Not Index based

② allows duplicate | ② doesn't allow duplicate

values

③ It has get | ③ It uses Iterator interface

method to | to iterator elements

Program

```
public class HtmlPopup {
    public static void main (String [] args) throws
        InterruptedException
```

Webdriver driver = new FirefoxDriver();

driver.get("http://www.mkyong.com/");

```

// Click on Flights link , which opens new window
Flight Element (By Link Text ("Flights")).click();

// Open child window active window ID's
Flight > Get New WindowHandle\(\)

// captures child window IDs using Iterator


```

Iterator<String> it = set.iterator();
String parentWindowID = it.next();
String childWindowID = it.next();

```


// pass driver control to child window
switchTo\(\).window(childWindowID);

```

// click child window to download

```

downloadManager\(\).timeouts\(0\).implicitlyWait
\(500, TimeUnit.SECONDS\);

```

// perform action on child window

```

close\(\);

```

// pass driver control back to parent window

```

switchTo\(\).window\(parentWindowID\);

```

// perform operation on parent window

```

click\("usernameText"\);
sendKeys\("admin"\);

```

Note

- * Whenever action performed to open the windows (HTML windows) webdriver cannot have control automatically
- * In order to work with new windows pass a driver control to new windows by using window ID as a reference
- * There are two ways we can get the window ID
 - ① driver.getWindowHandle()
 - ② driver.getwindowHandles()
- * String currentWindowID = driver.getWindowHandle();
 Set<String> set = driver.getWindowHandles();
- * → getwindowHandle() captures only one window ID at a time driver context is current active
 → getwindowHandles() captures all the windows for which is opened by webdriver for the current execution
 Assign (x) unique window ID
 * Application always generate unique execution for each window for the window handles
 * In order to perform action on your windows see the window handles It's using getwindowHandles() on the window ID's using
 method which returns WebElement set of strings, and pass a driver control to new windows using
 driver.getWindowHandle()

Q) One the operation performed with new window
pass or control back to parent window before performing operation on parent window

`driver.switchTo().window("childID")`

`driver.switchTo().window("parentID")`

d. `switchTo().window("parentID")`.

Q) There are two ways we can close the window.

① `driver.quit()` → close all the windows which is opened by web browser for current execution.

② `driver.close()` → close only one window release driver control of current tab active.

Press / configuration / Java -> script pop-up

Scenario : Verify the crossed configuration message

Step1: Navigate to `http://ebook-co.in` website

Steps : click a login button without username & password

Step3: Verify configuration message

Expected Result: Configuration pop-up should be displayed along with error message

username

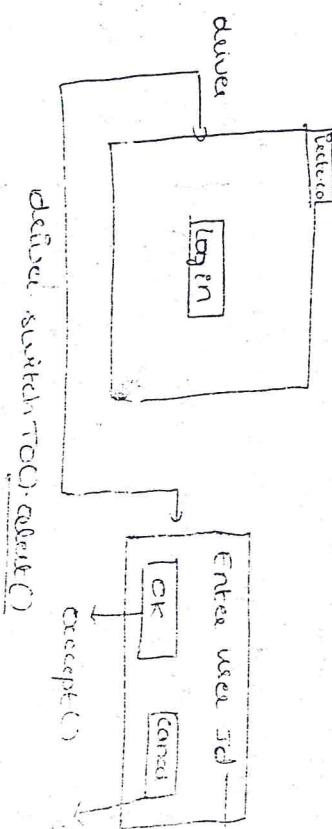
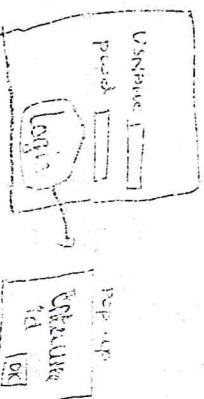
password

`Alert accept();`
`Alert dismiss();`
`Alert getText();`

```
public class AlertSample {
    public static void main(String[] args) {
        // test data
    }
}
```

```
String expctdVal = "Enter login id";
String actlVal = driver.getTitle();
System.out.println(expctdVal);
System.out.println(actlVal);
if(expctdVal.equals(actlVal)) {
    System.out.println("Test Case Pass");
} else {
    System.out.println("Test Case Fail");
}
```

We cannot expect
on this since still
not found pop-up
it is Java script.



new window
size (7d)
html lang

conformation pop-up → java script
it consists of only string
(ex) Ok / cancel

Ok / cancel

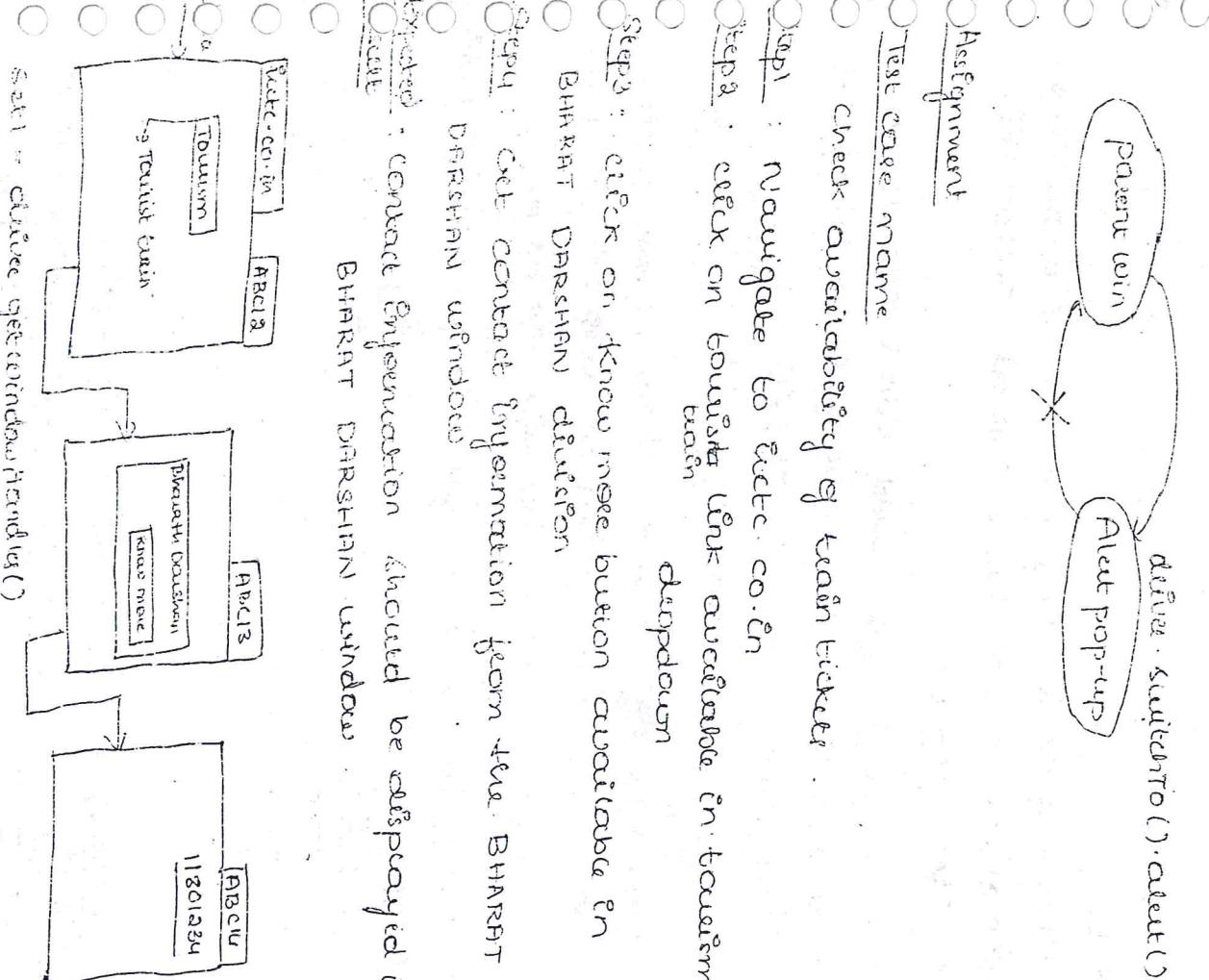
```

    driver.findElement(By.id("loginButton")).click();
    // press enter key to Alert pop-up
    Alert alert = driver.switchTo().alert();
    alert.sendKeys("abc");
    // capture the msg from alert
    String actualVal = alert.getText();
    if(actualVal.equals("Expected Val"))
        System.out.println("actual + " ==> msg is
                verified = PASS");
    else
        System.out.println("actual + " ==> msg is not
                verified = FAIL");
    // close the alert
    alert.accept();
    // dismiss the alert
    alert.dismiss();
    // get text of alert
    String text = alert.getText();
    System.out.println(text);
    // switch back to main window
    driver.switchTo().defaultContent();

```

- NOTE
- * whenever action performed to open an Alert, we web driver cannot have control on alert automatically on alert pop-up
 - * In order to work with alert pop-up, we should pass a direct control to alert box before performing an action
 - driver.switchTo().alert().
 - * Alert box are always implemented using Java's accept which always contain only OK & CANCEL & also Truncated message
 - * Using alert API interface perform specific click on alert
 - [Alert]
 - accept()
 - dismiss()
 - getText()
 - * Whenever operation is performed with alert we need to get control back to main window/ previous window because control automatically come to front to prevent windows.

Note word => msg. in verified = - pass



```
public class TicketTrainTickets
{
    public static void main (String [ ] args)
```

```
webdriver = webdriver.Chrome()  
driver.get("http://www.freelancer.com")
```

```
webElement wb = driver.findElement(By.  
id("recte-touism"));
```

Actions \leftarrow new Actions (deletion)

act. moveToElement(*el*).perform(*o*)

```
    .driver.findElement(By.xpath("//a[@text='Tourist Taxim']")).click();
```

set<Steing> set = driver . get window klein -
- aber ();

```

Iterate <Steing> (to = &ste, CleanUp());
Steing parent [id] = ste.next();
Steing current [id] = to.next();

```

dear Mr. Garrison, Washington, D. C., August 10, 1860.

device. manage(), timeOuts(), apply(λ , μ , ν)
(λ , TimeUnit, Seconds);

It is a main feature of the

```
click());
```

```
// step 3
```

```
driver.manage().timeouts().implicitlyWait(
```

```
20, TimeUnit.SECONDS);
```

```
driver.findElement(By.xpath("//input[@value="
```

```
'Create New Project']")).click();
```

```
Select sel = new Select(driver.findElement(By.
```

```
name("customerID")));
```

```
sel.selectByVisibleText(custName);
```

```
driver.findElement(By.name("name")).sendKeys
```

```
(proName);
```

```
driver.findElement(By.name("createProjectSubmit"))
```

```
.click();
```

```
// Step 4
```

```
driver.findElement(By.linkText("Open Tasks")).
```

```
click();
```

```
driver.manage().timeouts().implicitlyWait(
```

```
20, TimeUnit.SECONDS);
```

```
driver.findElement(By.xpath("//input[@value="
```

```
'Create New Task']")).click();
```

```
Select sel1 = new Select(driver.findElement
```

```
(By.name("customerID")));
```

```
sel1.selectByVisibleText(custName);
```

```
Select sel2 = new Select(driver.findElement
```

```
(By.name("projectID")));
```

```
// steps
```

```
driver.findElement(By.name("taskID.name"))
```

```
.sendKeys(taskName[0]);
```

```
driver.findElement(By.name("task[0].name"))
```

```
.sendKeys(taskName[1]);
```

```
driver.findElement(By.name("task[1].name"))
```

```
.sendKeys(taskName[2]);
```

```
driver.findElement(By.xpath("//input[@value="
```

```
'Create Task']")).click();
```

```
// step 6
```

```
String str = driver.findElement(By.xpath("//input[@value="
```

```
'new task']")).getText();
```

```
System.out.println(str);
```

```
if (str.contains(custName))
```

```
{ System.out.println("verified = pass");
```

```
else
```

```
{ System.out.println("not verified = fail");
```

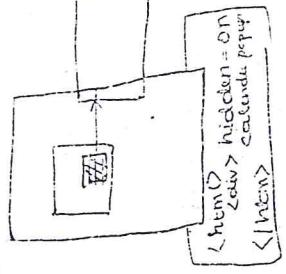
```
}
```

```
driver.findElement(By.id("logoutLink"))
```

```
.click();
```

Pre-condition: Flying houses to be available in AM Application Using DB query we can insert flight record

Output



3 new tasks were added to the customer

"OAHDFC", project "testing".

User feed = PASS

Invisible Pop-up

In invisible pop up there are 3 types,

① Crosses pop-up

② Expected invisible pop up

③ Unexpected invisible pop up (Advertisement pop up)

Scenario

Step 1: Navigate to makemytrip.com
Step 2: Search for a flight availability for the
particular specific day.

Test Data :

type = Domestic

from = Bangalore
to = De Lhi

Date = Dec/22/2013

Flight type = One way

Expected Result : Flight should be available & ticket
can be booked

variable it is checked
this hidden code is off
It is written within html
source code but it opens a
new window(s) does not
have a URL as in case
of html pop up

NOTE

④ Invisible pop up case always implemented using
either html tag or some windows built
which will be disabled initially.
⑤ In order to click with impossible pop up problem
it is to enable invisible pop up & clicking
action to enable a xpath and perform an operation

Sample code for calendar pop up

public static void main (String [] args)
{
 public static void SearchFlight Makemytrip
 {
 // test data
 String month = "9";
 String day = "1";
 String year = "2013";
 }
}

Abstract class

```
    [Anwesen] x dispay()
```

des piece () ;

```
    [Scanning] [watching] mobile
```

```
    [watching] [mobile]
```

IMPLEMENTATION NOTES

continue

WebDriver driver = new FirefoxDriver();
driver.get("http://www.makemytrip.com/");

// click on calendar icon, which opens a dropdown
driver.findElement(By.xpath("//span[@text='DEPARTURE']")).click();

String datePicXpath = "//td[@class='day-cell'][text=" +
+ day + " - " + month + " - " + year + "]/a";
text() = ' " + day + "]";

int count = 0;

while (count <= 18) {

tag

driver.findElement(By.xpath("//td[@class='day-cell'][text=" +
+ day + " - " + month + " - " + year + "]/a")).

click();

break;

} catch (Throwable e) {

driver.findElement(By.xpath("//td[@class='day-cell'][text=" +
+ day + " - " + month + " - " + year + "]/a")).

click();

count++;



Sample code for Expected Invisible pop up

Steps to write, Expected Invisible pop up

Step 1 : perform an action to enable invisible pop up

Step 2 : check whether the pop up is enabled or not is added.

Step 3 : press an action if the pop up is enabled.

Eg-

```
boolean flag = driver.findElement(By.id("DiscardChanges")).isEnabled();
```

```
if (flag)
```

```
{
```

```
driver.findElement(By.id("DiscardChanges")).click();
```

```
}
```

```
else
```

```
{
```

```
System.out.println("pop-up is not displayed  
== == FAIL");
```

```
}
```

Sample code for Advertisement / unexpected invisible pop up

Pop up

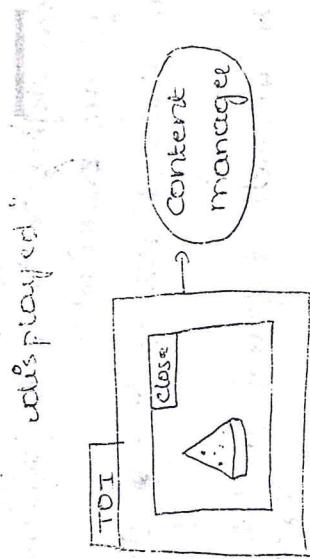
Vaidy Anandgopal Pop Up

Pre-condition :- Execute dbl Queen "Times of India".

Advertisement pop-up in front page of Times of India.

- ① navigate to TOI
- ② verify the "Advertise pop-up"

Expected Result :- "Pop up should be displayed"



Program

```
public class HandleAdvertisementPopUp {  
    static WebDriver main (String [] args) {
```

```
        WebDriver driver = new FirefoxDriver ();  
        driver .get ("http://lipkakat.com");
```

```
        try {  
            WebElement element = driver .findElement (By .xpath ("//span[@class="
```

```
                'close - id'])).click ();  
            System.out.println ("pop up is displayed");  
        } catch (Exception e) {  
            System.out.println ("pop up is not displayed");  
        }
```

```
        driver .get ("http://timesofindia.com");  
        try {  
            WebElement element = driver .findElement (By .xpath ("//button[@class="
```

```
                'close')").click ();  
            System.out.println ("pop up is displayed");  
        } catch (Exception e) {  
            System.out.println ("pop up is not displayed");  
        }
```

```
        System.out.println ("Advertisement pop up was displayed 5 to 10 times of India & pop up is going to dispaly only 5 or 6 times 1 to 15");  
    } catch (Exception e) {  
        System.out.println ("error");  
    }
```

```
    } catch (Exception e) {  
        System.out.println ("error");  
    }
```

```
    } catch (Exception e) {  
        System.out.println ("error");  
    }
```

try {
}

```
driver.findElement(By.xpath("//close Bttn")).click();
sop("pop up is displayed");
break;
}
```

```
} catch (Exception e)
```

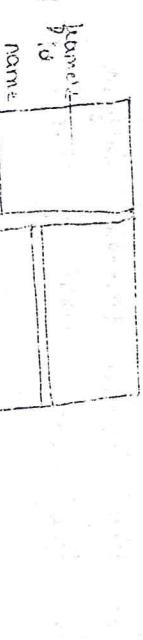
```
{ sop("pop up is not displayed");
```

```
Thread.sleep(1000);
count++;
```

Authentication pop up | Browser On load pop up

https://hatc.com

UN	<input type="text"/>
psw	<input type="password"/>



become
Edit
Name

To find where an object is in main (i.e) become window. 2 ways can be used
a) right click on target object & search name

- ① right click on target object & search name
- ② use `getAttribute("name")` each time its frame in Edit bar

TOP	<input type="text"/>
Frame	<input type="password"/>

<html>

```
<form id='frame' name='frame'>
```

<html>

=

</html>

</form>

=

</html>

[Embedded HTML page
in another html page]

```
String url = "https://admin:password@hatc.com";
driver.get(url);
```

* Authentication pop up always appears while loading
page & ask for username & password.

* In order to handle this Authentication pop up
pass credentials on url itself.

Frame handling

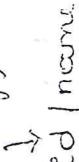
Section on browser

</frame>

Frame 3

→ To switch from main window to frame we can use 2 overrided methods

① driver.switchTo().frame(String frameName);



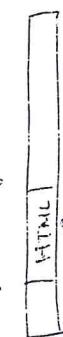
② driver.switchTo().frame(WebElement element);



③ driver.findElement(By.name("webElement"))

→ To identify frame id or name

Go to browser.



Select

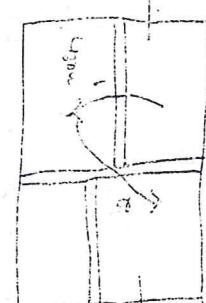


Search

try to find frame option & click

then we can find frame id / name

→ To perform action on one frame from another frame, we first switch to main window & then switch to another frame



Program

```
public class FrameHandle {  
    public static void main(String[] args) {  
        WebDriver driver = new FirefoxDriver();  
        driver.get("http://selenium.github.io/selenium/javascript.html");  
        // open javascript.html file  
        // pass a webdriver control to Frame - 3  
        driver.switchTo().frame("classFrame");  
        // perform an operation , wherein object present inside  
        // the frame - 3  
        driver.findElement(By.linkText("com.sinaenglishsite.  
selenium")) .click();  
        // pass control back to main window  
        driver.switchTo().defaultContent();  
        // pass control to Frame - 2  
        driver.switchTo().frame("packageFrame");  
        // perform an operation wherein object present inside  
        // the frame - 2  
        driver.findElement(By.linkText("Actions")) .click();  
    }  
}
```

one page next

Working with Link / multiple Elements

Scenario : Verify multiple links availability

Pre-conditions :- upload multiple links in the system

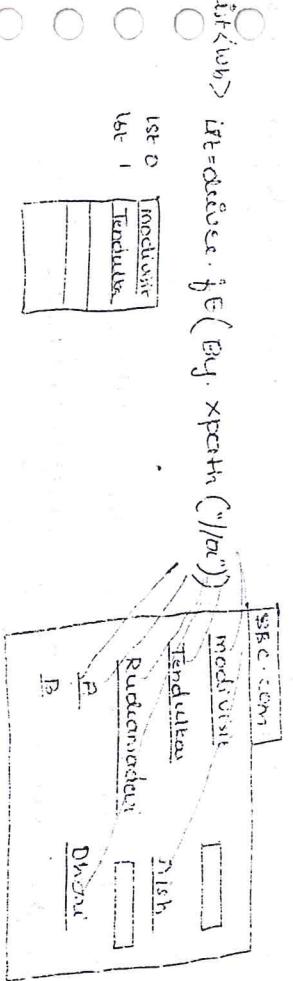
end.

Step1 : Navigate to BBC. com

Step2 : Check no of links available in UI

Step3 : Capture all the link names available in UI & place all the link name in excel sheet / Note pad.

Expected Result :- uploaded link should be displayed in UI



Scenario : Delete all inbox message

Pre-condition :- Message should be executable in inbox and user account should be created

Step1 : Navigate to Gmail application and login credentials.

Step2 : Select all the inbox message checkbox by clicking on select button

Expected Result : All the message selected are checked

public class MultipleElements

{

public static void main(

String[] args)

{

WebDriver driver = new FirefoxDriver();

driver.get("http://bbc.com");

// find all links available in UI, using find

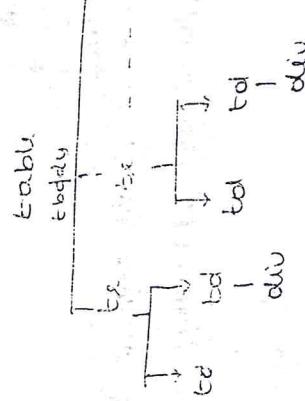
```
// find all the checkboxes available in Index
```

```
let<wd> = id('wd');
let<wd> = id('wd').findElements(By.xpath("//input[@type='checkbox']"));
let<wd> = id('wd').findElements(By.xpath("//input[@type='checkbox']"));



ck1
 ck2
 ck3


```



```
public class
```

```
public static void main(String[] args)
```

- NOTE
- * findElements methods used to identify multiple webElements in UI
 - * findElements always returns LIST of webElements
 - If multiple matching node in UI
 - * findElements always returns empty LIST, if no matching node in UI
 - * Prefer to get multiple elements webElements in UI [Eg only check in UI, we go for findElements in UI]
 - * Thread to capture value from the specific column from dynamic webTable findElements methods performed
- ```

public class Test {
 WebDriver driver = new FirefoxDriver();
 driver.get("http://gmail.com");
 WebElement loginElement(By.xpath("//input[@name='username']"));
 loginElement.sendKeys("selenium.selenium@gmail.com");
 loginElement.sendKeys("password").click();
 Thread.sleep(2000);
 WebElement findElement(By.id("rememberMe"));
 findElement.click();
 WebElement findElement(By.id("signIn"));
 findElement.click();
}

```

driver.findElement(By.tagName("a"));

(Identify the first webElement)



webElement wb = d.findElement(By.tagName("a"));

(Identify the first webElement)



Difference between findElements & findElement?

findElement

used to identify single webElement

used to identify multiple webElements

findElement method returns list available webElement

if multiple matching node in UI

find Element throws NoSuchElementException if no matching node in UI

find Elements

used to identify multiple webElements

find Element method returns collection of list

multiple matching node in UI

Assignment

String firstColumnxpath = d.findElements(By.xpath("//table[  
@name = 'tab1']/tbody/tr/td[1]/\*"))

but 1st column has 3 rows

| Transac No. | Name |
|-------------|------|
| ABC1        |      |
| ABC2        |      |
| ABC3        |      |

use mandatory attribute when writing xpath  
How to get first column data from Dynamic Webtable?

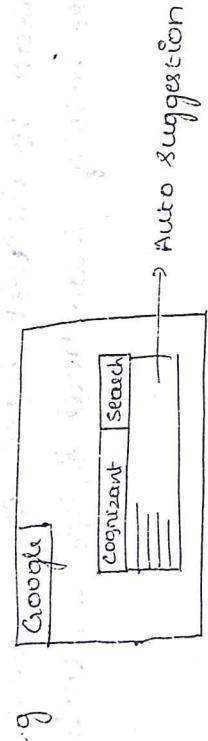
- ④ How to get first column data from Dynamic Webtable?
  - driver.findElement(By.tagName("a"));
  - driver.findElement(By.xpath("//a"));
  - Used findElements method from tagName / xpath
  - How to get all the links available in UI?
    - driver.findElement(By.tagName("a"));
    - driver.findElement(By.xpath("//a"));
  - How to get all the edit box available in UI?
    - driver.findElement(By.xpath("//input[@type = 'text']));

### Interview Question

- ① How to get all the links available in UI?
  - driver.findElement(By.tagName("a"));
  - driver.findElement(By.xpath("//a"));
- ② TestcaseName : Capture all the link available in UI
  - Step1 : Go to "Google"
  - Step2 : Find links available in Google
  - Step3 : Capture all the links available in UI & display in console

- ② Test Case Name :- Capture all message subject in inbox to Gmail Application
- Step 1 : Login to Gmail Application
- Step 2 : Capture all message subject available in Dynamic Web table.

### AUTO SUGGEST



But this is Empact not a drop down

<input type='text' autocomplete='on' />  
 ↳ due to this attribute its gives suggestion  
 ↳ suggestion list  
 ↳ on table written

</table>

### Install Xpath checker

- Download xpathchecker
- First link & click on Add to Firefox

In Google search,

(multiple editor is other editor)  
 ↳ while using Fire path we may not able to find  
 So we use xpath checker, & right click on that web element → view xpath → it generates id to use

### Program (Auto completed box)

```
public class GoogleSearch
{
 public static void main (String [] args)
 {
 WebDriver driver = new FirefoxDriver();
 driver.get("http://google.com");
 driver.findElement(By.id("q")).sendKeys
 ("cognizant careers", Keys.ENTER);
 }
}
```

Instead of using Actions  
 class & perform Enter operation  
 In WebElement sendKeys we can also use enum Keys class to perform Keyboard operation

```
public class GoogleSearch
{
 public static void main (String [] args)
 {
 WebDriver driver = new FirefoxDriver();
 driver.get("http://google.com");
 }
}
```

```
public static void main (String [] args)
{
 WebDriver driver = new FirefoxDriver();
 driver.get("http://google.com");
 driver.findElement(By.id("q")).sendKeys
 ("cognizant");
}
```

```
driver.findElement(By.xpath("//b[text()='Cognizant']"));
- etc])); click();
```

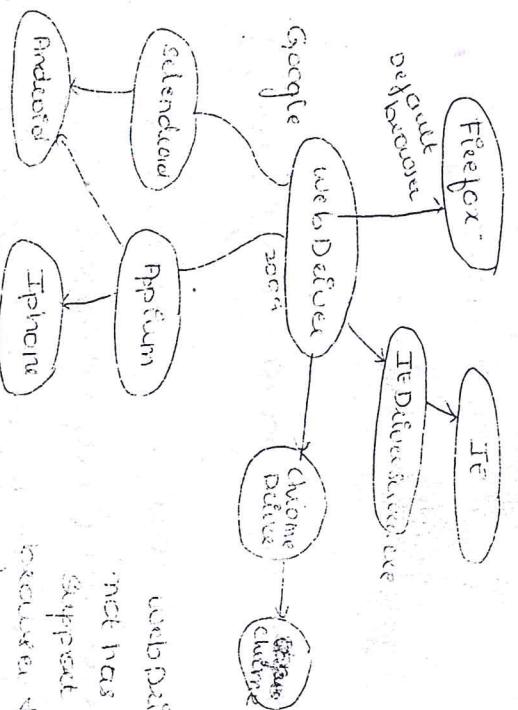
## NOTE

- \* Auto Complete Edit Box or dropdown box which is implemented using input html tag uses auto-complete attribute.
- \* True case two ways we can work with auto completed edit box

- o Write complete string using sendKeys method
- o Enter keyBoard operation

- o Write part of a string using sendKeys and go for click operation to perform click action on suggested list.

## Working with Multiple Browsers



## NOTE

- \* Firefox is a default browser for a webDriver, in order to work with other than Firefox we should download browser specific driver & run the server before launching the driver.

- \* Webdriver will not have direct support with mobile platforms. In order to interact mobile application we should take help of Selenium & Appium tools (ex) Java

## Steps to work with Internet Explorer Browser

Step 1: Download IE Driver Server from Selenium Community

- Go to Selenium community , Find Internet Explorer Driver Server version & click on windows ie link based on Operation System

→ Unzip the folder

Step 2: Run the IE Driver server before launching the browser

- Screenshot & Properties ("webdrivers\ie\_driver\ie\_driver.exe")
- Run "Desktop\New\IE Driver Server.exe"
- Used in Desktop\New\IE Driver Server.exe

Step 3: We have to configure few settings in IE Driver

- ① Browser zoom level should be 100%

③ Enclose protected mode checkbox unchecked  
checked for all other alone

F12 → Dev Tools

Tools → Security option in Internet Options

③ security level for all the zone should be Same  
Go to Tools of IE browser → Internet Option →  
click on security tab

Steps to work with Chrome Browser  
Pulaser class Google Search Box

public static void main (String[] args)

```
System.setProperty ("webdriver.chrome.driver" ,
"C:\\Users\\Deepu\\Desktop\\new\\chromedriver
exe");
Webdriver driver = new ChromeDriver();
driver.get ("http://google.com");
driver.findElement (By.name ("q")).sendKeys
("cognizant");
driver.findElement (By.xpath ("//b[text() =
'cognizant']")).click();
```

driver.pop up is known as Because native pop up

Firefox Profile

→ Set Profile ( )  
No need to change anything in chrome browser

Note

\* In order to view html source code in chrome  
g. IE (dev) browser better to go for Envelope  
Developer tool

\* In order to enable envelope tool, Press F12 to  
get inspect feature  


Using Developer tool, we cannot usually open

File Attachment and Download

Scenario : Download Zip file  
Step 1 : Navigate to Selenium community ,  
Download webdriver zip file

→ when we start to download a file, we get a  
pop up, which is not implemented using selenium  
webdriver does not know to work with  
this pop up but it can ignore this pop up.  
This pop up is known as Because native pop up.

Firefox Profile

→ Set Profile ( )  
→ Set Assume Unchecked Certificate Trust ( )  
→ Select Extension ( )