

February '19	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28		

## INDEX

Dates

\* Ref → 08-09/01 Month

January  
Tuesday  
2019

01

Wk-01 • 001-364

## CONTENTS (BASIC SELENIUM)

S.No	Content	Date
08	1 Introduction to Automation, Advantages, Disadvantages	08-09 /01
	2 Intro to Selenium, Versions, Different Tools used	10-12 /01
09	3 Types of Applications	12/01
	4 Java - Selenium Architecture	14/01
10	5 First Ever Selenium Test Script	15/01
	6 WebDriver Architecture & Example	16,17 /01
11	7 WebDriver Abstract Methods with Examples	18-25 /01
	8 Intro to HTML, Basics, HTML Source Codes	25-30 /01
12	9 Inspection Methods, Locators & its types,	31/01 - 25/02
*	10 Test Case to LOGIN into an Application to Test Script	26-28 /02
01	11 Synchronization, Types, Examples	28/02 - 07/03
12	12 Handling Dropdown	07-12 /03
03	13 Handling Duplicates, Dynamic Dropdown	13-18 /03
14	14 Handling Keyboard and mouse hover actions	18-22 /03
05	15 Handling Scroll Down	23-27 /03
16	16 Handling Disabled Element	27-29 /03
*	17 Taking Screenshots	30-31 to 04-4
18	18 Handling Web Table	04-06 /04
05	19 Handling Pop-ups	08-17 /04
20	20 Handling Frames	17-20 /04
21	21 Handling new Window   tab   child browser popup	20-25 /04
22	22 Methods of WebElement Interface	25-30 /04
07		

## ADVANCED SELENIUM

1.	Automation Framework: Advantages, stages, Types	02-03 /05
2.	Data-Driven Framework	03-13 /05
3.	Key-Word-Driven Framework	14-24 /05
4.	StaleElementReferenceException, Page Object Model (POM)	24-28 /05
5.	Method-Driven Framework	28/05 - 01/06
6.	TestNG : Introduction	01-05 /06
7.	Flags of TestNG	06-11 /06

02

January

Wednesday

Wk-01 • 002-363

2019

JANUARY '19

		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		
S	M	T	W	T	F	S

Ref.

## CONTENTS

8	Annotations of TestNG	11-18 /06
9	Assertion in TestNG	18-20 /06
10	Data-Driven Concept, Grouping Execution in TestNG	20-24 /06
11.	Data Parameterisation in TestNG	28/06
12	Parallel Execution in TestNG	29/06- 06/07
13.	Modular Framework with Example	08-17 /07
14.	Hybrid Framework Architecture & Notes	17-29 /07
15	GitHub	29/07- 01/08
16.	Maven	01-06 /08
17.	Jenkins	06-09 /08

				1	2
3	4	5	6	7	8
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	29
S	M	T	W	T	F

## AUTOMATION

### TESTING

January

Tuesday

08

25.08.21

ID1

2019

Wk-02 • 008-357

Process of testing the Functionality of an Application by the help of automation tools is known as Automation Testing.

(OR)

Converting manual test cases into automation scripts, executing them through a framework and getting test results is known as Automation Testing.

Q: Why do we Switch to Automation Testing? (or)

\* Advantages of Automation Testing.

- We Switch to Automation Testing when we have to do regression testing.
- While regression testing, there are lot of repetitive tasks and performing them manually is a boring job.
- Also in regression, when the product size increases, time taken to test also will increase.
- Hence we switch to automation to reduce the time taken for testing.
- To reduce the manpower / number of resources.
- To obtain the test results faster.
- To get a quick ROI (Returns on Investment).
- We can expect Accurate results.
- Due to tough Competition, Software Companies need to deliver high quality product within less time. Hence we switch to automation testing.

\* Drawbacks / Limitations of Automation Testing:

- 100% Automation testing is not possible
- Anything which is Completely dynamic can't be automated.

# 09

January  
Wednesday

Wk-02 • 009-356

2019

JANUARY '19		1	2	3	4	5
S	M	T	W	T	F	S
		6	7	8	9	10
		13	14	15	16	17
		20	21	22	23	24
		27	28	29	30	31

→ Anything which requires manual intervention (interaction) cannot be automated eg.

What all cannot be automated ..

1. OTP, Captcha

2. Making payments through swiping the debit/credit cards.

3. Barcode Scanner, QR Code Scanning

4. Animation

5. Testing the quality of audio/video files

6. Game testing.

→ We need skilled resources to do automation testing.

→ Cost involved in automation testing is more

Compared to manual testing, because tools are licenced and automation skilled resources

are paid higher.

→ We can't automate an unstable product, we have to wait for the product to become stable.

Note: Newly built features cannot be automated.

# SELENIUM AUTOMATION

January  
Thursday

10

19	3	4	5	6	7	8	9
20	10	11	12	13	14	15	16
21	17	18	19	20	21	22	23
22	24	25	26	27	28		
S	M	T	W	T	F	S	

TOOL

26.08.21

D2

2019

Wk-02 • 010-355

Q: What is Selenium?

Selenium is a free, Open-source automation tool. It can automate web-based applications, but can't automate stand-alone and Client-Server applications.

Appium Tool : Used to automate mobile apps.

Winnium Tool, Auto IT, QTP : Used to automate stand alone apps

QTP : Quick Test Professional

→ Selenium jar file can be downloaded from <https://www.selenium.dev/downloads>.

→ Source Code is visible from <https://github.com/SeleniumHQ/Selenium>  
<https://github.com/SeleniumHQ/java-client>  
<https://org/openqa/selenium>

Quick Notes :

Developed by - Jason Huggins

Year - 2004

Company (worked in) - Thoughtworks, Chicago, USA

Former Name - Javascript Test Runner

Renamed as - Selenium

Reason (for renaming) - To compete with HP - Mercury Automation Tool.

Selenium Versions / Components / Flavours :

1. Selenium Core
2. Selenium IDE Record and playback tool
3. Selenium RC (Remote Control), also called as Selenium 1.0,
4. Selenium Webdriver (2007) / Selenium 2.0.

11

January

Friday

2019

Wk-02 • 011-354

JANUARY '19	1	2	3	4	5	6	7	8	9	10	11	12
S	M	T	W	T	F	S						
1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31								

Current Stable Version - 3.141.59  
Upcoming (Alpha) - 4.0

## 09 5. Selenium Grid

Note :

Selenese : Selenium Commands which are used in Selenium IDE

→ Se Supports all the programming languages -  
Java, C#, Javascript, Perl, Ruby, Python, R, TCL, Elixir, Haskell.

→ Se Supports all the browsers :  
Google Chrome, Firefox, Opera, Safari

→ Se Supports all the Operating Systems :  
Windows, Mac, Linux, except for Unix?

### 04 TOOLS :

#### \* Automation Testing Tools :

Selenium, QTP, Test Optimize, Cypress, Silk Test, Appium, Winnium, Test Complete, Auto IT, TosCA

#### \* Defect Tracking Tools :

Jira, Bugzilla, Bugzero, Mantis, Bug-genic, rational clear, quest.

#### \* Test Management Tools :

Test Rail, Jira, QTest, Test Manager, QC (quality Center), ALM (Application Life Cycle Management)

FEBRUARY '19	
3	4
10	11
17	18
24	25
S	M
T	T
W	F
2	S

January

Saturday

2019

12

Wk-02 • 012-353

## \* Performance Testing Tools :

Load Runner, JMeter, Neoload, Web load.

## \* Read Softwares & Files for Automating :

JDK, IDE (Eclipse), Selenium jar file.

### Types of Applications : [D3] 27.08.21

1. Web based Applications : A web based application is any program that is accessed over a network connection using HTTP, rather than existing within a device's memory. It often runs inside a web browser. Web based applications are also known as Web Apps.

### 2. Client Server Applications :

It consists of a client program that consumes services provided by a server program. The client requests services from the server by calling functions in the server application.

### 3. Stand Alone Applications :

It is an application that runs locally on the device and doesn't require anything else to be functional. All the logic is built into the app, so it doesn't need an internet connection nor any other services installed. These type of applications are not bound to any specific platform.. 13 Sunday

14

January  
Monday  
2019

Wk-03 \* 014-351

00009221

PDA

	1	2	3	4	5
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	
S	M	T	W	T	F

## JAVA - SELENIUM ARCHITECTURE :

08

Client/Language  
Bindings

Selenium

09

JAVA

JSON

WIRE

PROTOCOL

webdriver

API

chromedriver

.exe

Google chrome

geckodriver

.exe

Mozilla

Firefox

10

11

12

JSON - Javascript Object Notation

API - Application programming Interface.

→ Selenium webdriver API Supports all the programming languages and it Communicates with language bindings like java by using JSON - Wire protocol.

→ These two Combined to Communicate with the browsers by using respective driver executable files like chromedriver.exe, geckodriver.exe etc.

				1	2
3	4	5	6	7	8 9
10	11	12	13	14	15 16
17	18	19	20	21	22 23
24	25	26	27	28	F S

01.09.21

D4

January

Tuesday

2019

15

Wk-03 • 015-350

## Q. Opening browser and closing it through automation.

```
08 package aisp;
```

```
09 import org.openqa.selenium.chrome.ChromeDriver;
```

```
10 import org.openqa.selenium.firefox.FirefoxDriver;
```

```
11 public class LaunchBrowserTest
```

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

```
13 // Open the browser
```

```
14 // Just Create an object of browser class
```

```
15 // Set the System property
```

```
16 // Key means type of browser
```

```
17 // Value means path of the driver.exe file
```

```
18 System.setProperty("webdriver.chrome.driver",
```

```
19 "D:\\kcsmc\\AutomationProject\\drivers\\chrome
```

```
20 driver.exe");
```

```
21 new ChromeDriver().close();
```

```
22 System.setProperty("webdriver.gecko.driver",
```

```
23 "D:\\kcsmc\\AutomationProject\\drivers\\geckodriver
```

```
24 .exe");
```

```
25 FirefoxDriver driver = new FirefoxDriver();
```

```
26 driver.close();
```

```
27 }
```

```
28 }
```

```
29 Output: Info: Detected dialect: W3CDOM
```

16

January  
Wednesday  
2019

Wk-03 • 016-349

02.09.21

DS

JANUARY '19	6	7	1	2	3	4	5
	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
	27	28	29	30	31		
	S	M	T	W	T	F	S



## WEB DRIVER ARCHITECTURE

08

09

10

11

12

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

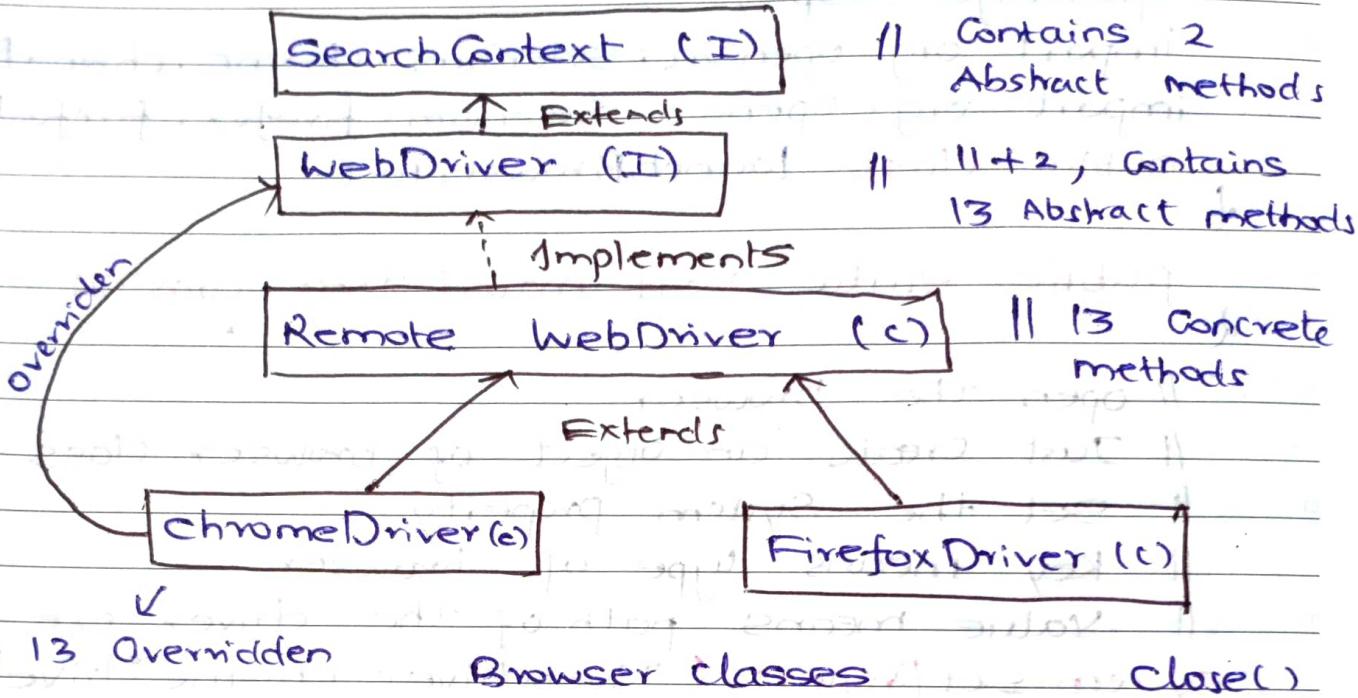
27

28

29

30

31



Explanation: SearchContext is the super most interface, WebDriver interface extends it and all the 13 abstract methods of WebDriver are given implementation in RemoteWebDriver class and all those concrete methods are overridden in respective browser classes like ChromeDriver class, FirefoxDriver class etc.

→ As per Se standards, we always upcast our browser classes to WebDriver interface to achieve

1. Generalization (not specifying any type)
2. Runtime Polymorphism (At runtime, we can decide in which browser my code will run)
3. To get all the 13 methods required for

				1	2
3	4	5	6	7	8
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	
S	M	T	W	T	F

January

Thursday

2019

17

Wk-03 • 017-348

## Automation Testing.

Q: How to open the browser?

To open any browser, we need to create an object of that browser class.

```
new ChromeDriver().close();
```

## Runtime Polymorphism: (Program)

```
package arsp;
import java.util.Scanner;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class RuntimePolymorphismInSelenium
{
    public static void main(String args[])
    {
        System.out.println("Enter Browser Name:");
        Scanner Sc = new Scanner(System.in);
        String Str = Sc.next();
        WebDriver driver = null;
        if (Str.equals("chrome"))
        {
            System.setProperty("webdriver.chrome.driver",
".\drivers\chromedriver.exe");
            driver = new ChromeDriver();
        }
    }
}
```

18

January  
Friday

Wk-03 • 018-347

2019

JANUARY '19	1	2	3	4	5
S	M	T	W	T	F
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	

8      }      else if ( str. equals ( " Firefox " ) )  
 9      {  
 10        System. SetProperty ( " webdriver. gecko. driver " ,  
 11                    ". \ drivers \ geckodriver. exe " );  
 12        driver = new FirefoxDriver();  
 13     }  
 14     else  
 15        {  
 16         System. out. println ( " Give proper browser name " );  
 17         driver. close();  
 18     }  
 19     }  
 20     }  
 21     }  
 22     }  
 23     }  
 24     }  
 25     }  
 26     }  
 27     }  
 28     }  
 29     }  
 30     }  
 31     }  
 32     }  
 33     }  
 34     }  
 35     }  
 36     }  
 37     }  
 38     }  
 39     }  
 40     }  
 41     }  
 42     }  
 43     }  
 44     }  
 45     }  
 46     }  
 47     }  
 48     }  
 49     }  
 50     }  
 51     }  
 52     }  
 53     }  
 54     }  
 55     }  
 56     }  
 57     }  
 58     }  
 59     }  
 60     }  
 61     }  
 62     }  
 63     }  
 64     }  
 65     }  
 66     }  
 67     }  
 68     }  
 69     }  
 70     }  
 71     }  
 72     }  
 73     }  
 74     }  
 75     }  
 76     }  
 77     }  
 78     }  
 79     }  
 80     }  
 81     }  
 82     }  
 83     }  
 84     }  
 85     }  
 86     }  
 87     }  
 88     }  
 89     }  
 90     }  
 91     }  
 92     }  
 93     }  
 94     }  
 95     }  
 96     }  
 97     }  
 98     }  
 99     }  
 100     }  
 101     }  
 102     }  
 103     }  
 104     }  
 105     }  
 106     }  
 107     }  
 108     }  
 109     }  
 110     }  
 111     }  
 112     }  
 113     }  
 114     }  
 115     }  
 116     }  
 117     }  
 118     }  
 119     }  
 120     }  
 121     }  
 122     }  
 123     }  
 124     }  
 125     }  
 126     }  
 127     }  
 128     }  
 129     }  
 130     }  
 131     }  
 132     }  
 133     }  
 134     }  
 135     }  
 136     }  
 137     }  
 138     }  
 139     }  
 140     }  
 141     }  
 142     }  
 143     }  
 144     }  
 145     }  
 146     }  
 147     }  
 148     }  
 149     }  
 150     }  
 151     }  
 152     }  
 153     }  
 154     }  
 155     }  
 156     }  
 157     }  
 158     }  
 159     }  
 160     }  
 161     }  
 162     }  
 163     }  
 164     }  
 165     }  
 166     }  
 167     }  
 168     }  
 169     }  
 170     }  
 171     }  
 172     }  
 173     }  
 174     }  
 175     }  
 176     }  
 177     }  
 178     }  
 179     }  
 180     }  
 181     }  
 182     }  
 183     }  
 184     }  
 185     }  
 186     }  
 187     }  
 188     }  
 189     }  
 190     }  
 191     }  
 192     }  
 193     }  
 194     }  
 195     }  
 196     }  
 197     }  
 198     }  
 199     }  
 200     }  
 201     }  
 202     }  
 203     }  
 204     }  
 205     }  
 206     }  
 207     }  
 208     }  
 209     }  
 210     }  
 211     }  
 212     }  
 213     }  
 214     }  
 215     }  
 216     }  
 217     }  
 218     }  
 219     }  
 220     }  
 221     }  
 222     }  
 223     }  
 224     }  
 225     }  
 226     }  
 227     }  
 228     }  
 229     }  
 230     }  
 231     }  
 232     }  
 233     }  
 234     }  
 235     }  
 236     }  
 237     }  
 238     }  
 239     }  
 240     }  
 241     }  
 242     }  
 243     }  
 244     }  
 245     }  
 246     }  
 247     }  
 248     }  
 249     }  
 250     }  
 251     }  
 252     }  
 253     }  
 254     }  
 255     }  
 256     }  
 257     }  
 258     }  
 259     }  
 260     }  
 261     }  
 262     }  
 263     }  
 264     }  
 265     }  
 266     }  
 267     }  
 268     }  
 269     }  
 270     }  
 271     }  
 272     }  
 273     }  
 274     }  
 275     }  
 276     }  
 277     }  
 278     }  
 279     }  
 280     }  
 281     }  
 282     }  
 283     }  
 284     }  
 285     }  
 286     }  
 287     }  
 288     }  
 289     }  
 290     }  
 291     }  
 292     }  
 293     }  
 294     }  
 295     }  
 296     }  
 297     }  
 298     }  
 299     }  
 300     }  
 301     }  
 302     }  
 303     }  
 304     }  
 305     }  
 306     }  
 307     }  
 308     }  
 309     }  
 310     }  
 311     }  
 312     }  
 313     }  
 314     }  
 315     }  
 316     }  
 317     }  
 318     }  
 319     }  
 320     }  
 321     }  
 322     }  
 323     }  
 324     }  
 325     }  
 326     }  
 327     }  
 328     }  
 329     }  
 330     }  
 331     }  
 332     }  
 333     }  
 334     }  
 335     }  
 336     }  
 337     }  
 338     }  
 339     }  
 340     }  
 341     }  
 342     }  
 343     }  
 344     }  
 345     }  
 346     }  
 347     }  
 348     }  
 349     }  
 350     }  
 351     }  
 352     }  
 353     }  
 354     }  
 355     }  
 356     }  
 357     }  
 358     }  
 359     }  
 360     }  
 361     }  
 362     }  
 363     }  
 364     }  
 365     }  
 366     }  
 367     }  
 368     }  
 369     }  
 370     }  
 371     }  
 372     }  
 373     }  
 374     }  
 375     }  
 376     }  
 377     }  
 378     }  
 379     }  
 380     }  
 381     }  
 382     }  
 383     }  
 384     }  
 385     }  
 386     }  
 387     }  
 388     }  
 389     }  
 390     }  
 391     }  
 392     }  
 393     }  
 394     }  
 395     }  
 396     }  
 397     }  
 398     }  
 399     }  
 400     }  
 401     }  
 402     }  
 403     }  
 404     }  
 405     }  
 406     }  
 407     }  
 408     }  
 409     }  
 410     }  
 411     }  
 412     }  
 413     }  
 414     }  
 415     }  
 416     }  
 417     }  
 418     }  
 419     }  
 420     }  
 421     }  
 422     }  
 423     }  
 424     }  
 425     }  
 426     }  
 427     }  
 428     }  
 429     }  
 430     }  
 431     }  
 432     }  
 433     }  
 434     }  
 435     }  
 436     }  
 437     }  
 438     }  
 439     }  
 440     }  
 441     }  
 442     }  
 443     }  
 444     }  
 445     }  
 446     }  
 447     }  
 448     }  
 449     }  
 450     }  
 451     }  
 452     }  
 453     }  
 454     }  
 455     }  
 456     }  
 457     }  
 458     }  
 459     }  
 460     }  
 461     }  
 462     }  
 463     }  
 464     }  
 465     }  
 466     }  
 467     }  
 468     }  
 469     }  
 470     }  
 471     }  
 472     }  
 473     }  
 474     }  
 475     }  
 476     }  
 477     }  
 478     }  
 479     }  
 480     }  
 481     }  
 482     }  
 483     }  
 484     }  
 485     }  
 486     }  
 487     }  
 488     }  
 489     }  
 490     }  
 491     }  
 492     }  
 493     }  
 494     }  
 495     }  
 496     }  
 497     }  
 498     }  
 499     }  
 500     }  
 501     }  
 502     }  
 503     }  
 504     }  
 505     }  
 506     }  
 507     }  
 508     }  
 509     }  
 510     }  
 511     }  
 512     }  
 513     }  
 514     }  
 515     }  
 516     }  
 517     }  
 518     }  
 519     }  
 520     }  
 521     }  
 522     }  
 523     }  
 524     }  
 525     }  
 526     }  
 527     }  
 528     }  
 529     }  
 530     }  
 531     }  
 532     }  
 533     }  
 534     }  
 535     }  
 536     }  
 537     }  
 538     }  
 539     }  
 540     }  
 541     }  
 542     }  
 543     }  
 544     }  
 545     }  
 546     }  
 547     }  
 548     }  
 549     }  
 550     }  
 551     }  
 552     }  
 553     }  
 554     }  
 555     }  
 556     }  
 557     }  
 558     }  
 559     }  
 560     }  
 561     }  
 562     }  
 563     }  
 564     }  
 565     }  
 566     }  
 567     }  
 568     }  
 569     }  
 570     }  
 571     }  
 572     }  
 573     }  
 574     }  
 575     }  
 576     }  
 577     }  
 578     }  
 579     }  
 580     }  
 581     }  
 582     }  
 583     }  
 584     }  
 585     }  
 586     }  
 587     }  
 588     }  
 589     }  
 590     }  
 591     }  
 592     }  
 593     }  
 594     }  
 595     }  
 596     }  
 597     }  
 598     }  
 599     }  
 600     }  
 601     }  
 602     }  
 603     }  
 604     }  
 605     }  
 606     }  
 607     }  
 608     }  
 609     }  
 610     }  
 611     }  
 612     }  
 613     }  
 614     }  
 615     }  
 616     }  
 617     }  
 618     }  
 619     }  
 620     }  
 621     }  
 622     }  
 623     }  
 624     }  
 625     }  
 626     }  
 627     }  
 628     }  
 629     }  
 630     }  
 631     }  
 632     }  
 633     }  
 634     }  
 635     }  
 636     }  
 637     }  
 638     }  
 639     }  
 640     }  
 641     }  
 642     }  
 643     }  
 644     }  
 645     }  
 646     }  
 647     }  
 648     }  
 649     }  
 650     }  
 651     }  
 652     }  
 653     }  
 654     }  
 655     }  
 656     }  
 657     }  
 658     }  
 659     }  
 660     }  
 661     }  
 662     }  
 663     }  
 664     }  
 665     }  
 666     }  
 667     }  
 668     }  
 669     }  
 670     }  
 671     }  
 672     }  
 673     }  
 674     }  
 675     }  
 676     }  
 677     }  
 678     }  
 679     }  
 680     }  
 681     }  
 682     }  
 683     }  
 684     }  
 685     }  
 686     }  
 687     }  
 688     }  
 689     }  
 690     }  
 691     }  
 692     }  
 693     }  
 694     }  
 695     }  
 696     }  
 697     }  
 698     }  
 699     }  
 700     }  
 701     }  
 702     }  
 703     }  
 704     }  
 705     }  
 706     }  
 707     }  
 708     }  
 709     }  
 710     }  
 711     }  
 712     }  
 713     }  
 714     }  
 715     }  
 716     }  
 717     }  
 718     }  
 719     }  
 720     }  
 721     }  
 722     }  
 723     }  
 724     }  
 725     }  
 726     }  
 727     }  
 728     }  
 729     }  
 730     }  
 731     }  
 732     }  
 733     }  
 734     }  
 735     }  
 736     }  
 737     }  
 738     }  
 739     }  
 740     }  
 741     }  
 742     }  
 743     }  
 744     }  
 745     }  
 746     }  
 747     }  
 748     }  
 749     }  
 750     }  
 751     }  
 752     }  
 753     }  
 754     }  
 755     }  
 756     }  
 757     }  
 758     }  
 759     }  
 760     }  
 761     }  
 762     }  
 763     }  
 764     }  
 765     }  
 766     }  
 767     }  
 768     }  
 769     }  
 770     }  
 771     }  
 772     }  
 773     }  
 774     }  
 775     }  
 776     }  
 777     }  
 778     }  
 779     }  
 780     }  
 781     }  
 782     }  
 783     }  
 784     }  
 785     }  
 786     }  
 787     }  
 788     }  
 789     }  
 790     }  
 791     }  
 792     }  
 793     }  
 794     }  
 795     }  
 796     }  
 797     }  
 798     }  
 799     }  
 800     }  
 801     }  
 802     }  
 803     }  
 804     }  
 805     }  
 806     }  
 807     }  
 808     }  
 809     }  
 810     }  
 811     }  
 812     }  
 813     }  
 814     }  
 815     }  
 816     }  
 817     }  
 818     }  
 819     }  
 820     }  
 821     }  
 822     }  
 823     }  
 824     }  
 825     }  
 826     }  
 827     }  
 828     }  
 829     }  
 830     }  
 831     }  
 832     }  
 833     }  
 834     }  
 835     }  
 836     }  
 837     }  
 838     }  
 839     }  
 840     }  
 841     }  
 842     }  
 843     }  
 844     }  
 845     }  
 846     }  
 847     }  
 848     }  
 849     }  
 850     }  
 851     }  
 852     }  
 853     }  
 854     }  
 855     }  
 856     }  
 857     }  
 858     }  
 859     }  
 860     }  
 861     }  
 862     }  
 863     }  
 864     }  
 865     }  
 866     }  
 867     }  
 868     }  
 869     }  
 870     }  
 871     }  
 872     }  
 873     }  
 874     }  
 875     }  
 876     }  
 877     }  
 878     }  
 879     }  
 880     }  
 881     }  
 882     }  
 883     }  
 884     }  
 885     }  
 886     }  
 887     }  
 888     }  
 889     }  
 890     }  
 891     }  
 892     }  
 893     }  
 894     }  
 895     }  
 896     }  
 897     }  
 898     }  
 899     }  
 900     }  
 901     }  
 902     }  
 903     }  
 904     }  
 905     }  
 906     }  
 907     }  
 908     }  
 909     }  
 910     }  
 911     }  
 912     }  
 913     }  
 914     }  
 915     }  
 916     }  
 917     }  
 918     }  
 919     }  
 920     }  
 921     }  
 922     }  
 923     }  
 924     }  
 925     }  
 926     }  
 927     }  
 928     }  
 929     }  
 930     }  
 931     }  
 932     }  
 933     }  
 934     }  
 935     }  
 936     }  
 937     }  
 938     }  
 939     }  
 940     }  
 941     }  
 942     }  
 943     }  
 944     }  
 945     }  
 946     }  
 947     }  
 948     }  
 949     }  
 950     }  
 951     }  
 952     }  
 953     }  
 954     }  
 955     }  
 956     }  
 957     }  
 958     }  
 959     }  
 960     }  
 9

				1	2
3	4	5	6	7	8
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	23
S	M	T	W	T	F

January  
Saturday

2019

19

Wk-03 019-346

Method Type	S.No	Method Name	Return type	Method Use
Data Capture Methods (5 & 6)	5	getWindowHandle()	String	Used to get the window handle of the current browser window
Browser Window Handling method (7)	6	getWindowHandles()	Set<String>	Used to get window handle of all the browser windows
Navigation Method (8)	7	manage()	Options	Used to manage the browser window
Inspection Methods (9 & 10)	8	navigate()	Navigation	Used to navigate from one page to another previous page, next page, Can refresh current webpage
Control Switching method (11)	9	findElement(By arg)	WebElement	Find the particular element on the web page
Termination methods (12 & 13)	10	findElements(By arg)	List<WebElement>	Find multiple elements on the webpage.
	11	SwitchTo()	Target Locator	Used to switch over location from webpage to pop-ups, frames, windows etc
	12	close()	void	close the current browser window
	13	quit()	void	close all the browser windows opened by Selenium

21

January  
Monday

Wk-04 • 021-344

2019

JANUARY '19	1	2	3	4	5
S	M	T	W	F	S
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	

## Program on 13 methods - Concept

```

08
09 package aisp;
10 import org.openqa.selenium.Dimension;
11 import org.openqa.selenium.Point;
12 import org.openqa.selenium.WebDriver;
13 import org.openqa.selenium.chrome.ChromeDriver;
14
15 public class WebDriverInterfaceMethods
16 {
17     public static void main(String args[]) throws
18         InterruptedException
19     {
20         System.setProperty("webdriver.chrome.driver",
21             "./drivers/chromedriver.exe");
22
23         WebDriver driver = new ChromeDriver();
24         driver.get("https://www.google.com/");
25         String pageURL = driver.getCurrentUrl();
26         System.out.println(pageURL);
27
28         if (pageURL.equals("https://www.monsterIndia.com"))
29         {
30             System.out.println("MonsterIndia URL is
31                 entered, PASS");
32         }
33         else
34         {
35             System.out.println("monster India URL is not entered, FAIL");
36         }
37
38         String Pgsrc = driver.getPageSource();
39         System.out.println(Pgsrc);
40     }
41 }

```

				1	2
3	4	5	6	7	8
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	
5	M	T	W	F	S

January

Thursday

2019

Wk-04 • 024-341

24

```

if (pgsrc .contains ("Gmail"))
{
    if (sop ("Gmail link is present, PASS"));
}
else
{
    sop ("Gmail link is not present, FAIL");
}

String title = driver.getTitle();
sop (title);
if (title.equals ("Google"))
{
    if (sop ("Google page is displayed, PASS"));
}
else
{
    sop ("Google page is not displayed, FAIL");
}

driver.manage().window().maximize();
Thread.sleep (3000);
Dimension d = new Dimension (300, 400);
driver.manage().window().setSize (d);
Thread.sleep (3000);
Point P = new Point (600, 400);
driver.manage().window().setPosition (P);
}

```

25

January  
Friday

Wk-04 • 025-340

2019

04.09.2021

D7

JANUARY '19	1	2	3	4	5
S	M	T	W	T	F
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	

## DIFFERENCE BETWEEN get() & navigate()

08

get()

1. Enter URL

2. get() will not take advantage of Browser history

3. get() will wait until Complete page is loaded.

navigate()

1. Enter URL

Navigate to Previous page

Navigate to Next page

Refresh the Current page

2. navigate() will take advantage of browser history

3. navigate() will not wait until the complete page is loaded

02

## HYPER TEXT MARKUP LANGUAGE (HTML)

03

Scripting language used to develop web element.

In HTML, there are 3 required things.

05

1. Tag

2. Attribute

3. Text

Rules:

- tag and attribute should be within <>
- Text should be outside <>
- Always start from <html>
- Once a tag is opened, close it by using /tag.

Any editor - Save as .html

FEBRUARY '19
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
1
2

January

Saturday

2019

26

Wk-04 • 026-339

```

08 <html>
09   <head> Text
10   <title> My First Page </title>
11 </head> Link
12 <body> Text
13   <a href = 'https://www.facebook.com'>
14     facebook </a>
15   </body>
16 </html>

```

→ href is an attribute of anchor tag [hyper text reference].

It is used to specify the URL of the page that the link goes to this attribute is used with anchor <a> tag

→ <a> - the anchor tag is used to create a hyperlink on the webpage. This hyperlink is used to link the webpage to other webpage. href attribute which indicates link's destination.

<a href = 'https://www.w3schools.com' target = '\_blank'

attribute Link

visit w3schools.com </a>.

27 Sunday

28

January

Monday

Wk-05 • 028-337

2019

JANUARY '19	1	2	3	4	5
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	
S	M	T	W	T	F

## BASICS OF HTML

08

Tag	Description
<!DOCTYPE>	Defines the document
<html>	Defines html document
<head>	Contains meta data / information for document.
<body>	Defines document body
<h1> to <h6>	Defines HTML heading
<p>	Defines a paragraph
 	Inserts a single line break
  [not next]	Defines a thematic change in the content.
<!-- -->	Defines a comment

03

## FORMS AND INPUT

Tag	Description
<input>	Defines an input control
<text area>	Defines a multiple input control (text area)
<button>	Defines a clickable button
<select>	Defines a dropdown list
<option>	Defines an option in dropdown.
<label>	Defines label for an <input> element.

FEBRUARY-'19							1	2
	3	4	5	6	7	8	9	
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28			
S	M	T	W	T	F			S

January

Tuesday

2019

29

Wk-05 • 029-336

Code - paragraph

<h1> heading 1 </h1>

< h2 > heading 2 </h2>

< h6 > heading 6 < /h6 >

**Bold:** < b > statements - < /b >

<sup>12</sup> Italic : <i> Statements -- </i>

Small (Text) tag : <small> Statements <

01 Small (text) tag : <small> Statements </small>

• **What are the main differences between the two types of government?**

Underline : <u> statements </u>

*Strewn about the ground were many small pieces of dried plant material.*

03 image : <img src = " imagename.jpg " width = 40  
height = "40" border = "0" >

04 *English* *English*

link : <head> ~~empty 'id' or 'name'~~ <input> <script>

05 <link ref = "stylesheet" type = "text"

`href = "style.css"`

06 head>body>head>body>head>body>

## 07 Radial Basis Function Networks

<form method="post" action="/caib/example

↳ Equation methods - for detailed analysis

Select an option : <br>

```
<input type = "radio" name = "option"> option 1
```

`<input type = "radio" name = "option"> option 2`

30

January  
Wednesday

Wk-05 • 030-335

2019

JANUARY '19	6	7	1	2	3	4	5
	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
	27	28	29	30	31		
S	M	T	W	T	F	S	

## Program :

```

08 <html>
09   <head>
10     <title> My Signup page </title>
11   </head>
12   <body>
13     <h1> This is my page </h1>
14     <h2> This is my page </h2>
15     <h3> This is my page </h3>
16     <h4> This is my page </h4>
17     <h5> This is my page </h5>
18
19 User Name <input type='text'> </input> <br>
20 Password <input type='password'> </input> <br>
21 <span> choose your gender </span> <br>
22 <input type='radio' name='gender'> MALE
23 </input> <br>
24 <input type='radio' name='gender'> FEMALE
25 </input> <br>
26 <input type='radio' name='gender'> OTHERS
27 </input> <br>
28
29 <div> choose Subjects , already completed </div>
30 <input type='checkbox'> Manual Testing </input> <br>
31 <input type='checkbox'> SQL </input> <br>
32 <input type='checkbox'> Core Java </input> <br>
33 <input type='checkbox'> Automation </input> <br>
34 <input type='button' value='signup'>
35 </input> <br>
36 </body>
37 </html>

```

FEBRUARY '19
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
1
2
S
M
T
W
T
F
S

07.09.21

D8

January  
Thursday

2019

31

Wk-05 • 031-334

## INSPECTION METHODS

Before performing actions on any element, first we should locate the element on webpage.

Hence to find its address we have to find element using `findElement()` and `findElements()` inside webdriver interface.

`findElement()` - method of webdriver interface is used to get the address of first matching element on the web page.

- Return type of `findElement()` is `<WebElement` which is an interface

- If `findElement()` is not able to find the element on the webpage then we get `NoSuchElementException`.

- `findElement()` takes locators as its argument

### Locators

- Locators are Static methods of abstract 'By' class

- Locators are used to locate one or more elements on the webpage

- Locators act as argument for `findElement()` and `findElements()`

01

February  
Friday

Wk-05 • 032-333

2019

FEBRUARY '19	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28		
	S	M	T	W	T	F	S

## Types of Locators :

1. tagName (String arg)
2. id (String arg)
3. className (String arg)
4. partialLinkText (String arg)
5. linkText (String arg)
6. cssSelector (String arg)
7. xPath (String arg)

## HTML Code to Create a Link :

```

<html>
  <body>
    <a href = 'https://www.google.com' id = 'i1',
      name = 'n1' class = 'c1' value = 'v1' title = 't1'>
      Google </a>
    </body>
  </html>

```

In the above Code, to create a link, we use `<a>` tag and mandatory attribute called `'href'` but other attributes like `id`, `name`, `class`, `value`, `title` are Optional.

So, in Selenium, we can use features like tag, attributes like `id`, `name` and `class` etc.

So, according to the above example,

MARCH '19	31	1	2
	3	4	5
	10	11	12
	17	18	19
	24	25	26
S	M	T	W

February

Saturday

2019

02

Wk-05 • 033-332

- 08 1. `tagName("a")` - we use tagname of the element
- 09 2. `id("ii")` - we use id of the element
- 09 3. `name("n1")` - we use name of the element
- 10 4. ~~className~~ `("Googly")` - Here we use text but linkText  
make sure element is link  
Only
- 11 5. `className("cl")` - we use class of the element
- 02 6. `partialLinkText ("Go")` - Here we use a part of the complete link text.

Note: Through partial LinkText() locator, we handle partially dynamic linktext.

Program:

```

package arsp;
import org.openqa.Selenium.By;
import org.openqa.Selenium.WebDriver;
import org.openqa.Selenium.WebElement;
import org.openqa.Selenium.Chrome.ChromeDriver;
public class InspectionMethods
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.chrome.driver",
                           "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("file:///D:/KCSMS/HTML/link.htm");
        //on the browser, find the element with
        //tagname as 'a',
        WebElement linkAddress = driver.findElement
            (By.tagName("a"));
    }
}

```

FEBRUARY '19							1	2
3	4	5	6	7	8	9		
10	11	12	13	14	15	16		
17	18	19	20	21	22	23		
24	25	26	27	28				
S	M	T	W	T	F	S		

```
// On the address, perform click action  
linkAddress.click();  
  
// On the webpage, find the element with  
the id as 'ii' And click on it.  
driver.findElement(By.id("ii")).click();  
driver.findElement(By.name("ni")).click();  
driver.findElement(By.className("ci")).click();  
driver.findElement(By.linkText("Facebook")).click();  
  
// Click on the partial link text "Fa"  
driver.findElement(By.partialLinkText("Fa")).click();  
driver.findElement(By.cssSelector("a[title='t1']")).click();  
  
// Get the URL  
driver.get("https://www.google.com/");  
driver.findElement(By.xpath("//input[@name='q']")).sendKeys("raghav");  
  
}  
}
```

## 7. CSS Selector

05 Here we identify any element by giving both tag and attribute.

Syn: tagName [AttributeName = 'AttributeValue']

Eg: (some functions) work faster without using `std::vector`

a [ href = 'https://www.google.com' ]

`a[id] = "in[id]"`

a [name] = 'ni']

`a [class = 'c1']`

a [value = 'v1']

```
a [title = 'hi']
```

• 25. amount

3.2 Shortcut for id is #

Shortcuts:  $\text{ctrl} + \text{shift} + \text{F}$  finds all occurrences of selected text

MARCH '19
31
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
S
M
T
W
T
F
S

February  
Tuesday

2019

05

Wk-06 • 036-329

Shortcut for Class is .

div[class=c2] drag and drop on

div.c2 remains on top of all

elements with class c2

↳ This is how we can select second element with

class name c2.

To verify the cssSelector expression and xpath expression, we download an add-on for Firefox browser called Try Xpath.

Steps:

- Open Firefox browser.
- Go to Google and type 'download try xpath for firefox'.
- Click on 'Add to Firefox' button and click on 'Add' button.
- Try xpath will be installed and visible at top right corner of firefox browser.
- Click on Tx icon on top right corner.
- For verifying cssSelector expression, change way to 'QuerySelectorAll'
- Write the cssSelector expression and click on execute button.

Limitation of cssSelector:

To use cssSelector minimum 1 attribute should be present which is unique.

If no attributes are present, we can't use cssSelector because cssSelector does not support text.

Note: Refer program on left page for example.

3	4	5	6	7	8	9	10	11	12	13	14
17	18	19	20	21	22	23	24	25	26	27	28
S	M	T	W	T	F	S					

## 8. Xpath:

- Xpath is the path travelled in the HTML tree to find an element.
- Xpath is one of the locators which covers all possible ways to find an element.

There are two types of Xpath:

1. Absolute Xpath
2. Relative Xpath

Absolute Xpath means Complete path we travel from Start (html) to element

- It is achieved by '/' (single forward slash) → immediate child / immediate descendant.

HTML Code

&lt;html&gt;

&lt;body&gt;

&lt;div&gt;

A &lt;input type = 'text'&gt; &lt;/input&gt;

B &lt;input type = 'text'&gt; &lt;/input&gt;

&lt;/div&gt;

&lt;div&gt;

C &lt;input type = 'text'&gt; &lt;/input&gt;

D &lt;input type = 'text'&gt; &lt;/input&gt;

&lt;/div&gt;

&lt;/body&gt;

&lt;/html&gt;

31		5	6	7	1	2
3	4	11	12	13	14	15
10			19	20	21	22
17				28	29	30
24	25	26	27			
S	M	T	W	T	F	S

MARCH '19

February

Thursday

2019

07

Wk-06 • 038-327

## HTML Tree Structure

html[1]

body[1]

input A[1]

input B[2]

div[2]

input C[1]

input D[2]

Note: In realtime applications, if we use absolute xpath, it will become very lengthy. So we use relative xpath.

Relative xpath - Shortest path --> achieved by '/' (double forward slash) -->  
 Any child / Any descendant.  
 - It is the shortest path in the HTML tree  
 (Directly jump to required element)

### Elements

ABVD

AB

A

AV

BV

AD

### Absolute Path

html/body/div/input

html/body/div[1]/input

html/body/div[1]/input[1]

html/body/div[1]/input[1]

html/body/div[1]/input[2]; html/body/div[2]/input[1]

html/body/div[1]/input[1]; html/body/div[2]/input[2]

FEBRUARY '19						
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		
S	M	T	W	T	F	S

Elements	Relative xpath
ABVD	// input
AB	// div[1] / input
A	// div[1] input[1]
AV	// input[1]
BV	// div[1] / input[2] ; // div[2] / input[1]
AD	// div[1] / input[1] ; // div[2] / input[2]

Note : Refer 04 Feb page for program.

### Cases of Relative xpath :

Case 1 : Xpath by @Unique attribute

Sym: //tagName[@AttrName='AttrValue']

Eg 1: // input[@name='or'] (Google.com)

Eg 2: // @[@href='https://www.facebook.com/praveendraomusic/] (https://www.yuvadwaja.in/index.html)

Eg 3: // img[@src='images/kunwar Singh\_Kannada.jpg'] (https://www.yuvadwaja.in/index.html)

Eg 4: // input[@name='pwd'] (demo.actitime.com)

Case 2 : Xpath by text() function

Sym: //tagName[text()='textvalue']

Eg 1: // div[text()='login']

Eg 2: // small[text()='Now']

Eg 3: // a[text()='Gmail']

MARCH 19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

February

Saturday

09

Case 3 : xpath by multiple ~~Attributes~~ 040-325

<div fn = 'Deepika' ...>

Sym:

Unique Element (and):

// tagName[@attr1 = 'attrval' and @attr2 = 'attrval' and ...]

Many Elements (or):

// tagName[@attr1 = 'attrval' Or @attr2 = 'attrval' or ...]

Concept:

<div fn = 'Deepika' ln = 'padukone' >

<div fn = 'Deepika' ln = 'Rai' >

<div fn = 'Aishwarya' ln = 'Rai' >

// div [@fn = 'Deepika'] // 2 Counts (but required only 1)

// div [@fn = 'Rai'] // 2 Counts (but required only 1)

So, // div [@fn = 'Deepika' and @ln = 'Rai'] // 1 count

↳ i.e. xpath by multiple attributes to achieve 1 count  
Or uniquely one element.

Eg 1: // input[@type = 'password' and @name = 'pwd']

(demo.actitime.com)

Eg 2: // a [@href = 'index.html' and @class = 'nav-link']

(https://yuvadhuaja.in/)

Eg 3: // div [@class = '\_4IiNRh\_2mtkou' and @title = 'coreiz'] (flipkart.com)

Eg 4: // div [@class = '\_1GEHlw' and text() = 'Coreiz']  
(flipkart.com)

Combination of Text and an attribute:

10 Sunday

Eg: // tagName[@AttrName = 'Attrvalue' and text() = 'textvalue']

Eg: // a [@href = 'index.html' and text() = 'Home']

11

February

Monday

Wk-07 • 042-323

2019

	3	4	5	6	7	8	9
FEBRUARY 19	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28		
	S	M	T	W	T	F	S

Case 4 : xpath by Contains() function

Sol: // tagName[Contains(@AttrName, 'Attrvalue')]

09 Situation:

```

08 < div id = 'ui-id-1'>
09 < div id = 'ui-id-2'>
10 < div id = 'ui-id-3'>
11 < div id = 'ui-id-4'>
12 < div id = 'ui-id-5'>
```

Raghavendra evaluates Raghavendra  
Raghavendra Contains Raghav

02 Write xpath expression to match all the above five:

03 1. a) Handle partially Dynamic Elements  
(partially dynamic attribute value)

Eg: // div [Contains(@id, 'ui-id')]

05 1. b) Handle partially Dynamic Elements  
(partially dynamic text value)

Eg: // div [Contains( text(), 'we have')]

for the example, <div> we have 1,24,456 products </div>

07 2. Avoid writing lengthy text

Eg: // p [Contains( text(), 'Uttara Karnataka')]

3. Avoid spaces present in the values

Eg: // div [Contains( text(), 'Login')]

MARCH '19	31	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	S	M	T	W	T	F	S

February

Tuesday

2019

12

Wk-07 • 043-322

## Case 5 : xpath by axis (relationship)

08

### HTML TREE

09    html    ~~elements that are not part of document tree~~  
      body

10    ~~elements~~ table    ~~child :: parent relationship~~  
      tbody

11    tr id='ii'

12    td A    ~~child :: parent relationship~~  
      td B

01    td C    ~~child :: parent relationship~~  
      td D

02    tr id='ii'    ~~parenting elements~~  
      td E

03    td F    ~~parenting elements~~  
      td G

04    td H

parent --> child

05 // parent / descendant :: which child

06 ① html --> body

// html / descendant :: body

07    ② table --> tr(ii)

// table / descendant :: tr[@id = 'ii']

Child --> parent

// child / ancestor :: which parent

① B --> tr(ii)

// td [text() = 'B'] / ancestor :: body

13

February  
Wednesday

Wk-07 • 044-321

2019

FEBRUARY '19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S	M	T	W	T	F	S																	

② F → body

// td[text() = 'F']/ ancestor :: body

From one element to next element of same element. parent.

// element/following-sibling :: which next element

① A → D

// td[text() = 'A']/ following-sibling :: td[text() = 'D']

From one element to previous element of same parent

// element/preceding-sibling :: which previous element

① C → A

// td[text() = 'C']/ preceding-sibling :: td[text() = 'A']

② i2 → i1

// tr[@id = 'i2']/ preceding-sibling :: tr[@id = 'i1']

Cousin

B → F

// td[text() = 'B']/ ancestor :: tr[@id = 'i1']/ following-sibling :: tr[@id = 'i2']/ descendant :: td[text() = 'F']

	31		4	5	6	7	8	9
MARCH '19	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

February

Thursday

2019

14

Wk-07 • 045-320

## ASSIGNMENT 1 : On Case 1 : xpath by unique attribute

Sym // tagName[@AttrName = 'AttrValue']

- 09 1. //input[@id = 'identifierId'] (Gmail.com)
2. //input[@placeholder = 'password'] (Facebook - login page)
- 10 3. //input[@name = 'verification'] (<https://www.patanjaliayurved.net/customer/login>)
- 11 4. //input[@name = 'Session[password]'] (Twitter login page)
- 5 //input[@id = 'Sign1nsubmit'] (Amazon prime video login page)
6. //input[@id = 'ap\_password'] (Amazon prime video login, passwd)
07. //input[@placeholder = 'Enter your mobile number'] (<https://www.hotstar.com/in/subscribe/signin?returnURL=/in/subscribe/my.account>)
- 02 8. //label[@class = 'checkbox'] (<https://zoom.us/signup>)
9. //a[@href = '/signup'] (<https://zoom.us/signin>)
10. //input[@id = 'join-Confir'] (<https://zoom.us/join>)

## ASSIGNMENT 2 : On Case 2: xpath by text() function

Sym // tagName[@text() = 'textvalue']

- 01 1. //span[text() = 'Get access to your orders, Wishlist and recommendations'] (<https://www.flipkart.com>)
- 02 2. //span[text() = 'Filters'] (<https://www.flipkart.com/inverters/pr?sid=jqe,abm,7no&otracker=undefined-footer-footer>)
- 03 3. //h1[text() = 'Self Drive Car Rental in'] (<https://www.zoomcar.com/bangalore/>)
- 04 4. //p[text() = 'Earn 7 points for every booking and redeem for additional discount'] (<https://www.zoomcar.com/bangalore/>)
- 05 5. //div[text() = 'Email or phone'] (Gmail Signup page)

15

February  
Friday

Wk-07 • 046-319

2019

FEBRUARY '19	
3	4
10	11
17	18
24	25
S	M

6. //button[@text='FACEBOOK'] (<https://www.zoomcar.com>)  
 7. //button[@text()='Login'] (<https://www.facebook.com>)  
 8. //span[@text()='create your account']  
     (<https://twitter.com/i/flow/signup>)  
 9. //span[@text()='Date of birth']  
     (<https://twitter.com/i/flow/signup>)  
 10. //label[@text()='Email or phone']  
     (<https://www.linkedin.com/login>)

## ASSIGNMENT 2 : On Case 3 : xpath by multiple attributes.

### a) Unique Element (and):

Ques: tagName[@attr1='attrval' and @attr2='attrval']

1. //input[@id='email' and @name='email']  
     (<https://www.facebook.com>)
2. //input[@id='pass' and @placeholder='password']  
     (<https://www.facebook.com/password>)
3. //button[@name='login' and @data-testid='royal\_login\_button']  
     (<https://www.facebook.com/login>)
4. //input[@title='Search for products, brands and more' or @placeholder='Search for products, brands and more']  
     (<https://www.flipkart.com/searchbar>)
5. //a[@class='\_2rpwqI' or @target='\_blank']  
     (count:127)  
     (<https://www.flipkart.com/search?q=hair+oil&otracker=Search%26otrackerf=Search&marketplace=Flipkart&as-show=on&as-off>)

31		1	2
3	4	5	6
10	11	12	13
17	18	19	20
24	25	26	27
S	M	T	W

February

Saturday

2019

16

Wk-07 • 047-318

6. //img[@class='figure-href' or @height='100']  
 (Count: 18)  
 (https://www.patanjaliayurved.net/products/all)
7. //img[@class='img-responsive' or @title='Mine Diamond Ring KDRHCR111007']  
 (https://www.malabargoldanddiamonds.com/diamond-jewellery.html)
8. Combination of a text and an attribute:  
 8. //h3[@class='formStandard\_subheading' and  
 text()='Additional information related to  
 your business needs']  
 (https://explore.zoom.us/en/contactSales)
9. //h2[@class='section-title' and text()='Our Collection']  
 (https://www.malabargoldanddiamonds.com)
10. //p[@class='page-help-link\_text' and text()='Looking to create a page for a business?']  
 (https://www.linkedin.com/signup)
- 05 **ASSIGNMENT 2: On Case 4: xpath by Contains() function**
- 06 1. //h4[contains(text(), 'Designed')]  
 (https://byjus.com/byjus-classes/)
- 07 2. //h1[contains(text(), 'Multi')]  
 (https://www.upgrad.com/blog/introduction-to-multivariate-regression-in-machine-learning)
3. //p[contains(text(), 'transcend')]  
 (https://www.gitam.edu)
4. //p[contains(text(), 'workplaces')]  
 (https://www.tcs.com/test-chroma)
5. //p[contains(text(), 'Capgemini operates in nearly')]  
 (https://www.Capgemini.com/in-en/contact-us/)

# 18

February

Monday

Wk-08 • 049-316

2019

FEBRUARY '19	
3	4
10	11
17	18
24	25
S	M
T	W
F	S

6. //span[contains(text(), 'Basic')]

(<https://www.amazon.com>)

7. //a[contains(@title, 'stupid')]

(<https://www.reddit.com>)

8. //a[contains(@href, '/kukatpally')]

(<https://www.Omnihospital.com>)

9. //span[contains(text(), '2 & 3 BHK')]

(<https://www.myhomes.com>)

10. //img[contains(@src, '/krjs10.1970x.jpg')]

(<https://www.malabar.com>)

16.09.21

D11

02 Case 6 : xpath by dependent and independent element. (OR) Reference of surrounding unique element

03 → First identify unique element which is nearby non-unique element.

04 → From that unique element travel to non-unique element.

Ex: //p[text() = 'Ruby']/following-sibling::p[4]/descendant::a

//li[@class = 'nav-item']/descendant::a[text() = 'Home']

(selenium.dev)

Ruby Download

Java Download

Python Download

C++ Download

JS Download

MARCH '19	31	1	2
	3	4	5
	10	11	12
	17	18	19
	24	25	26
S	M	T	W
		T	T
		F	S

February

Tuesday

2019

19

Wk-08 • 050-315

In our example,

1. Inspect independent element

//td[.= 'Java']

2. Travel back until dependent element within the boundary

//td[.= 'Java']/--

3. Inspect dependent and write continuously

//td[.= 'Java']/..//a[.= 'Download']

(or)

Alternate way to reach from Java --> Download.

//td[.= 'Java']/following-sibling::td[@data-label='Links']/descendant::a[.= 'Download']

//td[.= 'Java']/following-sibling::td[@data-label='Links']/a[.= 'Download']

//td[text()='Java']/following-sibling::td[@data-label='Links']/descendant::a[text()='Download']

//td[text()='Java']/ancestor::tr/descendant::a[text()='Download']

//td[text()='Java']/..//a[text()='Download']

(flipkart.com)

//div[text()='motorola']/preceding-sibling::div[@class='l1p7h2j']

20

February  
Wednesday

Wk-08 • 051-314

2019

February '19	3	4	5	6	7	8	9	10
S	M	T	W	T	F	S	S	

Case 7: Xpath by group index.

Here we group all similar elements and apply index on that group.

Q: What is the difference between  $\text{//input}$ ,  $\text{input[]}$ ,  $(\text{//input})[i]$ ? $\text{//input}$  - Matches with all the inputs on the webpage. $\text{//input[i]}$  - matches with all the first inputs of their parent element (tag) $(\text{//input})[57]$  - matches 57th input. $(\text{//input})[\text{last}()]$  - matches last input $(\text{//input})[\text{last}()-i]$  - matches second last input (input before the last one) $(\text{//input})[\text{position}() > (\text{last}() - 5)]$  - matches last 5 inputs. $(\text{//input})[\text{position}() \bmod 2 = 0]$  - matches all even inputs. $(\text{//input})[\text{position}() \bmod 2 = 1]$  - matches all the odd inputs.

Explanation: Case 7:

html 1

body 1

div 1

 $\text{//a}$  - matches all thea 1  $\rightarrow 6$ 

a 2

a 3

 $\text{//a}[i]$  - matches all thefirst 'a'  $\rightarrow 2$ 

div 2

a 1

a 2

a 3

 $(\text{//a})[i]$  - matches very first'a'  $\rightarrow 1$ 

a 1
a 2
a 3
a 4
a 5
a 6

MARCH '19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

February

Thursday

2019

21

Wk-08 • 052-313

## ASSIGNMENT 3 : On Case 6 : Xpath by dependant and independant element.

09. `//div[text()='INDUSTRIES']/ancestor::li[@id='om-leaf-om-u1-676429867-1']/following-sibling::li[@id='om-leaf-om-u1-676429867-2']/descendant::div[text()='SERVICES & PRODUCTS']` (<https://www.hcltech.com>)
10. `//a[text()='Industries']/ancestor::li[@class='menu-itemmain-menu-items']/following-sibling::li[@class='menu-itemmain-menu-items']/descendant::a[text()='Services']` (<https://www.tcs.com>)
11. `//span[text()='Helito']/ancestor::div[@class='a-Section a-spacing-none a-spacing-top-small']/following-sibling::div[@class='a-Section a-spacing-none a-spacing-top-small']/descendant::span[text()='399']` (<https://www.amazon.in>)
12. `//div[text()='SAMSUNG GALAXY F22']/ancestor::div[@class='_3PLY_Crow']/descendant::div[text()='14999']` (<https://www.flipkart.com>)
13. `//span[text()='Redme Note 10 (Aqua Green, 4GB RAM, 64 GB storage) - Amoled Dot Display 48MP Sony Sensor IMX582 | snapdragon 678 processor']/ancestor::div[@id='title_features_div']/following-sibling::div[8]/descendant::span[text()='13,999.00']` (<https://www.amazon.com>)

22

February

Friday

Wk-08 • 053-312

2019

FEBRUARY '19						
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
S	M	T	W	T	F	S

6. `//span[text() = 'TRADE ZIE (pack of 4) msWG-4 class set']/ancestor::div[@class = 'aMaAEs']//descendant::div[text() = '₹499']  
( https://www.amazon.com/)`
7. `//span[text() = 'VEEMIZO Waterproof Shoe bags for Travelling Bag']/ancestor::div[@id = 'title_feature_div']//following-sibling::div[8]//descendant::span[text() = '₹280.00']  
( https://www.flipkart.com/)`
8. `//div[text() = 'APPLE iPhone 12 (Black (28 GB))']//ancestor::div[@class = 'col col-7-12']//following-sibling::div[1]//descendant::div[text() = '₹68,999']  
( https://www.flipkart.com/)`
9. `//span[text() = 'Redmi Earbuds -']//ancestor::div[@id = 'title-feature-div']//following-sibling::div[8]//descendant::span[text() = '₹998.00']  
( https://www.amazon.com/)`
10. `//span[text() = 'realme Buds 2 Wired Headset']//ancestor::div[@class = 'aMaAEs']//descendant::div[text() = '₹599.00']  
( https://www.amazon.com/)`

31	1	2
3	4	5
10	11	12
17	18	19
24	25	26
S	M	T

17.09.21

D12

February

Saturday

2019

Wk-08 • 054-311

23

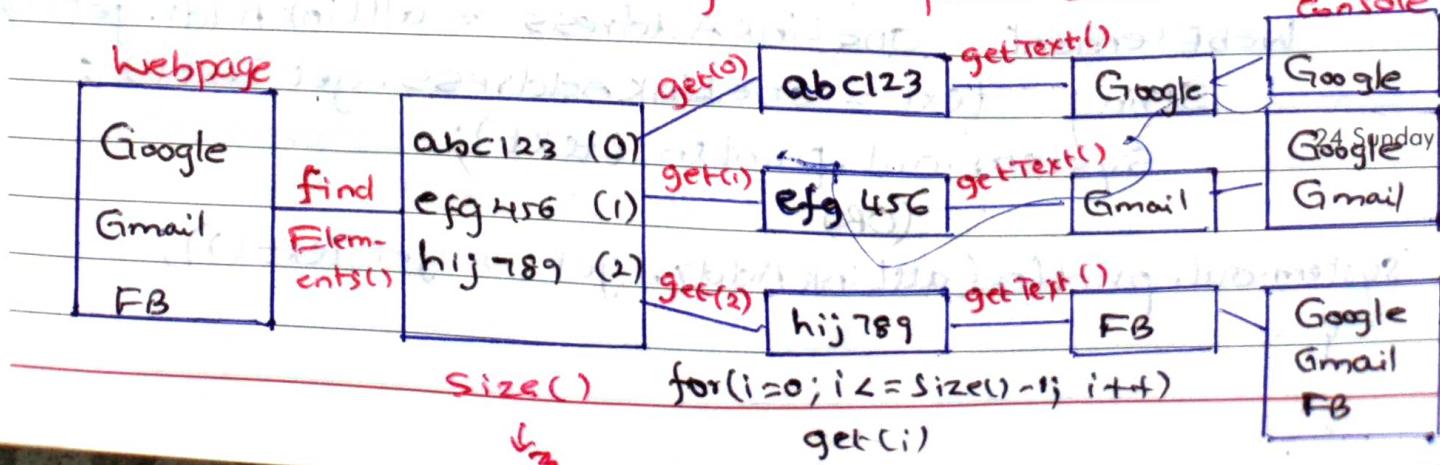
## Handling Multiple Elements :

- 08 1. We handle multiple elements by using findElements() method of webdriver Interface
- 09 2. Return type of findElements() is List<webE-  
lements>
- 10 3. If findElements() method is not able to find  
the elements on the webpage, then we get  
Empty list.

## 12 Difference between findElement() & findElements()

findElement()	findElements()
1. <u>findElement()</u> is used to get the address of first matching element on the webpage.	1. <u>findElements()</u> is used to get the address of all the matching elements on the webpage.
2. Return is <u>WebElement Interface</u> .	2. Return type is <u>List&lt; WebElement &gt;</u>
3. If <u>findElement()</u> method is not able to find the element on the webpage, we get, <u>NoSuchElementException</u> .	3. If <u>findElements()</u> method is not able to find the elements on the webpage then we get <u>empty List</u> .

## Explanation on Handling Multiple Elements :



25

February  
Monday  
2019

Wk-09 • 056-309

FEBRUARY 19	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28		
S	M	T	W	T	F	S	S

Program :

```

08 package arsp;
09 import java.util.List;
10 import org.openqa.selenium.By;
11 import org.openqa.selenium.WebDriver;
12 import org.openqa.selenium.chrome.ChromeDriver;
13
14 public class HandlingMultipleElements
15 {
16     public static void main(String[] args)
17     {
18         System.setProperty("webdriver.chrome.driver",
19             "./drivers/chromedriver.exe");
20         WebDriver driver = new ChromeDriver();
21         driver.get("file:///C:/KCSMG/HTML/MultipleElements.html");
22         List<WebElement> allLinkAddr =
23             driver.findElements(By.tagName("a"));
24
25         // 1. Count the number of links
26         int noofLinks = allLinkAddr.size();
27         System.out.println("No. of Links: " + noofLinks);
28
29         // 2. print the text of the all links
30         for (int i=0; i< allLinkAddr.size()-1; i++)
31         {
32             WebElement oneLinkAddress = allLinkAddr.get(i);
33             String text = oneLinkAddress.getText();
34             System.out.println(text);
35             // (OR)
36             System.out.println(allLinkAddr.get(i).getText());
37         }
38     }
39 }

```

MARCH 19	31	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	S	M	T	W	T	F	S

February

Tuesday

2019

26

Wk-09 • 057-308

11 3. Click on last link

08 allLinks Addr. get (allLinkAddr. size() - 1). click();

{

09 }

## \* TEST CASE to login to an Application

Step No.	Action Description	Input	Expected Result	Actual Result
1.	Open the browser	https://demo.actitime.com/	Login page should be displayed	
2.	Enter the test URL	login.do	(actiTime - Login)	
01	Enter valid User name, Password, & click on Login button	UN - admin Pwd - manager	Home Page (Enter Time Track page should be displayed)	
02				
03				

## \* TEST SCRIPT for the above test case

```

05 package arsp;
06 import org.openqa.selenium.By;
07 import org.openqa.selenium.WebDriver;
08 import org.openqa.selenium.chrome.ChromeDriver;
09
10 public class ActiTIMEloginTest {
11
12     public static void main(String[] args) throws
13         InterruptedException {
14
15         // Set the system property
16         System.setProperty("webdriver.chrome.driver",
17             ".\drivers\chromedriver.exe");
18
19     }
20
21 }

```

27

February  
Wednesday

Wk-09 • 058-307

2019

FEbruary '19	3	4	5	6	7	8	9
S	10	11	12	13	14	15	16
M	17	18	19	20	21	22	23
T	24	25	26	27	28	29	30

```

// Open the chrome browser
08 WebDriver driver = new ChromeDriver();
// maximize the browser window
09 driver.manage().window().maximize();
// Enter the test URL
10 driver.get("https://demo.actitime.com/login.do");
// verify login page is displayed.
11 String expectedTitle = "actiTIME - Login";
String actualTitle = driver.getTitle();
12 if (actualTitle.equals(expectedTitle))
{
    System.out.println("Login page is displayed, PASS");
}
else
{
    System.out.println("Login page is not displayed, FAIL");
}
Thread.sleep(2000);
// Enter Valid Username into Username Textbox
05 driver.findElement(By.id("username")).sendKeys("admin");
Thread.sleep(2000);
// Enter valid password into password Textbox
07 driver.findElement(By.name("pwd")).sendKeys("manager");
Thread.sleep(2000);
// click on login Button
driver.findElement(By.xpath("//div[@text()='Login']")).click();
Thread.sleep(2000);

```

MARCH '19	31					1	2
	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	S	M	T	W	T	F	S

February

Thursday

28

2019

Wk-09 • 059-306

```
1 // verify login page is displayed
2 String expectedHomeTitle = "actiTime-Enter Time-Track";
3 String actualHomeTitle = driver.getTitle();
4 if (actualHomeTitle.equals(expectedHomeTitle))
5 {
6     System.out.println("Home page is displayed, PASS");
7 }
8 else
9 {
10    System.out.println("Home page is not displayed, FAIL");
11 }
12 }
```

18.09.21

D13

**Synchronisation:** Matching the speed of Selenium according to the speed of application, to avoid Synchronisation issue is called Synchronisation.

Q. What is Synchronisation issue?

06 Selenium is too fast but application might take time to load, hence this time mismatch  
07 is called Synchronisation issue.

There are two types of waits to match Selenium speed.

1. Static wait: waiting time is fixed.
  2. Dynamic wait: waiting time is variable.

01

March  
Friday

Wk-09 • 060-305

2019

MARCH 19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

means, Compulsorily we have to wait for 15 seconds.

→ Dynamic wait means waiting period is not fixed, here it changes according to the application speed.

Two types of Dynamic wait

Implicitwait : Here, no need to give any condition, It will wait for findElement() method and findElements() method.

Syntax: driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

20 - indicates long arg

Here 20 means maximum time until control should wait before giving NoSuchElementException

Note: Implicitwait statement can be declared once at the top of the code which can work for all the findElement() method / findElements() method

### IMPLICITLY WAIT FLOW DIAGRAM

Explanation: When the control comes to findElement() / findElements() it will check whether the element is present or not. If the element is present, then findElement() will return the address.

APRIL 19	1	2	3	4	5	6	
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30				
S	M	T	W	T	F	S	

March  
Saturday

2019

02

Wk-09 • 061-304

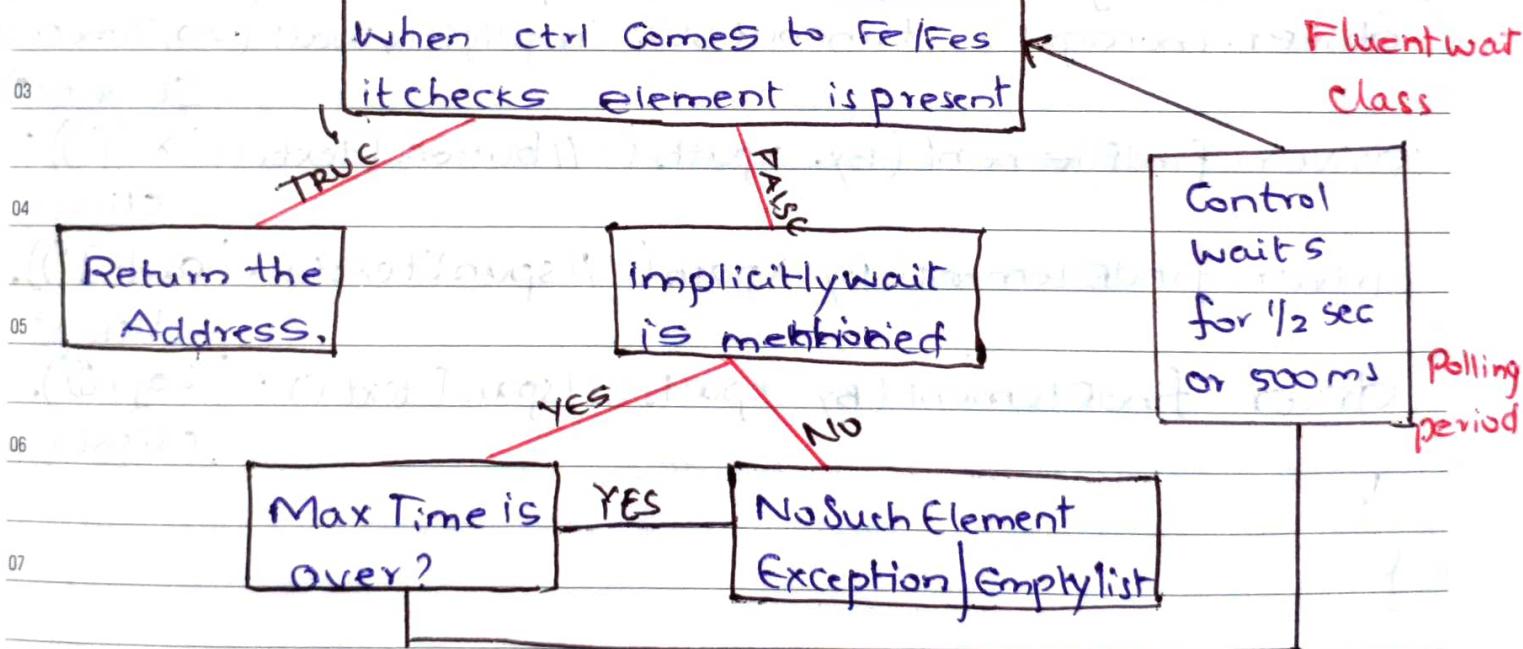
But, if element is not present then it will check whether implicitlyWait statement is mentioned or not.. If not then we will get NoSuchElementException / empty list. If mentioned, then it will check whether time is over or not?

10

If time is over but element not found, NoSuchElementException .. But if time is not over, then Control will wait for 1/2 second or 500ms and again check whether element is present or not.

This 500 ms duration is called polling period, and is mentioned in FluentWait class.

02



Program:

```

package org;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
  
```

03 Sunday

04

March  
Monday

Wk-10 • 063-302

2019

MARCH '19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

```

import org.openqa.selenium.support.ui.Expected
08 Conditions;
import org.openqa.selenium.support.ui.WebDriverWait;
09
10 public class FlipkartTest {
11
12     public static void main(String[] args) throws
13 InterruptedException {
14
15     System.setProperty("webdriver.chrome.driver", "./drivers/
16         chromedriver.exe");
17
18     WebDriver driver = new ChromeDriver();
19     driver.manage().window().maximize();
20
21     driver.get("https://www.flipkart.com/");
22
23     driver.manage().timeouts().implicitlyWait(20, TimeUnit.
24         SECONDS);
25
26     driver.findElement(By.xpath("//button[text()='X']"))
27         .click();
28
29     driver.findElement(By.xpath("//span[text()='Cart']"))
30         .click();
31
32     driver.findElement(By.xpath("//span[text()='Login']"))
33         .click();
34
35 }
36
37 }
```

ExplicitlyWait : Here we need to give Condition, It will work for any method.

Create an Object of 'WebDriverWait' class.  
 WebDriverWait wait = new WebDriverWait(reference  
 Variable of WebDriver, timeoutInSeconds);  
 ExplicitlyWait will work for any method including  
 findElement() and findElements(). Here we need

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

APRIL '19

March  
Tuesday

2019

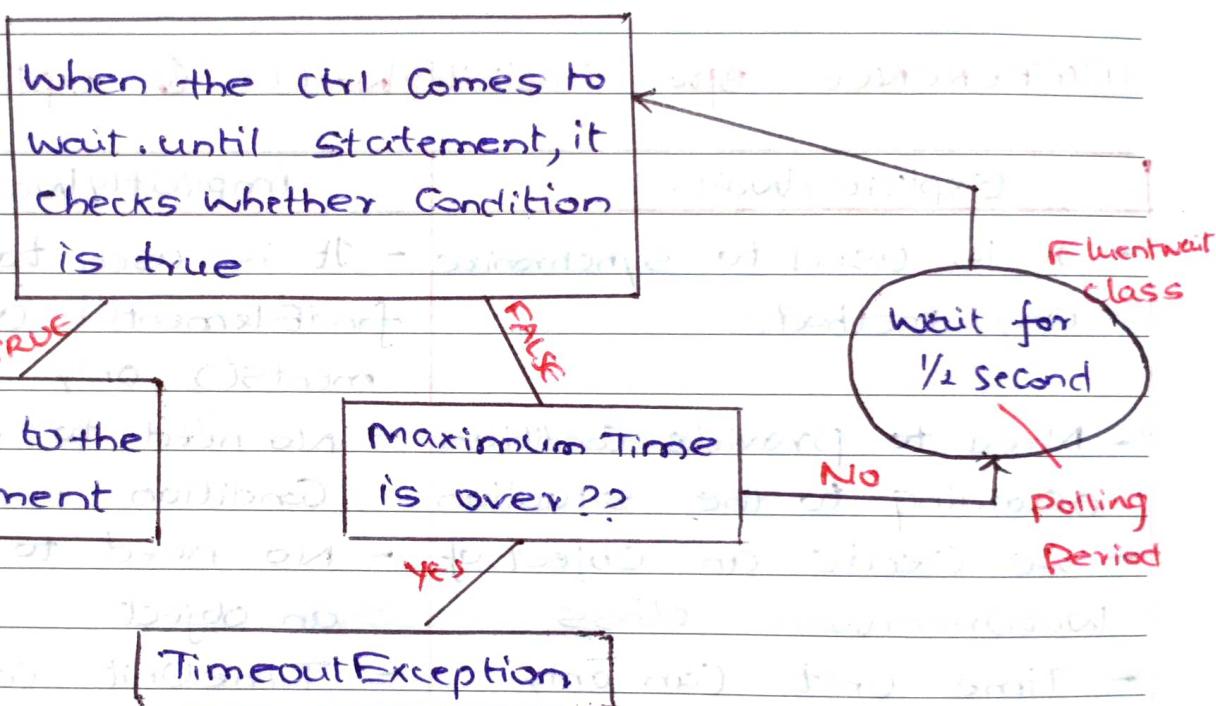
05

Wk-10 • 064-301

to create an object of WebDriverWait class and we use the static methods of ExpectedConditions class to give Condition.

WebDriverWait class and ExpectedConditions class is imported from 'org.openqa.selenium.support.ui' package.

## Explicitlywait Flow Diagram



When the Control Comes to wait.until() Statement, it will Check whether the Condition is true or false. If Condition is true, then Control goes to the next Statements and start executing them.. If Condition is false, then it will check maximum time is over or not, if yes, that means Condition never became true within the maximum time, it gives TimeOutException. If no, it waits for 1/2 sec or 500 ms and then checks again Condition became true or not,

MARCH '19	31	3	4	5	6	7	8	9	10
	10	11	12	13	14	15	16	17	18
	17	18	19	20	21	22	23	24	25
	24	25	26	27	28	29	30	31	
	S	M	T	W	T	F	S	S	

For 04 Mar program, we can also use ExplicitlyWait()

Last statement but one,

```
WebDriverWait wait = new WebDriverWait(driver, 20);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("By.xpath("//span[text()='']"));
```

## DIFFERENCE B/W implicitlyWait() & explicitlyWait()

explicitlyWait()	implicitlyWait()
- It is used to synchronize any method.	- It is used to synchronise findElement() and findElements() only.
- Need to provide condition according to the situation	- No need to mention any condition.
- We create an object of WebDriverWait class	- No need to create an object
- Time unit can only be SECONDS	- TimeUnit can be days, hours, minutes, seconds, milliseconds, microseconds, nano seconds.
- Here, if condition doesn't become true within maximum time, then we get TimeoutException.	- Here, if element is not found within maximum time, we get NoSuchElementException / Empty list
- It is a kind of "localwait" which has to be mentioned before every synchronization issue.	- It is a kind of "global wait" because we can mention implicitlyWait once at the top of the code which works for every findElement()

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Thursday

2019

07

Wk-10 • 066-299

### explicitlyWait()

08 - Syn: WebDriverWait = new WebDriverWait(WebDriver reference, TimeoutInSeconds) wait.until(ExpectedCondition Required Condition)

### implicitlyWait()

or findElements() through the code:  
- Syn: driver.manage().timeouts().implicitlyWait(long arg, Timeout). Eg: driver.manage().timeout().implicitlyWait(20, TimeUnit.SECONDS)

19.09.21

D14

## 01 Handling Dropdown (List Box / Web List / Custom Dropdown)

02 Drop-downs are used to save the space on the webpage.

03 Two types: 1. Static Dropdown  
2. Dynamic Dropdown

04 Static Dropdown: Options of the dropdown are fixed.

05 Two types of Static Dropdown.

06 1. Single-select Dropdown

2. Multi-select Dropdown

07 Created by Using <select>, <options> tags.

Q: How to handle the static dropdown?

Since it is created by using <select>, we

handle it by using Select class.

Select class is imported from org.openqa.

Selenium.Support.UI package.

Select class constructor is a parameterized constructor which takes WebElement arg, where we need to pass address of the dropdown.

# 08

March  
Friday

Wk-10 • 067-298

2019

MARCH '19	31	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	S	M	T	W	T	F	S
-----------	----	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---

## Select Class Methods:

### Selection methods

1. SelectByVisibleText (String arg) : void
2. SelectByValue (String arg) : void
3. SelectByIndex (int index) : void

### Deselection Methods

1. deselectByVisibleText (String arg) : void
2. deselectByValue (String arg) : void
3. deselectByIndex (int index) : void
4. deselectAll () : void

### Operational Methods

1. isMultiple () : boolean
2. getOptions () : List<WebElement>
3. getAllSelectedOptions () : List<WebElement>
4. getFirstSelectedOption () : WebElement
5. getWrappedElement () : WebElement

### Note :

1. SelectByIndex() can handle duplicate options
2. When we try to use deselection methods on a single-select dropdown, then we get UnsupportedOperationException. (Java) (RTE)

**isMultiple()**: Used to verify whether it is a single select dropdown (or) multi-select dropdown. Return type is boolean. Returns true if it is multi-select dropdown.

**getOptions()**: Used to get the address of all the options present in the dropdown. Return type is List<WebElement>.

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Saturday

2019

09

Wk-10 • 068-297

If no option is present in dropdown, then we get empty list.

**getAllSelectedOptions()**: Used to get the address of the selected options. Return type is List<WebElement>. If no option is selected in dropdown then we get Empty List.

**getFirstSelectedOptions()**: Used to get the address of first selected option in the dropdown. Return type is WebElement Interface. If no option is selected, then it returns NoSuchElementException.

**getWrappedElement()**: Used to get the address of all the options wrapped into a single address. Return type is WebElement Interface. If no option is present in dropdown, then we get NoSuchElementException.

**Note:** To create a multi-select dropdown, add the attribute `multiple` in select tag.

04

### Program:

```
05 package org.openqa.selenium;
import java.util.List;
06 import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
07 import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;
```

08 public class HandlingDropdown

10 Sunday

```
09 public static void main(String[] args) throws
InterruptedException.
```

11

March  
Monday

Wk-11 • 070-295

2019

MARCH '19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

System.setProperty("webdriver.chrome.driver",  
 08                    "./drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("file:///D:/KCSM6/HTML/dropdown.html");  
 09                    driver.get("file:///D:/KCSM6/HTML/multiselectdropdown.html");  
 10                    Thread.sleep(2000);

11        WebElement ddAddr = driver.findElement(By.id("SGH"));  
 12        Select sel = new Select(ddAddr);

13        // Sel. selectByVisibleText("CHAPATHI");

14        // Sel. selectByValue("idli");

15        // Sel. selectByIndex(8);

16        ddAddr.click();

17        if (sel.isMultiple())

18                    System.out.println("It's a Multi-select Dropdown");

19 } else

20        { system.out.println("It's a single-select dropdown");

21 }

22        "Sel. selectByVisibleText("Dosa");

23        List<WebElement> allOptions = sel.getOptions();

24        for (int i=0; i<=allOptions.size()-1; i++)

25        { System.out.println(allOptions.get(i).getText());

26        }

27        for (int i=0; i<=allOptions.size()-1; i++)

28        { sel.selectByIndex(i);

29        Thread.sleep(1000);

30        }

						6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Tuesday  
2019

12

Wk-11 \* 071-294

```

11 Thread.sleep(3000);
11 sel.deselectByVisibleText("DOSA");
11 sel.deselectByValue("dosa");
09/11 Sel.deselectByIndex(1);
11 Sel.deselectAll();
10/11 for(int i=1; i<=4; i++)
11 {
11   Sel.SelectByIndex(i);
11 }
11 System.out.println(sel.getFirstSelectedOption().getText());
11 System.out.println(sel.getAllSelectedOptions().size());
01 System.out.println(sel.getWrappedElement().getText());
}

```

### HTML Code for the above program.

```

03 <html>
04   <body>
05     <select id='novotel'> (OK) <select id='SGH' multiple>
06       <option value='idli'> IDLI </option>
05       <option value='dosa'> DOSA </option>
06       <option value='vada'> VADA </option>
06       <option value='poori'> POORI </option>
06       <option value='chapathi'> CHAPATHI </option>
07       <option value='gamosa'> SAMOSA </option>
06       <option value='panipoori'> PANI POORI </option>
06       <option value='thatti idli'> THATTI IDLI </option>
06       <option value='idli'> IDLI </option>
06       <option value='double ka meeta'> DOUBLE KA MEETA
06           </option>
06       <option value='irani chai'> IRANI CHAI </option>
06   </select>
06 </body>
06 </html>

```

13

March  
Wednesday

Wk-11 • 072-293

2019

20.09.21

D15

31	3	4	5	6	7	1	2
10	11	12	13	14	15	16	17
17	18	19	20	21	22	23	24
24	25	26	27	28	29	30	S
M	T	W	T	F	S	S	S

## Handling Duplicates

Inputs :

Idli
Dosa
Vada
poori
Idli
chai

List (I)

Idli
Dosa
Vada
poori
Idli
chai

1. Duplicates are allowed
2. Insertion order is maintained
3. Prints as per the given order  
(Expln. to 2 point)

## SET (I)

HashSet (c)

Dosa
poori
Idli
chai
Vada

LinkedHashSet(c)

Idli
Dosa
Vada
poori
Chai

TreeSet(c)

Chai
Dosa
Idli
poori
Vada

1. Remove Duplicate
2. Insertion order is not maintained

1. Remove Duplicate
2. Order is maintained

1. Remove Duplicate
2. Sorted order

Idli
Dosa
Vada
poori
Idli
chai
Vada

Idli
Dosa
Vada
poori
Chai

Chai
Dosa
Idli
poori
Vada

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Thursday

14

2019

Wk-11 • 073-292

## Program :

```

08 package arsp1;
09 import java.util.HashSet;
10 import java.util.LinkedHashSet;
11 import java.util.List;
12 import java.util.TreeSet;
13 import org.openqa.selenium.By;
14 import org.openqa.selenium.WebDriver;
15 import org.openqa.selenium.WebElement;
16 import org.openqa.selenium.chrome.ChromeDriver;
17 import org.openqa.selenium.support.ui.Select;

01
02 public class HandlingDuplicates
03 {
04     public static void main(String[] args)
05     {
06         System.setProperty("webdriver.chrome.driver",
07             "./drivers/chromedriver.exe");
08         WebDriver driver = new ChromeDriver();
09         driver.get("file:///D:/KCSMS/HTML/MultiselectDropdown.html");
10         WebElement dd1Addr = driver.findElement(By.id("SGH"));
11         Select sel = new Select(dd1Addr);
12         List<WebElement> allOptions = sel.getOptions();
13         for (int i=0; i<=allOptions.size()-1; i++)
14         {
15             String text = allOptions.get(i).getText();
16             System.out.println(text);
17         }
18         List<WebElement> allOptions = sel.getOptions();
19         HashSet<String> allText = new HashSet<String>();
20         for (int i=0; i<=allOptions.size()-1; i++)
21         {
22             String text = allOptions.get(i).getText();
23             if (!allText.contains(text))
24             {
25                 allText.add(text);
26             }
27         }
28     }
29 }
```

15

March  
Friday

Wk-11 • 074-291

2019

MARCH '19	31	3	4	5	6	7	1	2
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

```

11 String text = allOptions.get(i).getText();
08 allText.add(text);
11 }
09 11 List<WebElement> allOptions = sel.getOptions();
11 HashSet<String> allText = new HashSet<String>();
10 11 for(int i=0; i<=allOptions.size()-1; i++)
11 {
11   String text = allOptions.get(i).getText();
11   allText.add(text);
12 11 System.out.println(allText);
01 11 {
11   System.out.println(oneText);
01 }
11 List<WebElement> allOptions = sel.getOptions();
03 LinkedHashSet<String> allText = new LinkedHashSet<String>();
for (int i=0; i<=allOptions.size()-1; i++)
{
  String text = allOptions.get(i).getText();
  allText.add(text);
}
for (String oneText: allText)
{
  System.out.println(oneText);
}
System.out.println(allText);
}
List<WebElement> allOptions = sel.getOptions();
TreeSet<String> allText = new TreeSet<String>();
for (int i=0; i<=allOptions.size()-1; i++)
{
  String text = allOptions.get(i).getText();
  allText.add(text);
}

```

7	1	2	3	4	5	6
8	9	10	11	12	13	
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Saturday

16

2019

Wk-11 • 075-290

for (String oneText : allText)

System.out.println(oneText);

21.09.21

[D16]

## Dynamic Dropdown / Custom Dropdown

In this kind of dropdown, the options are changing..

To create it we use tags like input, div, ul, li, a etc.

### Q: How to handle dynamic dropdown?

Since the options are inspectable, we use findElement(), findElements(), sendKeys(), click().

Since it is not created by Select tag, hence we can't use Select class and its methods.

If we use, we get UnexpectedTagNameException.

TS:

```
package com.avspit;
```

```
import java.util.List;
```

```
{ public class HandlingDynamicDropdown
```

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

```
        InterruptedException
```

17 Sunday

```
        System.setProperty("webdriver.chrome.driver",
```

```
            "C:\Drivers\chromedriver.exe");
```

18

March  
Monday

Wk-12 • 077-288

2019

MARCH '19	31	3	4	5	6	7	1	2
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

```

08 WebDriver driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.flipkart.com/");
09 driver.findElement(By.xpath("//button[text()='X']"))
    .click();
10 WebElement ddAddr = driver.findElement(By.name("q"));
ddAddr.sendKeys("mobiles");
11 ddAddr.sendKeys(Keys.DOWN);
ddAddr.sendKeys(Keys.ENTER);
12 driver.get("https://www.google.com/");
driver.findElement(By.name("q")).sendKeys("selenium");
01 Thread.sleep(3000);
List allOptions = driver.findElements(
02 By.xpath("//div[@class='wM6Wrd']"));
Thread.sleep(3000);
03 System.out.println(allOptions.size());
04 }

```

22.09.2021

D17

## Handling Keyboard and Mouse Actions

Action class : Used to handle keyboard and mouse actions imported from org.openqa.selenium.interactions package.

- Actions class constructor takes WebDriver reference or Browser class reference as its argument.
- We handle mouse and Keyboard Actions by using non-static methods of Action class.
- Also note for every method of Action class, we need to compulsorily use •perform() at

APRIL '19
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
S M T W T F S

March  
Tuesday

19

2019

Wk-12 • 078-287

the end.

- 08 → Robot class is used to handle keyboard and mouse Actions. It is imported from java.awt package
- 09 → Robot class constructor throws a Comptime exception called AWTException
- 10 → Robot class uses KeyPress in which KeyEvent class is present.
- 11 → In KeyEvent class, all the Keyboard Keys are stored as Virtual keys and since they are static final variables we can access directly through classname.VariableName (KeyEvent.VK-T)
- 12 → Also remember, to release a key after it is pressed, so that it will be released virtually.

### 1. Handling mouse hover action

moveToElement(addr).perform()

### 2a. Perform Right Click

ContextClick(addr).perform()

### 2b. Handling Right Clicked Options

Robot class -> keyPress / keyRelease

### 3 Perform Double click

doubleClick(addr).perform()

### 4. Perform Drag & Drop action

dragAndDrop(fromAddr, toAddr).perform()

20

March  
Wednesday

Wk-12 • 079-286

2019

MARCH '19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	17
	17	18	19	20	21	22	23	24
	24	25	26	27	28	29	30	31
	S	M	T	W	T	F	S	S

S.No	Actions	methods	Mandatory method	Class
08	1. Mouse Hover Action	moveToElement (address of an element)	performs	Actions class
09	2. Right Click Action	ContextClick (address of an element)	performs	Actions class
10	3. Double click Action	doubleClick (address of an element)	performs	Actions class
11	4. Drag and Drop Action	dragAndDrop (from address, to address)	performs	Actions class
12	5. Handle Right clicked option	keyPress (KeyEvent.VK) keyRelease (KeyEvent.VK)		Robot class

## Example :

```

package . asp1 ;
import java.awt. AWTException;
import java.awt. Robot;
import java.awt. event. KeyEvent;
import org.openqa. selenium. By;
import org.openqa. Selenium. WebDriver;
import org.openqa. Selenium. WebElement;
import org.openqa. Selenium. Chrome. ChromeDriver;
import org.openqa. Selenium. interactions. Actions;
public class HandlingMouseAndKeyboardActions
{
    public static void main( String[ ] args ) throws
        InterruptedException, AWTException
    {
        // performing Mouse Hover Action
        System. setProperty( "webdriver. chrome. driver",
            "./drivers/ Chromedriver. exe" );
    }
}

```

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Thursday

21

2019

Wk-12 • 080-285

```
08 WebDriver driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.flipkart.com");
driver.findElements(By.xpath("//button[text()='X']")).click();
09
10
```

```
Actions ac = new Actions(driver);
```

```
11 WebElement fashion = driver.findElement(By.xpath("//div[text()='Fashion']"));
```

```
12 ac.moveToElement(fashion).perform();
```

```
Thread.sleep(3000);
```

```
01 // performing Double click Action
```

```
driver.get("https://demo.actitime.com/login.do");
```

```
02 WebElement untb = driver.findElement(By.id("username"));
untb.sendKeys("admin");
```

```
03 ac.doubleClick(untb).perform();
```

```
Thread.sleep(3000);
```

```
04 // Right Click And Handle Right Clicked Options.
```

```
driver.get("file:///D:/kcmg/HTML/Link.html");
```

```
05 Thread.sleep(3000);
```

```
06 WebElement linkAddress = driver.findElement(By.id("i"));
```

```
ac.contextClick(linkAddress).perform();
```

```
07 Thread.sleep(3000);
```

```
Robot r = new Robot();
```

```
• Key Press(KeyEvent.VK_T);
```

```
• keyRelease(KeyEvent.VK_T);
```

```
Thread.sleep(3000);
```

```
11 Perform Drag and Drop Actions
```

```
driver.get("https://www.dhtmlgoodies.com/scripts/
drag-drop-custom/demo-drag-drop-3.html");
```

22

March  
Friday

Wk-12 • 081-284

2019

MARCH '19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

```

08 WebElement src = driver.findElement(By.id("box1"));
09 WebElement dest = driver.findElement(By.id("box2"));
10 ac.dragAndDrop(src, dest).perform();
11 }
12
13 } catch (Exception e) {
14     e.printStackTrace();
15 }
16
17 }
18
19 }
```

23.09.2021

D18

One more way to do the Assignment 6:

```

01 System.setProperty("webdriver.chrome.driver", "C:/Users/91990/Downloads/chromedriver.exe");
02 WebDriver driver = new ChromeDriver();
03 driver.manage().window().maximize();
04 driver.get("https://demo.actitime.com/login.htm");
```

```

05 WebElement untb = driver.findElement(By.id("username"));
06 untb.sendKeys("admin");
07 untb.sendKeys(Keys.CONTROL, "a");
08 untb.sendKeys(Keys.CONTROL, "c");
09 (or)
10
11 Thread.sleep(3000);
12 untb.sendKeys(Keys.CONTROL, "v");
13 Thread.sleep(3000);
```

```

14 driver.findElement(By.name("pwd")).sendKeys(Keys.CONTROL, "v");
```

Keyboard combination: ~~ctrl + v~~ ~~ctrl + c~~ ~~ctrl + a~~

APRIL '19	1	2	3	4	5	6
	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30			
	S	M	T	W	F	S

March  
Saturday

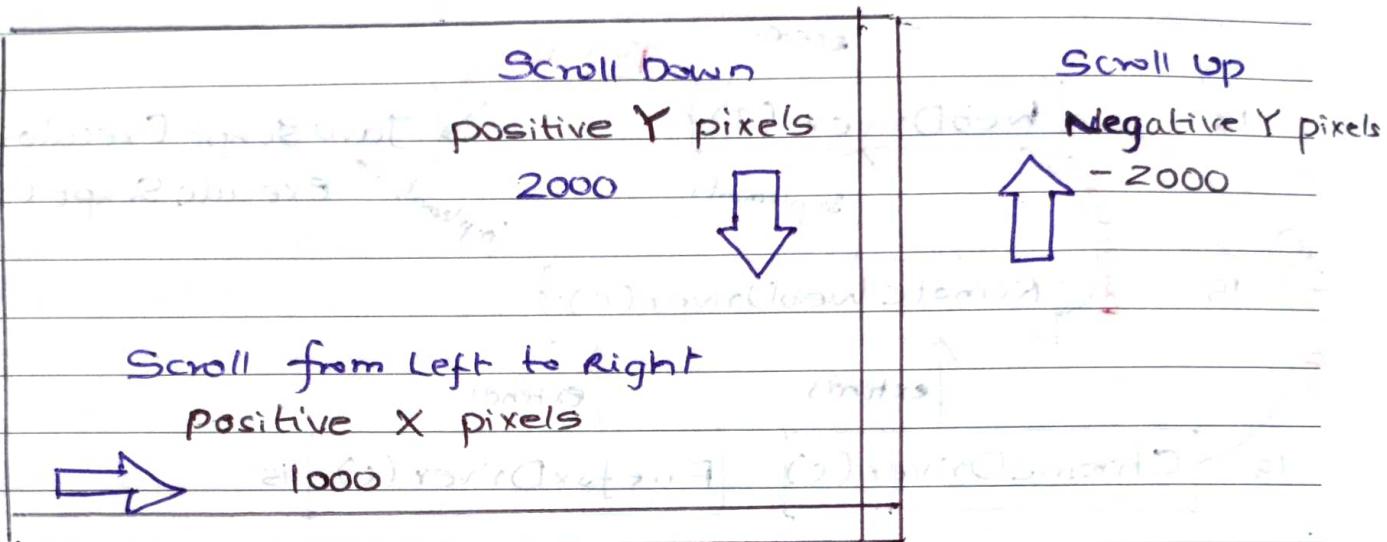
2019

23

Wk-12 • 082-283

Q: How do we scroll down?

we use javascript



Scroll from Right to Left  
Negative X pixels  
-1000

Q: How to scroll down manually?

Step 1: Go to the required webpage and

Press "Ctrl+Shift+I" Or, "F12 / Fn F12"  
"Developer Tools" will be displayed.

Step 2: Click on the "Console tab" and write the javascript as follows,

Sym: `window.scrollBy(x axis, y axis)`

Examp: `window.scrollBy(100, 800)`

Q: How to scroll down through Automation(selenium WebDriver)???

24 Sunday

In Selenium we use `executeScript()` of JavaScriptExecutor interface to execute the required javascript automatically...

25

March  
Monday

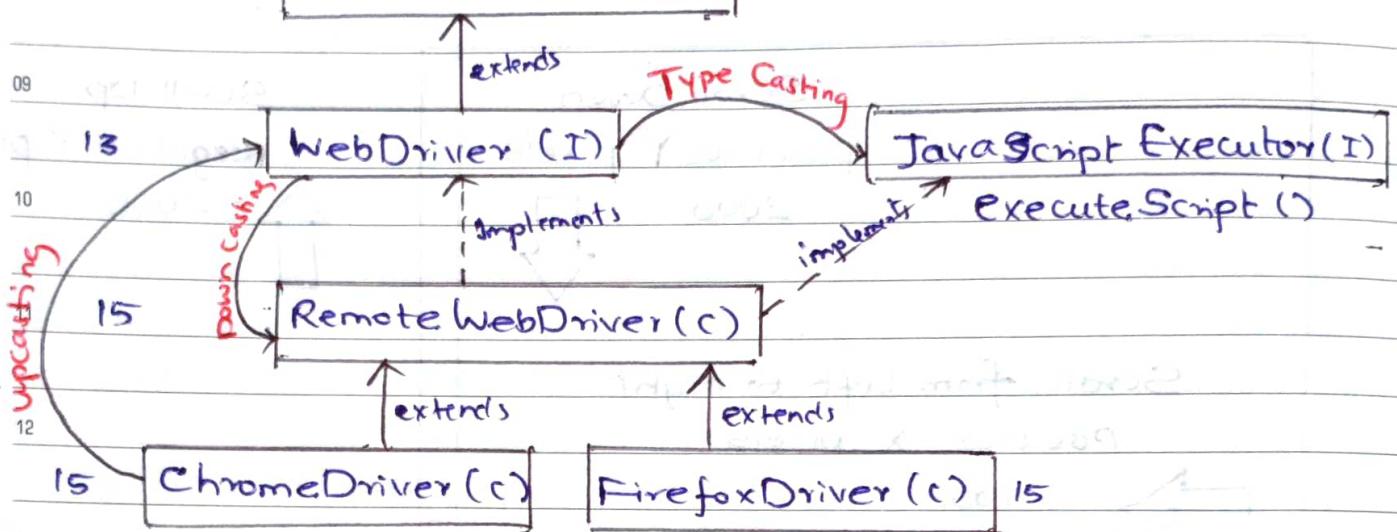
Wk-13 • 084-281

2019

MARCH '19	31	3	4	5	6	7	8	9
	10	11	12	13	14	15	16	
	17	18	19	20	21	22	23	
	24	25	26	27	28	29	30	
	S	M	T	W	T	F	S	

08 2

## SearchContext (I)



02 CD dr = new CD(); ✓ RWD dr = new CD();  
dr.executeScript(); ✓ dr.executeScript(); ✓

03 JSE dr = new CD(); ✓ WD dr = new CD();  
dr.executeScript(); ✓ dr.executeScript(); ✗ ✗

05 Since we are upcasting our driver to WebDriver interface, `executeScript()` method is not accessible.

Hence to access it we either,

07 TypeCast from WebDriver interface to JavascriptExecutor interface.

(OR) Downcast from WebDriver interface to RemoteWebDriver class.

Q: How to scroll down to a particular element?

If we want to scroll down to a particular element, then we get the location of the element (location means X axis and Y axis).

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

March  
Tuesday

2019

26

Wk-13 • 085-280

by using getlocation() method of WebElement interface.

→ getlocation() returns Point(x axis, y axis) of the element.

window.scrollBy + location of Element

Eg:

```
WebElement element = driver.findElement(By.xpath("//h2[.= 'Selenium Level sponsors']"));
```

```
Point loc = element.getLocation();
```

```
JavaScriptExecutor jse = (JavaScriptExecutor)driver;
jse.executeScript("window.scrollBy" + loc);
```

### Example :

```
package asp1;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.remote.RemoteWebDriver;
```

public class ScrollingDownAWebPage

```
public static void main(String[] args) throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "D:\chromedriver\chromedriver.exe");
}
```

// Upcasted from ChromeDriver class to WebDriver Interface

```
WebDriver driver = new ChromeDriver();
driver.manage().window().maximize();
driver.get("https://www.selenium.dev/downloads/");
Thread.sleep(2000);
```

27

March  
Wednesday

Wk-13 • 086-279

2019

MARCH '19	31	1	2
	3	4	5
	10	11	12
	17	18	19
	24	25	26
S	M	T	W
			T
			F
			S

// Through Type Casting

08 JavascriptExecutor js = (JavascriptExecutor) driver;  
to 09 js.executeScript("window.scrollBy(0,1000);");

// Through DownCasting

10 RemoteWebDriver rwd = (RemoteWebDriver) driver;  
11 rwd.executeScript("window.scrollBy(0,1000);");

12 WebElement selspans = driver.findElement(By.xpath("//h2[text()='Selenium Level Sponsors']);");

13 System.out.println(loc);

01 Point loc = Selspans.getLocation();

02 JavaScriptExecutor j = (JavascriptExecutor) driver;

03 j.executeScript("window.scrollBy"+loc);

4

03 You can't handle the element because it is disabled.

## 04 HANDLING DISABLED ELEMENT

05 → Disabled elements are created by 'Disabled' attribute.

06 → It is handled by Javascript.

07 If we try to handle disabled element without javascript, i.e., by findElement(), click(), sendKeys() then we get ElementNotInteractable Exception.

⇒ Handling Disabled element manually:

1. Go to the required webpage, where disabled element is present.
2. Press F12/ Function + F12 (ctrl+shift+I) - Developer tools will be opened.

APRIL '19
1
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30
S M T W T F S

March  
Thursday

28

2019

Wk-13 • 087-278

3. Click on console tab, write the required javascript and press enter (execute javascript)  
 document.getElementById('i2').value = 'manager'

09

Handling Disabled element through Automation  
 By using executeScript() of JavascriptExecutor interface.  
 Since, we upcast our browser class to WebDriver Interface, this method is hidden.  
 Hence we either Typecast from WebDriver Interface to JavascriptExecutor Interface  
 or

Downcast from

WebDriver Interface to RemoteWebDriver class

### HTML Document

```
<html>
  <body>
    Username<input type='text' id='i1'> </input> <br>
    Password<input type='text' id='i2' disabled>
  </body>
</html>
```

### EXAMPLE

```
package apsp1;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
```

29

March  
Friday

Wk-13 • 088-277

2019

	31					1	2
MARCH '19	3	4	5	6	7	8	9
	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	S	M	T	W	T	F	S

```
import org.openqa.selenium.chrome.ChromeDriver;
```

(90 mm) est le plus courant.

public class HandlingDisabledElement

```
public static void main(String[] args) throws InterruptedException {
    System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("file:///D:/kcsme/html/Disabled Element.html");
```

```
02 import java.util.concurrent.TimeUnit;
03 Thread.sleep(2000); // wait at login page
04 driver.findElement(By.id("username")).sendKeys("admin");
05 Thread.sleep(2000); // wait until login
06 WebElement pwtb = driver.findElement(By.id("password"));
07 if(pwtb.isEnabled())
08 {
09     pwtb.sendKeys("password");
10 }
```

```
System.out.println("Password textbox is enabled,  
Handle Normally");  
pwtb.sendKeys("manager");
```

else

```
System.out.println("Password textbox is " + disabled);
Handle through Javascript");
JavascriptExecutor jse = (JavascriptExecutor) driver;
jse.executeScript("document.getElementById('id').  

    value='manager'");
```

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				S

24.09.21

March

D19 Saturday

30

2019

Wk-13 • 089-276

## TAKING SCREENSHOTS THROUGH AUTOMATION

2

SearchContext (I)

extends

WebDriver (I)

TypeCasting

TakesScreenshot (I)

Upcasting  
Downcasting

implements

implements

RemoteWebDriver (C)

getScreenshotAs()

EventFiringWebDriver (C)

extends

extends

ChromeDriver (C)

FirefoxDriver (C)

WebElement (I)

extends

CD dr = new CD(); RWD dr = new CD();

dr.getSA(); dr.getSA(); ✓ ✓

PDS dr = new PDS(); WO dr = new WO();

TS dr = new TS(); dr.getSA(); ✓ ✓

dr.getSA(); X

We use `getScreenshotAs()` method of `TakesScreenshot` Interface to take a screenshot.

`getScreenshotAs()` is given implementation in `RemoteWebDriver` class and it is extended in browser classes.

But as a Selenium standard we always upcast our browser object to `WebDriver` interface,

```
WebDriver driver = new ChromeDriver();
```

As a result `getScreenshotAs()` is hidden and inaccessible.

31 Sunday

01

April  
Monday

Wk-14 • 091-274

2019

19	2	3	4	5	6
20	7	8	9	10	11
21	14	15	16	17	18
22	21	22	23	24	25
23	28	29	30	26	27
24	S	M	T	W	F
25					S

Hence to access it, we either typecast from WebDriver Interface to TakesScreenshot Interface.

`TakesScreenshot ts = (TakesScreenshot) driver;`

We downcast from WebDriver interface to RemoteWebDriver class,

`RemoteWebDriver rwd = (RemoteWebDriver) driver;`

Since EventFiringWebDriver class is also implementing it, we can create object and access it

`EventFiringWebDriver e = new EventFiringWebDriver(driver);`

By above ways we can take a full page Screenshot

But if we have to take a "Particular Element's" screenshot. Then, since WebElement interface extends

TakesScreenshot interface, we can access through WebElement by getting the address of element and then taking screenshot of it.

Eg:

`WebElement imgAddress = driver.findElement(By.id("hpLogo"))`

`FileInputStream src = imgAddress.getScreenshotAs(OutputType.FILE)`

## Explaining TakesScreenshot Program in Interview

1. Upcasting Browser class to WebDriver Interface
2. Typecasting from WebDriver to TakesScreenshot interface.
3. Getting the Screenshot by calling `getScreenshotAs()` which takes argument, `OutputType` which is file type.

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	S
	S	M	T	W	T	F

April

Tuesday

2019

02

Wk-14 • 092-273

4. we create an object of File class and in  
08 constructor we pass three things.
- a. Path where we need to store the screenshot
  - b. filename with which we want to store
  - c. Extension (.png / .jpg)

5. Using third party class called "Files" which is  
11 imported from com.google.common.io package.  
we copy the obtained screenshot and paste it  
12 in the specified path.  
Files.copy() throws a compile time exception  
01 called IOException.

## 02 Example:

```
03 package arsp1;
04 import java.io.File;
05 import java.io.IOException;
06 import org.openqa.selenium.By;
07 import org.openqa.selenium.OutputType;
08 import org.openqa.selenium.TakesScreenshot;
09 import org.openqa.selenium.WebDriver;
10 import org.openqa.selenium.WebElement;
11 import org.openqa.selenium.chrome.ChromeDriver;
12 import org.openqa.selenium.remote.RemoteWebDriver;
13 import org.openqa.selenium.support.events.EventFiringWebDriver;
14 import com.google.common.io.Files;
15 public class TakingScreenshot
16 {
17     public static void main(String[] args) throws
18         IOException
```

# 03

April  
Wednesday

Wk-14 • 093-272

2019

	1	2	3	4	5	6
1	7	8	9	10	11	12
2	14	15	16	17	18	19
3	21	22	23	24	25	26
4	28	29	30			
5	S	M	T	W	T	F

```
System.setProperty("webdriver.chrome.driver",  
08 "C:/drivers/chromedriver.exe");  
WebDriver driver = new ChromeDriver();  
09 driver.manage().window().maximize();  
driver.get("https://www.google.com/");  
10
```

// Through Type Casting

```
11 TakesScreenshot ts = (TakesScreenshot) driver;  
12 // Take Screenshot and store in heap area  
13 File src = ts.getScreenshotAs(OutputType.FILE);  
14 // But we want in our location  
15 // path + filename + extension  
File dest = new File ("D:\\KCSMS\\Screenshots\\  
02 GoogleImage.jpg");  
16 // Copy from Src and paste it in dest  
03 Files.Copy(src, dest);
```

// Through Downcasting

```
RemoteWebDriver rwd = (RemoteWebDriver) driver;  
05 File src = rwd.getScreenshotAs(OutputType.FILE);  
File dest = new File ("D:\\KCSMS\\Screenshots\\  
06 GooglePhoto.png");  
07 Files.Copy(src, dest);
```

// Taking screenshot of a particular element

```
WebElement googleLogo = driver.findElement(By.  
xpath("//img[@alt='Google']"));  
File src = googleLogo.getScreenshotAs(OutputType.  
FILE);  
File dest = new File ("D:\\KCSMS\\Screenshots\\  
08 GoogleLogo.png");  
Files.Copy(src, dest);
```

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	T	F	S

April  
Thursday

2019

04

Wk-14 • 094-271

// Through EventFiringWebDriver class

```
08 EventFiringWebDriver ef = new EventFiringWebDriver(driver);
09 File src = ef.getScreenshotAs(OutputType.FILE);
File dest = new File("D:\\\\KCSM6\\\\Screenshots\\\\")
10 GooglePhoto.png");
Files.copy(src, dest);
```

25.09.2021

D20

## 01 Handling Web Table:

02 Web table is created by using tags like

<table>

03 <th> / <thead> → table head

<tbody> → table body

04 <tr> → table row

05 <td> → table data

06 We can handle Web table by using findElement()

07 And findElements() of WebElement Interface

### 07 findElement() of

WebDriver (I)

It will search full page and get address of first matching element on the webpage.

webElement (I)

It will search within the current element and get the address of first matching element within the specified element.

05

April  
Friday

Wk-14 • 095-270

2019

		1	2	3	4	5	6
APRIL '19		7	8	9	10	11	12
		14	15	16	17	18	19
		21	22	23	24	25	26
		28	29	30			27
	S	M	T	W	T	F	S

findElements() of

08 WebDriver (I)

## WebElement (I)

09 It will get the address  
of all the matching  
10 elements after search-  
ing on full webpage

It will get the address of all the matching elements within the current element.

\*<sup>42</sup> XPath to handle dynamic webtable

```
//table[attName = 'attrval']//tr[*]//td[cols num]"
```

### EXAMPLE :

```
02 package arsp1;
03 import java.util.List;
04 import org.openqa.selenium.By;
05 import org.openqa.selenium.WebDriver;
06 import org.openqa.selenium.WebElement;
07 import org.openqa.selenium.chrome.ChromeDriver;
```

06 public class HandlingWebTables

```
07 public static void dynamicTable(WebDriver driver, int colNum){
```

```
List<WebElement> Col = driver.findElements(By.  
xpath("//table[@id='t2']//tr[*]//td["+colNum+"]");  
System.out.println(Col.size());  
}
```

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
-	-	-	W	T	F	S

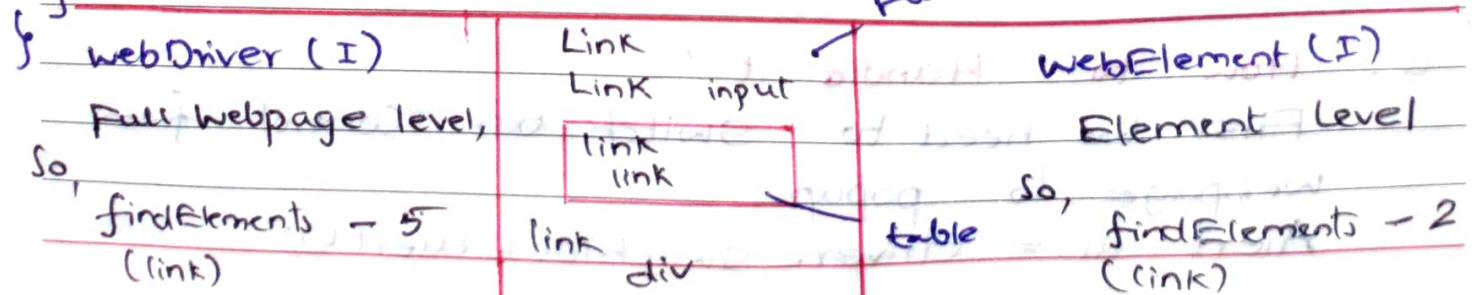
April  
Saturday

06

2019

Wk-14 • 096-269

```
public static void main(String[] args)
{
    System.setProperty("webdriver.chrome.driver",
        "./drivers/chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.get("file:///D:/KCSMG/HTML/webtable.html");
    List<WebElement> alltrs = driver.findElements(By.
        tagName("tr"));
    System.out.println("Total rows on webpage: "
        + alltrs.size());
    WebElement table2 = driver.findElement(By.id("t2"));
    List<WebElement> table2Rows = table2.findElements(
        By.tagName("tr"));
    System.out.println("table 2 rows: " + table2Rows.size());
    WebElement tr3Row = driver.findElement(By.xpath(
        "table[@id='t2']//tr[3]"));
    List<WebElement> tr3data = tr3Row.findElements(
        By.tagName("td"));
    System.out.println("3rd row data: " + tr3data.size());
    // Handling Dynamic webtable
    driver.get("file:///C:/Users/HP/Desktop/HTML/dynamicweb
        table.html");
    dynamicTable(driver, 7);
}
```



# 08

April  
Monday

Wk-15 • 098-267

2019

27.09.21

D21

APRIL '19	1	2	3	4	5	6
S	7	8	9	10	11	12
M	14	15	16	17	18	19
T	21	22	23	24	25	26
W	28	29	30			

## HANDLING POP-UPS

08

Pop-ups are anything which gives information to user or takes information / confirmation from the user.

They are of 2 types of pop-ups

1. Web-based Popup
2. Window-based Popup

Web-based Popup :

Here, we have,

1. Javascript popup
2. Hidden Division Popup
3. Browser Notification popup

→ Javascript Popup:

Created by using Javascripts.. hence the name

Types:

1. Alert (only OK button)
2. Confirmation (Both OK and Cancel button)

Characteristics:

1. Is it Colorful ? - No
2. Is it Movable ? - No
3. Is it inspectable ? - No
4. Is it Until we handle this popup we can't continue with webpage.

Q: How to Handle it?

First we need to switch our control from webpage to popup

Alert al = driver.switchTo().alert();

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	S
S	M	T	W	T	F	S

April

Tuesday

2019

09

Wk-15 • 099-266

After switching we handle by using the abstract methods of Alert Interface,

al.accept() - Clicks on OK button

al.dismiss() - Clicks on Cancel button

al.getText() - Used to get the message on popup

al.sendKeys("String"); - Used to type any message to popup.

### Note:

If there are no Javascript popups to handle, but still we try to switch and handle then we get NoAlertPresentException

→ Hidden Division Popup:

### Characteristics:

1. Is it Colorful? - YES

2. Is it Movable? - NO

3. Is it Inspectable? - YES

### How to handle it?

Since we can inspect it, we can handle by using findElement(), click() and sendKeys().

Ex:

```
<html>
<body>
<h2> JavaScript Alert </h2>
<button onclick="myFunction()"> Try It </button>
<script>
function myFunction()
{
    alert("I am an Alert box!");
}
```

10

April  
Wednesday

Wk-15 • 100-265

2019

APRIL '19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S

```

7 </script>
8 </body>
9 </html>
10 <html>
11 <body>
12 <h2> JavaScript Confirm Box </h2>
13 <button onclick="myFunction()"> I Love You </button>
14 <p id="demo"></p>
15 <script>
16 function myFunction()
17 {
18     var text;
19     if (confirm("Will you marry me ???")) {
20         txt = "You pressed OK! Aww... Cho Chweet... Love
21         you tooo;";
22     } else {
23         txt = "You pressed Cancel! Get lost Dabba, Fellow!";
24     }
25     document.getElementById("demo").innerHTML = txt;
26 }
27 </script>
28 </body>
29 </html>

```

		1	2	3	4	
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	T	F	S

MAY '19

April

Thursday

2019

11

Wk-15 • 101-264

EXAMPLE

```

08 package arsp1;           /* (over) 1930 to present */

09 import org.openqa.selenium.Alert;
10 import org.openqa.selenium.By;
11 import org.openqa.selenium.WebDriver;
12 import org.openqa.selenium.chrome.ChromeDriver;

13 public class Handling_Popups
14 {
15     public static void main(String[] args) throws
16         InterruptedException
17     {
18         System.setProperty("webdriver.chrome.driver", ".\chromedriver.exe");
19         WebDriver driver = new ChromeDriver();
20         driver.manage().window().maximize();
21         driver.get("file:///D:/KCSM6/HTML/Alert.html");
22         Thread.sleep(2000);
23         driver.findElement(By.xpath("//button[text()='Try It']"))
24             .click();
25         Alert al = driver.switchTo().alert();
26         Thread.sleep(2000);
27         System.out.println(al.getText());
28         Thread.sleep(2000);
29         al.accept();
30         driver.get("file:///D:/KCSM6/HTML/Confirm.html");
31         Thread.sleep(2000);
32         driver.findElement(By.xpath("//button[text()='I love
33         you']")).click();
34         Thread.sleep(1000);
35     }
36 }
```

12

April  
Friday

Wk-15 • 102-263

2019

APRIL '19	1	2	3	4	5	6
S	7	8	9	10	11	12
M	14	15	16	17	18	19
T	21	22	23	24	25	26
W	28	29	30			

Alert al = driver.switchTo().alert();

Thread.sleep(1000);

System.out.println(al.getText());

Thread.sleep(1000);

// To click on OK button

al.accept();

// To click on Cancel button

Thread.dismiss();

// Handling Hidden Division popup

System.setProperty("webdriver.chrome.driver",  
".\drivers\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

driver.get("https://www.cleartrip.com/");

Thread.sleep(3000);

driver.findElement(By.xpath("//button[@class =  
"flex flex-middle flex-between t-all fs-2 focus:bc-  
secondary - 500 \* bg-transparent bc-neutral - 100 c-pi-  
nter pr-2 pl-3 pt-2 ph-2 ba br-4 h-8 c-neutral-  
900 "])) .click();

Thread.sleep(3000);

driver.findElement(By.xpath("//div[contains(@aria-  
label, 'oct')]/descendant :: div[text() = '28']")).click();

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	T	F	S

28.09.21

D22

Saturday

2019

WK-15 • 103-262

April

13

→ Browser Notification Popup:

characteristics

1. It is not Colorful
2. It is not movable
3. It is not Inspectable
4. But, We can continue with a webpage without even handling it.
- This is the only difference between Javascript Popup and Browser Notification Popup.

Window Based Popup:

Selenium Can't automate window based applications. Hence we take help from third party tools like Robot Class or AutoIT.

Characteristics:

1. It is Colourful.
2. It is movable
3. It is not Inspectable.
4. We can't go further before handling it.

Examples:

File upload

File Download

Printscreen (Screenshot)

Scrolling (scrolling)

Q: How to handle File upload?

Trick: Use sendkeys() with complete path of the file which has to be uploaded.

If you try to handle file upload using Selenium, you get an exception - InvalidArgument Exception

15

April  
Monday

Wk-16 • 105-260

2019

	1	2	3	4	5	6
APRIL '19	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30			27
	S	M	T	W	T	F

So, To copy the complete path of window popup  
 " Shift + RightClick --> Copy as path.

Proper way: AutoIT

Q: How to handle File Download?

Trick: Use Robot class

Proper way: AutoIT

Note: No need to handle download popup in chrome browser. Because, it will not show popup but it auto-downloads any file.

But, to handle it in firefox browser, use Robot class.

### EXAMPLE:

```

03 import java.awt.event.KeyEvent;
04 import org.openqa.selenium.By;
05 import org.openqa.selenium.WebDriver;
06 import org.openqa.selenium.WebElement;
07 import org.openqa.selenium.chrome.ChromeDriver;
08 import org.openqa.selenium.firefox.FirefoxDriver;
09 import java.awt.Robot; import java.awt.AWTException;
10 public class HandlingWindowPopups
  
```

```

11 {
12   public static void main(String[] args) throws
13     InterruptedException, AWTException
  
```

```

14   // Handling File Upload popup
  
```

```

15   System.setProperty("webdriver.chrome.driver",
16     "./drivers/chromedriver.exe");
  
```

```

17   WebDriver driver = new ChromeDriver();
18   driver.manage().window().maximize();
  
```

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	T	F	S

April

Tuesday

2019

16

Wk-16 • 106-259

```

driver.get("https://www.monsterindia.com/");
08 driver.findElement(By.xpath("//span[text()='Upload
Resume']")).click();
09 Thread.sleep(2000);
10 // Hidden Division Popup will be opened.
WebElement uploadbtn = driver.findElement(By.id("file-
upload"));
11 uploadbtn.sendKeys("C:\\Users\\HP\\Desktop\\My Resume.docx");
12
13 // File Download Popup
01 System.setProperty("webdriver.gecko.driver",
".\\drivers/geckodriver.exe");
02 WebDriver driver = new FirefoxDriver();
driver.manage().window().maximize();
03 driver.get("https://www.selenium.dev/downloads/");
Thread.sleep(2000);
04 driver.findElement(By.xpath("//p[@text() = 'Java']/following-
sibling::p[@class='']"));
05 Thread.sleep(2000);
Robot r = new Robot();
06 r.keyPress(KeyEvent.VK_DOWN);
Thread.sleep(2000);
r.keyPress(KeyEvent.VK_ENTER);
r.keyRelease(KeyEvent.VK_DOWN);
r.keyRelease(KeyEvent.VK_ENTER);
11 Handling Browser Notification Popup
System.setProperty("webdriver.chrome.driver",
".\\drivers/chromedriver.exe");
12 // Change the browser settings
ChromeOptions co = new ChromeOptions(); //FirefoxOptions
//OperaOptions

```

		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30				
	S	M	T	W	T	F	S

```

08 Co. addArguments(" --disable-notifications");
09 Co. addArguments("start-maximized"); // instead of
10 // open the browser with the changes
11 WebDriver driver = new ChromeDriver(co);
12 }

13 } // main class
14 }
```

29.09.2021

D23

## Handling Frames

Frame: One webpage present inside another webpage is called inner webpage / nested webpage or frames.

Frames are created by using iframe tag.

By using src attribute we pass path of other webpage which we need on current webpage.

To check on the webpage whether an element is present inside frame, Right click and if we see word 'frame' in the options then the element is inside the frame (Firefox Browser).

Q: How to handle frames?

Handled by switching to frame by using overloaded frame() method.

Three ways to switch:

frame(int arg) - index of the frame, index - 0

frame(String arg) - either 'id' or 'name' of the frame

frame(webElement arg) - address of the frame

MAY '19		1	2	3	4
	5	6	7	8	9
	12	13	14	15	16
	19	20	21	22	23
	26	27	28	29	30
S	M	T	W	T	F
					S

April

Thursday

2019

18

Wk-16 • 108-257

Q: How to switch control back from a frame to parent frame?

To switch from a frame to its immediate parent frame, we use `driver.switchTo().parentFrame()`

Q: How to exit from all the frames and take our control back to the default webpage?

We use `driver.switchTo().defaultContent();`

Q: How to switch control back from one frame to another frame (not to immediate frame)?

To switch from one frame to another, first switch from a current frame to default webpage and then from default webpage to, switch to the desired frame.

→ If an element is present inside the frame and we are trying to handle it without switching we get `NoSuchElementException`.

→ If there is no frame, but still we try to switch (or)

→ If we are switching to a frame by giving incorrect frame details then we get, `NoSuchFrameException`.

### HTML Source Codes.

Frame : page2.html

```
<html>
<body>
    PASSWORD <input type='text' id='i2'>
</body>
</html>
```

19

April  
Friday

Wk-16 • 109-256

2019

APRIL '19	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				
S	M	T	W	T	F	S

Main page :

&lt;html&gt;

&lt;body&gt;

USERNAME &lt;input type='text' id='ii'&gt;&lt;/input&gt;

&lt;iframe src='page2.html' id='frid' name='frm1' class='frc1'&gt;&lt;/iframe&gt;

&lt;/body&gt;

&lt;/html&gt;

**EXAMPLE:**

package aisp1;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

03

public class HandlingFrames

public static void main(String[] args) throws

InterruptedException

System.setProperty("webdriver.chrome.driver",

"./drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().windows().maximize();

driver.get("file:///C:/Users/HP/Desktop/HTML/Page1.htm");

Thread.sleep(2000);

driver.switchTo().frame(0); // Through index

driver.switchTo().frame("frid"); // Through id or name

driver.switchTo().frame("frm1"); // Through class

driver.switchTo().frame("frc1"); // Through webpage address

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	T	F	S

April

Saturday

20

2019

Wk-16 • 110-255

WebElement frAddr = driver.findElement(By.id("frid"));  
 08 driver.switchTo().frame(frAddr);  
 driver.findElement(By.id("i2")).sendKeys("manager");  
 09 Thread.sleep(2000);

10 driver.switchTo().parentFrame();  
 driver.switchTo().defaultContent();  
 11 driver.findElement(By.id("ii")).sendKeys("admin");

12

## HANDLING NEW WINDOW / NEW TAB (OR)

### HANDLING CHILD BROWSER POPUP

#### Characteristics:

1. It is Colorful.
2. It is movable.
3. It is inspectable.
4. It has minimize and maximize buttons.

#### Q: How to handle it?

First we need to switch our control from webpage to child browser window.

Hence we use, driver.switchTo().window(window handle)

To get window handle we use

getWindowHandle()

getWindowHandles()

Window Handle: It is an alpha-numeric String which is nothing but the unique address generated for every browser window whenever we open it.

22

April  
Monday

Wk-17 • 112-253

2019

	1	2	3	4	5	6
APRIL '19	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30			27
S	M	T	W	T	F	S

getWindowHandle(): String

String - Used to get the window handle of the current browser. (Window.getHandle() doesn't work)

getWindowHandles(): Set<String>

Used to get the window handles of all the browser windows.

Closes vs quits()

Close() - closes the current browser window

quit() - closes all the browser windows which are opened through Selenium.

### Important Interview Questions:

1. Can you close a parent browser window only?  
Yes, by using driver.close()2. Can you close all the browser windows at once?  
Yes, by using driver.quit()

3. Can you close all the browser windows without using quit()?

Yes, first we get window handle of all the browsers by using getWindowHandles(). Then we switch to each and every browser and close it by using close().

		1	2	3	4
5	6	7	8	9	10
12	13	14	15	16	17
19	20	21	22	23	24
26	27	28	29	30	31
S	M	T	W	T	F

April

Tuesday

2019

23

Wk-17 • 113-252

4. Can you perform action on a particular browser?

08 Yes, we get the title of every browser we switch and compare to the one which we want.

5. Can you close all the child browsers only?

11 Yes, First we take parent browser window title, then we write Condition if title not matching, close the browser.

(Action on particular)

NoSuchWindowException : Selenium - Runtime

01 Reason: If we are trying to switch to a window which is closed and session is expired then we get this exception.

03 EXAMPLE:

```
package arsp1;
04 import java.util.Set;
import org.openqa.selenium.By;
05 import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

public class HandlingChildBrowserPopup

```
public static void main(String[] args) throws Interr-
uptedException
```

```
{ System.setProperty("webdriver.chrome.driver",
```

```
"./drivers/chromedriver.exe");
```

```
WebDriver driver = new ChromeDriver();
```

```
driver.manage().window().maximize();
```

```
driver.get("https://www.haukari.com/");
```

24

April  
Wednesday

Wk-17 • 114-251

2019

APRIL '19		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30				
	S	M	T	W	T	F	S

07 // closing current browser window only  
 08 driver.close(); // closing all browsers at once.

09 // To close all the browser windows without using quit():

```
10 Set<String> allwhs = driver.getWindowHandles();
11 System.out.println(allwhs.size()); // it will print
12 for (String wh : allwhs)
13 {
14     driver.switchTo().window(wh); // switch to that window
15     driver.close(); // close that window
16     Thread.sleep(1000); // wait for 1 sec
17 }
```

03 // To perform action on a particular browser?

```
04 Set<String> allwhs = driver.getWindowHandles();
05 for (String wh : allwhs)
06 {
07     String title = driver.switchTo().window(wh).getTitle();
08     if (title.equals("Cognizant"))
09     {
10         driver.manage().window().maximize(); // along
11         Thread.sleep(3000);
12         driver.findElement(By.xpath("//img[@alt='Cognizant']"));
13         Thread.sleep(3000);
14         driver.quit();
15         break;
16     }
17 }
```

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	S

MAY 19

April

Thursday

2019

25

Wk-17 • 115-250

// To close all the child browsers only:

```

08 // To close all the child browsers only:
09 String parentTitle = driver.getTitle();
10 Set<String> allwhs = driver.getWindowHandles();
11 for(String wh: allwhs)
12 {
13     String title = driver.switchTo().window(wh).getTitle();
14     if(!title.equals(parentTitle))
15     {
16         driver.close();
17     }
18 }
19 driver.switchTo().window(wh); // NoSuchElementException
20 
```

[01.1021]

[01.024]

[D24]

### WEB ELEMENT (I) - METHODS (Element Level)

Method Type	S.No	Method Name	Return Type	Method Use
Operational Methods	1.	click()	void	Used to click on any element.
	2.	clear()	void	Used to clear the content of the element.
	3.	sendKeys(CharSequence sequence)	void	Used to send character sequence into the element.
	4.	getTagName()	String	Used to get the tagName of the element.
Capture Methods	5	getAttribute(String attrName)	String	Used to get a particular attribute value of the element by giving the attribute name

26

April  
Friday

Wk-17 • 116-249

2019

	1	2	3	4	5	6
APRIL '19	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30			27
	S	M	T	W	T	F

Method Type	SDO	Method Name	Return Type	Method Use
08	6.	getText()	String	Used to get the text of the element
09	7.	getLocation()	Point	Used to get location of the element (means X pixel, Y pixel)
10	Data			
11	Capture	8. getRect()	Rectangle	Used to get X Axis, Y Axis, height, width
12	Methods	9. getCssValue(String cssName)	String	Used to get a particular CSS attribute value of the element by giving the css attribute name
13	10. getSize()	Dimension	Used to get the size of the element (i.e. length & breadth)	
14	11. getScreenshotAs(File, Output Type)	File	Used to get the screenshot of a particular element	
15	12. isDisplayed()	boolean	Used to verify if an element is present on the webpage.	
16	13. isEnabled()	boolean	Used to verify if an element is enabled or disabled (enabled means ready to interact)	
17	14. isSelected()	boolean	Used to verify if an element is selected or not.	

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	F	S	

April  
Saturday

2019

27

Wk-17 • 117-248

Method Type	S.No	Method Name	Return Type	Method Use
Inspector Methods	15	findElement (By arg)	WebElement	Used to find a particular element within the current element
Operational Method.	16	findElements (By arg)	List<WebElement>	Used to find all the matching elements within the current element
	17	Submit()	void	Used like click() method but only can be used on buttons with type = 'submit'

EXAMPLE : `new WebDriverWait(driver, 20).until(ExpectedConditions.elementToBeClickable(By.name("submit"))).click();`

```
package arsp1.selenium;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class WebElementMethods
```

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

28 Sunday

```
System.setProperty ("webdriver.chrome.driver", "C:\Drivers\chromedriver.exe");
```

"C:\Program Files\Java\jdk-11.0.1\bin" > javac -javap

29

April  
Monday

Wk-18 • 119-246

2019

APRIL '19		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30				
	S	M	T	W	T	F	S

```
08 WebDriver driver = new ChromeDriver();
09 driver.manage().window().maximize();
10 driver.get("https://demo.actitime.com/login.do");
11
12 WebElement untb = driver.findElement(By.id("username"));
13 untb.sendKeys("admin");
14 Thread.sleep(2000);
15 untb.clear();
16
17 System.out.println("TagName: " + untb.getAttribute("placeholder"));
18
19 WebElement pwtb = driver.findElement(By.name("pwd"));
20 pwtb.sendKeys("raghav");
21
22 System.out.println(pwtb.getAttribute("value"));
23 " " (untb.getLocation());
24 " " (untb.getRect().x);
25 " " (untb.getRect().y);
26 " " (untb.getSize());
27 " " (untb.getRect().width);
28 " " (untb.getRect().height);
29 " " (untb.getCssValue("font-size"));
30
31 WebElement checkbox = driver.findElement(By.id("keepLoginChecked"));
32 checkbox.click();
33
34 if (checkbox.isDisplayed())
35 {
36     System.out.println("checkbox is displayed");
37 }
```

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
S	M	T	W	T	F	S

April  
Tuesday

30

2019

Wk-18 • 120-245

08      if (checkbox. is ~~selected~~ enabled)

System.out.println (" checkbox is Enabled");

09      if (checkbox. isSelected())

10     {      System.out.println (" checkbox is Selected, Don't touch now");

11     }

12     {

System.out.println (" click now");

01     checkbox.click();

}

02     }

else

{

03     System.out.println (" Checkbox is Disabled");

}

04     }

else

{

05     System.out.println (" Checkbox is not displayed");

}

06     }

07     }

08     }

Java

						1
30	3	4	5	6	7	8
2	9	10	11	12	13	14
16	17	18	19	20	21	22
23	24	25	26	27	28	29
S	M	T	W	T	F	S

05.10.2021

D25

May

Thursday

2019

02

Wk-18 • 122-243

## AUTOMATION FRAMEWORK

- Framework is set of rules and/or guidelines, files and folder structure (standard folder template) that every automation engineer should follow while automating the application  
 (OR)

Framework is collection of reusable generic methods, which makes automation engineers life easy and also makes Coding efficient.

### Advantages of Automation Framework:

- Contains generic reusable methods which help in code reusability.
- Since every automation engineer follows the same format, we can achieve consistency in Automation testing.
- Since we follow framework which is very organised, it is easy to develop scripts and easy to understand framework.
- Maintaining the developed scripts is easy.
- Reduces Coding effort and reduces time.
- Increases Coding efficiency.

### Stages of Framework:

- Design :** (Generic)  
 Responsible for Design, mostly taken care by the any experienced member.
- Implementation :** (Specific to Project)  
 Responsible for implementation, done by Seniors

# 03

May

Friday

Wk-18 • 123-242

2019

MAY 19	5	6	7	8	9	10	11
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	
	S	M	T	W	T	F	S

### 3 Execution :

- Responsible for execution - Any Automation engineer.

### Types of Framework :

1. Data-Driven Framework
2. Keyword-Driven Framework
3. Method Driven Framework
4. TestNG
5. Modular Framework
6. Hybrid Framework

#### 02 Data - Driven Framework:

Storing test data in external files like

Excel sheets, XML, Database, Code, CSV files, Property file and fetching the data through generic reusable methods and passing that data to the application and testing the application is known as Data-Driven Framework.

#### 01 Why Data - Driven ?

As a part of functional testing, we test each and every Component of the application with both valid and invalid data.

And since there will be many components, the test data is huge.

And if we hard-code test data then code size will be more.

Also difficult to maintain the test data. Hence we keep it in external files and fetch them from here.

JUNE '19	30	31	1
	2	3	4
	5	6	7
	8	9	10
	11	12	13
	14	15	16
	17	18	19
	20	21	22
	23	24	25
	26	27	28
	29		
S	M	T	W
F	S		

May  
Saturday

2019

04

Wk-18 • 124-241

Q: Why do we prefer to store test data in Excel?

1. Maintaining the test data in Excel is very easy (Adding, modifying and deleting)
2. Already manual test engineers would have used Excel to store test data, so we just write generic reusable codes to fetch them.

To achieve Data-Driven (read data) from excel Sheet, we need to download Apache POI jars.

#### Steps:

1. Go to google, type 'Download Apache poi jars'.
2. Click on first link (<https://poi.apache.org/download.html>).
3. Click on 'The latest, stable release is Apache POI 5.0.0' link.
4. Click on 'poi-bin-5.0.0-20210120.tar.gz (56.01 MB)', under Binary distribution.
5. Click on the first link..
6. Once downloaded, uncompress and add all the jars to build path.
  - a. Right click on Project, go to build-path-> Configure build path.
  - b. Click on 'Libraries' tab.
  - c. Click on 'Add External Jars'.
  - d. Go to the location where poi jars are downloaded and select all the jars in poi-4.1.2 folder and also in 'lib', 'ooxml-lib'

# 06

May  
Monday

Wk-19 • 126-239

2019

MAY 19	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	S	S	T	W	T	F	S	S	T	W	T	F	S	S	T	W	T	F	S	S	T	W	T	F	S	S	T	F	S		

To achieve data driven testing from excel sheet,

First we need to create an excel sheet and write our test data inside it.

Create a folder called 'data' under the project and create an excel under data folder.

- Right click on Project → New → Folder (name as 'data')
- Right Click on data folder → Properties → Copy path → Open window explorer → paste it in the address bar..
- Right click inside folder → New → Excel sheet (name as inputData).

To read the data, we open the file in read mode, we use POI libraries and guide our control to a particular cell and get the value in it.

→ To read multiple data, use loop to get data one after another.

→ To write the data, we open the file in read mode, prepare the cell and once we write we open in write mode and save it.

Note: Create an Excel file, fill the data related to Freedom fighters.

30		1
2	3	4
9	10	11
16	17	18
23	24	25
S	M	T
	W	T
	F	S

JUNE '19

May  
Tuesday

2019

07  
Wk-19 • 127-238

Note:

Workbook(I) ← WorkbookFactory(C)

09 public static Sheet getSheet() { return workbook.createSheet("Sheet1"); }

10 Sheet(I) if sheet is created sheet is

11 i(left) getRow() returning dw = 2nd row

12 Row(I) no. of columns of sheet is 2

getCell("A2") returning dw = A2 cell

01 Cell(I) no. of rows in sheet is 2

getStringCellValue() for dw = A was

\* WorkbookFactory.create().getSheet().getRow().getCell().  
getStringCellValue()

04 EXAMPLE: (Read Data from Excel)

05 package org.datadrivenframework;

06 import org.apache.poi.ss.usermodel.Cell;  
07 " " : (row) along 100 . Row;  
08 " " " " " " . Sheet;  
09 " " " " " " . Workbook;  
10 " " " " " " . WorkbookFactory;

11 public class ReadDataFromExcel

12 public static void main(String args[]) throws  
13 . Exception { (Exception) handling dw = Throwable,  
14 (Exception) printStackTrace(); } 15 } 16 }

08

May  
Wednesday

Wk-19 • 128-237

2019

MAY '19	5	6	7	8	9	10	11
S	M	T	W	T	F	S	

// Open file in read mode

08

```

FileInputStream fis = new FileInputStream("D:\\KCSM6\\Automation Project\\data\\inputData.xlsx");
// Take Control of that file
Workbook wb = WorkbookFactory.create(fis);
// Go to that particular Sheet
Sheet sh = wb.getSheet("Sheet1");
// Go to that Particular Row
Row r = sh.getRow(8);
// Go to that Particular Cell
Cell c = r.getCell(1);
// Capture the String value present inside it
String x1Val = c.getStringCellValue();
// Print it
System.out.println(x1Val);
}
}

```

(OR)

06-10-2021

D26

```

FileInputStream fis = new FileInputStream("./data/inputData.xlsx");
Workbook wb = WorkbookFactory.create(fis);
System.out.println(wb.getSheet("sheet1").getRow(4).getCell(0).getStringCellValue());
}
}

```

						1
30	1	2	3	4	5	6
2	7	8	9	10	11	12
16	13	14	15	17	18	19
23	20	21	22	24	25	26
S	T	W	T	F	S	S

May  
Thursday

09

2019

Wk-19 • 129-236

## Generic Method for Excel related

### Read Multiple Data :

```

1 package datadrivenframework;
2 import java.io.FileInputStream;
3 import org.apache.poi.ss.usermodel.Workbook;
4 import org.apache.poi.ss.usermodel.WorkbookFactory;
5
6 public class ReadMultipleData
7 {
8     public static void main(String[] args) throws Throwable
9     {
10         for(int i=0; i<10; i++)
11         {
12             FileInputStream fis = new FileInputStream("./data/input-
13                 multiple.xlsx");
14             Workbook wb = WorkbookFactory.create(fis);
15             System.out.println(wb.getSheet("sheet1").getRow(i).-
16                 getCell(0).getStringCellValue());
17         }
18     }
19 }
```

### Write Data into Excel :

```

Sc, import java.io.FileOutputStream;
public class WriteDataintoExcel
{
    public static void main(String[] args) throws Throwable
    {
        // Open file in read mode
        FileInputStream fis = new FileInputStream("./data/inputData.xlsx");
    }
}
```

10

May

Friday

Wk-19 • 130-235

2019

MAY '19	5	6	7	1	2	3	4
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	
	S	M	T	W	T	F	S

// Go to particular file

08 Workbook wb = workbookFactory.create(fis);

09 // Go to particular sheet

Sheet sh = wb.getSheet("Sheet1");

10

11 // Go to particular Row

11 Row r = sh.getRow(6);

12 // Keep the cell ready for writing purpose

Cell c = r.createCell(2);

01

13 // Provide data

02

c.setCellValue("Raghav Sir");

03

14 // Open file in write mode

04

FileOutputStream fos = new FileOutputStream

("./data/inputData.xlsx");

05

15 // Save the changes made in the file

06

wb.write(fos);

System.out.println("Done writing");

07

File Library: (Generic Methods)

package datadrivenframework;

Sc

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

JUNE '19	30	1
	2	3
	4	5
	6	7
	8	
	9	10
	11	12
	13	14
	15	
	16	17
	18	19
	20	21
	22	
	23	24
	25	26
	27	28
	29	
S	M	T
	W	T
	F	S

May

Saturday

2019

11

Wk-19 • 131-234

08 { public class FileLib

09     public String readExcelData(String excelPath, String sheetName,  
                      int row, int cell) throws Throwable

10         FileInputStream fis = new FileInputStream(excelPath);  
11         Workbook wb = WorkbookFactory.create(fis);  
12         String excelValue = wb.getSheet(sheetName).getRow(row).  
                       getCell(cell).getStringCellValue();  
13         return excelValue;

01 }  
02     public int getRowCount(String excelPath, String sheetName)

03         throws Throwable  
04         FileInputStream fis = new FileInputStream(excelPath);  
05         Workbook wb = WorkbookFactory.create(fis);  
06         int rowCount = wb.getSheet(sheetName).getLastRowNum();  
07         return rowCount;

05 }  
06     public void writeExcelData(String excelPath, String sheetName,  
                      int row, int cell, String data)  
07         throws Throwable

FileInputStream fis = new FileInputStream(excelPath);  
Workbook wb = WorkbookFactory.create(fis);  
wb.getSheet(sheetName).getRow(row).createCell(cell).  
setCellValue(data);

12 Sunday

FileOutputStream fos = new FileOutputStream(excelPath);  
wb.write(fos);

{}

13

May  
Monday

Wk-20 • 133-232

2019

MAY '19	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

## 08 Data - Driven Main Class:

```
09 package datadrivenframework;
10 public class DataDrivenMainClass
11 {
12     public static void main(String[] args) throws
13         Throwable
14     {
15         FileLib fib = new FileLib();
16         int rc = fib.getRowCount("./data/inputData.xlsx",
17             "Sheet1");
18         System.out.println(rc);
19         for (int i=0; i<rc; i++)
20         {
21             String value = fib.readExcelData("./data/inputData.xlsx",
22                 "Sheet1", i, 1);
23             System.out.println(value);
24             fib.writeExcelData("./data/inputData.xlsx", "Sheet1",
25                 i, 2, "Raghav Sir");
26         }
27     }
28 }
```

						1
30	3	4	5	6	7	8
2	9	10	11	12	13	14
16	17	18	19	20	21	22
23	24	25	26	27	28	29
S	M	T	W	T	F	S

07.10.2021

D27

May

Tuesday

2019

14

Wk-20 • 134-231

## KEYWORDDRIVEN FRAMEWORK

08

Storing the test data which is commonly or frequently used, as key-value pair and reading it by using generic reusable method is known as KeywordDriven Framework.

11

Here we store test data which we use very frequently like browserName, url, valid user-name and password to login etc, in every test case. As Key-value pair in property file and we use a java class called Properties class and its method getProperty to get the data.

02

getProperty() takes key as argument and returns its corresponding value.

04

Creating property file in our project.

1. Right click on the folder in which we want to create it ..> New --> File (name it as 'Config' and give extension as .property or .properties)
2. Store test data in key-value pair as below.

Key	Value
browser	chrome
url	<a href="https://www.actitime.com">https://www.actitime.com</a>
username	admin
password	manager

15

May

Wednesday

Wk-20 • 135-230

2019

MAY '19	5	6	7	8	9	10	11
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	
	S	M	T	W	T	F	S

## Property File: ( config.properties)

08

Trainer Raghav Sir  
 browser chrome  
 url <https://demo.actitime.com/login.do>  
 username admin  
 password manager

11

12

## IAutoConsts: Interface

```

01 package hybrid2.datadriven.keywordDriven;
02 public interface IAutoConsts
03 String PROP_PATH = ". /data/Config.properties";
04 String EXCEL_PATH = ". /data/inputData.xlsx";
05 String CHROME_KEY = "webdriver.chrome.driver";
06 String CHROME_VALUE = ". /drivers/chromedriver.exe";
07 String GECKO_KEY = "webdriver.gecko.driver";
08 String GECKO_VALUE = ". /drivers/geckodriver.exe";
09 }
  
```

JUNE '19
30
2
9
16
23
S
M
T
W
T
F
S

May

Thursday

2019

16

Wk-20 • 136-229

## Read Data From Property File

08

```
package keywordDrivenFramework;
```

```
import java.io.FileInputStream;
```

```
import java.util.Properties;
```

10

```
public class ReadDataFromPropertyFile
```

11

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

12

```
// Open file in read mode
```

```
01 FileInputStream fis = new FileInputStream(".\data\\Config.properties");
```

02

```
// Create an Object of Properties class
```

```
03 Properties prop = new Properties();
```

04 // Keep the file ready for reading purpose

```
prop.load(fis);
```

05

06 // Give the key to get its value

```
String value = prop.getProperty("browser");
```

07 (OR) String value = prop.getProperty("browser", "Incorrect  
key");

08 // Print it

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

09 }

10 }

17

May  
Friday

Wk-20 • 137-228

2019

MAY 19	5	6	7	1	2	3	4
	12	13	14	15	16	17	18
	19	20	21	22	23	24	25
	26	27	28	29	30	31	
S	M	T	W	T	F	S	

## FileLib

08

package keywordDrivenFramework;

import java.io.FileInputStream;

import java.util.Properties;

10

public class FileLib

11 }

public String readPropertyData(String propPath,  
String key) throws Throwable{  
FileInputStream fis = new FileInputStream(propPath);  
Properties prop = new Properties();  
prop.load(fis);  
String propValue = prop.getProperty(key, "Incorrect key");  
return propValue;  
}

04

## KeywordMainClass

05

package keywordDrivenFramework;

public class KeywordMainClass

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

FileLib lib = new FileLib();

String br = lib.readPropertyData("./data/config.properties",  
"Username");  
System.out.println(br);

}

}

JUNE '19	30	1
	2	3
	4	5
	6	7
	8	9
	10	11
	12	13
	14	15
	16	17
	18	19
	20	21
	22	23
	24	25
	26	27
	28	29
S	M	T
F	S	S

May

Saturday

18

2019

Wk-20 • 138-227

## (HYBRID DRIVEN)

## BaseTest

```

08 package hybrid_datadriven_keywordDriven;
09 import org.openqa.selenium.WebDriver;
09 import org.openqa.selenium.chrome.ChromeDriver;
10 import org.openqa.selenium.firefox.FirefoxDriver;
11
12 public class BaseTest implements IAutoConsts
13 {
14     public static WebDriver driver;
15     public void openBrowser() throws Throwable
16     {
17         FileLib flib = new FileLib();
18         String browserValue = flib.readPropertyData(PROP_PATH,
19                                         "browser");
20         if (browserValue.equalsIgnoreCase("chrome"))
21         {
22             System.setProperty(CHROME_KEY, CHROME_VALUE);
23             driver = new ChromeDriver();
24         }
25         else if (browserValue.equalsIgnoreCase("firefox"))
26         {
27             System.setProperty(GECKO_KEY, GECKO_VALUE);
28             driver = new FirefoxDriver();
29         }
30         else
31         {
32             System.out.println("Enter proper Browser name");
33         }
34         driver.manage().window().maximize();
35         String appUri = flib.readPropertyData(PROP_PATH, "url");
36         driver.get(appUri);
37     }

```

20

May

Monday

Wk-21 • 140-225

2019

MAY '19	5	6	7	1	2	3
	12	13	14	15	16	17
	19	20	21	22	23	24
	26	27	28	29	30	31
S	M	T	W	T	F	S

public void closeBrowser()

08}

driver.quit();

09}

## 10 FileLib (Add-ons)

11 package hybrid.datadriven.KeywordDriven;

12 import java.io. FileInputStream;

import java.io. FileOutputStream;

01 import java.util. Properties;

import org.apache.poi.ss.usermodel. Workbook;

02 import org.apache.poi.ss.usermodel. WorkbookFactory;

03 public class FileLib

04 public String readExcelData( String excelPath, String sheetName, int row, int cell) throws Throwable  
 FileInputStream fis = new FileInputStream(excelPath);  
 Workbook wb = WorkbookFactory.create(fis);  
 String excelValue = wb.getSheet(sheetName).getRow(row).getCell(cell).getStringCellValue();  
 return excelValue;

05 }  
 public int getRowCount( String excelPath, String sheetName) throws Throwable

FileInputStream fis = new FileInputStream(excelPath);  
 Workbook wb = WorkbookFactory.create(fis);  
 int rowCount = wb.getSheet(sheetName).getLastRowNum();  
 return rowCount;

}

JUNE 19	30						1
	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	S	M	T	W	T	F	S

May  
Tuesday

2019

21

2019

2019

2019

Wk-21 • 141-224

```
08 public void writeExcelData(String excelPath, String
09 sheetName, int row, int cell, String data) throws
10           {
11     FileInputStream fis = new FileInputStream(excelPath);
12     Workbook wb = WorkbookFactory.create(fis);
13     wb.getSheet(sheetName).getRow(row).createCell(cell);
14     wb.setCellValue(data);
15   }
16
17   FileOutputStream fos = new FileOutputStream(excelPath);
18   wb.write(fos);
19 }
```

02 Valid Login Test

```
03 package hybrid.datadriven.keywordDriven;
04 public class ValidLoginTest extends BaseTest
05 {
06   public static void main(String[] args) throws Throwable
07   {
08     BaseTest bt = new BaseTest();
09     bt.openBrowser();
10   }
11 }
```

08.10.202

D28

## WebDriverCommonLib

```
import org.openqa.selenium.support.ui.ExpectedConditions;  
import org.openqa.selenium.support.ui.WebDriverWait;
```

22

May  
Wednesday

Wk-21 • 142-223

2019

MAY '19	5	6	7	1	2	3
	12	13	14	8	9	10
	19	20	21	15	16	17
	26	27	28	22	23	24
S	M	T	W	30	31	F

```

public class WebDriverCommonLib
08 {
    public String getPageTitle()
09 {
10     String pageTitle = BaseTest.driver.getTitle();
11     return pageTitle;
12 }
13 public void waitForTitle(String title)
14 {
15     WebDriverWait wait = new WebDriverWait(BaseTest,
16                                         driver, 20);
17     wait.until(ExpectedConditions.titleContains(title));
18 }

public void verify(String expected, String actual,
03                                     String page)
04 {
05     if (actual.equals(expected))
06         System.out.println(page + " is Displayed, PASS");
07     else
08         System.out.println(page + " is not Displayed, FAIL");
09 }
10 }
```

Config. properties file (Add-ons)

loginTitle actiTIME - Login

homeTitle actiTIME - Enter Time-Track

						1
30	3	4	5	6	7	8
2	9	10	11	12	13	14
16	17	18	19	20	21	22
23	24	25	26	27	28	29
S	M	T	W	T	F	S

JUNE '19

May

Thursday

2019

23

Wk-21 • 143-222

## Base Test - Add ons before closing Browser

08

```

09 WebDriverCommonLib wlib = new WebDriverCommonLib();
10 wlib.verify(flib.readPropertyData(PROP_PATH, "loginTitle"),
11 wlib.getPageTitle(), "Login");
}

```

## File Lib - (Add ons)

```

12 String excelValue = wb.getSheet(sheetName).getRow(row).
13    getCell(cell).toString();
}

```

## Invalid Login Test:

```

01 package hybrid-dataDriven-keywordDriven;
02 import org.openqa.selenium.By;
03 public class InvalidLogintest extends BaseTest
{
04     public static void main(String[] args) throws Throwable
{
05     BaseTest bt=new BaseTest();
06     bt.openBrowser();
07     FileLib fib= new FileLib();
08     int rc=fib.getRowCount(EXCEL_PATH, "Invalid");
09     for(int i=1; i<=rc; i++)
{
10         String un=fib.readExcelData(EXCEL_PATH, "Invalid", i, 0);
11         String pw=fib.readExcelData(EXCEL_PATH, "Invalid", i, 1);
12         driver.findElement(By.id("username")).sendKeys(un);
13         Thread.sleep(2000);
14         driver.findElement(By.name("pwd")).sendKeys(pw);
15         driver.findElement(By.xpath("//div[text()='Login']"));
}
}

```

24

May

Friday

Wk-21 • 144-221

2019

12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

08      click();  
      Thread.sleep(6000);

10      input Data (EXCEL)

	Username	Password
11	ad123	manager
12	ad#%\$%	manager
13		manager
14	ad123#%\$%	manager
15	12345	manager
16	ad min	manager

03

11.10.2021

D29

04      StateElementReferenceException

Q: What is it?

05      The reference or address of the element is old.

06 Q: Why do we get this exception?

When we use findElement() / findElements() to get the address of an element, store it in a reference variable, but before performing action, if the webpage is refreshed, then on the webpage element's address will change, but in reference variable we still have old address, and we will be trying to perform action on the old address only.

Hence we get this exception.

	30		1			
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
S	M	T	W	T	F	S

May  
Saturday

2019

25

Wk-21 • 145-220

Q: How to handle it?

By Using Page Object Model (POM)

Q: What is Page Object Model?

Page Object Model is one of the Java Design Patterns.

Mainly it's an Object Repository which is used to store all the webElements in one place.

Hence it is also called Element Repository / Page Object Repository.

Q: How does POM handle StaleElementReferenceException?

In page object model, we use @FindBy to get the address of the elements. @FindBy will always get the new address / current address / latest address of the elements therefore address will never become old. Hence we can avoid getting this exception.

Q: How do we declare elements in POM class?

POM class means any class which contains @FindBy is known as POM class...

In this class we declare by using @FindBy.

~~So:~~ @FindBy(locatorName = "locatorValue") AccessSpecifier  
ReturnType ElementName;

Eg: @FindBy(id = "username") private WebElement unTB;

26 Sunday

Q: How do we initialize all the elements in POM Class?

We use initElements() of PageFactory class to initialize all the elements at once. initElements()

27

May  
Monday

Wk-22 • 147-218

2019

MAY '19	5	6	7	8	9	10	11
S	M	T	W	T	F	S	E

gives instructions to all the elements (@FindBy) stored to fetch the address and store it inside reference variable.

initElements() takes two Arguments

1. WebDriver reference variable - means browser class object reference.

Eg: driver

2. Object Page - means in which class we have stored the elements. If the elements are in the same class/ current class only then we use 'this'.

PageFactory.initElements(driver, this);

If the elements are in some other class, Just create an object of that class and pass reference variable as argument

PomClass pc = new PomClass();

PageFactory.initElements(driver, pc);

Q: Is user defined constructor compulsory to initialize elements?

No, but initializing the elements with initElements is mandatory. If not initialized then we get NullPointerException.

Q: How to handle multiple Elements in POM Class?

Use the return type as List<WebElement>

JUNE '19	30	1
	2	3
	4	5
	6	7
	9	10
	11	12
	13	14
	16	17
	18	19
	20	21
	23	24
	25	26
	27	28
	29	S
	T	F
	W	S

May

Tuesday

2019

28

Wk-22 • 148-217

**Q: How do we do utilization in POM class?**

By creating public getter and Setter methods.

Address returning method

```
public WebElement getUntb()
```

```
{
```

```
return untb;
```

```
}
```

Action Performing method

```
public void setUsername(String un)
```

```
{
```

```
untb.sendKeys(un)
```

```
}
```

02

## Uses of Page Object Model

- 03 1. It is used to store all the elements in one place. Hence it acts as Element Repository / Object Repository.
- 04 2. It can avoid StaleElementReferenceException
- 05 3. Code Reusability is achieved, which increases the efficiency of test script development.
- 06 4. Maintaining automation scripts is easy (Especially when requirements are frequently changing)
- 07 5. Helps in encapsulating the elements.
- 08 6. Can achieve abstraction
- 09 7. Can achieve method Driven testing

## METHOD DRIVEN FRAMEWORK

Writing the implementation of the methods in different classes like generic classes or POM class and calling the methods in test class and executing the scripts by the help of these methods.

29

May

Wednesday

Wk-22 • 149-216

2019

MAY '19	5	6	7	1	2	3	4
	12	13	14	8	9	10	11
	19	20	21	15	16	17	18
	26	27	28	22	23	24	25
S	M	T	W	T	F	S	S

is known as Method Driven Framework.

08

Login Page :

09

```

01 package Sere;
02 import org.openqa.selenium.WebElement;
03 import org.openqa.selenium.support.FindBy;
04 import org.openqa.selenium.support.PageFactory;
05 import hybrid-dataDriven-keywordDriven.BaseTest;
06
07 public class LoginPage
08 {
09     // Declaration
10     @FindBy(id = "username") private WebElement Untb;
11
12     // Initialization
13     public LoginPage
14     {
15         PageFactory.initElements(BaseTest.driver, this);
16     }
17
18     // Utilization
19     public WebElement getUntb()
20     {
21         return Untb;
22     }
23
24     public void typeUn(String un)
25     {
26         Untb.sendKeys(un);
27     }
28 }
```

JUNE 19
30
2
9
16
23
S

3	4	5	6	7	1
10	11	12	13	14	15
17	18	19	20	21	22
24	25	26	27	28	29
M	T	W	T	F	S

May  
Thursday

2019

30

Wk-22 • 150-215

## Handling StaleElementReferenceException

08

```
import org.openqa.selenium.By;
```

09

```
public class HandlingStaleElementReferenceException
```

```
extends BaseTest
```

10

```
11 public static void main(String[] args) throws Throwable
```

12

```
BaseTest bt = new BaseTest();
bt.openBrowser();
```

01

```
LoginPage lp = new LoginPage();
```

```
driver.navigate().refresh();
lp.typeUn("admin");
```

02

03

Login Page - (Add ons)

```
04
05 @FindBy(name="pwd") private WebElement pwtb;
@FindBy(xpath = "//div[text()='Login ']") private
WebElement loginBtn;
```

06

```
07 public LoginPage()
```

```
{ PageFactory.initElements(BaseTest.driver, this); }
```

```
08 public WebElement getPwtb()
```

```
{ return pwtb; }
```

```
09 public void setPwtb(String pwd)
```

```
{ pwtb.sendKeys(pwd); }
```

31

May  
Friday

Wk-22 • 151-214 2019

MAY '19	5	6	7	8	9	10	11
S	12	13	14	15	16	17	18
M	19	20	21	22	23	24	25
T	26	27	28	29	30	31	
W							
F							
S							

```

08 } public WebElement getLoginBtn()
09 {
10     return loginBtn;
11 }
12 public void clickLoginBtn()
13 {
14     loginBtn.click();
15 }
16 public WebElement getUntb()
17 {
18     return untb;
19 }
20 public void setUntb(String un)
21 {
22     untb.sendKeys(un);
23 }
24 public void login(String un, String pw)
25 {
26     untb.sendKeys(un);
27     pwtb.sendKeys(pw);
28     loginBtn.click();
29 }
30

```

Valid Login Test.

package methodDrivenFramework;

public class ValidLoginTest

{ public static void main(String[] args) throws Throwable  
//open Browser, enter URL, Verify login page

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June

## Saturday

01

2019

Wk-22 • 152-213

```
08     BaseTest bt = new BaseTest();  
09     bt.openBrowser();  
10     // Login into Application and verify HomePage  
11     LoginPage lp = new LoginPage();  
12     lp.login("admin", "manager");  
13 }
```

12.10.2021

D30

## 01 Unit Testing Tools :

Java	-	JUnit
.NET	-	NUnit

03 Java + .NET - TestNG (TestNextGeneration)

## PHP - PHPUnit

# 04 Python - PyDev

# Ruby - Rspec

05

## TestNG :

- It is a unit testing tool which supports Java and .NET.
  - It is a third party tool developed by Cedric Beust which is the next version of JUnit.
  - Used by Developers in WBT and Automation Test Engineers in BBT which makes automation test engineers life easy.

## Uses:

1. Obtain the output in the form of emailable HTML reports.

# 03

June  
Monday

Wk-23 • 154-211

2019

JUNE '19	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	S	M	T	W	T	F	S
----------	----	----	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---	---	---	---	---	---	---

2. Perform Batch execution.

08. (Running multiple Scripts with Single Click by Converting all the test Scripts into a test Suite (Collection of multiple Test Scripts → Test Suite)).

09. Provides lot of Annotations and flag which are very helpful in test Scripts development.

10. Perform Grouping Execution

11. Perform Data Parameterization

12. Achieve data-driven Concept

13. Perform Parallel Execution

14. Perform cross Browser testing

15. Provides better Assertion features

16. Provides many flags and test methods dependency features.

17. Provides listeners feature which can capture Screenshots dynamically.

## Download and Install TestNG:

Since TestNG is an eclipse plug-in

06. In Eclipse → Help → Install New Software → Click on 'Add' button.

07. Provide name as TestNG and location as <https://testing.org/testing-eclipse-update-site>

→ Select testNG and click on Install, follow instructions till end..

Restart the eclipse..

Adding TestNG to our project

1. Right click on the project → Go to BuildPath → Add Libraries → Select TestNG → Click on

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
5	M	T	W	T	F	S

June  
Tuesday

2019

Wk-23 • 155-210

04

## Next and Finish button

### Rules of TestNG:

01. Never use "Default Package"  
→ Always create a class under some package
02. Never use main()  
→ Use @Test
03. Never use System.out.println() for printing  
→ Use Reporter.log()

There are two types of reports

01. High-Level Reports : Overview Report which gives Status of the testscripts, Contains no. of tests passed, failed, skipped .. etc.
02. Low-level Reports : Detailed Report which is given by the programar.

Reporter.log("abc") → abc will be printed in report only.

Reporter.log("abc", true) → abc will be printed in both report as well as Console.

### Test Class :

Any class which contains @Test is called Test Class  
(Class Demo)

Q: How to see the TestNG Report?

1. Run the above Test Class
2. Refresh the Project (Right click on Java Project → refresh).

# 05

June  
Wednesday

Wk-23 • 156-209

2019

JUNE '19	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	F	S

09

3. New folder called test-output will appear
4. Expand it → Go to emailable-report.html  
→ Right click → Open with → Web Browser

Batch Execution: Means to execute multiple Test Classes with single click by Converting all the required test classes into Test Suite.

Q: How to generate Test Suite?

Right Click on the project →

01 TestNG →

Converting to TestNG →

02 Finish.

A .xml file is generated as below which is nothing but the test suite / Batch Executor.

```
04 <suite name="Suite">
    05   <test name="Test">
        06     <classes>
            07       <class name="testNG.Sumo"/>
            08       <class name="testNG.Demo"/>
            09       <class name="testNG.Remo"/>
        10     </classes>
    11   </test>
12 </suite>
```

To execute the test suite, Right click on testNG.xml → Run as → TestNG suite.

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June  
Thursday

2019

06

Wk-23 • 157-208

## Flags of TestNG.

1. priority - Used to control the order of execution of test methods within a class.
2. enabled - Controls the execution of test methods, will not execute if enabled = false.
3. invocationCount=3 - means test method will loop itself and run for 3 times.
4. dependsOnMethods - adds dependency feature, wherein a test method executes only if its dependant method is PASS.
5. parameters - Used to pick a specific parameter.
6. groups - Used in grouping required test methods and executing them.
7. name - Used for identification.
8. dataprovider - Used in data driven concept of TestNG.
9. description - Used to give brief description about a test method.
10. alwaysRun - If true, then it will run always even if dependent methods are failed.

Q: Can we have multiple test methods in a test class?

YES

Q: What is the order of execution of test methods if multiple are present?

Alphabetical order of their method name.

# 07

June  
Friday

Wk-23 • 158-207

2019

JUNE '19	30	31	1
	2	3	4
	9	10	11
	16	17	18
	23	24	25
S	M	T	W
			T
			F
			S

Q: Can you execute a test method after another test method is executed?  
YES, Using priority flag.

Q: Can you execute a test method only after another test method is PASS?  
YES, by using dependsOnMethods flag.

dependsOnMethods Flag:

By using this flag we can add dependency feature, means One test method can be dependent on 1 or more methods and only if all the methods are executed and passed, then dependent method will execute.

Example: If there are 2 methods, transfer1() And transfer2() and one more method called delete() is dependent on transfer1() and transfer2() then delete will execute only if transfer1() and transfer2() gets executed and passed.

Either or both of the methods are fail, then delete() gets "Skipped" and note that if alwaysRun flag is present then it will ignore dependsOnMethods flag .. thereby even if transfer1() fails delete() will execute..

When dependsOnMethods flag is present then priority flag will not have importance,

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	T	F	S	

June  
Saturday

08

2019

Wk-23 • 159-206

we cannot disable register() because delete()  
is dependent on register(). So if register()  
won't run, then delete() also won't run.

Demo:

```
10 package testing;
import org.junit.Test;
11 import org.junit.runner.RunWith;
```

```
12 public class Demo
```

```
01 @Test
```

```
02 public void raghav1()
```

```
03 System.out.println("I am SOP message");
```

```
04 Reporter.log("I am report message");
```

```
05 Reporter.log("Report + Console", true);
```

Remo

```
06 public class Remo
```

```
07 @Test
```

```
08 public void raghav2()
```

```
09 Reporter.log("Raghav2", true);
```

09 Sunday

10

June  
Monday

Wk-24 • 161-204

2019

JUNE '19	30						
	2	3	4	5	6	7	1
	9	10	11	12	13	14	8
	16	17	18	19	20	21	15
	23	24	25	26	27	28	22
S	M	T	W	T	F	S	S

Sumo.

```
08 public class Sumo
09 {
10     @Test
11     public void raghav35()
```

```
Reporter.log("raghava", true);
```

## Flags of Testing:

```
package testing; // Main class  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.junit.runners.JUnit4;
```

```
public class FlagsOfTestNG
```

@Test(priority=1)

```
public void run()
```

```
Reporter.log("run method", true);
```

```
@Test (priority = 2, enabled = false)
```

public void bun()

```
    Reporter.log("bun method", true);
```

```
@Test (priority = 3, invocationCount = 10)  
public void guns()
```

public void gun()

```
Reporter.log("fun method", true);
```

1	2	3	4	5	6
7	8	9	10	11	12
14	15	16	17	18	19
21	22	23	24	25	26
28	29	30	31		
S	M	T	W	T	F

## Dependency Feature

June  
Tuesday

11

2019

Wk-24 • 162-203

```
08 package testing; // import org.junit.*
09 import org.junit.Test;
10 import org.junit.runner.RunWith;
11 import org.junit.runners.JUnit4;
12
13 @RunWith(JUnit4.class)
14 public class DependencyFeature {
15     @Test(priority = 1)
16     public void register() {
17         Reporter.log("Registered successfully", true);
18     }
19
20     @Test(priority = 2)
21     public void register2() {
22         Assert.fail();
23         Reporter.log("Registered successfully", true);
24     }
25
26     @Test(priority = 3, dependsOnMethods = {"register1",
27                                         "register2"})
28     public void delete() {
29         Reporter.log("Deleted successfully", true);
30     }
31 }
```

13.10.2021

D31

## ANNOTATIONS

@BeforeSuite : It executes before start of every test Suite , Connects to Database Program.

# 12

June  
Wednesday

Wk-24 • 163-202

2019

JUNE 19	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	

⑧ @BeforeTest : It executes before every test runner, Custom report - keep ready.

⑨ @ BeforeClass : It executes before every class, Open the browser, enter the URL.

⑩ @ BeforeMethod : It executes before every test method, Login to App program.

⑪ @ Test : This annotation is used for execution purpose, we write Test Cases.. CreateCustomerTest, EditCustomerTest.

⑫ @ AfterMethod : It executes after every test method, logout program.

⑬ @ AfterClass : It executes after every class, Close Browser program

⑭ @ AfterTest : It executes after every testRunner, Save the report program.

⑮ @ AfterSuite : It executes after the end of every test suite, Disconnect from database.

@BS - connectDB()

@BT - reportInitialization

@BC - openBrowser()

@BM - login()

@Test - CreateCustomer()

@AM - logout()

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June  
Thursday

2019

13

Wk-24 • 164-201

- 08 @ BM - login()
- 08 @ Test - editCustomer()
- 08 @ AM - logout()
- 09 @ BM - login()
- 09 @ Test - deleteCustomer()
- 10 @ AM - logout()
- 11 @ AC - closeBrowser()
- 11 @ BC - openBrowser()
- 12 @ BM - login()
- 12 @ Test - CreateProject()
- 01 @ AM - logout()
- 01 @ AC - closeBrowser()
- 02 @ AT - SaveReports
- 02 @ AS - DisconnectDB()

### 03 AnnotationsOfTesting

```
package testing;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;
import org.junit.runners.MethodSorters;
import org.junit.runners.Parameterized.Parameters;
import org.junit.runners.Parameterized.Parameters;
import org.junit.runners.Parameterized.Parameters;
import org.junit.runners.Parameterized.Parameters;
import org.junit.runners.Parameterized.Parameters;
import org.junit.runners.Parameterized.Parameters;
```

```
public abstract class AnnotationsOfTesting
```

```
@BeforeSuite
```

```
public void ac()
```

14

June  
Friday

Wk-24 • 165-200

2019

JUNE '19	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
S	S	M	T	W	T	F	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S			

08 } Reporter.log ("BeforeSuite", true);

09 @AfterSuite

10 public void b()

11 } Reporter.log ("Aftersuite", true);

12 @BeforeTest

13 public void c()

14 } Reporter.log ("BeforeTest", true);

15 @AfterTest

16 public void d()

17 } Reporter.log ("AfterTest", true);

18 @BeforeClass

19 public void e()

20 }

21 } Reporter.log ("BeforeClass", true);

22 @AfterClass

23 public void f()

24 }

25 } Reporter.log ("AfterClass", true);

26 @BeforeMethod

27 public void g()

28 }

29 } Reporter.log ("BeforeMethod", true);

30 }

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June  
Saturday

2019

15

Wk-24 • 166-199

• @AfterMethod

08      public void h()

09      Reporter.log("AfterMethod", true);

10  
AnnotationsMainClass1

11 package testing;  
import org.junit.Test;

12 public class AnnotationMainClass1 extends AnnotationsOfTestNG

}

02      @Test

03      public void runAnn1()

04      Reporter.log("I am runAnn test method1", true);

05      @Test

06      public void runAnn2()

07      Reporter.log("I am runAnn test method2", true);

AnnotationsMainMethodClass2

package testing;

import org.junit.Test;

16 Sunday

public class AnnotationMainClass2 extends AnnotationsOfTestNG

}

@Test

17

June  
Monday

Wk-25 • 168-197

2019

JUNE '19	30	31	4	5	6	7	8
	2	3	10	11	12	13	14
	9		17	18	19	20	21
	16		24	25	26	27	28
	23		30	31	1	2	3
	S	M	T	W	T	F	S

```

08 } public void runAnn1() {
09 }     Reporter.log("I am runAnn test method3");
10 }
  
```

Suite :

```

11 <Suite name = "Suite">
12   <test name = "Test">
13     <classes>
14       <class name = "testing.AnnotationMainClass1">
15         </classes>
16       </test>
17     </suite>
  
```

Suite :

```

18 <Suite name = "Suite">
19   <test name = "Test">
20     <classes>
21       <class name = "testing.AnnotationMainClass2">
22       <class name = "testing.AnnotationMainClass1">
23     </classes>
24   </test>
25 </suite>
  
```

Suite :

```

26 <Suite name = "Suite">
27   <test name = "TestRunner1">
28     <classes>
29       <class name = "testing.AnnotationMainClass1">
30     </classes>
31   </test>
  
```

JUN	1	2	3	4	5	6
	8	9	10	11	12	13
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30	31		27
S	M	T	W	T	F	S

June  
Tuesday

2019

18

Wk-25 • 169-196

1. <test name = "TestRunner2">

2. <classes> at line 10 of TestNG.xml

3. <class name = "testing.AnnotationMainClass2"/>

4. </classes> at line 11 of TestNG.xml

5. </test> at line 12 of TestNG.xml

6. </suite> at line 13 of TestNG.xml

14.10.2021

D32

## Assertion in TestNG:

Assertion means verification, it means verifying the actual result is same as Expected Result.

There are two types of Assertion in TestNG.

1. Assert class (HardAssert)

2. SoftAssert class (Verify)

Difference between Assert and SoftAssert (Verify)

Assert: Change - (with fail) and no exception

1. We use Assert to verify mandatory or critical features.

2. In Assert, if verification fails then execution stops of that test method, and we get Assertion Error.

3. All the methods are static, so no need to create any object.

Assert.assertEquals(actual, expected);

4. Here we don't use assertAll().

19

June  
Wednesday

Wk-25 • 170-195

2019

JUNE '19	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S

## SoftAssert (verify)

1. We use SoftAssert to Verify non-mandatory or non-critical features.
2. In SoftAssert, even if verification fails the execution continues and at the end we get Assertion error.
3. All the methods are non-static, so we need to create object.

SoftAssert sa = new SoftAssert();

sa.assertEqual(actual, expected);

4. Here assertAll() is Compulsory, and use as the last statement of the test method.

## Methods of Assert:

1. Assert.assertEqual(actual, expected) - Compares actual with expected, and stops the execution if not matching.
2. Assert.fail() - Used to intentionally fail a method for testing purpose.
3. Assert.assertTrue(condition) - Expects condition to become true, stops execution if false.
4. Assert.assertFalse(condition) - Expects condition to become false, stops execution if true.

## Group Execution:

planning our execution in such a way that, we group the required test methods and execute the group to run those particular test scripts that we need.

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June

Thursday

2019

Wk-25 • 171-194

20

We achieve grouping by two things

1. By groups flag:

We use groups flag to add test method to a particular group by giving groupName

`@Test(groups = "smokeTests")`

2. In XML file, we use tags like

`<groups>` - Used for executing groups

`<define>` - Used for define all the groupNames

`< run>` - Used for executing particular group included in it.

`< include>` - Used to include a group for execution purpose.

`< exclude>` - Used to exclude a group so that we don't want it to execute.

## DATA-DRIVEN CONCEPT in Testing

process of storing the data in data banks/ data Repositories and fetching them into the test Scripts is Data Driven Concept in Testing.

To achieve this, we use two things

1. `@DataProvider` (DataProvider Annotation)

2. `dataProvider` flag

1. `@DataProvider` (DataProvider Annotation)

This annotation is used to create a data repository and supply it to test method.

Here we create an object of 2 Dimension Array And Store data inside it.

21

June

Friday

Wk-25 • 172-193

2019

30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	M	T	W	T

Object[][] Obj = new Object[2][3]

08      obj[0][0] = "admin1";  
 09      obj[0][1] = "manager1";  
 09      obj[0][2] = "Goa";

10      obj[1][0] = "admin2";  
 11      obj[1][1] = "manager2";  
 11      obj[1][2] = "vizag";

12      2 --> 2 rows also means 2 set of data present.

01      3 --> 3 Columns also means no. of data present in each set.

03      Here 2 arrays are present, one for rows and one for columns.

- Rows represent how many set of data
- Columns represent how many parameters we want in test method.
- We need to give return type of databank method as Object[][]

07      @DataProvider (name = "Set2")  
 public Object[][] dataBank2()

2. Data Provider flag:

This flag is used to identify a particular dataBank.

Here we can directly use databank method name / @Test (dataProvider = "dataBank2")

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June

Saturday

2019

Wk-25 • 173-192

22

We can use name flag's value of DataBase method / @Test (dataProvider = "set2")

we need to parameterize the test method to catch the data from DataBase and pass it to the test method.

```
11 public void getCredentials(String un, String pwd,
                           String city)
```

Assertion in TestNG:

```
01 package testing;
```

```
02 import org.testng.Reporter;
```

```
03 import org.testng.annotations.Test;
```

```
04 import org.testng.asserts.SoftAssert;
```

```
05 public class AssertionInTestNG {
```

```
06     @Test
```

```
07     public void verifyLogin() {
```

```
08         System.setProperty("webdriver.chrome.driver",
                           ".\Drivers\chromedriver.exe");
```

```
09         WebDriver driver = new ChromeDriver();
```

```
10         driver.get("https://demo.actitime.com/login.do");
```

```
11         Assert.assertEquals(driver.getTitle(), "actiTIME-Logi");
```

```
12         Reporter.log("Login page is Displayed, PASS", true);
```

```
13         SoftAssert sa = new SoftAssert();
```

```
14         sa.assertEquals(driver.getTitle(), "actiTIME-Logi");
```

24

June  
Monday

Wk-26 • 175-190

2019

JUNE '19	30								
	2	3	4	5	6	7	8	9	10
	9	10	11	12	13	14	15	16	17
	16	17	18	19	20	21	22	23	24
	23	24	25	26	27	28	29	30	S
	S	M	T	W	T	F	S		

```
driver.findElement(By.id("username")).sendKeys("admin");
```

08 Sa.assertAll();

۹

09

## Grouping Execution :

## 11 package testing;

12 import org. testing. Reporter;

```
import org.junit.Test;
```

## public class GroupingExecution

@Test (groups = "Smoke Gps") {

03 public void smoke()

14 } Reporter.log ("Smoke Test1", true);

05 , @Test (groups = "FTGrp")

{ public void ft1()

06      Reporter.log ("FT1", true);

07

```
@Test(groups = "ITGrp")
```

public void it1()

Reporter.log("IT1", true);

④ Test (groups = "Smoke Grp")

public void Smoke2()

```
} Reporter.log("Smoke Test2", true);
```

		1	2	3	4	5	6	
		7	8	9	10	11	12	13
		14	15	16	17	18	19	20
		21	22	23	24	25	26	27
		28	29	30	31			
	S	M	T	W	T	F	S	

June  
Tuesday

25

2019

Wk-26 • 176-189

@Test(groups = "RegrGrp")

public void reg1()

{

Reporter.log("Regr Test1", true);

}

@Test(groups = "FTGrp")

public void ft2()

{

Reporter.log("FT2", true);

}

@Test(groups = "ITGrp")

public void it2()

{

Reporter.log("IT2", true);

}

@Test(groups = "SmokeGrp")

public void Smoke3()

{

Reporter.log("Smoke Test3", true);

}

@Test(groups = "RegrGrp")

public void reg2()

{

Reporter.log("Regr Test2", true);

}

@Test(groups = "SmokeGrp")

public void smoke4()

{

Reporter.log("Smoke Test4", true);

}

26

June  
Wednesday

Wk-26 • 177-188

2019

JUNE '19	30	31	4	5	6	7	8
	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	S	M	T	W	T	F	S

Suite :

```

08 <suite name="Suite">
09   <test name="Test">
10     <groups>
11       <define name="allgroups">
12         <include name="SmokeGrp"> </include>
13         <include name="FTGGrp"> </include>
14         <include name="ITGGrp"> </include>
15         <include name="RegrGrp"> </include>
16       </define>
17     </run>
18   <include name="allgroups"> </include>
19   <exclude name="SmokeGrp"> </exclude>
20   </run>
21   </groups>
22   <classes>
23     <class name="testing.GroupingExecution"/>
24   </classes>
25 </test>
26 </suite>

```

06 Data driven in TestNG

```

07 package testing;
import org.testng.Reporter;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

```

```
{ public class DataDrivenInTestNG
```

```
@DataProvider
```

```
public Object[] database1()
```

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

June  
Thursday

2019

27

Wk-26 • 178-187

```

08 Object[][] Obj = new Object[2][2];
09     obj[0][0] = "admin1";
10     obj[0][1] = "manager1";
11
12 @DataProvider(name = "db2")
13 public Object[][] dataBank2()
14 {
15     Object[][] Obj = new Object[2][3];
16     Obj[0][0] = "admin1";
17     Obj[0][1] = "manager1";
18     Obj[0][2] = "Goa";
19
20     Obj[1][0] = "admin2";
21     Obj[1][1] = "manager2";
22     Obj[1][2] = "Hyd";
23
24     return Obj;
25 }
26
27 @Test(dataProvider = "db2")
28 public void getCredit(String un, String pw, String city)
29 {
30     Reporter.log(un + " --- " + pw + " --- " + city, true);
31 }
```

# 28

June  
Friday

Wk-26 • 179-186

2019

JUNE '19	30	31	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W	T	F	S	M	T	W		

15.10.2021

D33

## Data Parameterisation:

Fetching the data through the parameters of test method.

Reading Data from XML,

we use

1. <parameters> tag

2. @Parameters (Parameters annotation)

1. <parameters> tag :

It is used to store the data.

We use attributes called 'name' and 'value'

<parameter name="browser"> value = "chrome" </parameter>

04

@Parameters - Used to identify which parameter

value we require. This is also Method Level

Annotation just like @Test which should be written above the method.

07 @Parameters("Parameter Name") -

It will give parameter value which we can catch by parameters of test method.

@Parameters("browser")

@Test

```
public void getParameterValue(String browserName)  
Reporter.log(browserName, true)
```

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

JULY 19

June  
Saturday

2019

29

Wk-26 • 180-185

Note: Always run the .xml file because data is coming from it  
 write XML + Program

## Parallel Execution:

Planning our execution so as run the Scripts parallelly.

### Uses of Parallel Execution:

- To execute test scripts within less time, hence reducing execution time.
- To obtain the test execution results faster.

Two types:

- Distributed Parallel Execution
- Cross Browser Parallel Execution

Distributed Parallel Execution: In this type, we distribute the test scripts across multiple threads and run all the threads parallelly.

### Cross Browser Parallel Execution:

Cross browser means testing application across different browsers.

Cross Browser parallel Execution means executing the same set of test scripts in different browsers parallelly.

### Q: How to run parallelly?

By using the attribute called 'parallel' in <Suite> tag.

30 Sunday

01

July  
Monday

Wk-27 • 182-183

2019

	1	2	3	4	5	6	7
JULY '19	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31			
S	M	T	W	T	F	S	S

To run test methods parallelly.

&lt;Suite name = "Suite" parallel = "methods"&gt;

To run test classes parallelly

&lt;Suite name = "Suite" parallel = "classes"&gt;

To run test runners parallelly

&lt;Suite name = "Suite" parallel = "tests"&gt;

To Stop running parallelly / to run Sequentially

&lt;Suite name = "Suite" parallel = "none"&gt;

Note: Maximum thread limit in testing is '5'  
Because if we try to use more than 5 threads,  
execution performance / speed becomes slow.

Reading Data from XML

package testing;

import org.testng.Reporter;

import org.testng.annotations.Parameters;

import org.testng.annotations.Test;

public class ReadingDataFromXML

@Parameters({ "browser", "username" })  
@Testpublic void ~~get~~ getDataFromXML(String br, String un){  
Reporter.log(br + " == " + run, true);  
}

2	3	4	5	6
9	10	11	12	13
16	17	18	19	20
23	24	25	26	27
30	31			
T	W	T	F	S

AUGUST '19	4	5	6	7	1	2	3
	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
S	M	T	W	T	F	S	

July

Tuesday

2019

02

Wk-27 • 183-182

suite:

```

08 <suite name="Suite">
  <test name="Test">
    <parameter name="browser" value="chrome">
    </parameters>
    <parameter name="username" value="admin">
    </parameters>
  <classes>
    <class name="testing.ReadingDataFromXML"/>
  </classes>
  </test>
</suite>
```

02 Parallel methods

package testing;

```

03
import org.openqa.selenium.chrome.ChromeDriver;
04 import org.junit.Test;
```

```

05 public class ParallelMethods
```

```

06 @Test
public void runChrome1()
```

```

07 {
  System.setProperty("webdriver.chrome.driver",
    "./drivers/chromedriver.exe");
  new ChromeDriver();
}
```

```

08 @Test
```

```

public void runChrome2()
{
}
```

03

July

Wednesday

Wk27 • 184-181

2019

	1	2	3	4	5	6
S.	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30	31		27
S.	M	T	W	T	F	S

## @ Test

```
08 public void runChrome3() {
09     {
10         "Test" > suite
11     }
12     <test name="Test">
13     <classes>
14         <class name="testing.ParallelMethods"/>
15     </classes>
16     </test>
17 </suite>
18
19 Parallel Classes
20 package testing;
21 import org.openqa.selenium.Chrome, ChromeDriver;
22 import org.testng.annotations.Test;
23
24 public class Parallel {
25     @Test
26     public void runChrome1() {
27         System.setProperty("webdriver.chrome.driver",
28             "./drivers/chromedriver.exe");
29         new ChromeDriver();
30     }
31 }
```

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
S	M	T	W	T	F	S

July  
Thursday

04

2019

Wk-27 • 185-180

Just like previous program, test's

```

08   " "
public class Parallel2 {
09 }
    @Test
10 public void runChrome2() {
11 }
}

```

Just like,

```

01 public class Parallel3 {
02     = parallelTestParallel("Parallel3")

```

```

03 Suite<"first"> = following "suite" = simple suite
    <suite name="Suite" parallel="classes">
04     <test name="Test">
05         <classes>
06             <class name="testing.Parallel1"/>
07             <class name="testing.Parallel2"/>
08             <class name="testing.Parallel3"/>
09         </classes>
10     </test>
11 </suite>

```

Parallel Tests:

```

<suite name="Suite" parallel="tests">
    <test name="TestRunner1">
        <classes>
            <class name="testing.Parallel1"/>
        </classes>
    </test>

```

05

July  
Friday

Wk-27 • 186-179

2019

		1	2	3	4	5	6
7	8	9	10	11	12	13	14
14	15	16	17	18	19	20	21
21	22	23	24	25	26	27	28
28	29	30	31				5
S	M	T	W	T	F	S	S

08

```
<test name="TestRunner2">
<classes>
  <class name="testing.Parallel2"/>
</classes>
</test>
```

10

```
<test name="TestRunner3">
```

11

```
<classes>
  <class name="testing.Parallel3"/>
</classes>
</test>
```

```
</suite>
```

## 02 Suite: CrossBrowser Parallel Testing

```
<suite name="Suite" parallel="tests">
  <test name="ChromeTest">
    <parameter name="browser" value="chrome">
      </parameter>
    <classes>
      <class name="testing.CrossBrowserParallelTesting">
        </classes>
      </test>
```

```
</suite>
```

```
<test name="FirefoxTest">
  <parameter name="browser" value="firefox">
    </parameter>
  <classes>
    <class name="testing.CrossBrowserParallelTesting">
      </classes>
    </test>
  </suite>
```

			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
S	M	T	W	T	F	S

AUGUST '19

July

Saturday

2019

06

Wk-27 • 187-178

## Cross Browser Parallel Testing

```

08 package testing;
09 import org.openqa.selenium.firefox.FirefoxDriver;
10 import org.openqa.selenium.chrome.ChromeDriver;
11 import org.testng.annotations.Parameters;
12 import org.testng.annotations.Test;
13
14 public class CrossBrowserParallelTesting
15 {
16     @Parameters("browser")
17     @Test
18     public void openBrowser(String br)
19     {
20         WebDriver driver;
21         if(br.equalsIgnoreCase("chrome"))
22         {
23             System.setProperty("webdriver.chrome.driver",
24                 "./drivers/chromedriver.exe");
25             driver = new ChromeDriver();
26         }
27         else if(br.equalsIgnoreCase("firefox"))
28         {
29             System.setProperty("webdriver.gecko.driver",
30                 "./drivers/geckodriver.exe");
31             driver = new FirefoxDriver();
32         }
33         else
34         {
35             System.out.println("Enter proper Browser Name");
36         }
37     }
38 }

```

07 Sunday

08

July  
Monday

Wk-28 • 189-176

2019

18.10.2021

D34

	1	2	3	4	5	6
MON.	7	8	9	10	11	12
TUE.	14	15	16	17	18	19
WED.	21	22	23	24	25	26
THU.	28	29	30	31		
FRI.	S	M	T	W	T	F
SAT.						S

## Modular Framework

For each and every module in the application we create a separate package in the framework and we write all the test scripts related to that module inside that package.

Package Syntax: `com.actitime.domain.project.module`

`domain.ApplicationName.ModuleName`

### Modules

TimeTrack

Package names in our framework

`com.actitime.timeTrack`

Tasks

`com.actitime.tasks`

Reports

`com.actitime.reports`

Users

`com.actitime.users`

Settings

`com.actitime.settings`

List of classes in `com.actitime.genericLib`

`BaseTest.java` (copy from previous project)

`FileLib.java` (copy from previous project)

`JAutoConsts.java` (copy from previous project)

`WebDriverCommonLib.java` (copy from previous project)

Add-ons in `BaseTest`.

Add `@BeforeClass` annotation before `openBrowser()` and at the end of the method, add `@AfterClass` annotation for `closeBrowser()`.

					1	2	3
AUGUST '19							
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	31	
S	M	T	W	T	F	S	

July  
Tuesday

2019

09

2019

Wk-28 • 190-175

## Add-Ons for Config. properties file

08

## taskTitle actITIME - Task List

09

## Addons for WebDriver Common Lib class

10

```
package com.actitime.genericlib;
```

11

```
import java.io.File;
import java.io.IOException;
import org.openqa.selenium.OutputType;
// " " " (Custom) WebDriver TakesScreenshot;
// " " " " . WebElement;
// " " " " " . ExpectedConditions;
// " " " " " . Select;
// " " " " " . WebDriverWait;
import org.junit.Assert;
import org.junit.Reporter;
import com.google.common.io.Files;
```

\* Copy - methods like `getTitle()`, `waitForTitle()`, from previous project.

07 public void verify(String expected, String actual, String page)

Assert.assertEquals(actual, expected);

```
Reporter.log (page + " is Displayed, PASS", true);
```

```
public void selectOption(WebElement element, int index)
```

Select sel = new Select(element);

~~Sel. SelectByIndex(index);~~

JULY 19		1	2	3	4	5	6
		7	8	9	10	11	12
		14	15	16	17	18	19
		21	22	23	24	25	26
		28	29	30	31		
	S	M	T	W	T	F	S

08 public void selectOption(WebElement element, String  
value)

{

09 Select sel = new Select(element);

10 sel.selectByValue(value);

}

11 public void selectOption(String text, WebElement element)

12 Select sel = new Select(element);

13 sel.selectByVisibleText(text);

}

01 public void switchToFrame(int index)

2

3 BaseTest.driver.switchTo().frame(index);

4

02 public void switchToFrame(String value)

5

03 BaseTest.driver.switchTo().frame(value);

6

04 public void switchToFrame(WebElement element)

7

BaseTest.driver.switchTo().frame(element);

8

public void fullPageScreenshot(String path)

TakesScreenshot ts = (TakesScreenshot) BaseTest.driver;

File src = ts.getScreenshotAs(OutputType.FILE);

File dest = new File(path);

try

{ Files.copy(src, dest); } catch (IOException e) { e.printStackTrace(); }

Files.copy(src, dest);

}

AUGUST '19
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
S
M
T
W
T
F
S

July

Thursday

2019

11

Wk-28 • 192-173

```

01 catch ( IOException e )
02 {
03     e.printStackTrace();
04 }
05
06 public void getElementScreenshot ( WebElement element,
07                               String path )
08 {
09     File src = element.getScreenshotAs ( OutputType.FILE );
10
11     File dest = new File ( path );
12
13     try
14     {
15         Files.copy ( src, dest );
16     }
17     catch ( IOException e )
18     {
19         e.printStackTrace();
20     }
21
22     public void acceptAlert ()
23     {
24         BaseTest.driver.switchTo().alert().accept();
25     }
26
27     public void dismissAlert ()
28     {
29         BaseTest.driver.switchTo().alert().dismiss();
30     }
31
32 }

```

List of classes in com.actitime.pages.

EnterTimeTrackPage.java

LoginPage.java

12

July  
Friday

Wk-28 • 193-172

2019

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

## Login Page:

package com.actitime.pages;

```
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import com.actitime.GenericLib.BaseTest;
```

12 public class LoginPage {

```
01 @FindBy(id = "username") private WebElement untb;
02 @FindBy(name = "pwd") private WebElement pwtb;
03 @FindBy(xpath = "//div[text()='Login ']") private
    WebElement loginBtn;
```

04 public LoginPage()

05 PageFactory.initElements(BaseTest.driver, this);

06 public WebElement getPwtb()

07 return pwtb;

08 public void setPwtb(String pwd)

09 pwtb.sendKeys(pwd);

10 public WebElement getLoginBtn()

11 return loginBtn;

AUGUST 19	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
	S	M	T	W	T	F	S

July  
Saturday

13

2019

Wk-28 • 194-171

```

public void clickLoginBtn() {
    loginBtn.click();
}

public WebElement getUntb() {
    return untb;
}

public void setUntb(String un) {
    untb.sendKeys(un);
}

public void login(String un, String pw) {
    untb.sendKeys(un);
    pwtb.sendKeys(pw);
    loginBtn.click();
}

```

List of classes in com.actitime.TimeTrack

ValidLoginTest.java

```

package com.actitime.TimeTrack;
import org.testng.annotations.Test;
import com.actitime.genericLib.BaseTest;
import com.actitime.genericLib.FileLib;
import com.actitime.genericLib.WebDriverCommonLib;
import com.actitime.pages.LoginPage;

```

15

July  
Monday

Wk-29 • 196-169

2019

		1	2	3	4	5	6
JULY 19	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31			
	S	M	T	W	T	F	S

public class ValidLoginTest extends BaseTest

public void loginToApp() throws throwable

LoginPage lp = new LoginPage();

FileLib fib = new FileLib();

lp.login(fib.readPropertyData(Prop\_PATH, "username"),  
fib.readPropertyData(Prop\_PATH, "password"));

WebDriverCommonLib wlib = new WebDriverCommonLib();

wlib.waitForTitle(fib.readPropertyData(Prop\_PATH, "homeTitle"));

wlib.Verify(fib.readPropertyData(Prop\_PATH, "homeTitle"));

wlib.getPageTitle(), "Enter Time Track Page");

}

g

04 Enter Time Track Page:

05 package com.actitime.pages;

06 import com.actitime.genericLib.BaseTest;

07 public class EnterTimeTrackPage extends BaseTest

@FindBy(xpath = "//div[text()='Tasks']") private  
WebElement tasksTab;

public EnterTimeTrackPage(WebDriver driver) {

PageFactory.initElements(BaseTest.driver, this);

}

AUGUST '19
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
S
M
T
W
T
F
S

July

Tuesday

2019

16

Wk-29 • 197-168

```

08 public WebElement getTasksTab() {
09     return tasksTab;
10 }
11 public void clickTasksTab() {
12     tasksTab.click();
13 }

List of classes in Com.actitime.tasks
01 VerifyTasksListPageTest.java

02 Verify Tasks List Page Test :
03 package Com.actitime.tasks;
04
05 import Com.actitime.genericLib.WebDriverCommonlib;
06 import Com.actitime.pages.EnterTimeTrackPage;
07
08 public class VerifyTasksListPageTest extends BaseTest {
09
10     @Test
11     public void verifyTasks() throws Throwable {
12
13         // Login to actitime and verify homepage
14         ValidLoginTest vl = new ValidLoginTest();
15         vl.loginToApp();
16
17         // Click on Tasks tab
18         EnterTimeTrackPage et = new EnterTimeTrackPage();
19         et.clickTasksTab();
20
21     }
22
23 }

```

17

July  
Wednesday

Wk-29 • 198-167

2019

	1	2	3	4	5	6
19	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30	31		
	S	M	T	W	T	F

## // Verify Tasks Tab is Displayed

08

```
WebDriverCommonLib wlib = new WebDriverCommonLib();
FileLib flib = new FileLib();
wlib.verify(flib.readPropertyData(Prop.PATH, "taskTitle"));
wlib.getPageTitle(), "Task Page");
```

{  
}  
14

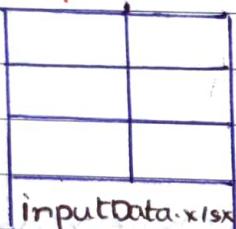
12

19.10.2021

D35

01

③ Data  
Repositories



① Generic  
Libraries

AutoConsts

② Object  
Repositories

LoginPage

02

Key - Value

Config.properties

FileLib

UserListPage

03

MyListeners

TaskPage

04

BaseTest

05

WebDriver  
CommonLib

④ Resources

Sel jars

Poi Jars

driver exe files

HYBRID FRAMEWORK ARCHITECTURE ↑ → ↘

			1	2	3
4	5	6	7	8	9 10
11	12	13	14	15	16 17
18	19	20	21	22	23 24
25	26	27	28	29	30 31
S	M	T	W	T	F S

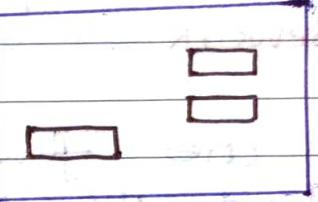
July  
Thursday

18

2019

Wk-29 • 199-166

## ⑤ Test Scripts ⑥ Batch Runner ⑦ Emailable HTML Report

Create CustTest	<suite>  <test>  <class> create edit delete </class> </test> </suite>	#Ran 3	#PASS 2	#FAIL 1	#SKIP 0
Edit CustTest	LoginPage Displayed → PASS HomePage not Displayed → FAIL				
Delete CustTest	<b>⑧ Screenshot</b>				
					

testing.xml

## HYBRID FRAMEWORK ARCHITECTURE ↑

### Rules of Writing Scripts in Framework:

1. Run the test case manually
2. Identify the elements required for the test
3. Script and Store it in respective pages.
4. Create getters and setters methods.
5. Use generic reusable methods in your Scripts and run them.

### Hybrid Framework Notes:

There are '8' Components in a framework.

1. Generic Library
2. Object Repository Library
3. Test Data

19

July  
Friday

Wk-29 • 200-165

2019

	1	2	3	4	5	6
JULY '19	7	8	9	10	11	12
	14	15	16	17	18	19
	21	22	23	24	25	26
	28	29	30	31		27
S	M	T	W	T	F	S

08 4. Resources

5. Test Scripts

09 6. Batch Runner

7. HTML Report

10 8. Screenshot

\*\*\*

Q: Explain your hybrid framework /

How to develop test scripts in our hybrid framework.

12

→ There are few things which will never change throughout the whole project. Such Constants I have stored inside JAutoConsts interface Eg: Path of excel file and property file, key and value of the driver executable files.

→ There are few things which are repeated like open browser, enter url, verify loginPage and closeBrowser etc. These repeated actions I am storing inside a Common class for all test scripts called as BaseTest, and I ensure that all the test scripts extends this BaseTest (Inheriting Common properties from same place)

→ While automating the application, we require huge set of test Data.. So we store it in a "data repositories" like Excel Sheets (since easy to maintain data) and we write generic reusable file handling libraries to handle data from them,

AUGUST '19	4	5	6	7	1	2	3
	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
	S	M	T	W	T	F	S

July  
Saturday

20

2019

Wk-29 • 201-164

- All the WebDriver related actions will be written in one place as generic reusable libraries once and reused in test scripts whenever required.
- All the elements on the webpage are stored in one place called Object repository (POM) wherein we declare, initialize and utilize the elements by writing public getters, setters and action methods.
- In test scripts, we just call generic methods, getters, setters, action methods and execute the automation Scripts through these methods. (Test Scripts become light-weight)
- Since we need to run all my scripts with one click we converted all the test scripts into test suite (Batch executor) and run the suite. (testing.xml file)
- Finally, we get HTML reports and Screenshots of failed test cases.

**Q:** Where have you used Java Concepts in your framework?

Inheritance:

Commonly repeated actions we have stored in BaseTest and we make sure every test class extends the BaseTest.

(start prints some standard text, like browser opening, start prints some standard text, like browser opening, start prints some standard text, like browser opening)

22

July  
Monday

Wk-30 • 203-162

2019

JULY '19		1	2	3	4	5
S	M	T	W	T	F	S
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

## Encapsulation And Abstraction:

We use Object repository to store the web-elements and we make the elements as private. So as to encapsulate them within the POM class also we provide public getters & setters to access them outside the class but cannot be modified. Also, here since we are hiding the functionality but just providing the public getters and setters, hence we use abstraction concept.

## Polymorphism:

While selecting the browser, in the BaseTest, we provide if-else condition wherein depending upon the browser value it decides whether to load ChromeDriver Object or FirefoxDriver object into the same reference variable called driver.

## Interface:

In my framework, there were few things, which were never changing like key and value of the driver exe files and excel path and property path. To indicate these are constants, we have created an interface and stored it inside it.

## Method - overloading:

In WebDriverCommonLib class, we write overloaded methods for selecting an option from the dropdown.

```
public void select (WebElement ele, int index)  
public void select (WebElement ele, String text)
```

			1	2	3
4	5	6	7	8	9 10
11	12	13	14	15	16 17
18	19	20	21	22	23 24
25	26	27	28	29	30 31
S	M	T	W	T	F S

July  
Tuesday

2019

23

Wk-30 • 204-161

public void select(String value, WebElement ele)

Method - Overriding :

In Listener, implementation class called MyListener, we have implemented the methods of ITestListener interface. And, all the methods are getting overridden inside MyListener class.

Abstract Class :

In BaseTest class, though we have two methods, openBrowser() & closeBrowser() [tearDown()] but they cannot run independently. They are dependent on @Test methods for execution. Since they can't run independently to denote that no automation engineer should run BaseTest class, we made BaseTest class as Abstract class.

Collection :

While handling multiple elements we use List < WebElement > interface. Also, in a situation of printing elements text without duplicate etc, we use Set interface. So, List, Set are Collection topic.

Array :

To store the data in databanks, we create an object of two dimensional array in DataProvider Concept.

Static variable / Global public Variable:

public static WebDriver driver;

24

July  
Wednesday

Wk-30 • 205-160

2019

		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31			
S	M	T	W	T	F	S	S

In BaseTest, Once we declare driver as WebDriver and make sure every test Script inherits And understands this declaration from BaseTest. Hence to access it everywhere, we declare as public.

To load this declaration first, before every other class wants to access it to make it as static.

MyListeners : (Refer next pages for the notes)

```

01 package com.actitime.genericLib;
02 import org.testng.ITestContext;
03 import org.testng.ITestListener;
04 import org.testng.ITestResult;
05 import org.testng.Reporter;
06
07 public class MyListeners implements ITestListener {
08     @Override
09     public void onStart(ITestContext context) {
10         Reporter.log(context.getName() + " started", true);
11     }
12     @Override
13     public void onTestStart(ITestResult result) {
14         Reporter.log(result.getName() + " method started", true);
15     }
16 }
```

AUGUST '19
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
S
M
T
W
T
F
S

July  
Thursday

2019

25

Wk-30 • 206-159

@override

```
08 public void onTestSuccess(ITestResult result)
09 Reporter.log(result.getName() + " method PASSED, true);
```

10 @Override
11 public void onTestFailure(ITestResult result)

```
12 Reporter.log(result.getName() + " method FAILED, true);
13 WebDriverCommonLib wlib = new WebDriverCommonLib();
14 wlib.fullPageScreenshot("./screenshots/" + result.getName()
15 + ".png");
```

02 @Override

```
03 public void onTestSkipped(ITestResult result)
```

Reporter.log(result.getName() + " method SKIPPED, true);

04 @Override

```
05 public void onFinish(ITestContext context)
```

06 Reporter.log(context.getName() + " ended", true);

07 VerifyTasksListPageTest extends BaseTest; it is  
+ VerifyTasksListPageTest extends BaseTest; it is

package com.actitime.tasks; import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement; import org.openqa.selenium.support.ui.ExpectedConditions;

@Listeners (com.actitime.genericLib.MyListeners.class)

public class VerifyTasksListPageTest extends BaseTest

26

July

Friday

Wk-30 • 207-158

2019

JULY '19	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

@Test

```
08 public void verifyTasks() throws throwable
{
```

```
09 // Login to actitime and verify homepage
```

```
10 ValidLoginTest vl = new ValidLoginTest();
  vl.loginToApp();
```

```
11 // Click on Tasks tab
```

```
12 EnterTimeTrackPage et = new EnterTimeTrackPage();
  et.clickTasksTab();
  Assert.fail();
```

```
02 // Verify Tasks Tab is Displayed.
```

```
03 WebDriverCommonLib wlib = new WebDriverCommonLib();
  FileLib flib = new FileLib();
  wlib.verify(flib.readPropertyData("PROP_PATH", "taskTitle"),
  wlib.getPageTitle(), "Task Page");
```

06 TestNG Listeners:

- 07 → Listeners "listen" to every action / event defined in the Selenium script and behave accordingly,
- It is used in Selenium by implementing ITestListener interface of TestNG.
- It allows customizing TestNG reports or logs.

AUGUST '19
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
S
M
T
W
T
F
S

July  
Saturday

2019

27

Wk-30 • 208-157

Q: How do you integrate TestNG Listeners with our test methods?

We use a class level annotation called @Listeners (path of listener implementation class)

Eg: @Listeners (com.actitime.genericLib.MyListener.class)  
public class CreateUser extends BaseTest,

⇒ TestNG Listener has following methods:

onStart - is called when every TestRunner Execution Starts.

onTestSuccess - is called on the success of any test.

onTestFailure - is called on the failure of any test.

onTestSkipped - is called on skipped off any test.

onFinish - is called after every TestRunner execution is finished.

Steps :

1. Create a class and make sure it implements ITestListener (interface).
  2. Right click inside class, source and click on 'Add Unimplemented Methods'. All the incomplete (abstract) methods will automatically come and falls in this class.
  3. Start writing the action you need to perform at each and every listener level.
- OnTestFailure - we can write taking screenshot program so that it will take a screenshot

29

July  
Monday

Wk-31 • 210-155

2019

JULY '19		1	2	3	4	5	6
		7	8	9	10	11	12
		14	15	16	17	18	19
		21	22	23	24	25	26
		28	29	30	31		
	S	M	T	W	T	F	S

every time when a test method fails.

4. There are interfaces like ITestContext (Test Runner level) and ITestResult (Test Method level) which is helpful in getting test method names by using `getName()` method.

20.10.2021

D36

## SOURCE CONTROL TOOL / CONFIGURATION MANAGEMENT

### TOOL / VERSION CONTROL TOOL / SOURCE CODE MANAGEMENT TOOL

- GitHub
- SVN (Sub Version)
- CVS (Concurrent Version System)
- Perforce.

It is a decentralised repository in which Source Code of the framework can be stored. It provides web UI, can be easily accessible anywhere in the world through internet.

There are 2 plug-ins available in Git.

1. GitHub
2. Git

GitHub: It's a cloud based decentralised repo, where we should create an account with github community in order to use it.

Basically, GitHub is a storage where we can create our own repos, that repos can hold the data (Based on money you pay, you get space).

			1	2	3
4	5	6	7	8	9
11	12	13	14	15	16
18	19	20	21	22	23
25	26	27	28	29	30
S	M	T	W	T	F

July  
Tuesday

2019

30

Wk-31 • 211-154

10 GB - 10 TB etc). that is being shared across the world.

Advantages of GitHub:

1. It's a cloud based de-centralised repo.
2. No need to invest money for physical hardware.
3. No need of any admin to maintain the storage.
4. It provides Web Interface.
5. Can be easily accessible anywhere in the world through internet.
6. It provides security and backup facility.

Git: It's a client plug-in which should be installed in local system which is used to access github over the net (internet).

Without Git plug-in we can't communicate to github.

There are many plugins available in the market.

1. E-Git (Eclipse-Git) - Default plugin for latest eclipse.
2. Git-Bash (cmd prompt)
3. Git-Desktop (cmd prompt)

**Q: How to Create Account with GitHub?**

→ Go to google, Search for Github login.

→ Navigate to <https://github.com/login>

→ Click on create an account by providing the details.

31

July  
Wednesday

Wk-31 • 212-153

2019

1	2	3	4	5	6	7
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
S	M	T	W	T	F	S

Q: How to create a repo in Github?

- 08 → Login to GitHub
  - 09 → Go to '+' dropdown menu and click on new repos.
  - 10 → Give repo name & and click on checkbox, initialize repo with read me.
  - 11 → Go to Code dropdown and copy the git repo URL.

<https://github.com/raghavendravinyb/ActitimeRepos.git>

12 Username : username , Password : password

Q: How to share existing framework in eclipse through git?

- 03 Select the project, Right click → Team → Share project, Create (Path) - finish - finish,  
No-HEAD means Successfully Shared.

Q: How to transfer framework from working directory to git?

- 06 Select Project → Right click → Team → Commit.  
Go to Git Staging window, drag all the files from Unstaged area to Staged area →
  - 07 Give Commit message → Commit.

Q: Transferring Framework from local repo to Global  
Right click on project → Team → push branchmaster

Q: How to get the framework from Global repo into local system?

Pre-Condition: Make sure you have git username and password.

SEPTEMBER '19
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
S
M
T
W
T
F
S

August  
Thursday

01

2019

Wk-31 • 213-152

1. Go to Eclipse, file → import
2. expand git → Project from git
3. clone Url → URL → UNC PWD → next-next-finish

**Q:** How to transfer newly created file from local repo to global repo?

- Select New file → right click → Commit.  
 Go to git Staging window, drag all the files from unstaged area to Staged area.  
 → Commit message → click on Commit and push.

**Q:** How to get new files available in github to local?

- Select the project → Right click → team → pull

**Q:** When will you get conflicts in github and how to resolve?  
 Whenever two engineers modifies the same file, when you try to commit that file into github, you'll get conflict.

In order to resolve this problem, pull the framework using rebase option.

## BUILD TESTING TOOLS:

Ant - previously used  
 Maven - Currently using

**Maven:** It is basically a dev tool used to test the applications build and create a build.

02

August

Friday

Wk-31 • 214-151

2019

AUGUST '19	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	

## Maven Usage in Automation:

- ⑧ Maven is a build testing tool, which will check the integration issue with the framework component.
- ⑨ Whenever multiple engineers integrate their work with the same framework.

10 Integration issues might get when engineers modify the code, which might affect other java file, which is created by other engineers.

12 Maven identifies such kind of issues and provide a message called build failure / build success.

01 There are 2 plug-ins available in framework

- ② 1. Maven cmd line plugin
- 2. Maven Integration plugin.

⇒ Maven Cmd line plugin: It should be explicitly installed in the local system which is used to handle the maven project in the Command line.

### Installation Steps:

- ⑥ → Go to Google, search for download maven
- Click on first link, navigate to maven Community.
- ⑦ → Beside the binary, click on apache-maven-3.6.1-bin
- Download And extract the zip file.
- Get inside the folder and copy the bin location.
- Go to environment variables window in local system.
- Go to System Variable division, find and edit path variable, enter new path as C:\apache-maven-3.6.1\bin And click on OK.

SEPTEMBER '19
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

August  
Saturday

2019

03

Wk-31 • 215-150

- Go to user variable div, click on new, and then create new M2\_HOME = C:\Apache-maven-3.6.1.
- Ensure JAVA\_HOME is also available otherwise maven will not work.
- Go to cmd, type mvn -version to check if maven is installed or not.

### → Maven - Integration Plugin:

It's a default plug-in, available in latest eclipse which is used to create maven project through eclipse.

### Advantages :

1. Has dependency feature, where we can download all dependent jar directly from the internet.
2. It is a build testing tool used to check the integration issue and provide the status of the build.
3. Using maven, we can run the test scripts in cmd line without eclipse.
4. Maven provides standard framework folder structure.

Q: What is maven dependency and how to handle maven dependency?

Dependency is a feature available only in Maven which is used to download all the dependency jar into local repo (C:/user/m2).

Global Repository (Google → mvn repository):

In order to add dependency we should write dependency code for webdriver, apache poi, testing etc., inside POM.xml.

04 Sunday

# 05

August

Monday

Wk-32 • 217-148

2019

AUGUST '19	4	5	6	7	1	2	3
S	11	12	13	14	15	16	17
M	18	19	20	21	22	23	24
T	25	26	27	28	29	30	31
W							

POM.xml :

Project Object Model is a configuration file for the maven which is used to handle maven project in cmd line.

→ Dependency code include group id, artifact id and version.

Group-id - Name of the Group / Community from where you are downloading.

Artifact-id - Name of the project file in that group / community - which file you are download.

Version - indicates version of the file which you want to download.

## Maven Life Cycle - Maven Goals

Clean - This phase involves cleaning of the project (for a fresh build and deployment).

Compile - Compilation of the project source code.

Install - Installing the application in the local repository. Any other project can use this as a dependency.

Test - Executing / Running the tests using some suitable test framework.

Note : For Compiling, we download Compiler plugin for test, we need to download Mvn Surefire plugin.

SEPTEMBER '19	1	2	3	4	5	6	7
	8	9	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28
	29	30					
S	M	T	W	T	F	S	

August

Tuesday

2019

06

Wk-32 • 218-147

Q: How to solve MojoFailureException when it says you are running on JRE rather than JDK?

- 9 Right click on JRE → Properties → Installed JRE's → Add → Next → Directory → Select path of JDK → Finish → Select JDK instead of JRE → Click on apply and close → Select workspace default JRE(JDK)
- 11 Click on apply and close.
- 12 **JENKINS:** It is a continuous integration and continuous deployment tool.  
→ In case of development, jenkins will be used to automate build creation and build testing and build deployment process. So, it is called Continuous deployment tool.

- 14 Jenkins Usage in Automation:
  - In Automation, jenkins will be used in Continuous integration process, it means Jenkins will always monitor the framework in Git location, automatically creates a build and provides email notification if any changes had happened in the build.
  - Jenkins can also be used to execute all the test scripts whenever new build arrives to testing environment.

- Jenkins provides 3 levels of execution:
1. On-demand (at user request or after a trigger)
  2. On-Schedule (at regular intervals or after a trigger)
  3. Poll - SCM (Source Code Management) (periodically or after a trigger)

# 07

August  
Wednesday

Wk-32 • 219-146

2019

AUGUST '19	4	5	6	7	1	2	3	4
	11	12	13	14	15	16	17	18
	18	19	20	21	22	23	24	25
	25	26	27	28	29	30	31	
S	M	T	W	T	F	S	S	S

On-demand: Based on the customer demand,  
08 login to Jenkins and start the execution.

09 On-Schedule: Based on the schedule time, which is given in the Configure Area, Jenkins automatically triggers build.

11 Poll- SCM: Poll means Continuously monitor the framework or application in git location, it automatically starts the execution whenever any changes happen in the framework or testing server.

01

**Note:** Whenever jenkins starts the job execution, it will connect to git and download the entire framework (build) into the local system and test the build by taking help of maven and execute all the test scripts and sends out an email.

04

2 Installation steps of Jenkins:  
05 → Go to Google, Search download jenkins, click on first link and click on windows option.  
06 → Download the exe file and install.  
→ After installation, in order to open jenkins, <http://localhost:8080>.

07

Jenkins Plugins:

→ Before creating a project in Jenkins make sure below plugins are installed in Jenkins  
→ Login to Jenkins, go to manage Jenkins, click on manage plugin → click on "Available" tab, Download  
a) Maven Integration Plugin b) Github Integration Plugin

SEPTEMBER '19	1	2	3	4	5	6	7
	8	9	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28
	29	30					
	S	M	T	W	T	F	S

August  
Thursday

08

2019

Wk-32 • 220-145

## Jenkins Environment Variables

- After the installation of the plugin, make sure environment variables are added for Java, maven and git.
- Login to Jenkins → manage Jenkins → Global tool Configuration → JDK → Unselect Install automatically → give name as JAVA\_HOME and location of JDK.
- Git : → C:\Program Files\Git\bin\git.exe
- Do the same as above for Git and maven
- Git.exe path
- MAVEN\_HOME
- Click on Save to save all the changes.

Q: How to create a project or Job in Jenkins?

1. Login to Jenkins
2. Click on new item, Select maven project and click on OK.

Q: How to Configure git project/ framework location in Jenkins project or job?

- Login to Jenkins → click on Project Name → click on Configure link.
- Find the click git radio button under 'Source Code Management' section
- provide git url (Take it from 'Code' of Github)
- Click on add button, provide git username and password.
- Select username, password from the credential dropdown.
- Scroll down, find build division.
- Enter in root POM: Project Name(Path)/POM.xml
- Goals and Options: test

09

August

Friday

Wk-32 • 221-144

2019

AUGUST '19	4	5	6	7	8	9	10
	11	12	13	14	15	16	17
	18	19	20	21	22	23	24
	25	26	27	28	29	30	31
	S	M	T	W	T	F	S

→ Click on save and click on build now.

08

Q: How to Configure Schedule?

- 09 1. Login to Jenkins
2. Go to Configure file of the job
- 10 3. Scroll down and Select build periodically checkbox under Build Triggers section.
- 11 4. Provide a time with respect to (min hour date month day)

(31 09 04 07 06)

- 01 Here it indicates 4th July Saturday 9:31 hours
- 02 → Minutes (0-59)
- 03 → Hours (0-23)
- 04 → Date of Month (1-31)
- 05 → Months (1-12)
- 06 → Day (0-7) where 0 and 7 are Sundays

Q: How to Configure Poll SCM?

- 05 1. Go to project configure file.
- 06 2. Select Poll SCM checkbox and provide 5 star separated by space (\* \* \* \* \*)
- 07 ' \* means any minute, any day, any hour etc.

Q: How to Configure email?

- 01 1. Go to Jenkins project configure file.
- 02 2. Select email notification checkbox at the end of the stage.
- 03 3. Provide recipients

Note: In order to Configure email notification, we should provide Company SMTP credentials in global Configuration file.