

TOPIC 1: INTRODUCTION TO VERSION CONTROL SYSTEM

Introduction

Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds for Linux kernel development. Git is a free software distributed under the terms of the GNU General Public License version 2.

Objectives

Objectives by the end of this topic you should be able to:

- Install and configure git on windows/linux
- Configure local and remote repository
- Use git with real project
- Create a github account

Learning activities

Learning Activity 1.1: Reading

Read more on the history of version control system.

Learning Activity 1.2: Journal

Summarize important git commands.

Learning Activity 1.3: Discussion

In groups of three discuss in detail how to push code to a remote repository with different branches.

Assessment

Topic resources

1. Pipinellis, A. (2018). *GitHub essentials: unleash the power of collaborative development workflows using GitHub*. Birmingham, U.K.: Packt Publishing Limited.
2. Bell, P., & Beer, B. (2014). *Introducing GitHub: a non-technical guide*. Beijing: O'Reilly.
3. Loeliger, J., & McCullough, M. (2012). *Version control with Git*. Beijing: O'Reilly.
4. Tsitoara, M. (2020). *Beginning Git and GitHub: a comprehensive guide to version control, project management, and teamwork for the new developer*. New York, NY: Apress.
5. Loeliger, J. (2009). *Version control using Git*. O'Reilly: Cambridge.

URL Links

<https://www.geeksforgeeks.org/ultimate-guide-git-github/>

<https://www.geeksforgeeks.org/list-useful-github-commands/?ref=rp>

<https://www.tutorialspoint.com/git/index.htm>

<https://www.geeksforgeeks.org/ultimate-guide-git-github/?ref=rp>

<https://www.geeksforgeeks.org/top-10-useful-github-repos-that-every-developer-should-follow/?ref=rp>

<https://www.geeksforgeeks.org/5-github-repositories-that-every-new-developer-must-follow/?ref=rp>

TOPIC 1 NOTES

Nowadays software development takes place in a distributive way. This topic focuses on one such technology that supports distributed software development i.e GIT.

What GIT is about?

- Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- Git relies on the basis of distributed development of a software where more than one developer may have access to the source code of a specific application and can modify changes to it which may be seen by other developers..
- Initially designed and developed by Linus Torvalds for Linux kernel development in 2005.
- Every git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server.
- Git allows a team of people to work together, all using the same files. And it helps the team cope up with the confusion that tends to happen when multiple people are editing the same files.

Why Use Version Control Software?

- Version control software allows the user to have “versions” of a project, which show the changes that were made to the code over time, and allows the user to backtrack if necessary and undo those changes.
- This ability alone – of being able to compare two versions or reverse changes, makes it fairly invaluable when working on larger projects.
- In a version control system, the changes would be saved just in time – a patch file that could be applied to one version, in order to make it the same as the next version.
- All versions are stored on a central server, and individual developers checkout and upload changes back to this server.

Characteristics of Git

1. Strong support for non-linear development

- Git supports rapid branching and merging, and includes specific tools for visualizing and navigating a non-linear development history.
- A major assumption in Git is that a change will be merged more often than it is written.

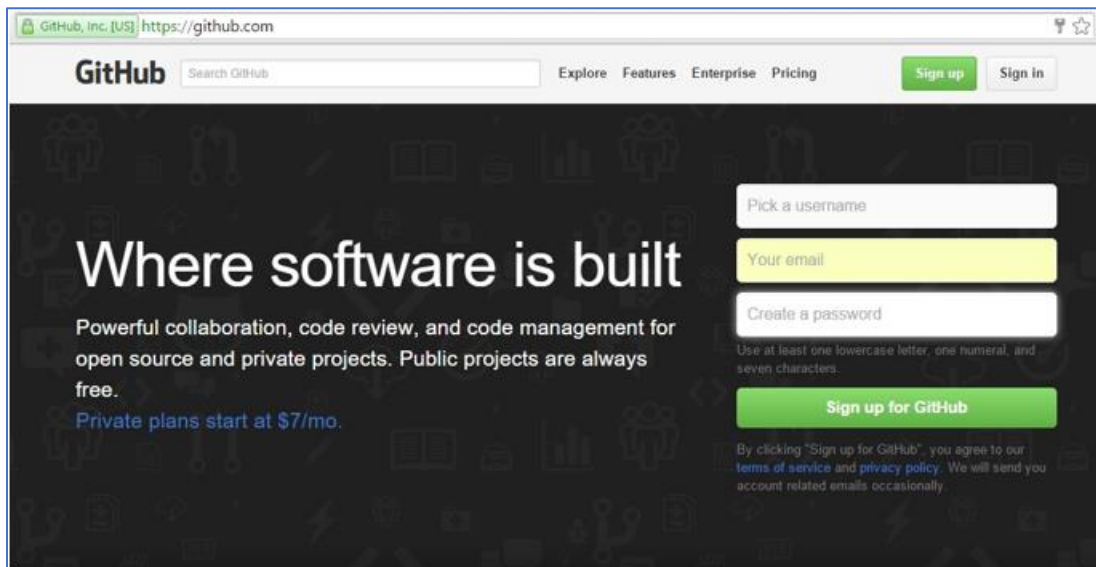
- Branches in Git are very lightweight.
- 2. **Distributed development**
 - Git **provides** each developer a **local copy** of the entire development history, and changes are copied from one such repository to another.
 - The changes can be merged in the same way as a locally developed branch very efficiently and effectively.
- 3. **Compatibility with existing systems/protocol**
 - Git has a **CVS** server emulation, which enables the use of existing **CVS** clients and IDE plugins to access Git repositories.
- 4. **Efficient handling of large projects**
 - Git is very fast and scalable compared to other version control systems.
 - The fetching power from a local repository is much faster than what possible with remote server.
- 5. **Data Assurance**
 - The Git history is stored in such a way that the ID of a particular version depends upon the complete development history leading up to that commit.
 - Once published, it is not possible to change the old versions without it being noticed.
- 6. **Automatic Garbage Collection**
 - Git automatically performs **garbage collection** when enough loose objects have been created in the repository.
 - Garbage collection can be called explicitly using `git gc --prune`.
- 7. **Periodic explicit object packing**
 - Git stores each newly created object as a separate file. It uses *packs* that store a large number of objects in a single file (or network byte stream) called packfile, delta-**compressed** among themselves.
 - A corresponding index file is created for each pack file, **specifying** the **offset** of each object in the packfile.
 - The process of **packing** can be very **expensive computationally**.
 - Git allows the expensive pack operation to be deferred until later when time does not matter.
 - Git **does periodic repacking automatically** but manual repacking can be done with the `git gc` command.

How GIT Works

1. A Git repository is a **key-value** object store where all objects are indexed by their SHA-1 hash value.
2. All commits, files, tags and filesystem tree nodes are different types of objects living in this repository.
3. A Git repository is a large **hash table** with no provision made for hash collisions.
4. Git specifically works by taking “snapshots” of files

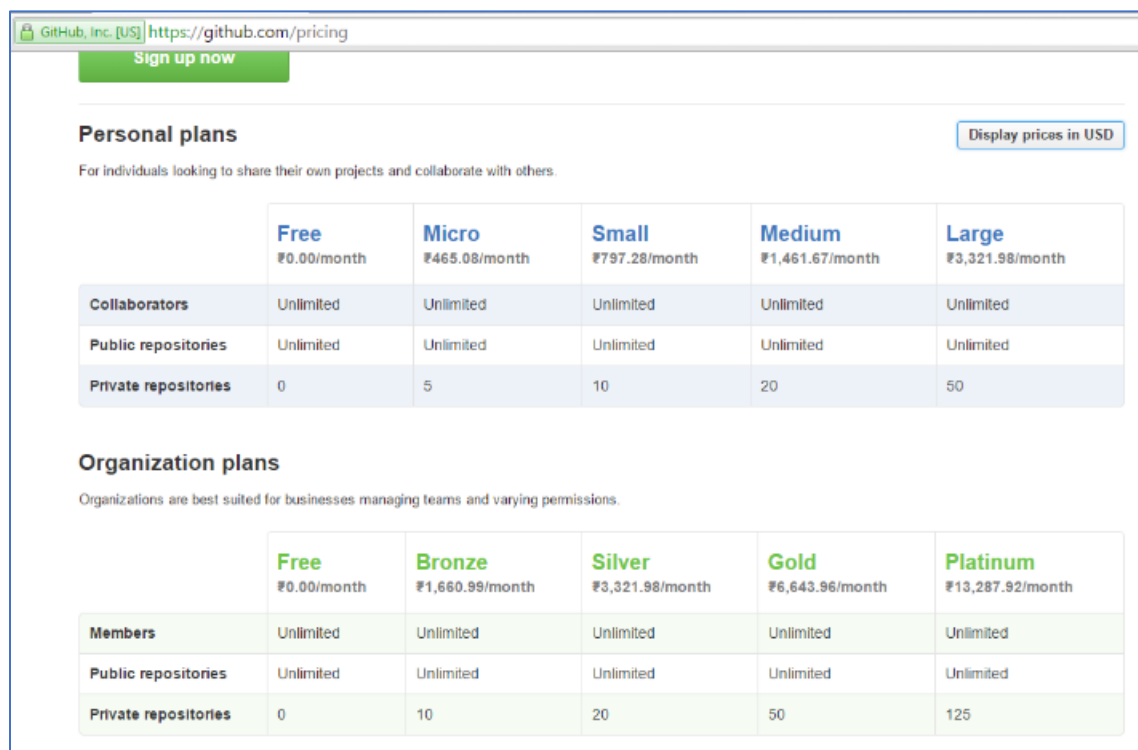
Creating a GitHub Account:

1. Go to github.com and enter the required user credentials asked on the site and then click on SignUp for GitHub button.



The screenshot shows the GitHub homepage with a dark background. The main heading is "Where software is built". Below it, the text says "Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free." and "Private plans start at \$7/mo." On the right side, there is a sign-up form with fields for "Pick a username", "Your email", and "Create a password". Below the password field, there is a note: "Use at least one lowercase letter, one numeral, and seven characters." A green "Sign up for GitHub" button is at the bottom of the form. At the very bottom, there is a small disclaimer: "By clicking 'Sign up for GitHub', you agree to our terms of service and privacy policy. We will send you account related emails occasionally."

2. Choose a plan that best suits you. The following plans are available:-



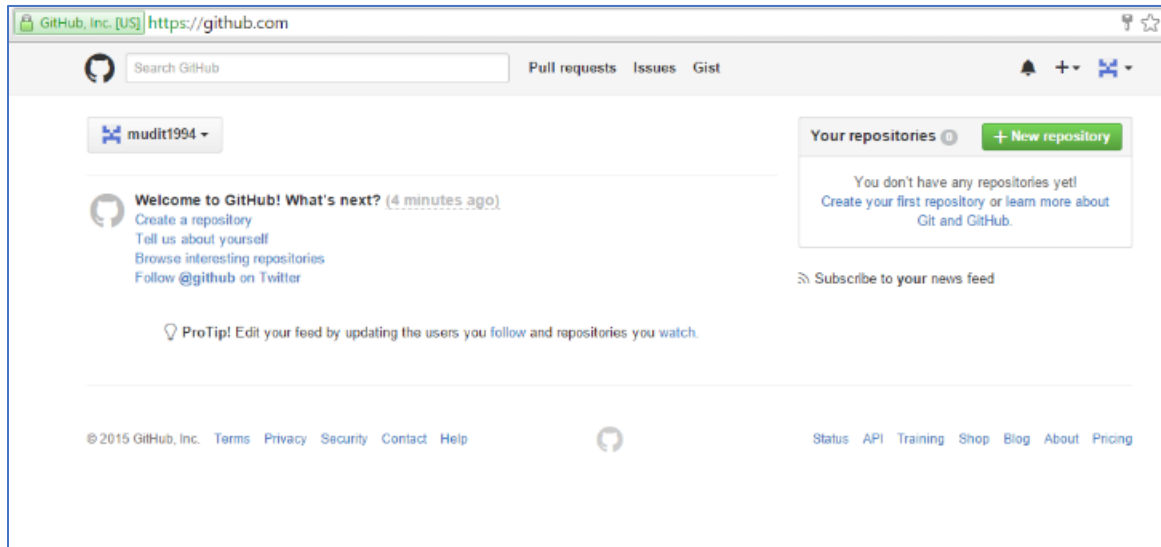
The screenshot shows the GitHub pricing page. At the top, there is a "Sign up now" button. Below it, the "Personal plans" section is highlighted. It includes a sub-header "For individuals looking to share their own projects and collaborate with others." and a table with five plans: Free, Micro, Small, Medium, and Large. The prices are listed in Indian Rupees (₹). To the right of the table, there is a button "Display prices in USD". Below the personal plans, the "Organization plans" section is shown. It includes a sub-header "Organizations are best suited for businesses managing teams and varying permissions." and a table with five plans: Free, Bronze, Silver, Gold, and Platinum. The prices are also listed in Indian Rupees (₹).

	Free ₹0.00/month	Micro ₹465.08/month	Small ₹797.28/month	Medium ₹1,461.67/month	Large ₹3,321.98/month
Collaborators	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Public repositories	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Private repositories	0	5	10	20	50

	Free ₹0.00/month	Bronze ₹1,660.99/month	Silver ₹3,321.98/month	Gold ₹6,643.96/month	Platinum ₹13,287.92/month
Members	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Public repositories	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Private repositories	0	10	20	50	125

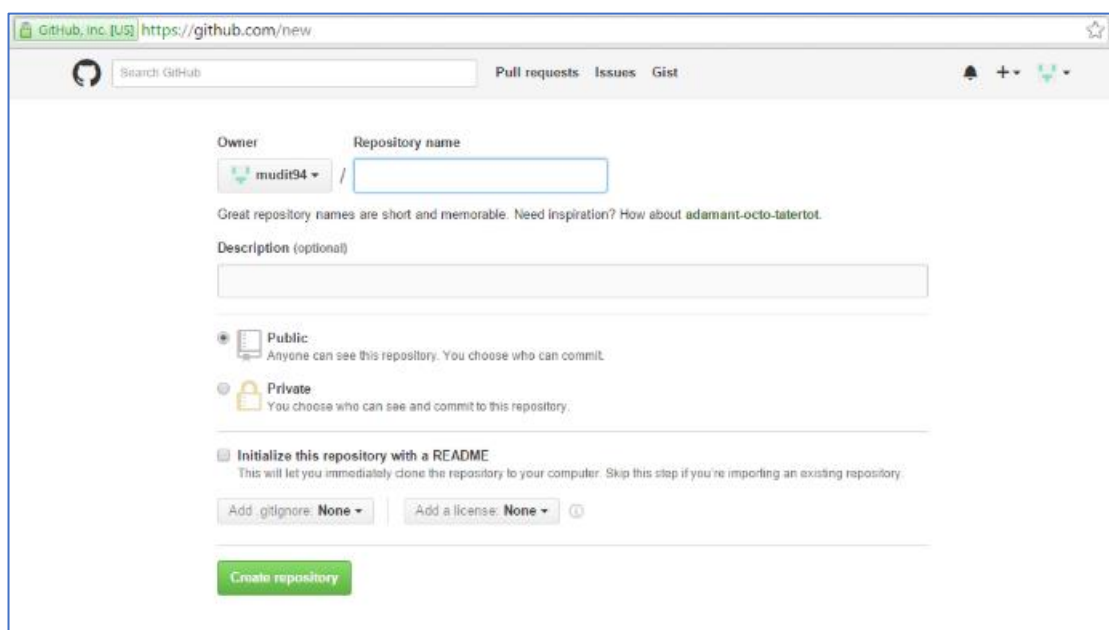
Then Click on Finish Sign Up.

The account has been created. The user is automatically redirected to your Dashboard.

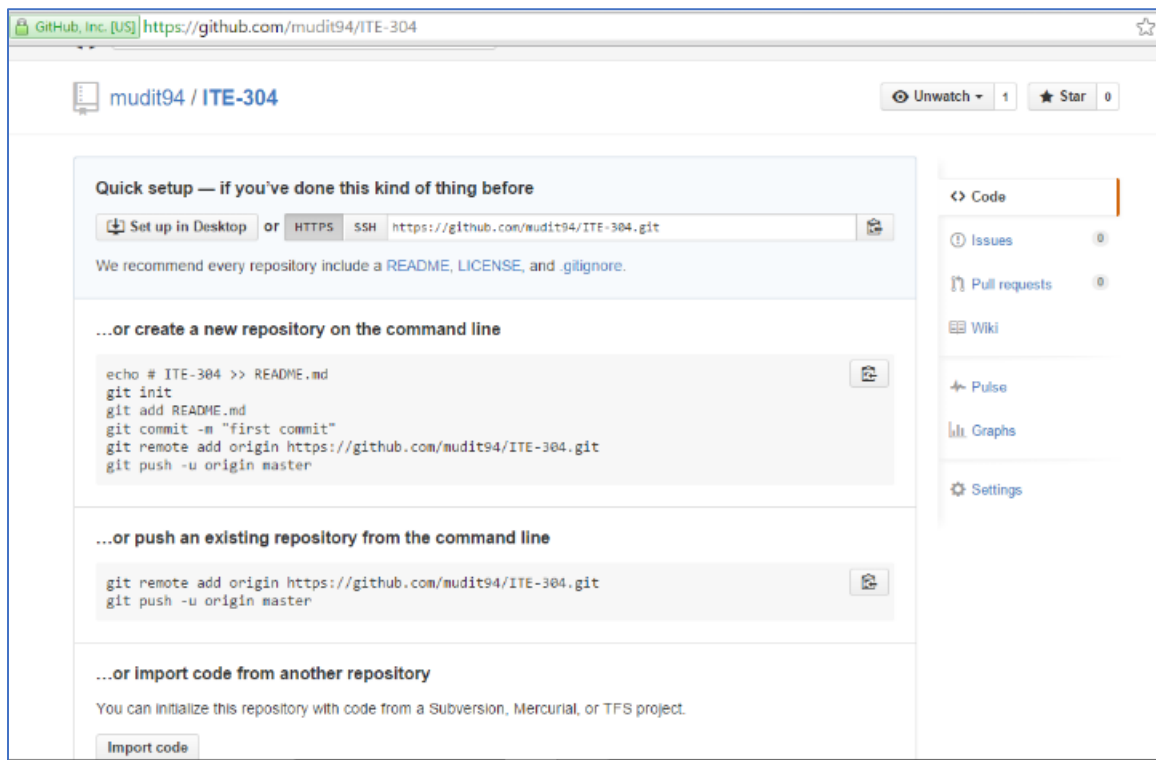


Creating a new Repository

1. Login to your Git Hub account
2. On the dashboard click on the Green Button stating New repository.
3. Make sure to **verify the Git hub account** by going into the mail which was provided when creating the account.
4. Once verification has been done, the following screen comes



5. Start by giving a repository name, description(optional) and select the visibility and accessibility mode for the repository
6. **Click** on Create repository
7. The repository (in this case ITE-304 is the repository) is now created. Repository can be created



Uploading existing le to GitHub

1. The **system should have git installed** in it. For installing git please refer <https://git-scm.com/downloads>. Make sure to choose Run git from Windows Command prompt option during installation. Otherwise open git bash in place of step 2.
2. Open Terminal (for Mac users) or the command prompt (for Windows and Linux users).
3. Change the current working directory to your local project

4. Initialize the local directory as a git repository in different ways as described in the image.

```
$ git init
```

A new **.git folder** is **created** in the directory which is by **default hidden**.

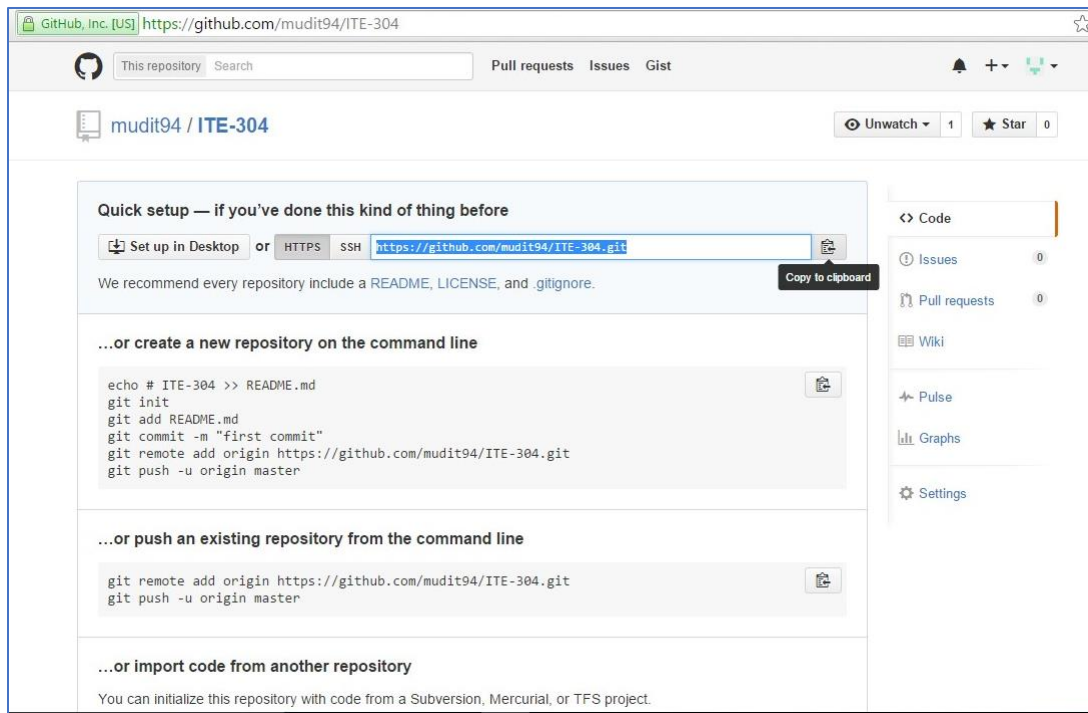
Add the les in your new local repository. This stages them for the rst commit.

```
$ git add .  
# Adds the files in the local repository and stages them for commit. To unstage a file use  
git reset HEAD <filename>
```

Commit the les that you've staged in your local repository.

```
$ git commit --m "first commit"  
#commits the tracked changes and prepares them to be pushed to a remote repository. to remove  
this commit and modify the file use  
$ git reset --soft head~1  
#and commit and add the file again
```

At the top of the GitHub repository's Quick Setup page, click on the icon shown and copy the remote repository URL.



In the Command prompt, **add the URL for the remote repository** where your local repository will be pushed.

```
$ git remote add origin <remote repository url>
```

#connects to the remote repository

```
$ git remote -v
```

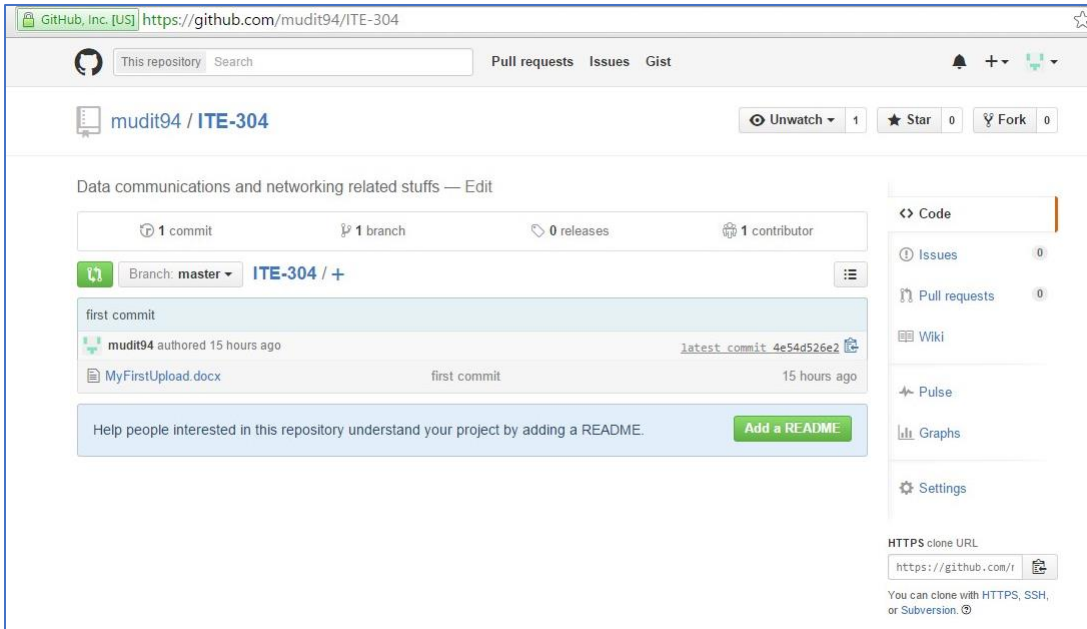
#verifies the new remote URL

Push the changes in your local repository to **GitHub**.

```
$ git push origin master
```

#pushes the changes in your local repository up to the remote repository you specified as the origin

And here you go...



Git Commands

1. An Ultimate Guide to Git and Github (<https://www.geeksforgeeks.org/ultimate-guide-git-github/>)
2. List of useful Github Commands (<https://www.geeksforgeeks.org/list-useful-github-commands/>)

Revision questions

1. What is GIT stash?
2. How can you create a repository in Git?
3. What is a repository in GIT?
4. (T/F) Pull requests should be fully functional and not contain any bugs before getting teammates to look through them.
5. (T/F) Git stores the history of changes made to the codebase over time, and information about who made those changes
6. (multiple choice) What is a branch?
 - a. A pointer to a specific commit.
 - b. A link between the local and remote histories.

- c. c. The centralized location where repositories are stored.
 - d. d. A version of a file at a specific time.
- 7. (multiple choice) Which of the following commands will create a new branch?
 - a. a. git checkout new-branch
 - b. b. git checkout -b new-branch
 - c. c. git clone new-branch
 - d. d. git create-branch new-branch
- 8. (T/F) Staging, or git add, is required before creating a commit.
- 9. (multiple choice) Which of the following commands will allow you to change branches?
 - a. git checkout
 - b. git clone
 - c. git add
 - d. git comm

TOPIC 2: PYTHON 3 BASICS AND ENVIRONMENT SETUP

Introduction.

Python 3.0 was released in 2008. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Objectives

Objectives by the end of this topic you should be able to:

- Download and install python 3 on:
 - ✓ Windows
 - ✓ Linux
 - ✓ MacOSx

Learning activities

Learning Activity 2.1: Reading

Read further on python 3 documentation.

Learning Activity 2.2: Journal

Describe procedures of installing python on linux and windows.

Learning Activity 2.3: Discussion

In groups of three discuss the differences between python 2 and python 3.

Assessment

Topic resources

1. The Python Tutorial¶. (n.d.). Retrieved from <https://docs.python.org/3/tutorial/index.html>
2. Mueller, J. P. (n.d.). *Beginning Programming with Python For Dummies*. S.I.: For Dummies.
3. (n.d.). Python 3.7.4 documentation. Retrieved from <https://docs.python.org/3>
4. (n.d.). Git Handbook. Retrieved from <https://guides.github.com/introduction/git-handbook/>
5. Shaw, Z. (2017). *Learn Python 3 the hard way: a very simple introduction to the terrifyingly beautiful world of computers and code*. Boston: Addison-Wesley.
6. Bader, D. (2018). *Python tricks: the book*. Vancouver, BC: Dan Bader.
7. Downey, A. B. (2015). *Think Python*. Sebastopol: O'Reilly.
8. Ramalho, L. (2016). *Fluent Python*:Beijing: O'Reilly.

URL Links

<https://docs.python.org/3/tutorial/index.html>

<https://www.tutorialspoint.com/python3/index.htm>

<https://pythonprogramming.net/introduction-learn-python-3-tutorials/>

<https://www.w3schools.com/python/>

<https://docs.python.org/3/tutorial/index.html>

TOPIC 2 NOTES

Python was developed by Guido van Rossum in the early 1990s and its latest version is 3.8.1, we can simply call it as Python3.

To understand how to install Python You need to know *What Python is* and where it is actually installed in your system.

Let's consider a few points:

- Python is a widely-used general-purpose, high-level programming language.
- Every Release of Python is open-source. Python releases have also been GPL-compatible.
- Any version of Python can be downloaded from **Python Software Foundation** website at python.org.
- Most of the languages, notably Linux provide a package manager through which you can directly install Python on your Operating System

In this Python tutorial of Installation and Setup, you'll see how to install Python on Windows, macOS, Linux, iOS, and Android.

Python Latest Version Installation and Setup

Here you can choose your OS and see the corresponding tutorial,

- [Windows](#)
- [Linux](#)
- [macOS / Mac OS X](#)
- [Android](#)
- [iOS \(iPhone / iPad\)](#)
- [Online Interpreters of Python](#)

How to install Python on Windows?

Since windows don't come with Python preinstalled, it needs to be installed explicitly. Here we will define step by step tutorial on How to install Python on Windows.

Follow the steps below :

Download Python Latest Version from python.org

- First and foremost step is to open a browser and open <https://www.python.org/downloads/windows/>

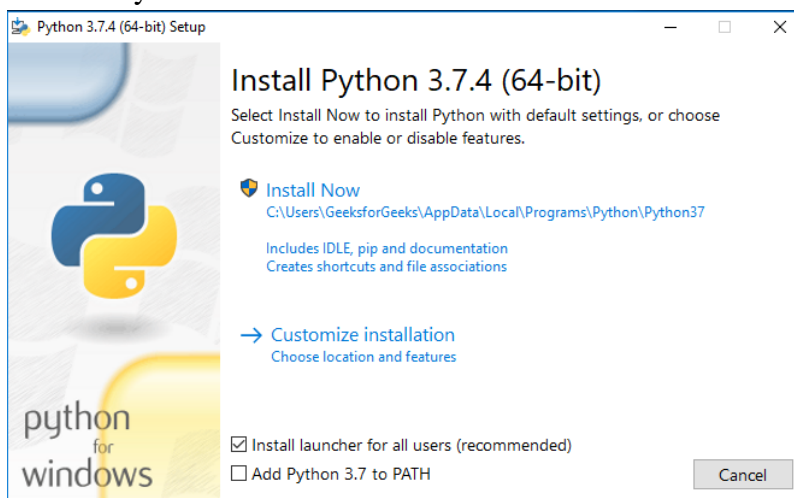


- Underneath the **Python Releases for Windows** find **Latest Python 3 Release – Python 3.7.4** (latest stable release as of now is Python 3.7.4).
- On this page move to **Files** and click on **Windows x86-64 executable installer** for 64-bit or **Windows x86 executable installer** for 32-bit.

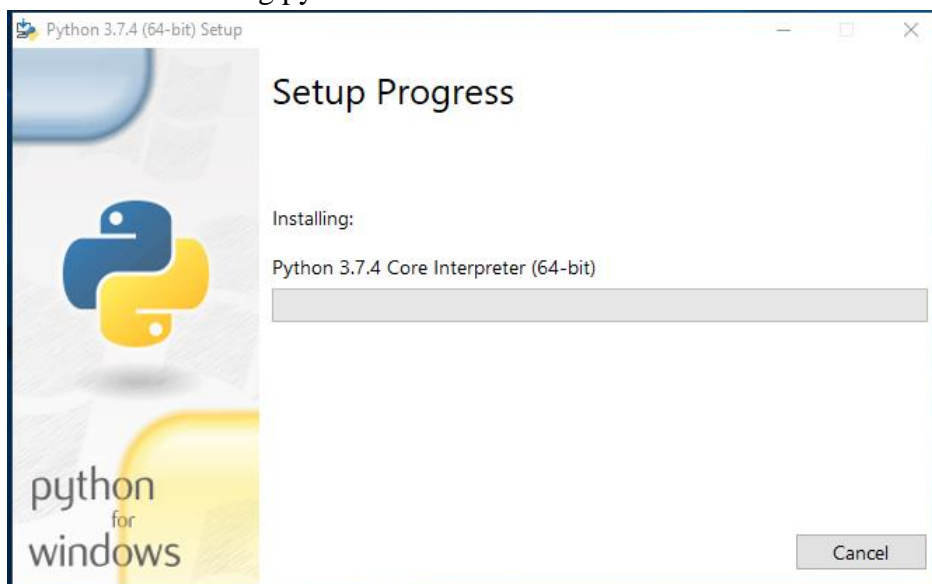
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Install Python 3.7.4 Latest Version on Windows

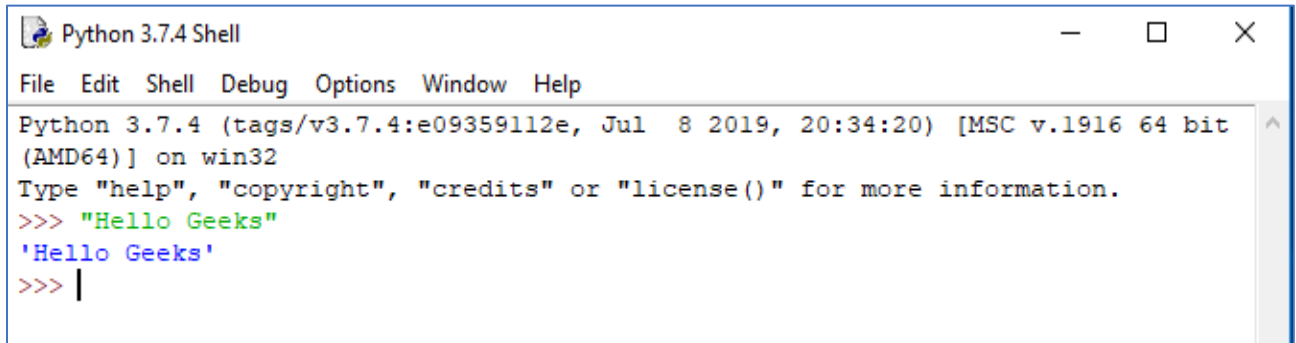
- Run the Python Installer from downloads folder



- Make sure to mark **Add Python 3.7 to PATH** otherwise you will have to do it explicitly. It will start installing python on windows.



- After installation is complete click on **Close**.
Bingo..!! Python is installed. Now go to windows and type IDLE.

A screenshot of a Windows application window titled "Python 3.7.4 Shell". The window has a standard menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content: "Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32", followed by "Type 'help', 'copyright', 'credits' or 'license()' for more information." Then, a prompt ">>>" is followed by the input string "Hello Geeks" in green text. The output "'Hello Geeks'" is shown in blue text. A second prompt ">>>" is followed by a vertical bar cursor "|".

```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> "Hello Geeks"
'Hello Geeks'
>>> |
```

This is Python Interpreter. I printed Hello geeks, python is working smoothly.

How to install Python on Linux?

On every linux system including following OS,

- ☐ Ubuntu
- ☐ Linux Mint
- ☐ Debian
- ☐ openSUSE
- ☐ CentOS
- ☐ Fedora
- ☐ Arch Linux.

You will find Python already installed. You can check it using the following command from the terminal

```
$ python --version
```

To check latest version of python 2.x.x :

```
$ python2 --version
```

To check latest version of python 3.x.x :

```
$ python3 --version
```

```
File Edit View Search Terminal Help
naveen@naveen-HP-Notebook:~$ python --version
Python 2.7.15+
naveen@naveen-HP-Notebook:~$ _
```

Clearly it won't be the latest version of python. There can be multiple methods to install python on a linux base system and it all depends on your linux system. For almost every Linux system, the following commands would work definitely.

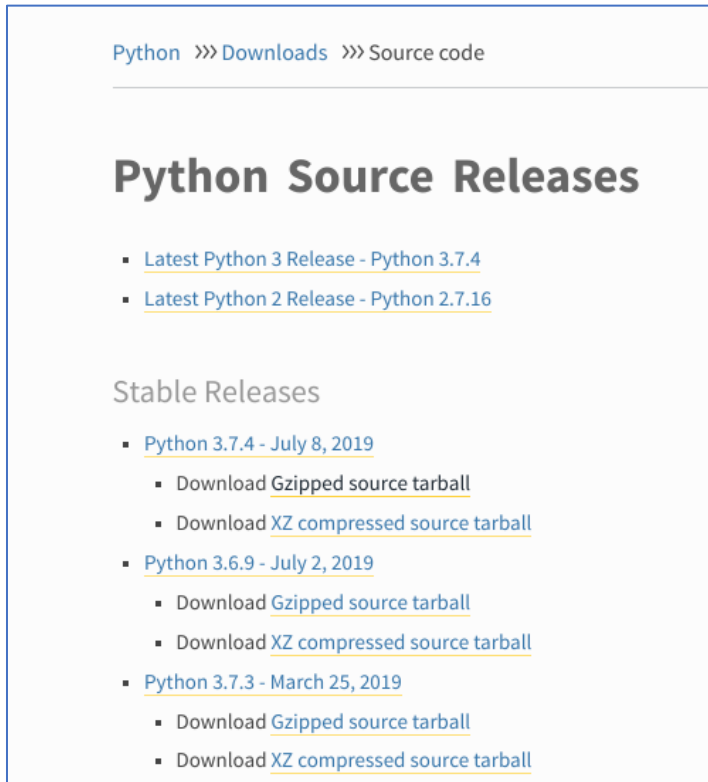
```
$ sudo add-apt-repository ppa:deadsnakes/ppa
$ sudo apt-get update
$ sudo apt-get install python3.7
```

Download and install Python Latest Version on Linux

To install the latest version from source code of Python follow below steps

Download Python Latest Version from python.org

- First and foremost step is to open a browser and open <https://www.python.org/downloads/source/>



- Underneath the **Stable Releases** find **Download Gzipped source tarball** (latest stable release as of now is Python 3.7.4).

You can do all the above steps in a single command

```
$ wget https://www.python.org/ftp/python/3.7.4/Python-3.7.4.tgz
```

Install Python 3.7.4 Latest Version on Linux

For installing Python successfully on Linux, Enter Following command to get the prerequisites and other source files

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

```
$ sudo apt-get install -y make build-essential libssl-dev zlib1g-dev libbz2-dev libreadline-dev  
libsqlite3-dev wget curl llvm libncurses5-dev libncursesw5-dev xz-utils tk-dev
```

Now we are all ready to unpack the file downloaded from the python official website'

Move to downloads directory using cd downloads in terminal and then enter following commands

```
$ tar xvf Python-3.6.5.tgz
```



```
$ cd Python-3.6.5
$ ./configure --enable-optimizations --with-ensurepip=install
$ make -j 8
$ sudo make altinstall
```

Bingo..!! The latest version of Python language is installed on your Linux system. You can confirm it using the below command.

```
python --version
```

How to install Python on macOS / Mac OS X ?

Like Linux, macOS also comes with Python pre-installed on the system. It might be Python version 2 or some similar outdated version. To update to the latest version, we will use the Homebrew Package manager. It is one of the best and convenient methods to install Python on macOS.

To know more about Homebrew Package manager, [visit here](#)

- **Download and install Homebrew Package Manager**

If you don't have homebrew installed on your system, follow the steps below

Open the Terminal Application of macOS from Application -> Utilities. Bash terminal will open where you can enter commands

Enter following command in macOS terminal

```
/usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```

sierra — ruby -e #!/usr/bin/ruby\012# This script installs to /usr/local only. To install elsewhere (which is\012# unsupported) you can
Last login: Mon Sep  9 22:28:02 on console
Sierras-MBP:~ sierra$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following existing directories will be made group writable:
/usr/local/lib
==> The following existing directories will have their owner set to sierra:
/usr/local/lib
==> The following new directories will be created:
/usr/local/bin
/usr/local/etc
/usr/local/include
/usr/local/sbin
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks
==> The Xcode Command Line Tools will be installed.

Press RETURN to continue or any other key to abort

```

Enter the system password if prompted. This will install the Homebrew package Manager on your OS.

After you see a message called “Installation Successful”. You are ready to install python version 3 on your macOS.

```

https://github.com/Homebrew/brew#donations
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Enumerating objects: 5042, done.
remote: Counting objects: 100% (5042/5042), done.
remote: Compressing objects: 100% (4841/4841), done.
remote: Total 5042 (delta 46), reused 565 (delta 10), pack-reused 0
Receiving objects: 100% (5042/5042), 4.03 MiB | 1.59 MiB/s, done.
Resolving deltas: 100% (46/46), done.
Tapped 2 commands and 4830 formulae (5,083 files, 12.5MB).
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
==> Next steps:
- Run `brew help` to get started
- Further documentation:
https://docs.brew.sh
Sierras-MBP:~ sierra$

```

- **Install Python Latest Version on macOS / macOS X**

To install python simple open Terminal app from Application -> Utilities and enter following command

```
brew install python3
```

After command processing is complete, Python's version 3 would be installed on your mac.

```
echo 'export PATH="/usr/local/opt/sqlite/bin:$PATH"' >> ~/.bash_profile

For compilers to find sqlite you may need to set:
export LDFLAGS="-L/usr/local/opt/sqlite/lib"
export CPPFLAGS="-I/usr/local/opt/sqlite/include"

==> python
Python has been installed as
/usr/local/bin/python3

Unversioned symlinks `python`, `python-config`, `pip` etc. pointing to
`python3`, `python3-config`, `pip3` etc., respectively, have been installed into
/usr/local/opt/python/libexec/bin

If you need Homebrew's Python 2.7 run
brew install python@2

You can install Python packages with
pip3 install <package>
They will install into the site-package directory
/usr/local/lib/python3.7/site-packages

See: https://docs.brew.sh/Homebrew-and-Python
Sierras-MBP:~ sierra$
```

To verify the installation enter following commands in your Terminal app

```
python
```

```
pip3
```

```
Last login: Tue Oct  1 12:07:58 on ttys000
Sierras-MBP:~ sierra$ python3 --version
Python 3.7.4
Sierras-MBP:~ sierra$
```

Bingo..!! Python is installed on your computer.

Revision questions

1. [How do I run a Python program under Windows?](#)

2. [How do I make Python scripts executable?](#)
3. [Why does Python sometimes take so long to start?](#)
4. [How do I make an executable from a Python script?](#)
5. [Is a *.pyd file the same as a DLL?](#)
6. [How can I embed Python into a Windows application?](#)
7. [How do I keep editors from inserting tabs into my Python source?](#)