

UCSB, Physics 129AL, Computational Physics: Section Worksheet, Week 1

Zihang Wang (UCSB), zihangwang@ucsb.edu

January 5, 2025

Section Participation

Section attendance is required, but you do not need to complete all the work during the section. At each section, the TA will answer any questions that you might have, and you are encouraged to work with others and look for online resources during the section and outside of sections.

Unless otherwise stated, the work will be due one week from the time of assignment.

To get participation grades for the section, before Leaving the section, you must have the following, and inspected by the TA.

- You have been added to the Github organization by your TA.
- Show to the TA that you have a working interface on <http://localhost:xx80/>.
- Show your git command history within the container to the TA that you have followed and completed Task 3. Show the remote repository created via Task 3.

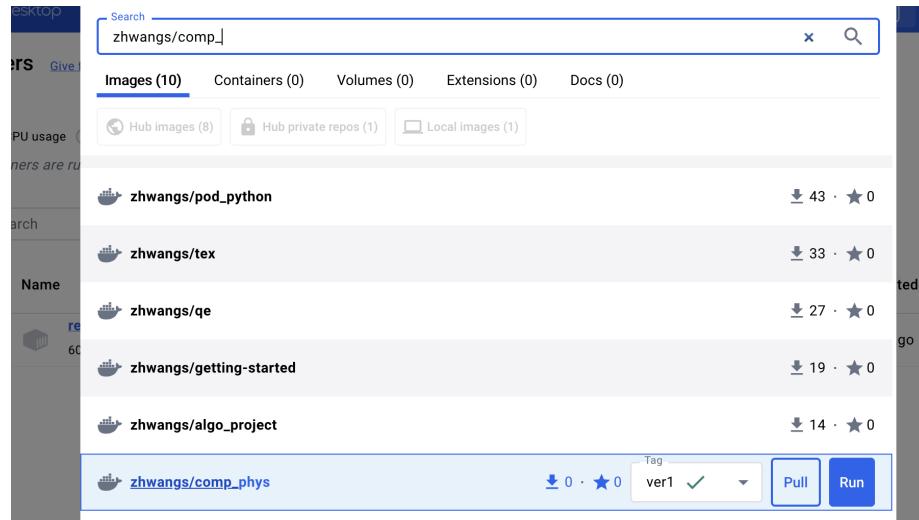
The TA will give you 1 point for each task completed. You can see your grades on Canvas.

Task 1: Github and Docker Account

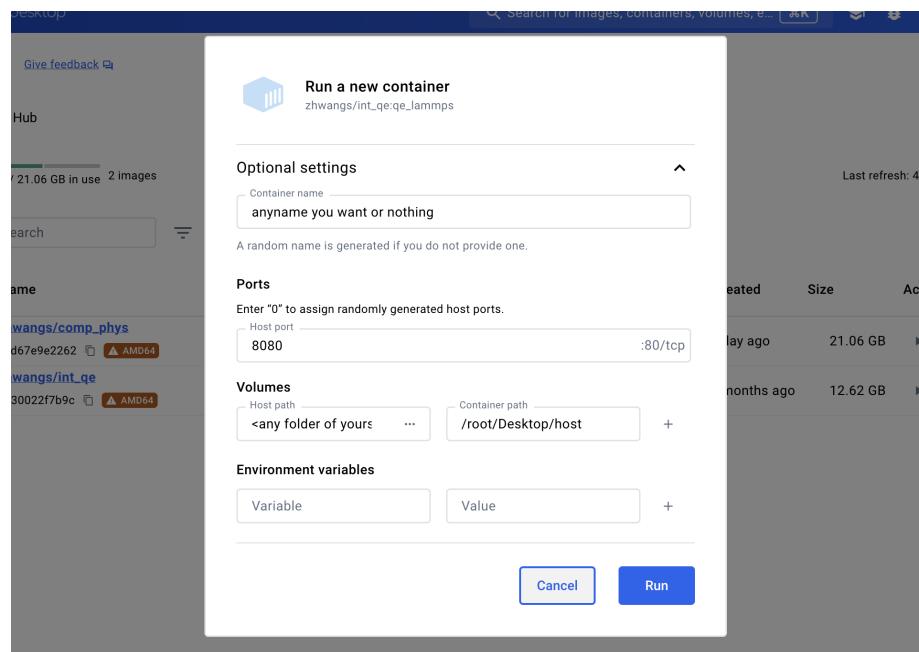
In the class, we discussed the importance of Github and Docker in modern computation. In here, you will create both Github and Docker accounts with your @ucsb.edu email. Provide your Github username to the TA, and the TA will invite you to the Github organization. Accept the invitation, and show it to the TA before you leave the section.

Task 2: Docker Image

Download Docker Desktop and install it on your computer, and pull the following image via the docker Desktop: zhwangs/comp_phys:ver1.



Do the following to create an instance (container) of the above image.



If you encounter problems, please go visit the following: <https://docs.docker.com/get-started/docker-concepts/the-basics/what-is-a-container/>

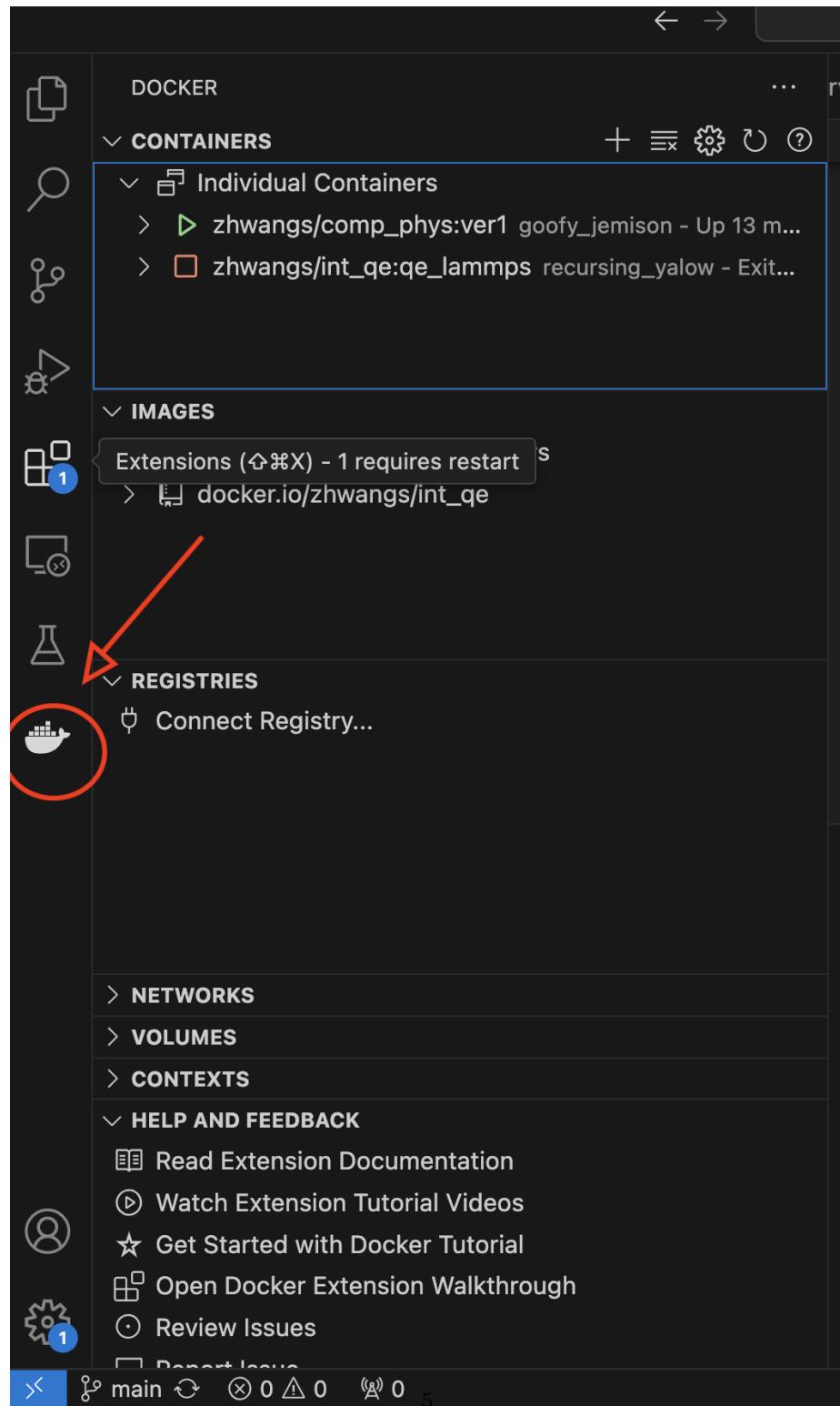
To access the docker container, open your web browser, and copy the following: <http://localhost:xx80/>, where xx is the TCP address you entered previously.

Task 2.5 Attach Visual Studio Code

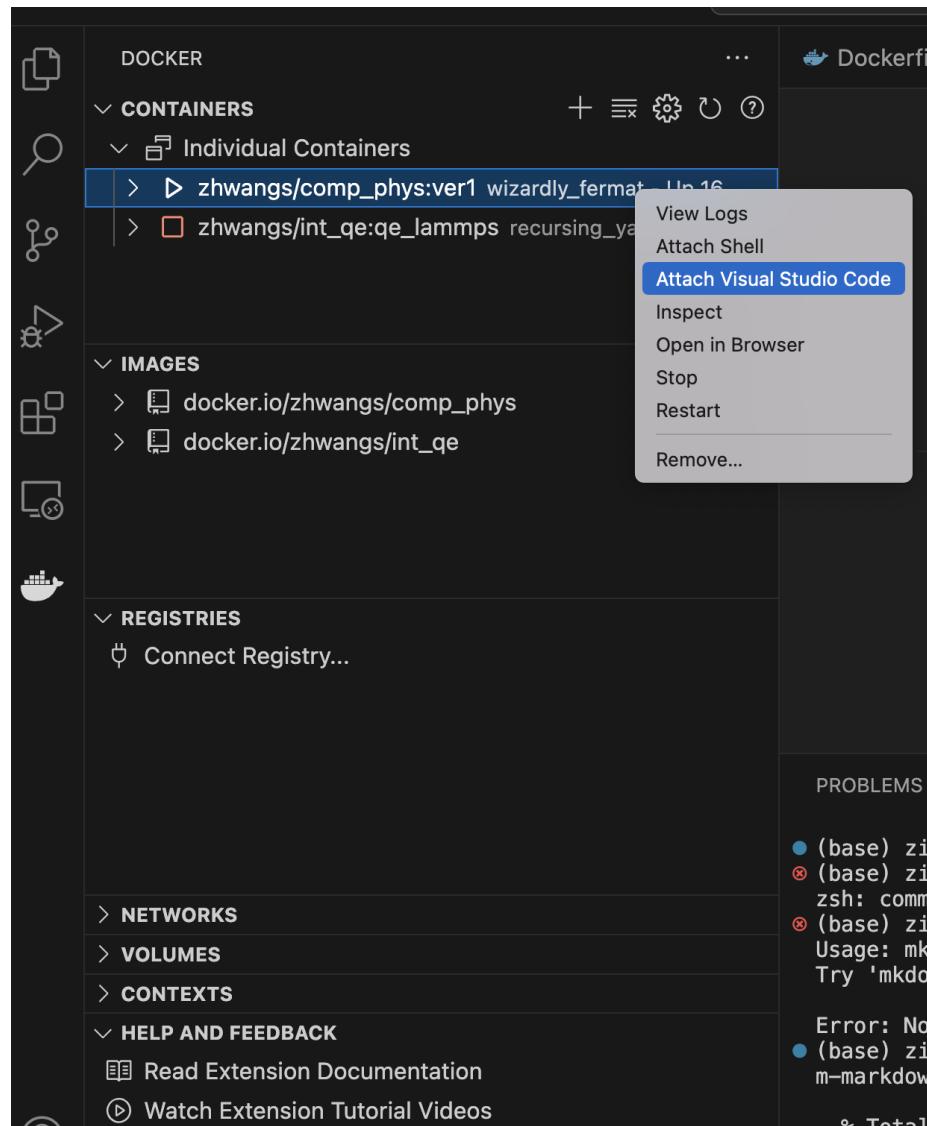
Download Visual Studio Code, and add the following extensions:

- Dev Containers
- Docker
- Python
- Jupyter
- Pylance

While keeping the above container excited, open Visual Studio Code, and navigate to **the Docker icon on the sidebar**, and open it. (for cerntain computers with some versions of the Visual Studio Code, you should navigate to Dev Containers icon instead, i.e. if you did not see the Docker icon, go to Dev Containers icon.)



You can attach shell to the terminal without each time going to the localhost interface, e.g. look at the following,



Task 3: Git Example Usages

Before you start, you need to generate a token for the terminal access, go to the following, <https://github.com/settings/tokens>

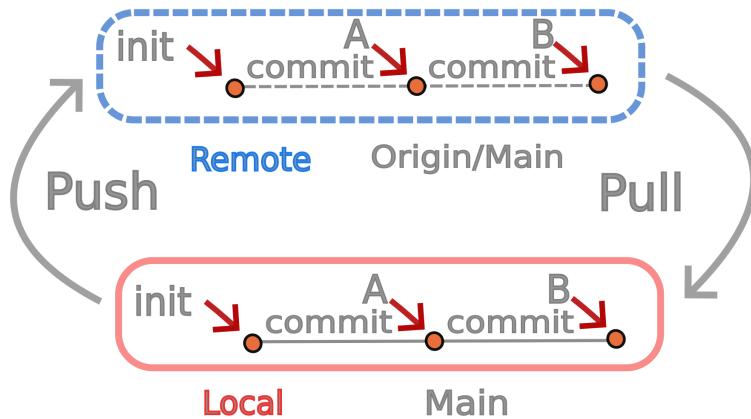
Click generate new token (classic), and assign permissions.

Here are some example usages on Git with remote origin on Github. Initial-

ize a git repository walk-through the follow example locally and then push to your own Github remote repository (make sure you create it first on Github). In the end, you should expect something like this example, https://github.com/Physics-129AL/git_example. **You must work within the container you excite previously.**

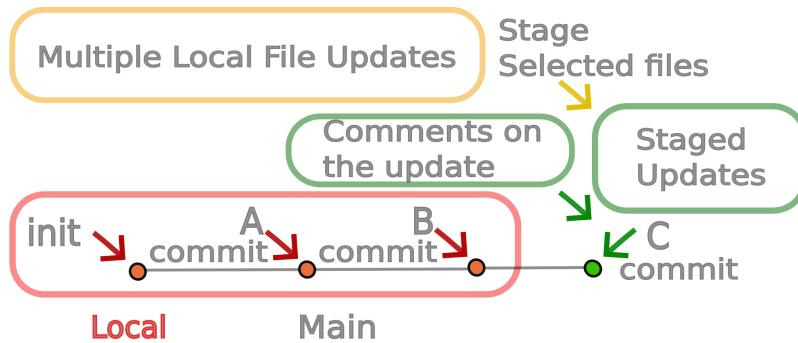
As one of the most influential packages, Git tracks changes in the source code during software development. It is a powerful and widely adopted tool that allows multiple developers to collaborate on the same project and manage versions efficiently. Here are some basic ideas of Git to get you started. A Git repository is a directory that contains all the files and the entire history of a project. It can be a local repository or a remote one like GitHub, e.g. Fig.1. The basic git version control contains a few key concepts, including Branch, Commit, Push, Pull, and Pull request. Version management is a complex task currently undergoing intensive research and development efforts. Although we won't go into details, we will briefly touch upon conflict control, e.g. Merge or Rebase. We will introduce them sequentially.

0.1 Synchronization of the Local and Remote Branch



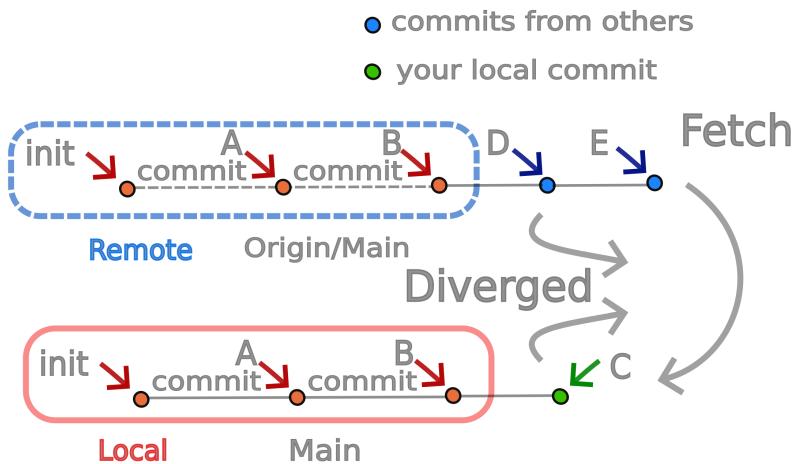
A graph illustrates the interaction between a local branch named "main" and a remote branch called "origin/main." The orange dots on the graph represent commit events, which are labeled as "init," "A," and "B," respectively. Synchronization between two branches can be achieved through push and pull operations.

0.2 Commit



A graph illustrates the commit operation to the local branch named "main". The orange dots on the graph represent previously commit events, which are labeled as "init," "A," and "B," respectively. The green dot on the graph represents the newly commit event in the local main branch. To commit a local file updates to the main branch, we first select all files we want to append to the main branch, and stage them. We then can commit the change with a short descriptions of the changes. Please note that it only applies to those staged changes. Synchronization between two branches can be achieved through push and pull operations.

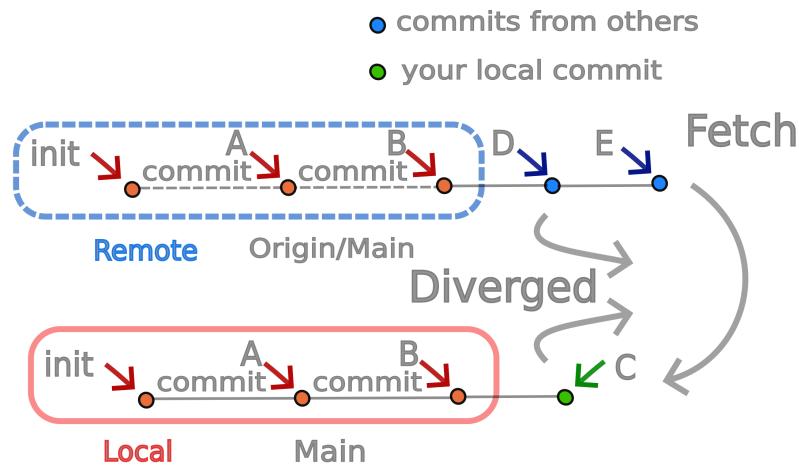
0.3 Fetch



A graph illustrates the general fetch operation to the local branch named "main". The orange dots on the graph represent previously commit events, which are

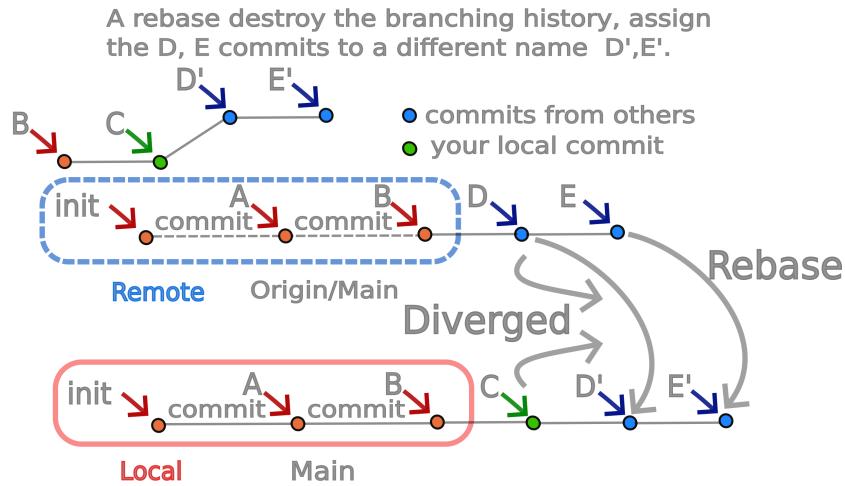
labeled as "init," "A," and "B," respectively. The green dot on the graph represents the newly commit event. The blue dots on the graph represent the changes have made by other collaborators. Let's say you have made some changes in your local main branch based on the orange dots (commits). You also want to update your local main branch with the remote main branch, updated by your collaborators, e.g. blue dots, D,E. The Fetch operations are able to acquire all remote commits (blue dots) that are absent in the local main branch. However, we must tell git how to group them.

0.4 Merge Commit



A graph illustrates the general fetch operation to the local branch named "main". The orange dots on the graph represent previously commit events, which are labeled as "init," "A," and "B," respectively. The green dot on the graph represents the newly commit event. The blue dots on the graph represent the changes have made by other collaborators. The violet dot on the graph represent the commit for the merge operation. Merge operation saves the the divergent between the local main branch and remote origin/main into a new commit, e.g. dot F. In other words, the merrge operation perserves the branch difference at the dot F in version history.

0.5 Rebase



A graph illustrates the general fetch operation to the local branch named "main". The orange dots on the graph represent previously commit events, which are labeled as "init," "A," and "B," respectively. The green dot on the graph represents the newly commit event. The blue dots on the graph represent the changes have made by other collaborators. As the name suggested, rebase operation appends commits from others, D,E to the existing local main branch, and assign them new name D',E'. We see that the history of branching is destroyed.

0.6 Example Usage

```
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:           git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/zhang/Desktop/git_example_physics129/.git/
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git config --global user.name zhangwang
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git config --global user.email zhangwang@ucsb.edu
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git checkout -b main
Switched to a new branch 'main'
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git add .
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git commit -m "init"
[main (root-commit) 0a22fa9] init
 4 files changed, 4 insertions(+)
 create mode 100755 cache/image_info.txt
 create mode 100755 readme
 create mode 100755 src/src_info.txt
 create mode 100755 test/test_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git remote add origin https://github.com/zhangw/Physics129_git_example.git
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ git push -u origin main
Username for 'https://github.com': zhangw
Password for 'https://zhangw@github.com':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (9/9), 623 bytes | 124.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/zhangw/Physics129_git_example.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$
```

Annotations on the right side of the terminal output:

- Set user name (points to `git config --global user.name zhangwang`)
- Set user email (points to `git config --global user.email zhangwang@ucsb.edu`)
- Build the main branch (points to `git checkout -b main`)
- Stage all files in git repository (points to `git add .`)
- Commit with comments (points to `git commit -m "init"`)
- Add remote origin from github. (points to `git remote add origin https://github.com/zhangw/Physics129_git_example.git`)
- Push to the origin/main (points to `git push -u origin main`)

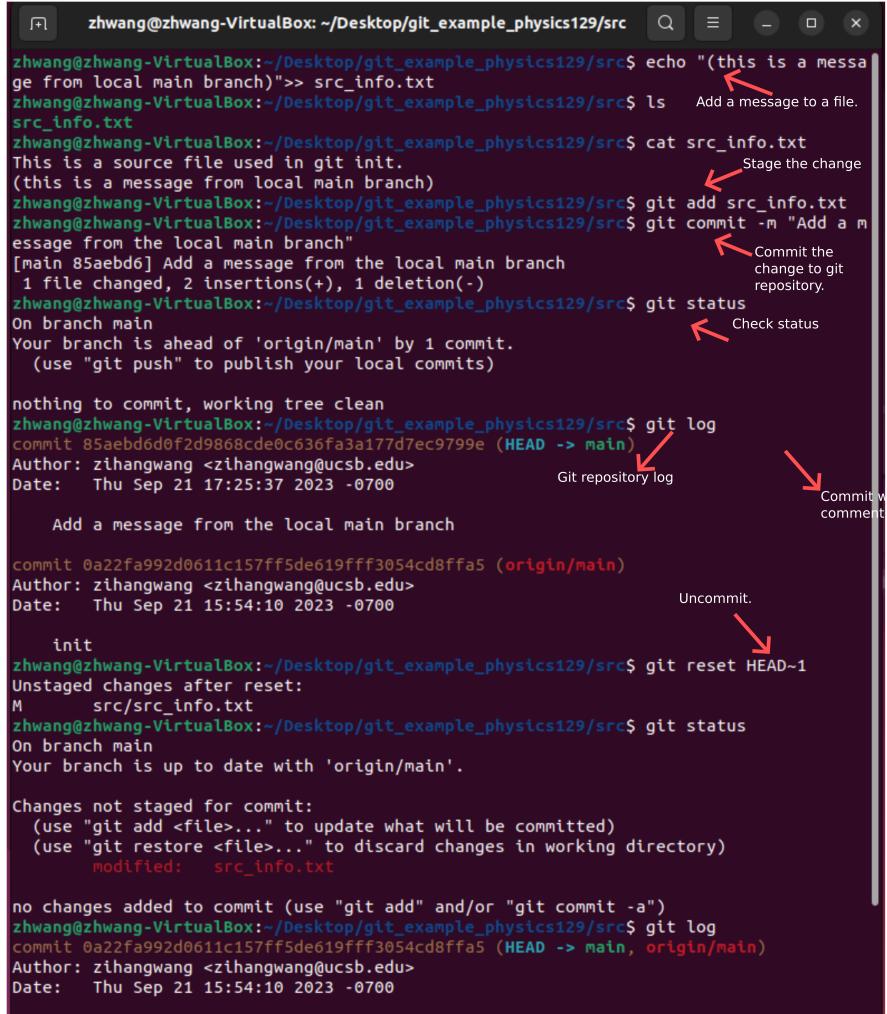
Initialize Git repository and push the first commit to the remote. For additional examples, refer to supplementary material *SI git example usage.pdf*.

```

zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:           git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:           git branch -m <name>
Initialized empty Git repository in /home/zhangwang/Desktop/git_example_physics129/.git/
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git config --global user.name zhangwang
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git config --global user.email zhangwang@ucsb.edu
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git checkout -b main
Switched to a new branch 'main'
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git add .
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git commit -m "init"
[main (root-commit) 0a22fa9] init
 4 files changed, 4 insertions(+)
 create mode 100755 cache/image_info.txt
 create mode 100755 readme
 create mode 100755 src/src_info.txt
 create mode 100755 test/test_info.txt
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git remote add origin https://github.com/zhwangs/Physics129_git_example.git
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ git push -u origin main
Username for 'https://github.com': zhwangs
Password for 'https://zhangwang@github.com':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (9/9), 623 bytes | 124.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/zhwangs/Physics129_git_example.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
zhangwang@zhangwang-VirtualBox:~/Desktop/git_example_physics129$ 

```

Initialize Git repository and push the first commit to the remote.



```

zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ echo "(this is a message from local main branch)">> src_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ ls      Add a message to a file.
src_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ cat src_info.txt
This is a source file used in git init.
(this is a message from local main branch)                                     Stage the change
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git add src_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git commit -m "Add a message from the local main branch"
[main 85aebd6] Add a message from the local main branch
 1 file changed, 2 insertions(+), 1 deletion(-)
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)                                Check status
nothing to commit, working tree clean
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git log
commit 85aebd60f2d9868cde0c636fa3a177d7ec9799e (HEAD -> main)
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 17:25:37 2023 -0700
          Git repository log
          Commit w/ comment
          Uncommit.
          Changes not staged for commit:
          (use "git add <file>..." to update what will be committed)
          (use "git restore <file>..." to discard changes in working directory)
            modified:   src_info.txt
no changes added to commit (use "git add" and/or "git commit -a")
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git log
commit 0a22fa992d0611c157ff5de619fff3054cd8ffa5 (HEAD -> main, origin/main)
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 15:54:10 2023 -0700

```

Example of staging, committing, and uncommitting.

```
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ echo "(this is a message from local main branch)">> src_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ cat src_info.txt
This is a source file used in git init.
(this is a message from local main branch)
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git add src_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git commit -m "Add a message from the local main branch"
[main 91da6e4] Add a message from the local main branch
 1 file changed, 2 insertions(+), 1 deletion(-)
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git push origin main
Username for 'https://github.com': zhang
Password for 'https://zhang@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 391 bytes | 195.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/zhang/Physics129_git_example.git
  0a22fa9..91da6e4  main -> main
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$
```

Push an update commit to the remote.

```
zhang@zhang-VirtualBox: ~/Desktop/git_example_physics129/src$ git checkout -b feature_branch
Switched to a new branch 'feature_branch'          Checkout(and create) →
new branch
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ echo "(this is a
message from local feature branch)">> src_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ echo "(this is a
message from local feature branch to test_info)">> ../test/test_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git status
On branch feature_branch                         Check status →
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   src_info.txt
    modified:   ../test/test_info.txt  ← Not staged
                                         stage all
                                         changes in src
no changes added to commit (use "git add" and/or "git commit -a") ←
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git add .
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git status
On branch feature_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   src_info.txt  ← One file staged
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   ../test/test_info.txt      stage changes in a
                                         different location. →
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git add ../test/test_info.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git status
On branch feature_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   src_info.txt
    modified:   ../test/test_info.txt
```

Create a new branch and commit changes in it.

```
zhangwang@zhangwang-VirtualBox: ~/Desktop/git_example_physics129/src$ git commit -m "new branch with two changes"
[feature_branch bb88779] new branch with two changes Commit changes to a new branch
 2 files changed, 2 insertions(+), 1 deletion(-)
zhangwang@zhangwang-VirtualBox: ~/Desktop/git_example_physics129/src$ git push -u origin feature_branch
Username for 'https://github.com': zhangwang Push to new branch (use -u for linking local and the newly created remote branch.
Password for 'https://zhangwang@github.com': 
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 600 bytes | 600.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 Check status
remote: Create a pull request for 'feature_branch' on GitHub by visiting:
remote:     https://github.com/zhangwang/Physics129_git_example/pull/new/feature_branch
remote:
To https://github.com/zhangwang/Physics129_git_example.git
 * [new branch]      feature_branch -> feature_branch
Branch 'feature_branch' set up to track remote branch 'feature_branch' from 'origin'.
zhangwang@zhangwang-VirtualBox: ~/Desktop/git_example_physics129/src$ git log
commit bb88779e06ef5f6ca8efee7b9864794fefbd50c8 (HEAD --> feature_branch, origin/feature_branch)
Author: zhangwang <zhangwang@ucsb.edu>
Date:   Thu Sep 21 18:44:30 2023 -0700

    new branch with two changes

commit 91da6e43a192cbb8d3567c8d088979e8f1f939b7 (origin/main, main)
Author: zhangwang <zhangwang@ucsb.edu>
Date:   Thu Sep 21 18:01:30 2023 -0700

    Add a message from the local main branch

commit 0a22fa992d0611c157ff5de619fff3054cd8ffa5
Author: zhangwang <zhangwang@ucsb.edu>
Date:   Thu Sep 21 15:54:10 2023 -0700

    init
```

Create new remote branch.

```

zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git merge feature_branch
Updating 91da6e4..bb88779
Fast-forward
src/src_info.txt | 1 + Since there is no file conflicts, the history is fast
test/test_info.txt | 2 +- forwarded (e.g. no need fo the F dot)
 2 files changed, 2 insertions(+), 1 deletion(-)
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git log
commit bb88779e06ef5f6ca8fee7b9864794fefebd50c8 (HEAD -> main, origin/feature_branch)
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 18:44:30 2023 -0700

    new branch with two changes

commit 91da6e43a192cbb8d3567c8d088979e8f1f939b7 (origin/main)
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 18:01:30 2023 -0700

    Add a message from the local main branch

commit 0a22fa992d0611c157ff5de619fff3054cd8ffa5
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 15:54:10 2023 -0700

    init
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git push origin main
Username for 'https://github.com': zhangw
Password for 'https://zhangw@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/zhangw/Physics129_git_example.git
  91da6e4..bb88779  main -> main

```

Merge two branches and remote updates.

```

zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git fetch origin
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
nothing to commit, working tree clean
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git diff main origin/main
diff --git a/remote_created.txt b/remote_created.txt
new file mode 100644
index 000000..a45cf0
--- /dev/null
+++ b/remote_created.txt
@@ -0,0 +1 @@
+This is a remote created file on the main branch.
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git reset --hard
HEAD~1
HEAD is now at 91da6e4 Add a message from the local main branch
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ cat src_info.txt
This is a source file used in git init. ← A file change restored.
(this is a message from local main branch)
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git reset --hard
bb88779e06ef5f6ca8efee7b9864794efebd50c8 ← Set to a commit
HEAD is now at bb88779 new branch with two changes ID.
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ cat src_info.txt
This is a source file used in git init.
(this is a message from local main branch) ← A file change recovered
(this is a message from local feature branch)
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ git pull origin main
From https://github.com/zhangw/Physics129_git_example
 * branch      main      -> FETCH_HEAD
Updating bb88779..392bfda ← Pull from remote
Fast-forward
 remote_created.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 remote_created.txt
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129/src$ cd ..
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ ls
cache readme remote_created.txt src test
zhang@zhang-VirtualBox:~/Desktop/git_example_physics129$ █

```

Local are updated one commit according to the remote

Pull and version restoration.

```
zhwang@zhwang-VirtualBox: ~/Desktop/git_clone/Physics129$ cd ~/Desktop/
zhwang@zhwang-VirtualBox:~/Desktop$ mkdir git_clone
zhwang@zhwang-VirtualBox:~/Desktop$ cd git_clone/
zhwang@zhwang-VirtualBox:~/Desktop/git_clone$ git clone https://github.com/zhwangs/Physics129_git_example.git
Cloning into 'Physics129_git_example'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 22 (delta 3), reused 17 (delta 1), pack-reused 0
Receiving objects: 100% (22/22), done.
Resolving deltas: 100% (3/3), done.
zhwang@zhwang-VirtualBox:~/Desktop/git_clone$ ls
Physics129_git_example
zhwang@zhwang-VirtualBox:~/Desktop/git_clone$ cd Physics129_git_example/
zhwang@zhwang-VirtualBox:~/Desktop/git_clone/Physics129_git_example$ ls
cache  readme  remote_created.txt  src  test
zhwang@zhwang-VirtualBox:~/Desktop/git_clone/Physics129_git_example$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
zhwang@zhwang-VirtualBox:~/Desktop/git_clone/Physics129_git_example$ git log
commit 392bfda97478d6ad208fc4971389e6adf6add6ca (HEAD -> main, origin/main, origin/HEAD)
Author: zhwangs <39713086+zhwangs@users.noreply.github.com>
Date:   Thu Sep 21 23:59:53 2023 -0700

    Create remote_created.txt

commit bb88779e06ef5f6ca8efee7b9864794efebd50c8 (origin/feature_branch)
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 18:44:30 2023 -0700

    new branch with two changes

commit 91da6e43a192cbb8d3567c8d088979e8f1f939b7
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 18:01:30 2023 -0700

    Add a message from the local main branch

commit 0a22fa992d0611c157ff5de619fff3054cd8ffa5
Author: zihangwang <zihangwang@ucsb.edu>
Date:   Thu Sep 21 15:54:10 2023 -0700

    init
zhwang@zhwang-VirtualBox:~/Desktop/git_clone/Physics129_git_example$
```

Clone a git repository.