**Table Contents:**

**Lambok**

**Maven**

**Apache POI**

**Thymleaf**

**GitHub**

**Java8features**

**Lambok :-**

Kind of tool hain and sath hi library bhi hain isko apne project main add krne se private data ke liye public getter setter banana ki jaroorat nahi padegi

**Lambok ke banana ke do tarike hote hain**

* Lambok ki plugin ko add krna
* Lambok ki dependcy ko apne project main add krke project ko banana

Note -bolwler plate code ko lambok remove kr deta hain jise hum lengthi code bolate hain use remove kr deta hain

Bolwler plate code wo code hota hain jis se program lengthi lage lambok hum kuch annotation deta hain jise lagane per ye code nahi likhana padta hain

**Maven-** hum maven main do tarike se project bana sakate hain

* Agar app desktop app bana rahe ho to quick start lelo
* Agar app web app bana rahe ho to maven webapp le sakate ho

**Maven:-**maven project ko mange krne ka ek tool hota hain

Maven tutorial provides basic and advanced concepts of **apache maven** technology. Our maven tutorial is developed for beginners and professionals.

Maven is a powerful *project management tool* that is based on POM (project object model). It is used for projects build, dependency and documentation.

It simplifies the build process like ANT. But it is too much advanced than ANT.

Current version of Maven is 3.

**Maven ki jab hum use nahi krte hain the to project ko use krne main bahut si problem ati thi usme jar war add krne main**

## **Understanding the problem without Maven**

There are many problems that we face during the project development. They are discussed below:

**1) Adding set of Jars in each project:** In case of struts, spring, hibernate frameworks, we need to add set of jar files in each project. It must include all the dependencies of jars also.

**2) Creating the right project structure:** We must create the right project structure in servlet, struts etc, otherwise it will not be executed.

**3) Building and Deploying the project:** We must have to build and deploy the project so that it may work.

What it does?

Maven simplifies the above mentioned problems. It does mainly following tasks.

1. It makes a project easy to build
2. It provides uniform build process (maven project can be shared by all the maven projects)
3. It provides project information (log document, cross referenced sources, mailing list, dependency list, unit test reports etc.)
4. It is easy to migrate for new features of Maven

Apache Maven helps to manage

* Builds
* Documentation
* Reporing
* SCMs
* Releases
* Distribution

What is Build Tool

A build tool takes care of everything for building a process. It does following:

* Generates source code (if auto-generated code is used)
* Generates documentation from source code
* Compiles source code
* Packages compiled code into JAR of ZIP file
* Installs the packaged code in local repository, server repository, or central repository

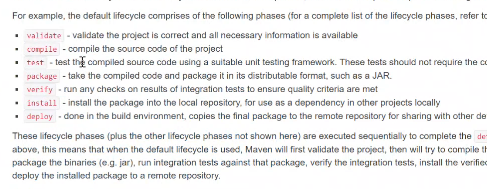
Ye language specific tool nahi hota hain ise kisi main use kr sakate hain

**Pahla to advantages ye hain ki agar apko koi jar ko add krna hain apne project main agar maven nahi hoga to apko download krna padega aur phir add krna hoga but agar maven hoga to maven ke pass poxm.xml file hoti hain jisme hum dependency dal dege to wo dependency se wo jar file add ho jayegi wo dependency maven ki maven repository central website per sabaki dependency mil jayegi**

**Maven project management tool hota hain**

**Command hain pata krna ki maven install hain ya nahi-mvn archetype:generate -DgroupId=com.javatpoint -DartifactId=CubeGenerator-DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false**

**Life cycle of Maven :-**

****

**Apache POI:-** ye word file and excel se data ko read write krne ke liye hota hain specially **-hum normally input output se bhi kr sakate hain but isme jada features hote hain**

[Apache POI](https://www.geeksforgeeks.org/apache-poi-introduction/) is an open-source java library to create and manipulate various file formats based on Microsoft Office. Using POI, one should be able to perform create, modify and display/read operations on the following file formats. For Example, Java doesn’t provide built-in support for working with excel files, so we need to look for open-source APIs for the job.

[Apache POI](https://www.geeksforgeeks.org/apache-poi-introduction/) provides Java API for manipulating various file formats based on the Office Open XML (OOXML) standard and OLE2 standard from Microsoft. Apache POI releases are available under the Apache License (V2.0).

**Some important features of**[**Apache POI**](https://www.geeksforgeeks.org/apache-poi-introduction/)**are as follows:**

* Apache POI provides stream-based processing, that is suitable for large files and requires less memory.
* Apache POI is able to handle both XLS and XLSX formats of spreadsheets.
* Apache POI contains HSSF implementation for Excel ’97(-2007) file format i.e XLS.
* Apache POI XSSF implementation should be used for Excel 2007 OOXML (.xlsx) file format.
* Apache POI HSSF and XSSF API provide mechanisms to read, write or modify excel spreadsheets.
* Apache POI also provides SXSSF API that is an extension of XSSF to work with very large excel sheets.
* SXSSF API requires less memory and is suitable when working with very large spreadsheets and heap memory is limited.
* There are two models to choose from – the event model and the user model. The event model requires less memory because the excel file is read in tokens and requires processing. The user model is more object-oriented and easy to use.
* Apache POI provides excellent support for additional excel features such as working with Formulas, creating cell styles by filling colors and borders, fonts, headers and footers, data validations, images, hyperlinks, etc.

**Commonly used components of Apache POI**

1. **HSSF (Horrible Spreadsheet Format)**: It is used to read and write xls format of MS-Excel files.
2. **XSSF (XML Spreadsheet Format)**: It is used for xlsx file format of MS-Excel.
3. **POIFS (Poor Obfuscation Implementation File System)**: This component is the basic factor of all other POI elements. It is used to read different files explicitly.
4. **HWPF (Horrible Word Processor Format)**: It is used to read and write doc extension files of MS-Word.
5. **HSLF (Horrible Slide Layout Format)**: It is used for read, create, and edit PowerPoint presentations.

# **How to Read Excel File in Java**

In this section, we are going to learn how we can read data from an excel file.

In Java, reading excel file is not similar to read word file because of cells in excel file. JDK does not provide direct API to read or write Microsoft Excel or Word document. We have to rely on the third-party library that is Apache POI.

## **What is Apache POI?**

**Apache POI** (Poor Obfuscation Implementation) is a Java API for reading and writing Microsoft Documents in both formats **.xls** and **.xlsx**. It contains classes and interfaces. The Apache POI library provides two implementations for reading excel files:

* **HSSF (Horrible SpreadSheet Format) Implementation:** It denotes an API that is working with Excel 2003 or earlier versions.
* **XSSF (XML SpreadSheet Format) Implementation:** It denotes an API that is working with Excel 2007 or later versions.

## **Interfaces and Classes in Apache POI**

### Interfaces

* **Workbook:** It represents an **Excel Workbook**. It is an interface implement by **HSSFWorkbook** and **XSSFWorkbook**.
* **Sheet:** It is an interface that represents an **Excel worksheet**. A sheet is a central structure of a workbook, which represents a grid of cells. The Sheet interface extends **java.lang.Iterable**.
* **Row:** It is also an interface that represents the **row** of the spreadsheet. The Row interface extends **java.lang.Iterable**. There are two concrete classes: **HSSFRow** and **XSSFRow**.
* **Cell:** It is an interface. It is a high-level representation of a **cell** in a row of the spreadsheet. **HSSFCell** and **XSSFCell** implement Cell interface.

### Classes

**XLS Classes**

* **HSSFWorkbook:** It is a class representing the XLS file.
* **HSSFSheet:** It is a class representing the sheet in an XLS file.
* **HSSFRow:** It is a class representing a row in the sheet of XLS file.
* **HSSFCell:** It is a class representing a cell in a row of XLS file.

**XLSX Classes**

* **XSSFWorkbook:** It is a class representing the XLSX file.
* **XSSFSheet:** It is a class representing the sheet in an XLSX file.
* **XSSFRow:** It is a class representing a row in the sheet of XLSX file.
* **XSSFCell:** It is a class representing a cell in a row of XLSX file.

## **Steps to read data from XLS file**

**Step 1:** Create a simple Java project in eclipse.

**Step 2:** Now, create a lib folder in the project.

**Step 3:** Download and add the following jar files in the lib folder:

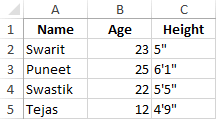
* commons-collections4-4.1.jar [**Click Here**](https://bit.ly/2SG4r3Y)
* poi-3.17.jar [**Click Here**](https://bit.ly/2Y6HRY9)
* poi-ooxml-3.17.jar [**Click Here**](https://bit.ly/2LJ1leO)
* poi-ooxml-schemas-3.17.jar [**Click Here**](https://bit.ly/2LHjeL9)
* xmlbeans-2.6.0.jar [**Click Here**](https://bit.ly/2ybqxBR)

**Step 4:** Set the Class Path:

Right-click on the project ->Build Path ->Add External JARs -> select all the above jar files -> Apply and close.

**Step 5:** Now create a class file with the name **ReadExcelFileDemo** and write the following code in the file.

**Step 6:** Create an excel file with the name "student.xls" and write some data into it.



**Step 7:** Save and run the program.

**Example of reading excel file (.xls) file**

1. **import** java.io.File;
2. **import** java.io.FileInputStream;
3. **import** java.io.IOException;
4. **import** org.apache.poi.hssf.usermodel.HSSFSheet;
5. **import** org.apache.poi.hssf.usermodel.HSSFWorkbook;
6. **import** org.apache.poi.ss.usermodel.Cell;
7. **import** org.apache.poi.ss.usermodel.FormulaEvaluator;
8. **import** org.apache.poi.ss.usermodel.Row;
9. **public** **class** ReadExcelFileDemo
10. {
11. **public** **static** **void** main(String args[]) **throws** IOException
12. {
13. //obtaining input bytes from a file
14. FileInputStream fis=**new** FileInputStream(**new** File("C:\\demo\\student.xls"));
15. //creating workbook instance that refers to .xls file
16. HSSFWorkbook wb=**new** HSSFWorkbook(fis);
17. //creating a Sheet object to retrieve the object
18. HSSFSheet sheet=wb.getSheetAt(0);
19. //evaluating cell type
20. FormulaEvaluator formulaEvaluator=wb.getCreationHelper().createFormulaEvaluator();
21. **for**(Row row: sheet)     //iteration over row using for each loop
22. {
23. **for**(Cell cell: row)    //iteration over cell using for each loop
24. {
25. **switch**(formulaEvaluator.evaluateInCell(cell).getCellType())
26. {
27. **case** Cell.CELL\_TYPE\_NUMERIC:   //field that represents numeric cell type
28. //getting the value of the cell as a number
29. System.out.print(cell.getNumericCellValue()+ "\t\t");
30. **break**;
31. **case** Cell.CELL\_TYPE\_STRING:    //field that represents string cell type
32. //getting the value of the cell as a string
33. System.out.print(cell.getStringCellValue()+ "\t\t");
34. **break**;
35. }
36. }
37. System.out.println();
38. }
39. }
40. }

**GitHub:-**

* gitHub ek website hoti hain jaha per hum sara code rakh sakate hain

**Git**:

* git version control system hain is se code main hum changes kr sakate hain(VCS)
* git madad se hum apne code ko github main store krayenge
* git software se tracking maintain krke rakh sakate hainfor example humlog ek project bana rahe muje dekhana hain kisne kitana kaam kiya
* version controlling bhi kr sakate hain-code ko role back kr sakate hain -for example jaise aap company main 1.1 version per kaam kr rahe ho aur ab 1.2 version lunch hua but wo abhi kaam nhai kr raha hain aur wo problem bhi nahi solve ho raha hain to hum github se 1.1 version uthakar 1.1 version per phir se kaam kr sakate hain git
* git hume bahut sari command deta hain jis se hum opration perform krenge

**README.md**: md-markdon -jab hum koi project banate hain to uska sara information readme main hota hain

**Humare pass do tarke hain**

**1.hum git per repository banaye waha per sara coding kre**

**2.hum git per sara code dal de**

**Command :-**

**git clone –**

* hum gitMyProject folder tha waha per humne gitMyforthe project ko clone kiya hain jis gitMyproject repositry ban gaya hain
* .hume github ka project liya hain aur usko clone kr liya hain
* Clone ka matlab hota hain jab github per ya kisi bhi server per project Rakha ho to hum apne system per la sakate hain clone command se
* Jab hum koi bhi project github se clone krte hain hain aur apne system main jis bhi location per late hain agar usme .git a jaye to jo github project hain wo clone ho gaya hain
* .git hain iske folder ka name myprojecte hain isko tracking ho skati hain but aur jitane bhi project banaye hain unki tracking nahi ho sakati hian

**git status :-**is se hum status dekhte hain

**Branch :**

* ek project per kai log kaam karate hain to unki sub branch bana dete hainaur ek main branch hoti hain aur usi ke andar se hum sub branch banate hain

**git add . :** ise se sari file add ho jayegi jo apne change ki thi

**Git stages :-**git alag -2 stage main work krta hain

* **modified**: apne na project main kuch change kiya hian
* **untracked: git** abhi is file ko track nahi kr pa raha hain
* **staged :**ready hain file commint krne ke liye
* **unmodified :**file main unchanged hain

**git commit** **-m**:- actual main jo apne change kiya hain wo git hub main jaye to isliye hum git commit ka use krenge

**git push origin main :-**

* jab git push krenge to waha git hub per pahuch jayega aur jaha change krena hain wo likho jaise muje main change krna hain to maim
* **maine** likha hain aur agar iske sath hum -u laga de to hume baar baar origin main likhne ki jaroot nahi padegi

**git init :**git repository ban jayegi jis hum track krwa sakate hain git hub per

**ls -a** :- is se hidden file bhi a jati hain

**git branch -M Kuraishi :-**hum branch ka name change kr sakate hain

**git checkout -b Rahul :** hum branch bana bhi sakate hain

**GitHub Project :-**

* git clone URLfrom location and change project
* cd project location
* git add .
* git commit -m “My first commit”
* git push

**System project :-**

**Case 1**

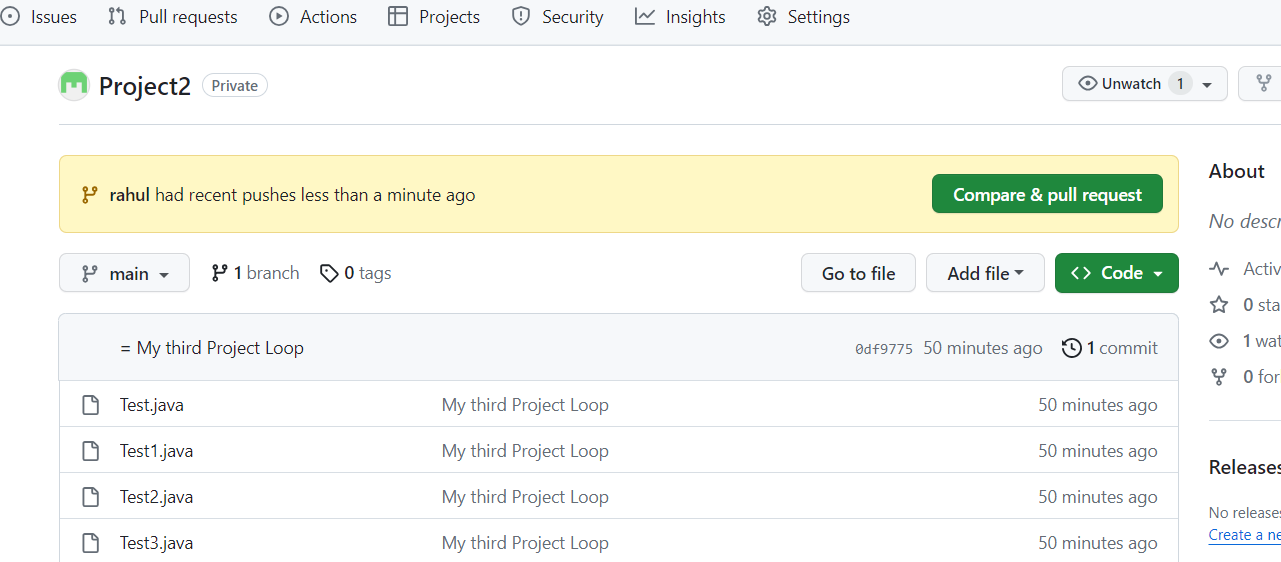
* **git init**
* **git remote add origin URL**
* **git add .**
* **git commit -m “My branch change same repository”**
* **git push --set-upstream origin master**

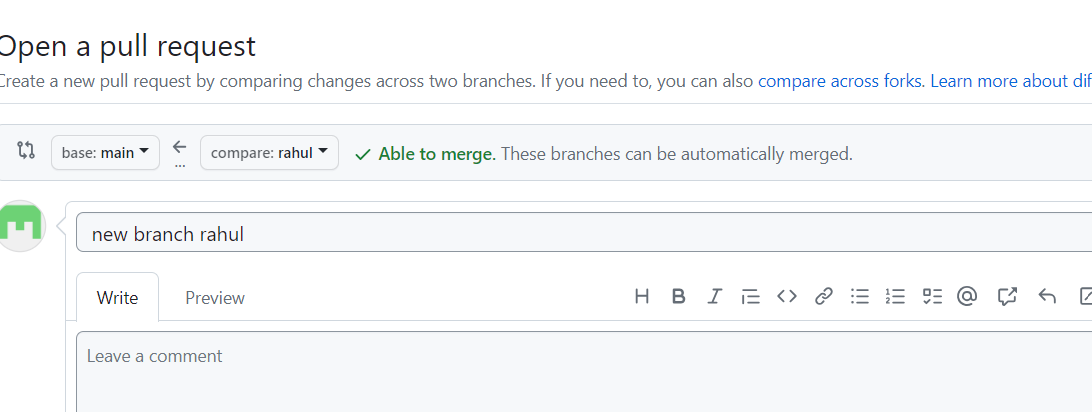
**case 2-system by**

* **git clone URL**
* **git add .**
* **git commit -m “My same branch change repository”**
* **git push --set-upstream origin master**

**Branch :-**

* Main branch main no change krna chahate hain to hum ek naya branch bana lenge aur usme change krenge aur check kra lenge
* git checkout -b rahul
* changes and git add .
* git commit -m “ “
* git push



****

**Merge rahul code and main**

**git branch -d – branch delete command**

**two branch main difference dekhana hain -command git diff rahul**

**branch merge krna hain folder wala to -command git merge rahul**

**Eclipse repo:-**Right click eclpse src and team click ,share and add index and commit and past URL finish.

**Git commands**

**-------------------------**

**1.git clone**

**----to copy the already existing repo from github**

**2.git status**

**----to track the status of git repo**

**3.git add [./foldername/file]**

**----to stage the file inside git repo**

**4.git commit -m "[commit message]"**

**-----to commit the changes in repo before push**

**5.git checkout -b "branchname"**

**---to create new branch and switch to it.**

**6.git remote add origin "specify URL"**

**--to link the origin with the repo before push**

**7.git push origin [branch name]**

**-----Push a branch to your remote repository**

**8.git config ----global user.name ="specify username"**

**----to set username for authorization**

**9.git config ---global user.email="specify email"**

**--to set email for authorization**

**10.git init**

**---to initialize the git repo**

**11.git branch**

**-----to check how many branches exists**

**12.git branch -D "Branch name"**

**--- to delete the existing branch forcefully**

**13.git branch -d "Branch name"**

**--- to delete the existing branch**

**14.git checkout "branch name"**

**--to switch to another branch**

**15.git branch -m [oldbranch] [new Branch]**

**----rename a branch**

**16.git checkout -**

**------Switch to the branch last checkedout**

**17.git merge [branchname]**

**---Merge a branch into a active branch**

**18.git checkout --[file\_name.txt]**

**----Discard changesto a file**

**19.git checkout -b [branch name] origin/[branch name]**

**----Clone a remote branch and switch to it**

**20.git branch -m [old branch name] [new branch name]**

**------Rename a local branch**

**21.git push origin --delete [branch name]**

**-------Delete a remote branch**

**22.git push -u origin [branch name]**

**-----Push changes to remote repository (and remember the branch)**

**23.git push**

**----Push changes to remote repository (remembered branch)**

**24.git push origin --delete [branch name]**

**-----Delete a remote branch**

**25.git pull**

**-----Update local repository to the newest**

**26.git pull origin [branch name]**

**----Pull changes from remote repository**

**27.git remote add origin ssh://git@github.com/[username]/[repository-name].git**

**----Add a remote repository**

**28.git remote set-url origin ssh://git@github.com/[username]/[repository-name].git**

**-----Set a repository's origin branch to SSH**

**29.git log --summary**

**----View changes (detailed)**

**30.git log --oneline**

**----View changes (briefly)**

**31.git diff [source branch] [target branch]**

**----Preview changes before merging**

**32.git revert commitid**

**-----Revert commit changes**

**33.git rm -r [file-name.txt]**

**-----Remove a file (or folder)**

**34.git add -A**

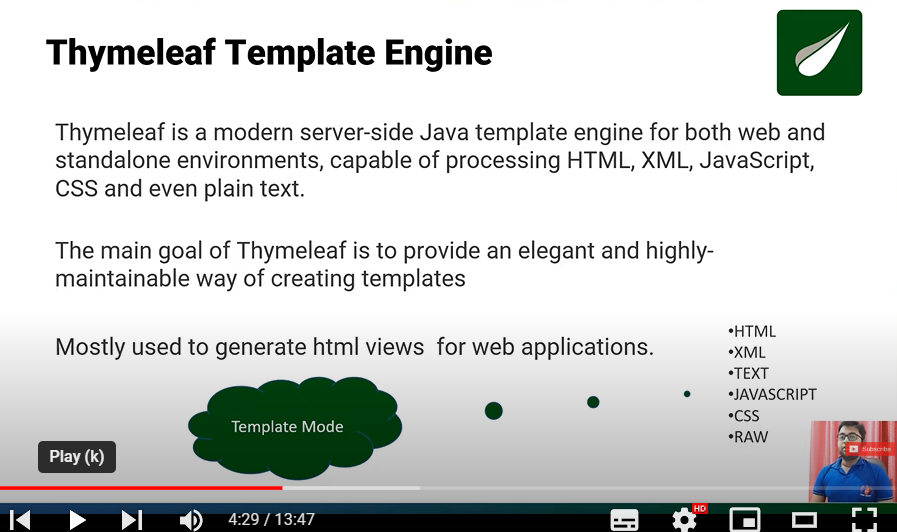
**-----Add all new and changed files to the staging area**

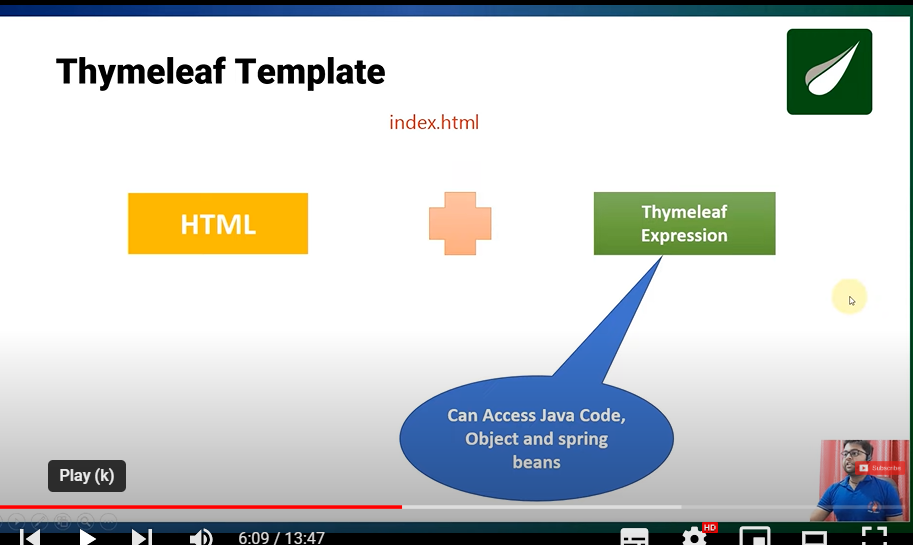
**35.git config --global --list**

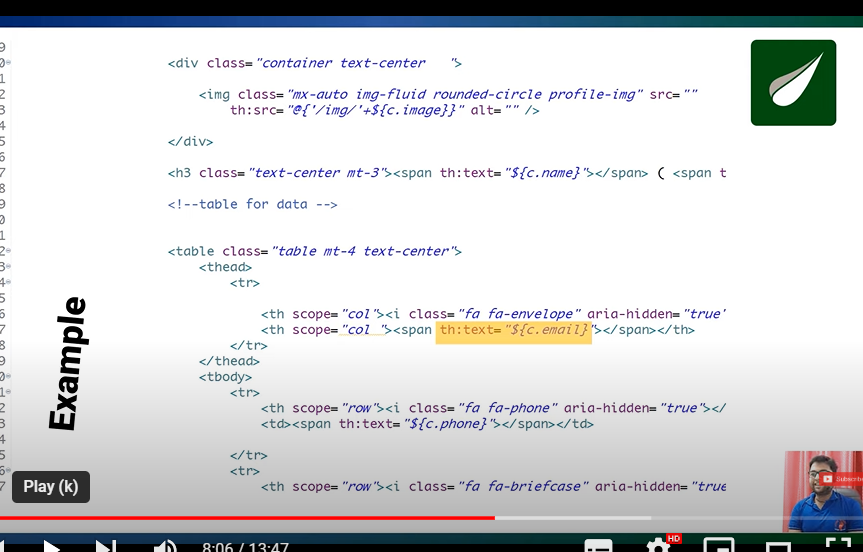
**----Get global config**

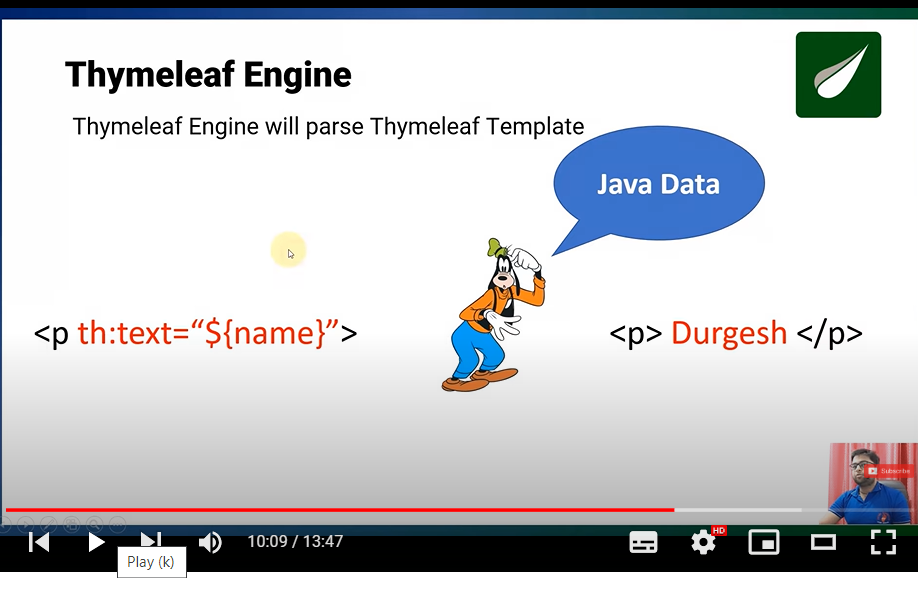
**Note:If any commands are missing anyone can add and also new commands of git as well**

**Thmeleaf Template Engine:-**

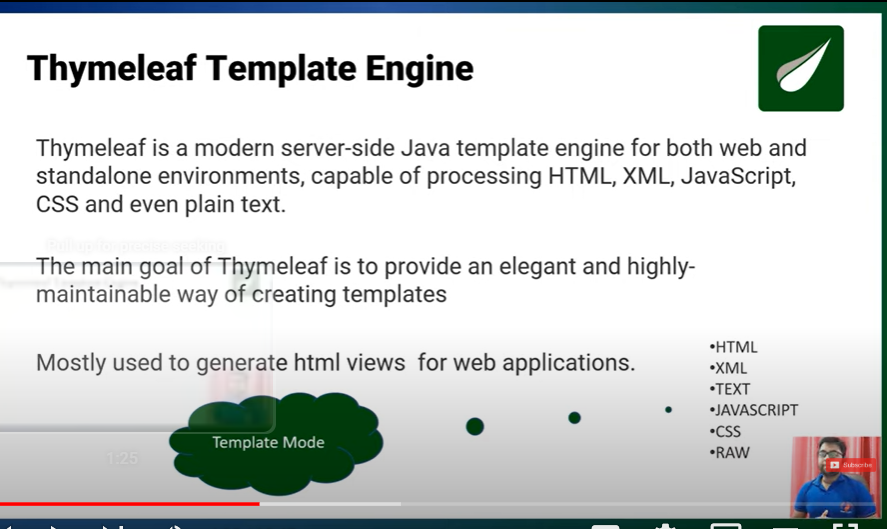
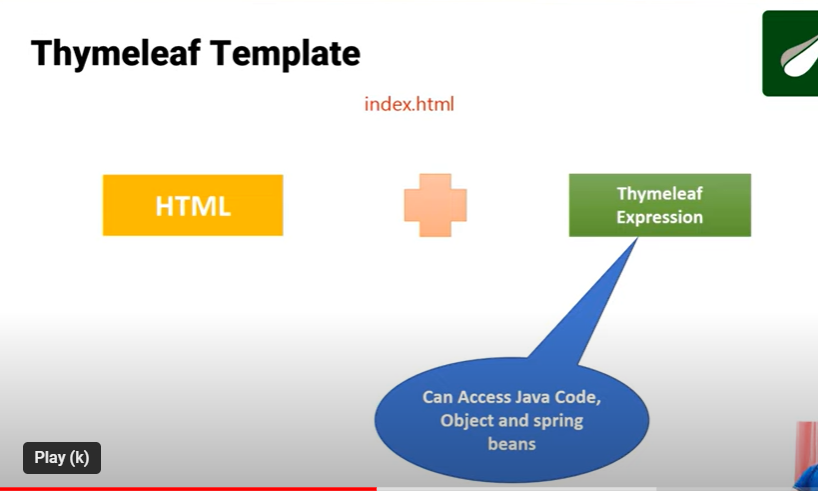
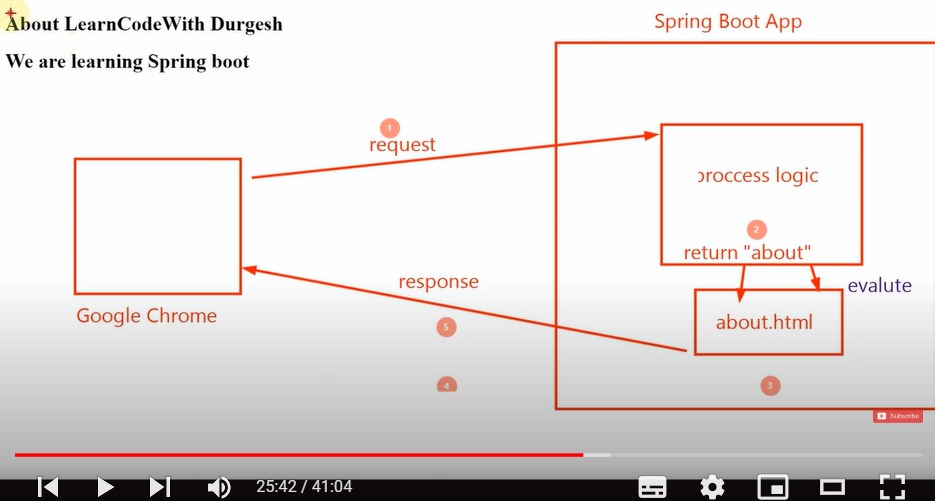
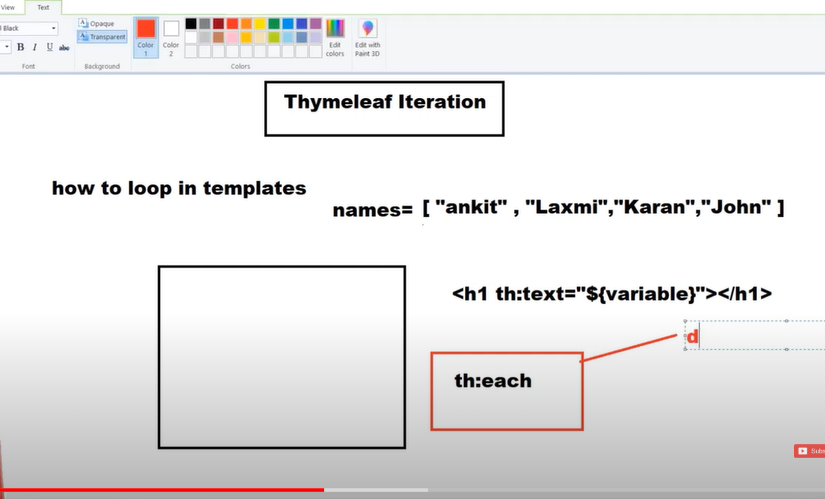


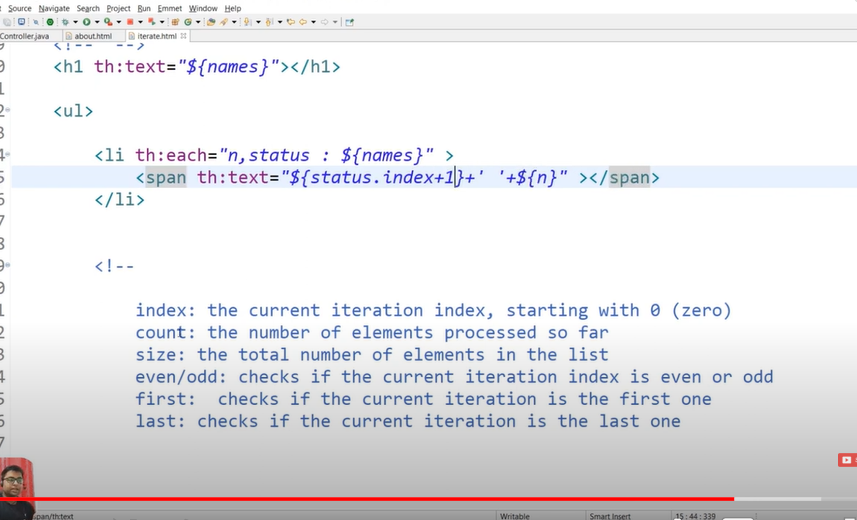


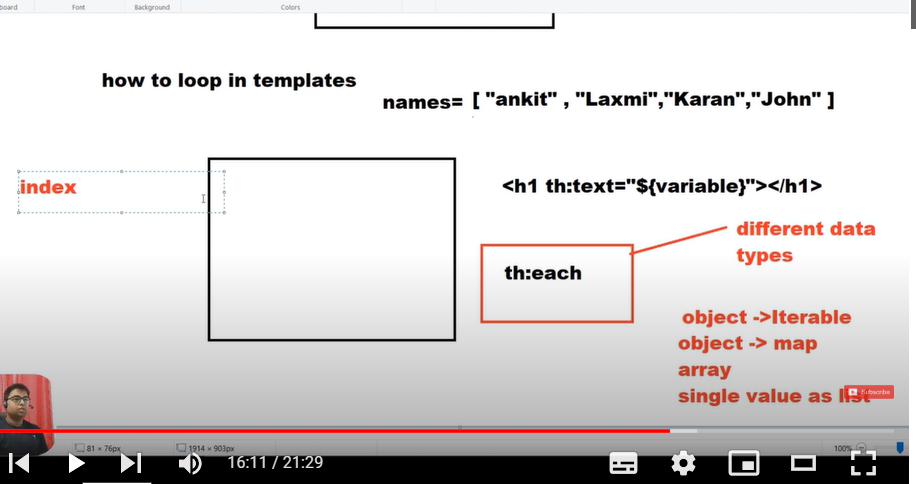


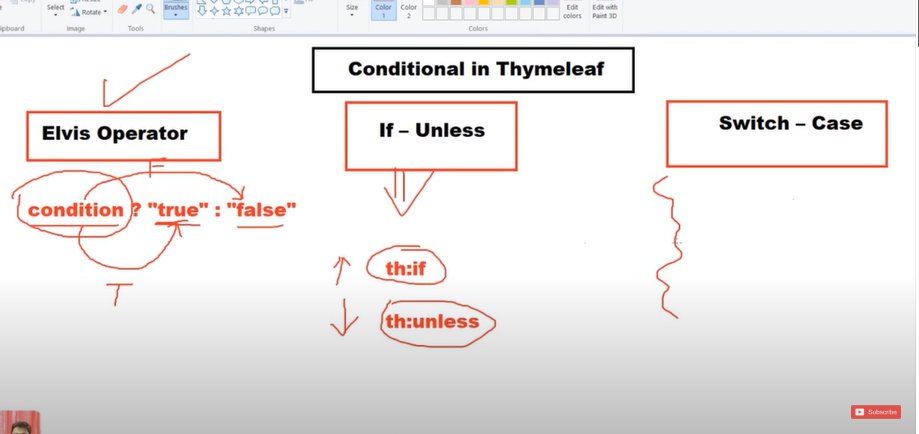


**Thymleaf:-**

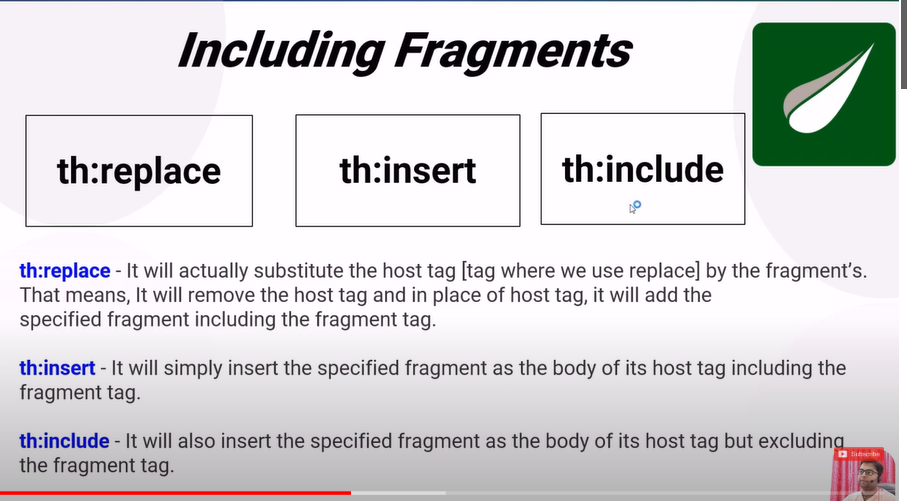
* Simler to html performance is better
* Mostly used to generate html views for web applications
* 
* 
* 
* Jsp main likha code pahle sevlet main change hota hain
* 
* 

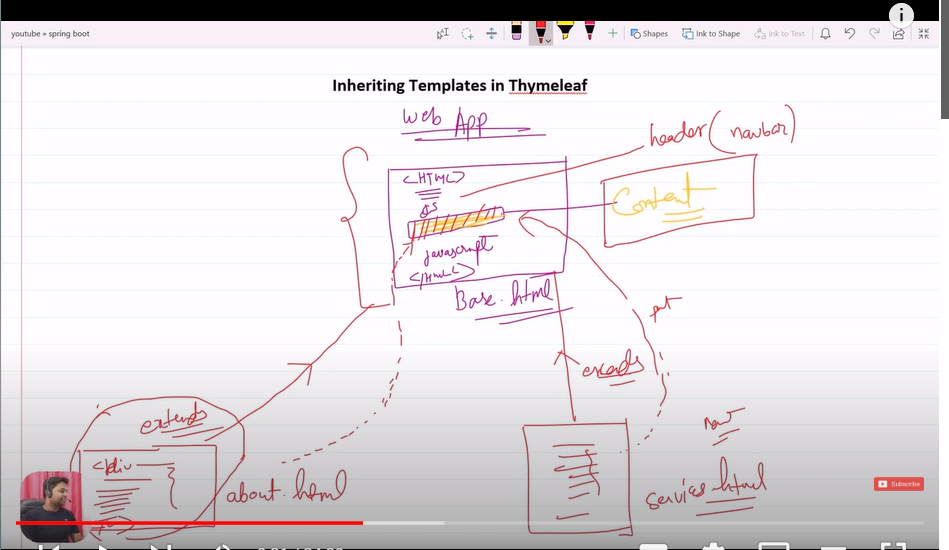
****

****

****

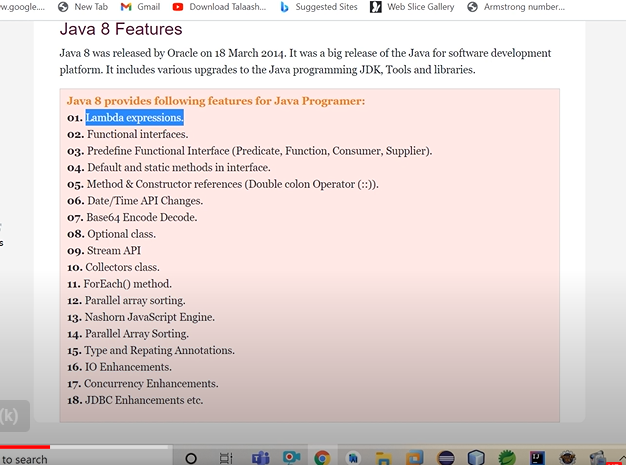
**Fragments:-**jo code baar reapeate hota hain use ek jagah rakh dete hain aur use jaha chahe waha use kr lete hain

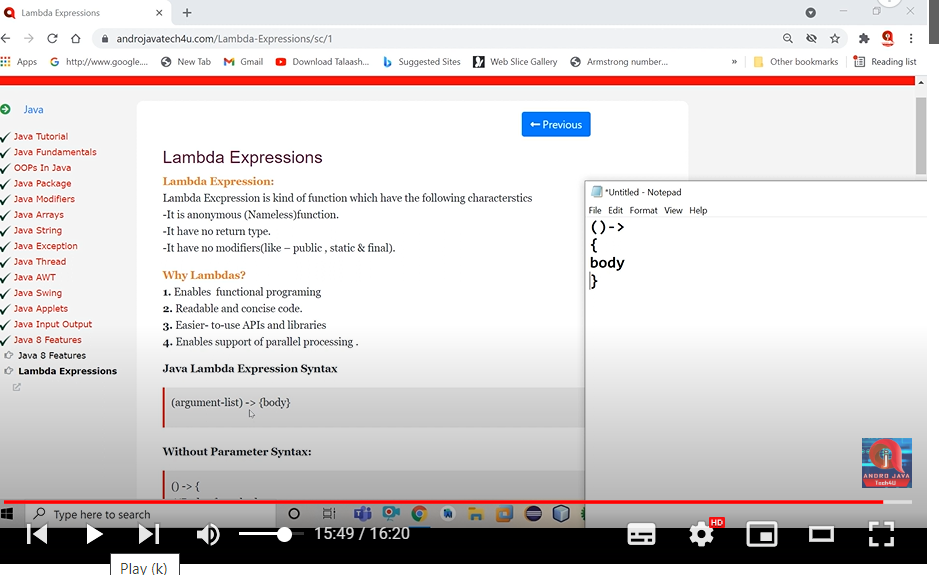


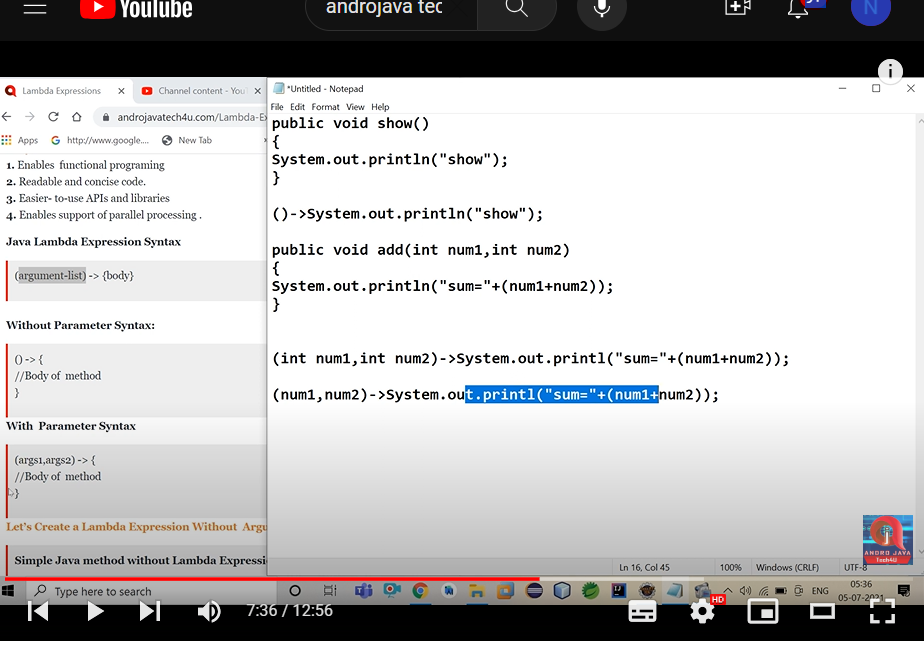


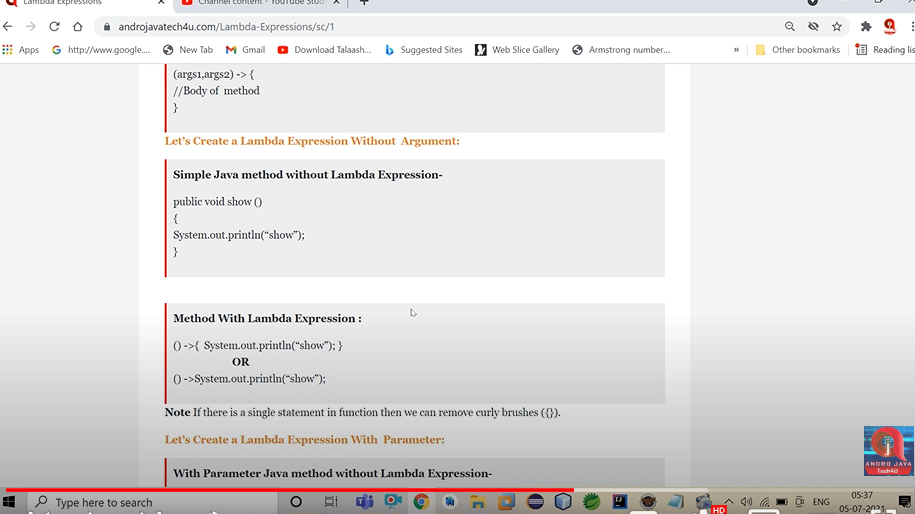
DOCKER:- ek application jiski madad se hum application ko ek jagah se dusri jagah shift krte hain wah per le jake run kr sakate hain

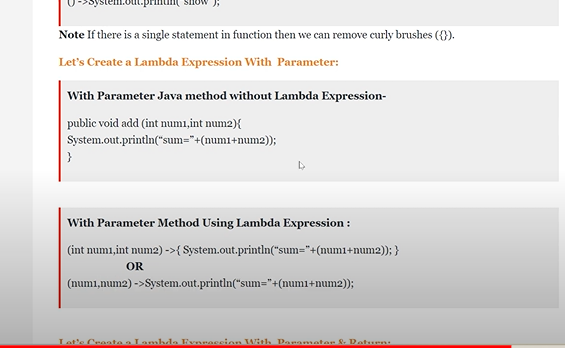
**Java8Features:-**ye code ko kam krne ke liye use hota hain

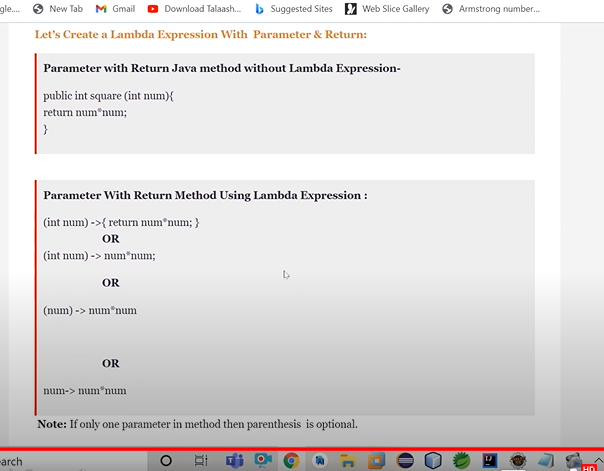


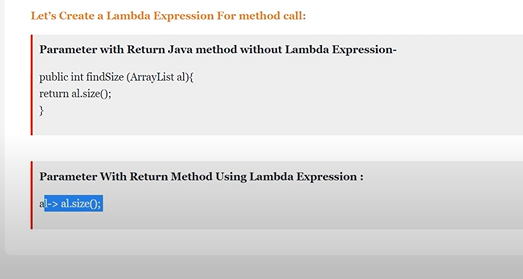


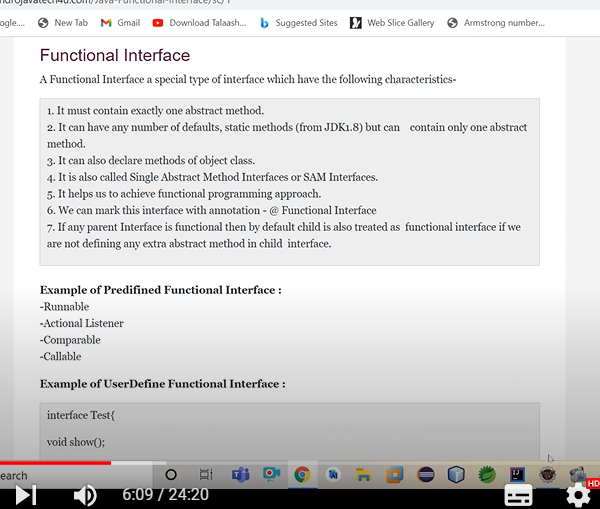




****

****

****

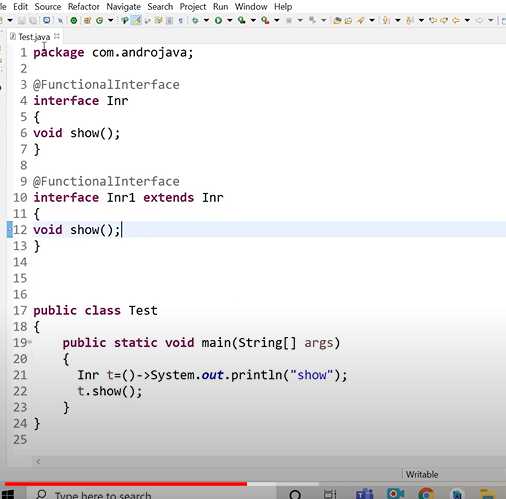
****

**Aur lambda expression ka use krte hain to anonymous ka byte code ke lliye alag class nahi banati hain jab hum lambda expression use krte hain aur nahi performance per kuch fark padega**

**Lambda expression anonymous ko reduce krne ke liye use kiya jata hain**

Normal Function--void show(){}**Lambda Function or Expression**:-()->

**what is functional interface?**An interface which contain exactly one abstract method in and can have n number of static and default method inside it.**why there is static and default method inside functional interface?**The default method has brought in the functional interface to provide some default value and programmer canuse that default values rather than passing their own values.**There are many pre define functional interface like**Runnable PredicateFunctionConsumerSupplierand much moreThe static method is just similar to default method but we need to call that static method explictly .**what is lamda expression or lamda function?**Lamda function is a anonymus function which does not have need to methion any return type of provide any parametertype . it has a syntax like ()-> by this we mention lamda function.It a also a way to achieve to functional interface. Where ever we need anonymus object then in such case we can use lamda function over it.It also provide the code reduncy and does not take any memory or create any byte code for it.it is also faster.



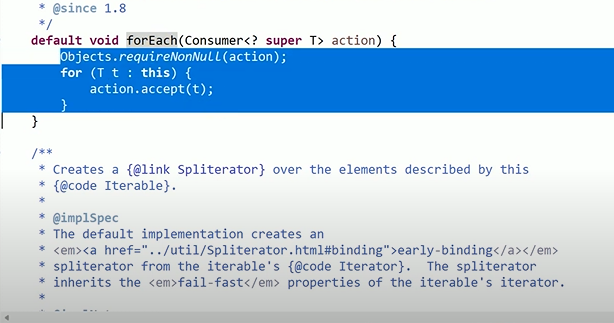
Agar ek interface dusre interface ko extend kre to agar dono main ek method hain aur same hain to donon interface functional interface honge kyuki waha per override ho raha hoga aur agar aisa nahi dono differ hain to waha per jiske pass ek method hoga aur child ke pass ek jada ho jayega isliye child interface functional nahi hoga

**##Java ke andar bane banaye functional interface hote hain jaise consumer ,predicate,function,supplier hain**

**\*\*\*Consumer ke pass accept naam ka method hota hain jo kisi bhi type ki value accept krta hain**

**\*\*forEach() method main consumer type ka parameter ka bhejna padta hain**

**Aur for each iterable ka method hota hain aur iterable ek interface hota hain aur jo iterable method hota hain defulte non static method hota hain jiske parameter main consumer interface hota hain in static and default method ka kaam hota hian ki uske andar kisi interface ki implemention ko provide krke de de diya hain aur for each main ye diya gaya hain kaise hum traverse kraye element ko**

****

**StreamAPI:-**collection ko sort, duplicate number, random number and even number Nikalo to ye kaam hum stream se kr sakate ho Stream and lambda ka use krke kr sakate hain

* Stream per ek baar main ek hi activtity kr sakate ho
* Ye immutable feature hota hain
* Isme ek se jada activity isliy nahi kr sakate ho for example apne stream banake sorted kiya aur ab aap sorted ho gayi ab us per kuch aur operation kr rahe ek time per to nahi kr sakte ho kyuki
* Wo usi stream ko replace krke ek nahi stream bana le rahi hain
* We can say that old object ko hata ke naya object la de raha hain

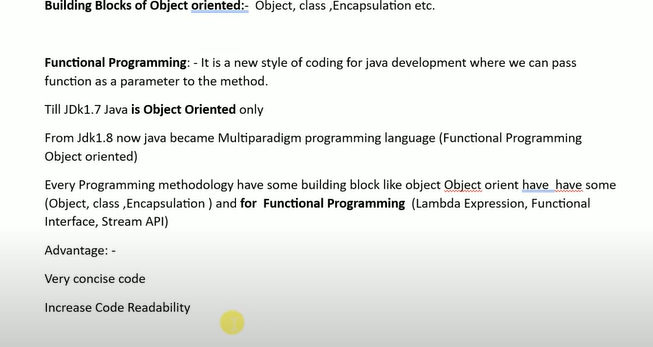
**Pradicate interface:** ye functional interface hota hain aur iske pass jo method hota hain uska return type **Boolean** hota hain aur ye **filter** ke parameter main jata hain hamesha

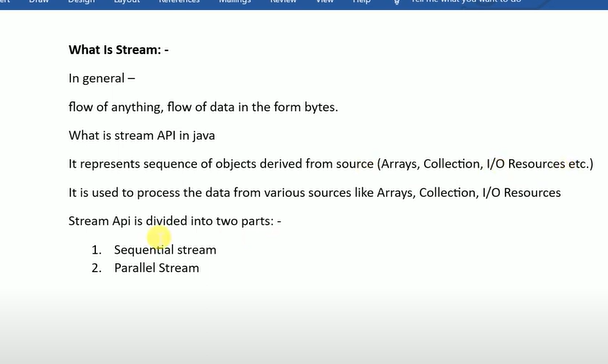
**Filter:**Relational operator ko perform krne ke liye hota hain

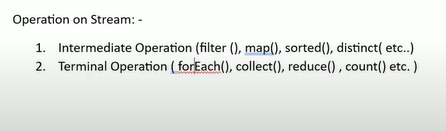
Relational operator Boolean ke liye hota hain

**Functional Programming:-**enable functional programming in this below –

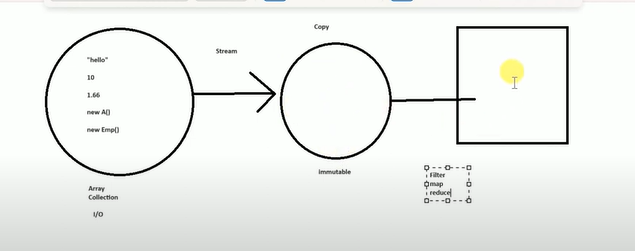
* StreamApi
* Functional Interface
* Lambda Expression







Terminal end point hota hain

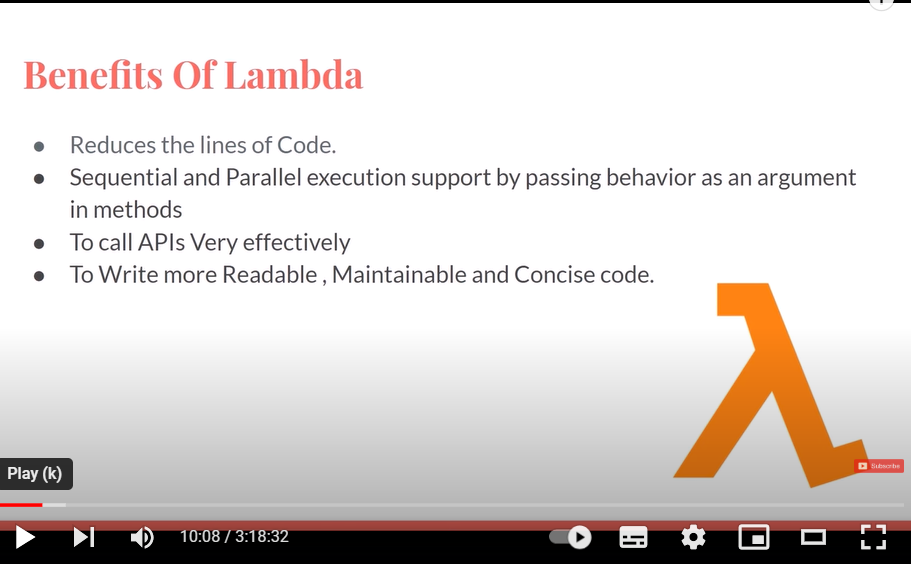
* **For jaise kisi collection se hum data ko nikale aur uske baad jo naya collection bana to wo immutable ho usme koi change nahi to hum stream ka use krenge to us per ek baar kewal ek hi operaion kr payenge**
* **Jaise maine stream kiya aur uske baad filter kiya ab main usi data per aap reduce nahi kr sakte hain ho ye jo operaion hoga wo sequential hoga**
* **Stream internal autoclosobale hoti hain isliye hum us per change nahi kr sakte hain**
* ****
* Double Sa=*employees*.stream().map(t->t.getSalary()).reduce((num1, num2)->num1+num2).get();
* System.***out***.println(Sa);
* reduce iska kaam hota hain pahale aapne sari salery nikali phir reduce kr deta hain aur do num1 and num2 phir aise hi 1,2 and 2,3 etc hoga

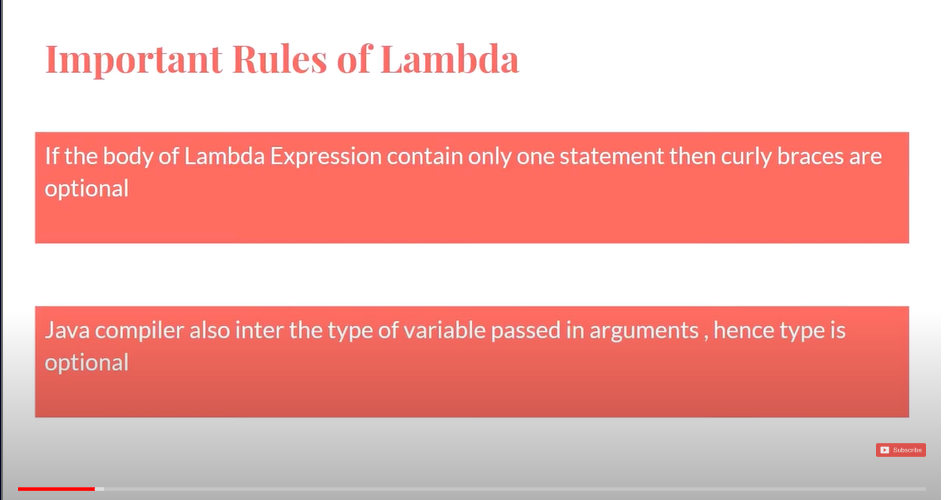
**What is difference b/w map and flatmap?**

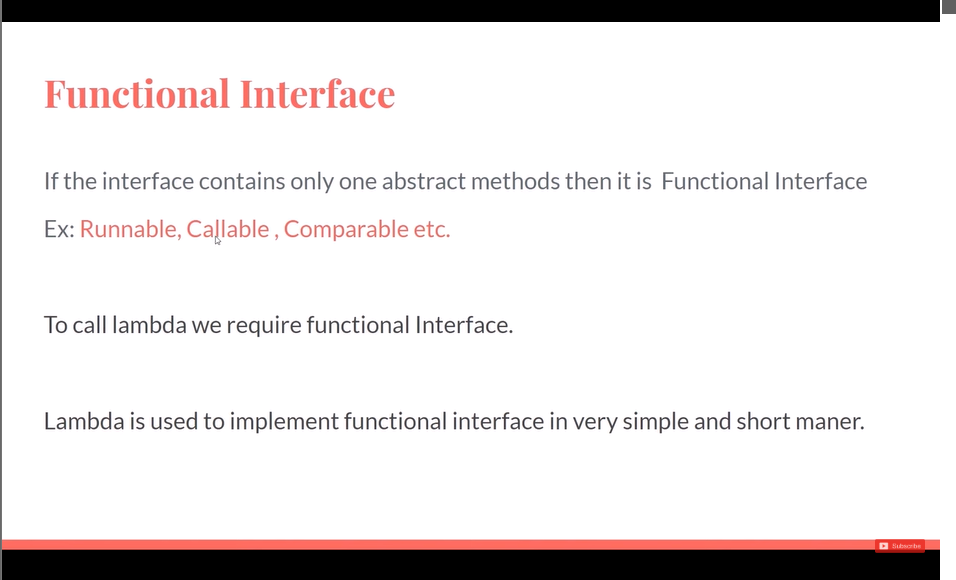
* Map list ke liye use hota hain -employee
* flatmap list ke andar further list ho to waha per use krte hain -employee ka

address

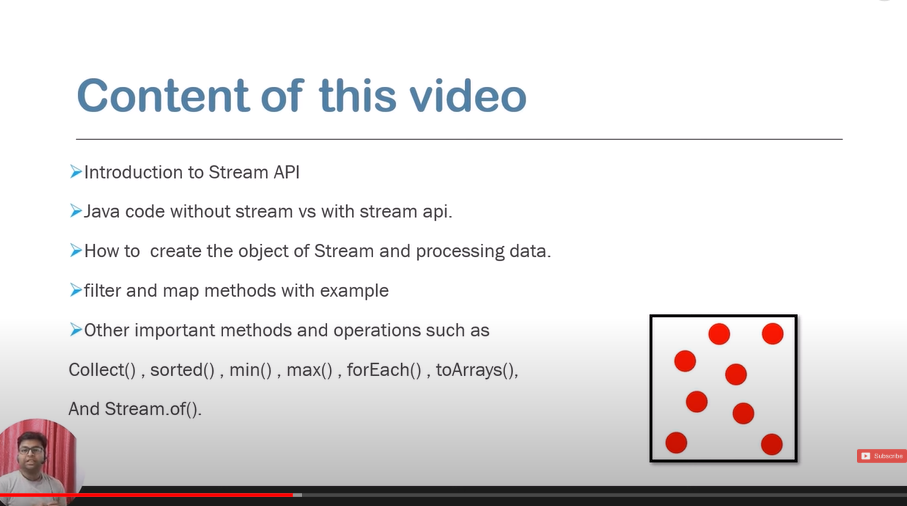
****

****

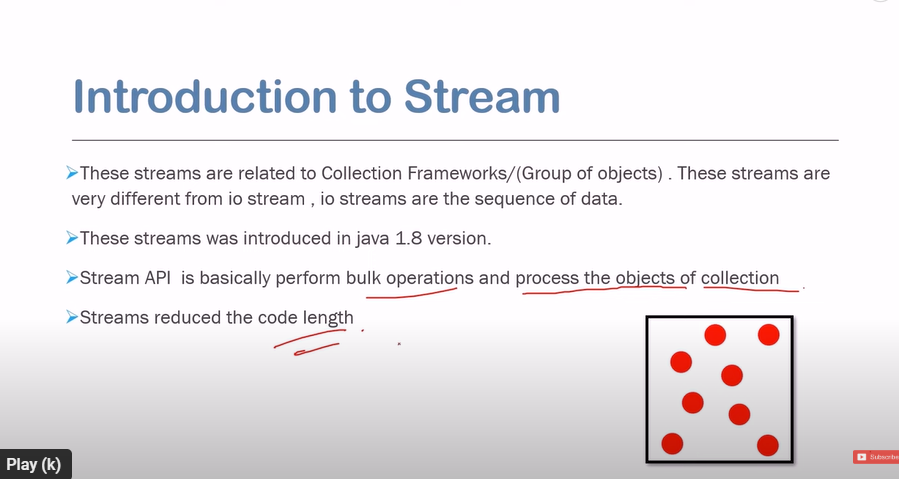
****

****

**StreamApi:-**

****

**Java 1.8 intruduced**

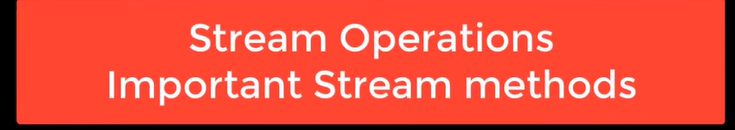
****

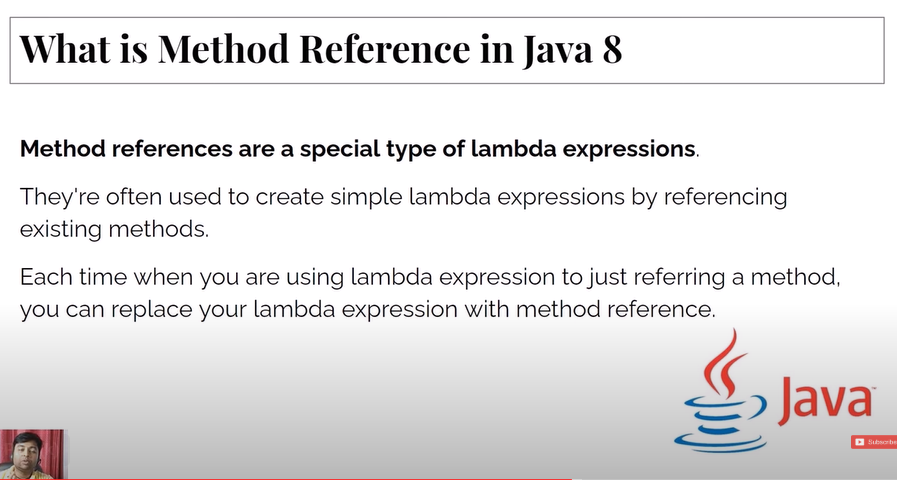
List<Integer>list1=List.of(4,5,6,7,9);*//imputable list hain*

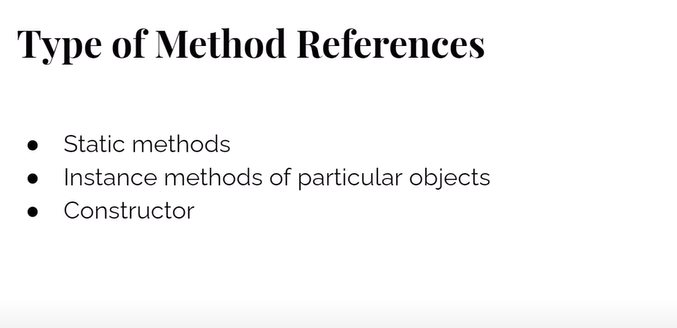
**Ye immutable list hain jisme hum kuch add nahi kr sakte hain**

**Stream is interface \*\*\*\***

**How to create object of stream**

****

****

****

**Note:-**jo method refere kr rahe ho aur jis interface ke liye refer kr rahe ho dono ke argument same ho ne chahiye and return type alag ho sakte hain

**OptionalExample:-**

Null pointer exception ko avoid and handle krne ke liye use kiya jata hain