##
##################################
Git:
======
Git is a free and open source distributed version control system.
Generating a new SSH key and adding it to the ssh-agent using git bash

install git-scm.com
Then we need to create account GithHub.com
*Create a Oraganization name to the account then create new repositort and name it.

**************************Commands Used**************
==============
How to access using SSH keys:

SSH steps key Generation:

Step1 :
ssh-keygen -t rsa -b 4096 -C (gmail id)

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin
$ ssh-keygen -t rsa -b 4096 -C vigneshpoojary198@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/vigne/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/vigne/.ssh/id_rsa
Your public key has been saved in /c/Users/vigne/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:mrKF5C5zAUgWsayrnwgxQI7O9go3qApw/E2OmfTel7E vigneshpoojary198@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
| +o              |
|=o.              |
|==               |
|*..              |
|=+o.o . S        |
|o*.=.X o   .     |
|* o.O.B     +    |
|*o++.= .   E     |
|*oo+o . ..       |
+----[SHA256]-----+
```

Then u will press enter and at default location shown there: (/c/Users/XXX/.ssh/id_rsa)

Step2:

start the ssh-agent in the background :

```
  $ eval $(ssh-agent -s)

  VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin
$ eval $(ssh-agent -s)
Agent pid 1173
```

Now add it to agent :

```
ssh-add ~/.ssh/id_rsa

VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin
$ ssh-add ~/.ssh/id_rsa
Identity added: /c/Users/vigne/.ssh/id_rsa (vigneshpoojary198@gmail.com)
```

After this we go to github --> settings || SSH and GPG keys | then create new SSH keys (title & Key)u need to add.
In SSH u need to add Public key ssh-rsa ***** ends with vigneshpoojary198@gmail.com.

LOCAL TO REMOTE REPOSITORY:
(ADMIN LEVEL)
=========================

$ git init
With this command we will initialise local directory as git repository. Till now our code is in working area

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin/ngx-admin-master
$ git init
Initialized empty Git repository in D:/Installed/git/admin/ngx-admin-master/.git/
```

$ git add .   -- with dot
# Adds the files in the local repository and stages them for commit. These files are prepared to be commited. So now files are moved from working area to staging area

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin/ngx-admin-master
(master)
$   git add .
```

$ git status
to see ur files. In staged, unstaged or untracked status. After commiting u will not see the file here.

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin/ngx-admin-master
(master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

$ git commit -m "First commit"
# Commits your tracked changes on your local repository and prepares them to be pushed to a remote repository. So now files moves from stage to local repo.

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/admin/ngx-admin-master
```

```
(master)
$ git commit -m "First commit"
[master (root-commit) b21ce4f] First commit
  Committer: Vignesh C <vigne@Vignesh-ST107.sakshath-technologies.com>

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
$ git status
On branch master
nothing to commit, working tree clean
```

git remote add origin git@github.com:Code-decode-learning/git-demo.git
  add url for remote repository in which your local repo will be pushed .

```
  VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
$ git remote add origin git@github.com:SoftwareDevelopmentCRUD/AngularSourceCode.git
```

git remote -v
  Verify new remote url.

```
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
$ git remote -v
origin    git@github.com:SoftwareDevelopmentCRUD/AngularSourceCode.git (fetch)
origin    git@github.com:SoftwareDevelopmentCRUD/AngularSourceCode.git (push)
```

$ git push 杣 origin master
  Pushes the changes in your local repository up to the remote repository you specified as the origin

```
  VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
$ git push -u origin master
Enumerating objects: 625, done.
Counting objects: 100% (625/625), done.
Delta compression using up to 8 threads
Compressing objects: 100% (612/612), done.
Writing objects: 100% (625/625), 3.47 MiB | 1.17 MiB/s, done.
Total 625 (delta 73), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (73/73), done.
To github.com:SoftwareDevelopmentCRUD/AngularSourceCode.git
  * [new branch]          master -> master
branch 'master' set up to track 'origin/master'.
```

U have to create Fork with user account name
=========================
REMOTE REPO TO LOCAL
(USER LEVEL)
============

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/user
$ git clone git@github.com:itsvigneshrp07/AngularSourceCode.git
Cloning into 'AngularSourceCode'...
```

```
remote: Enumerating objects: 625, done.
remote: Counting objects: 100% (625/625), done.
remote: Compressing objects: 100% (539/539), done.
remote: Total 625 (delta 73), reused 625 (delta 73), pack-reused 0
Receiving objects: 100% (625/625), 3.47 MiB | 1.74 MiB/s, done.
Resolving deltas: 100% (73/73), done.
```

Git and github tutorial for beginners that we cover here uses concept of forking created at github. git commands will be same for forked one and for organisational level repositories.

For forked repos :

Go to your .config file in .git and check upstream and origin.

Upstream should be from where u can take pull. Means your main repo of organisation.

Origin must be your forked where you will push your code and raise a pull request so that on approval those changes can be merged to upstream branch or organisation.

git remote add upstream (url).

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/user/AngularSourceCode (master)
$ git remote add upstream git@github.com:SoftwareDevelopmentCRUD/AngularSourceCode.git
```

git diff : It shows wt are changes we are done at working area. what changes were made in working area but not added in staging area.

git status : List which files are staged,unstaged and untracked.
((**unstaged and untracked files are always RED COLOR and staged filed will come in GREEN COLOR**))

```
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/user/AngularSourceCode (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md.bak

nothing added to commit but untracked files present (use "git add" to track)


VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/user/AngularSourceCode (master)
$ git add JavaCode.java.txt
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/user/AngularSourceCode (master)
```

```
$ git status

On branch master
Your branch is up to date with 'origin/master'.

Changes to be commited:
  (use "git restore –staged <file>..." to unstage)     //green color
        modified :JavaCode.java.txt

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git restore –staged JavaCode.java.txt
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git status


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git commit -m "my message that can be altered"
how can i revert commit message:
--------------------------------
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git commit --amend (hit enter it goes to notepad++ their you can change commit message
eg:my message that now altered)
It will change the   last commit message.
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git commit --amend
     my message that now altered (1 file is changed)

git log : Display the entire   commit history.
---------------
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git log
commit b21ce4f56bc068021196c22e73f60f540f1501ad (HEAD -> master, origin/master,
origin/HEAD)
Author: Vignesh C <vigne@Vignesh-ST107.sakshath-technologies.com>
Date:    Mon Feb 6 22:40:15 2023 +0530

     First commit


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git push origin master
Everything up-to-date


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git branch "myBranchName"

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
```

```
(master)
$ git checkout myBranchName
Switched to branch 'myBranchName'

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(myBranchName)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master) (local deletion of Branch)
$ git branch -d "myBranchName"
Deleted branch myBranchName (was b21ce4f).
```

//Rather then telling you in a 2 steps we can use this git checkout -b <Branch-name>commnand(directly is will created branced with name).

```
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git checkout -b newBranch
Switched to a new branch 'newBranch'
```

//Currenyly i pushed to my branch to remote repo.This is my current branch now with name newBranch -> newBranch

```
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(newBranch)
$ git push origin newBranch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newBranch' on GitHub by visiting:
remote:      https://github.com/itsvigneshrp07/AngularSourceCode/pull/new/newBranch
remote:
To github.com:itsvigneshrp07/AngularSourceCode.git
 * [new branch]      newBranch -> newBranch
```

//Deleted from remote repository

```
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(newBranch)   (Delete a particular remote Branch)
$ git push origin --delete newBranch
To github.com:itsvigneshrp07/AngularSourceCode.git
 - [deleted]         newBranch
```

//creating a new Branch (It is in a Local Branch)

```
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(newBranch)
$ git checkout -b develop
Switched to a new branch 'develop'
```

//Push Branch to remote Branch

```
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(develop)
$ git push origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
```

```
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:          https://github.com/itsvigneshrp07/AngularSourceCode/pull/new/develop
remote:
To github.com:itsvigneshrp07/AngularSourceCode.git
 * [new branch]          develop -> develop


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(develop)
$   git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$   git merge   master


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git merge master
Already up to date.
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(develop)
$   git checkout master
$   git status
$   git add JavaCode.java.txt
$   git commit -m "Third commit to user"
$   git push origin master
$   git pull
$   git push origin master
$   git checkout newBranchAgain
$   git rebase master


$   git checkout master
$   git pull upstream master
VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/user/AngularSourceCode
(master)
$ git pull upstream master
From github.com:SoftwareDevelopmentCRUD/AngularSourceCode
 * branch                master        -> FETCH_HEAD
 * [new branch]          master        -> upstream/master
Already up to date.
```

```
git remote add origin(url).
```

We have already seen:
Git init [repo name] : This command is used to start a new repository.
Git clone : used to obtain a repository from an existing URL into local repository.
Now:

Git add [file]   - This command adds a file to the staging area.
Git add .(dot) -   This command adds one or more to the staging area.

git diff: Show unstaged changes between your index and working directory.

git commit -m 擊 Type in the commit message]�
Usage : git commit 朼 m 搈 y message�

git status : List which files are staged, unstaged, and untracked.

Git log : Display the entire commit history using the default format. For customization see additional options.

git push origin $branchname:$remote_branchname
like --- git push origin master

Git pull -) this command will pull changes from default remote repository which is origin n not upstream, if wanna pull from upstream then use command git pull upstream master.

git fetch: fetches the changes from remote repository but will not affect your local so will not give u any merge confict,
The interesting thing about the fetch command is that it doesn't actually affect anything in your local repo. No working changes will be lost, and you'll see no direct affect on your local branches. This is because Git keeps fetched content separate from your own repo's content until it is merged in.

git fetch (remote-repo)
$ git merge FETCH_HEAD

So obviously the big difference between fetch and pull is that pull actually performs a fetch in addition to a merge.
##
######################################

##
##
 LOG4J:
 =====
* Logging : getting messages from application (Production server) like success, errors, warnings,..etc
* Moniter the process, find bugs from realtime environment.
*) Tools :
1. Log4J 2.x
2. Java Logging
3. Commons Logging
Repository , Service, Controller, Entity..etc

*) Log Messages can be stored at different places
Files, Database, Email , Console..etc
===
3 componets
1.Logger   (which classed u want to activate)

*) Logger : it is a base object in Log4J. It enables/activates
Logging/Moniter of a class/process.

```
class EmployeeController {
Logger log = LogManager.getLogger(EmployeeController.class);
}
```

*) Do not create Logger object for Entity/DTO/POJO classes.
*) Recomanded Controllers/Services..etc

--- Priority Methods ------------------------
These are pre-defined methods exist in Logger object
which prints messages based on type.

TRACE   :To find messages from multiple stages/env/apps.
DEBUG   :To print final result of a process/ internal details
INFO   :Current Step
WARN :App related messages
ERROR :To give details about exception
FATAL :Environment related issues
    ex:Database Connection is closed.
order of elements:

    TRACE > DEBUG > INFO > WARN > ERROR > FATAL

==
2.Appender (where to store log messages)
  Appender:- Location of messages to be stored. (classes)
1. ***   FileAppender : Write data to Log File (__.log)
2. ConsoleAppender : Prints data at console
3. JDBCAppender : Store messages at Database table
4. SMTPAppender : Write messages to Email.

*) 1 Project can have even multiple appenders.
==
3.Layout   (How you want to print)
Layout: Message Format/Pattern.
1. Default Layout (Message and NextLine)
2. HTML Layout (___.html)
3. XML Layout (___.xml)
4.*** PatternLayout (Your own format)
====================
program eg:
=========
pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <groupId>com.app.raghu</groupId>
        <artifactId>Log4J2ExamplesEx</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <properties>
```

```
		<maven.compiler.source>1.8</maven.compiler.source>
		<maven.compiler.target>1.8</maven.compiler.target>
	</properties>

	<dependencies>
		<dependency>
			<groupId>org.apache.logging.log4j</groupId>
			<artifactId>log4j-api</artifactId>
			<version>2.11.2</version>
		</dependency>
		<dependency>
			<groupId>org.apache.logging.log4j</groupId>
			<artifactId>log4j-core</artifactId>
			<version>2.11.2</version>
		</dependency>

	</dependencies>

</project>
```

Test.java (src/main/java)

```
package com.app.raghu;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

public class Test {

	private static final Logger log = LogManager.getLogger(Test.class);

	public static void main(String[] args) {
		processData();
	}

	public static void processData() {
		log.trace("FROM TRACE METHOD");
		log.debug("FROM DEBUG METHOD");
		log.info("FROM INFO METHOD");
		log.warn("FROM WARN METHOD");
		log.error("FROM ERROR METHOD");
		log.fatal("FROM FATAL METHOD");
		// .. read inputs
		// .. validate them
		// .. store in db
		// .. return results

	}
}
```
----

log4J2.xml (src/main/resouces)

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
```

```
	<Appenders>
		<Console name="LogToConsole" target="SYSTEM_OUT">
			<PatternLayout>
				<Pattern>%d - %C [%M] -(%p) %m %n</Pattern>
			</PatternLayout>
		</Console>
		<File name="LogToFile" fileName="logs/myapp.log">
			<PatternLayout>
				<Pattern>%d - %C [%M] -(%p) %m %n</Pattern>
			</PatternLayout>
		</File>
	</Appenders>
	<Loggers>
		<Root level="trace">
			<AppenderRef ref="LogToConsole" />
			<AppenderRef ref="LogToFile" />
		</Root>
	</Loggers>
</Configuration>
```

==

D:\Workspace_Eclipse\Log4J2ExamplesEx\logs
=========
myapp.log

```
2023-02-08 12:29:44,942 - com.app.raghu.Test [processData] -(INFO) FROM INFO METHOD
2023-02-08 12:29:44,945 - com.app.raghu.Test [processData] -(WARN) FROM WARN METHOD
2023-02-08 12:29:44,945 - com.app.raghu.Test [processData] -(ERROR) FROM ERROR METHOD
2023-02-08 12:29:44,945 - com.app.raghu.Test [processData] -(FATAL) FROM FATAL METHOD
2023-02-08 12:32:38,325 - com.app.raghu.Test [processData] -(TRACE) FROM TRACE METHOD
2023-02-08 12:32:38,327 - com.app.raghu.Test [processData] -(DEBUG) FROM DEBUG METHOD
2023-02-08 12:32:38,328 - com.app.raghu.Test [processData] -(INFO) FROM INFO METHOD
2023-02-08 12:32:38,328 - com.app.raghu.Test [processData] -(WARN) FROM WARN METHOD
2023-02-08 12:32:38,328 - com.app.raghu.Test [processData] -(ERROR) FROM ERROR METHOD
2023-02-08 12:32:38,338 - com.app.raghu.Test [processData] -(FATAL) FROM FATAL METHOD
```

##
######################

STEPS:
=====
(SSH KEYS STEPS FIRST TIME)

```
1)**	C:\Users\vigne\Desktop\Documents\web>ssh-keygen -t rsa -b 4096 -C vigneshpoojary198@gmail.com
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\vigne/.ssh/id_rsa):
C:\Users\vigne/.ssh/id_rsa already exists.
Overwrite (y/n)? n

2) ** $ eval $(ssh-agent -s)
** $ ssh-add ~/.ssh/id_rsa

  3) ** C:\Users\vigne\Desktop\Documents\web>git init
Initialized empty Git repository in C:/Users/vigne/Desktop/Documents/web/.git/
```

```
  4)** C:\Users\vigne\Desktop\Documents\web>git add .

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
5) ** $ git status

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
6)** $ git commit -m "First commit"

  VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
7)** $ git remote add origin git@github.com:SoftwareDevelopmentCRUD/AngularSourceCode.git

VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
8)** $ git remote -v


VIGNESH-ST107+vigne@Vignesh-ST107    MINGW64    /d/Installed/git/admin/ngx-admin-master
(master)
9)** $ git push -u origin master


=============================
Remote to Local
------------------
VIGNESH-ST107+vigne@Vignesh-ST107 MINGW64 /d/Installed/git/user
$ git clone git@github.com:itsvigneshrp07/AngularSourceCode.git
```