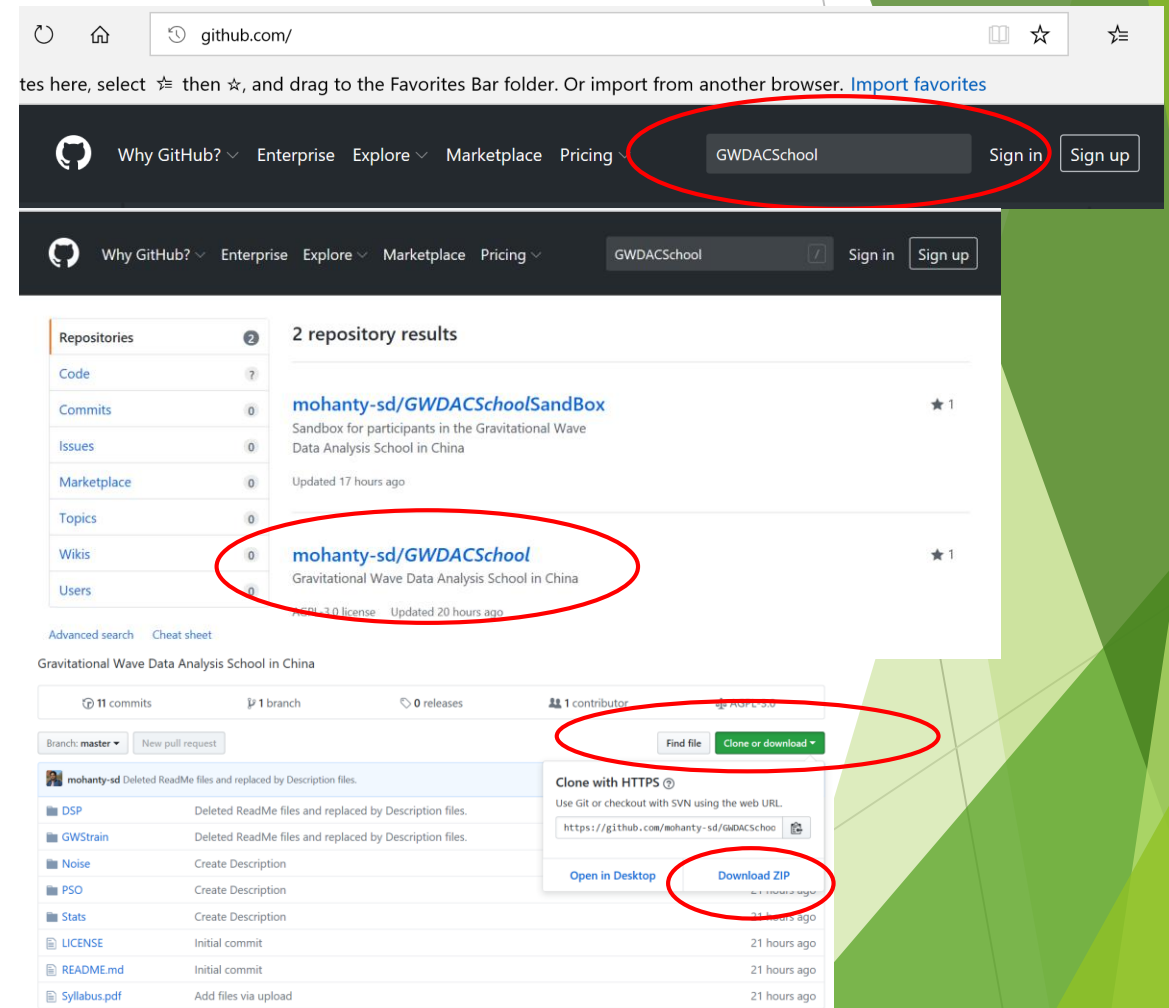


# Lab slides and Example codes

- ▶ Lab slides and associated code examples will appear in the GitHub repository: **GWDACSchool**
- ▶ Go to : `github.com`
- ▶ Search for the above repository name
- ▶ Download the contents
  - ▶ For just one file (e.g., Syllabus.pdf), click on the file name followed by “Download”

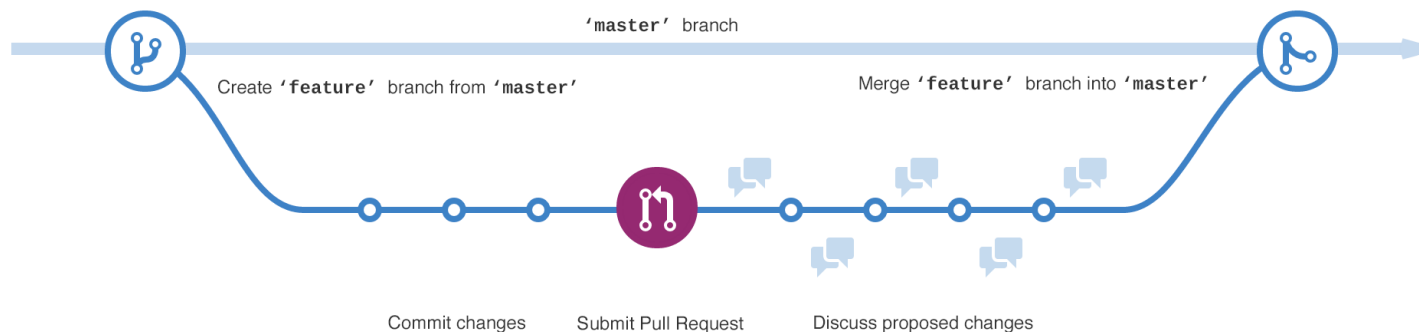


# Git and GitHub

- ▶ **Version control**
  - ▶ Maintaining a record of *all* changes made to a file, allowing any version of the file to be reconstructed
  - ▶ Allowing multiple users to make local changes to the same file and to merge the changes safely
  - ▶ Usually used for codes but also for any other content
- ▶ Everyone involved with coding should become familiar with some version control system
  - ▶ Especially important if you are working in a team on some project
- ▶ Git is a popular version control system
- ▶ GitHub is a web-based version control system
  - ▶ Public repository: Free
  - ▶ Private repository: Paid
- ▶ Tutorial: <https://guides.github.com/activities/hello-world/>

# Basic idea behind Git

- ▶ **Repository:** like a master copy of the files under version control
  - ▶ “A **repository** can contain folders and files, images, videos, spreadsheets, and data sets - anything your project needs” - GitHub tutorial
- ▶ **Branch:** A local copy of the repository in which you make changes
- ▶ Multiple branches can be active at a time opened by multiple developers
- ▶ Git will take care of safely merging all the changes made in different branches



# Teams

- ▶ **GWDACSchoolSandBox**: is the repository where you will add codes using Git
- ▶ Form teams of 3 to 4 people each
  - ▶ Members of a team will collaborate with each other codes
  - ▶ Different teams will work on different codes in most cases
  - ▶ Matlab is the preferred coding language since it will allow you to use codes developed by others
  - ▶ If you are not used to programming, make sure to join a team that has at least one good Matlab programmer
- ▶ Each team should create one account on Git
  - ▶ Designate a member of your team to handle Git
  - ▶ You can share the Git username and password but **remember to change the password** after the school!
- ▶ Email the account name to Xiao-bo who will add it as a “Collaborator” to the repository **GWDACSchoolSandBox**

# Lab exercise #1

- ▶ Clone the repository **GWDACSchoolSandBox** from GitHub
- ▶ Below is an example of cloning a different repository

The screenshot illustrates the process of cloning a repository on GitHub. It is divided into two main sections by a large orange arrow pointing downwards.

**Top Section (GitHub Homepage):**

- The browser address bar shows `https://github.com/`, which is circled in red.
- The navigation bar includes links for "Enterprise", "Explore", "Marketplace", and "Pricing". A search bar contains the text "SDMBIGDAT19" and is circled in yellow.
- A "Sign in" button is located to the right of the search bar.

**Bottom Section (Repository Page):**

- The repository page shows "7 commits", "1 branch", "0 releases", and "1 contributor".
- A "Find file" button is present, and next to it is a green "Clone or download" button, which is circled in red.
- A dropdown menu is open from the "Clone or download" button, showing the "Clone with HTTPS" option. The URL `https://github.com/mohanty-sd/SDMBIGDAT19` is displayed in the input field, and a red circle highlights the copy icon to the right of the URL.
- Below the URL, there are two buttons: "Open in Desktop" and "Download ZIP".
- On the left side of the repository page, there is a list of files: "CODES", "SLIDES", "README.md", and "README.pdf". Each file has an "Add files via upload" or "Update README.md" button next to it.

# Clone repository

- ▶ Git should already be available on Mac and Linux
- ▶ Windows users:
  - ▶ Search for “git on windows10” in Bing (or your favorite search engine)
  - ▶ Or URL: <https://git-scm.com/download/win>
  - ▶ Note: Accept the default choices but check with instructor first
- ▶ Clone the repository in a convenient folder as shown in the example

```
MINGW64:/c/Users/soumy/My Documents/TEACHING

soumya@DESKTOP-8PEE79N MINGW64 ~
$ cd c:

soumya@DESKTOP-8PEE79N MINGW64 /c
$ cd Users/soumy/My Documents/TEACHING

soumya@DESKTOP-8PEE79N MINGW64 ~/My Documents/TEACHING
$ ls
08CHINASCHOOL/      2010INDIGO/        BIGDAT19/           SHREYA/
09CHINASCHOOL/      3DMODELING/        DOWNLOADS/          SyncToy_6093ecf1-
17CHINASCHOOL_MLDC/ ANOMALOUSPULSAR/   GWDACSchool/       'Teaching Ideas'/
18CHINASCHOOL/      ARTICLES/           KAZAKHSCHOOL2014/  Thumbs.db
19CHINASCHOOL/      BIGDAT17/           PICTURES/

soumya@DESKTOP-8PEE79N MINGW64 ~/My Documents/TEACHING
$ git clone https://github.com/mohanty-sd/GWDACSchoolSandBox.git
Cloning into 'GWDACSchoolSandBox'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

soumya@DESKTOP-8PEE79N MINGW64 ~/My Documents/TEACHING
$
```



# Lab Exercise #2

- ▶ When instructed ...
- ▶ **git pull**
- ▶ Edit the file called Teams.md in your local branch and add the name of your team followed by the name of the team members **on a separate line**
  - ▶ <Team Name>: <name1>,<br><name2>,...
- ▶ Confirm with instructor ...
- ▶ Save the file and push your changes to the main repository as shown

The screenshot shows a terminal window with the following commands and output:

```
$ git config --global user.name mohanty-sd
$ git config --global user.email soumya.mohanty@utrgv.edu
soumya@DESKTOP-8PEE79N MINGW64 ~/My Documents/TEACHING/GWDACSchoolSandBox (master)
$ git commit Teams.md
```

The terminal then displays the commit message editor (vim) with the following text:

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   new file:   Teams.md
#
```

An orange callout box points to the vim editor with the following instructions:

- Type 'I' to 'Insert'
- Write a message
- Press 'Esc'
- Type 'wq'

A GitHub Login dialog box is overlaid on the terminal. It has the title "GitHub Login" and contains two input fields: "Username or email" and "Password". Below the fields are "Login" and "Cancel" buttons. At the bottom, it says "Don't have an account? Sign up" and "Forgot your password?". An orange arrow points from the dialog box to the terminal output below it.

```
$ git commit -a
[master c8cf795] First import.
1 file changed, 3 insertions(+)
create mode 100644 Teams.md
soumya@DESKTOP-8PEE79N MINGW64 ~/My Documents/TEACHING/GWDACSchoolSandBox (master)
$ git push https://github.com/mohanty-sd/GWDACSchoolSandBox.git
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 370 bytes | 185.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/mohanty-sd/GWDACSchoolSandBox.git
83b17d2..c8cf795 master -> master
```

A red circle highlights the command `$ git push https://github.com/mohanty-sd/GWDACSchoolSandBox.git` in the terminal output.

# Basic Git commands

- ▶ For any file that is edited locally ...
- ▶ You must first commit the change
  - ▶ **git pull**
    - ▶ This step should always precede git commit: it merges changes made by others into your local copies
  - ▶ **git commit <filename>** (Or, “git commit -a” to commit changes to all files)
    - ▶ This tells git that you are ready to push the changes to the main repository
    - ▶ Enter a short message describing the change
  - ▶ **git push**
    - ▶ This pushes your changes to the main repository, allowing others to see your changes
- ▶ If you do not wish to keep your changes, then **before doing a commit**
  - ▶ **git checkout <filename>**
    - ▶ This will revert to the version of the file before your made your changes



# Lab Exercise #3

- ▶ Verify that you have access to a working copy of Matlab
- ▶ `plot(1:10)`