Git Commands for Local Repos

Part 1: Repos with Only a `master` Branch

Consider this Git log graph (log messages omitted).

```
* a3b25e7 (HEAD -> master)
* e41c5b6
* 9d63832
* 40f26d8
```

What does each `*` represent?

A commit. Each one is associated with a different snapshot (version) of the project.

What do the hexadecimal numbers represent?

They are cryptographic hashes of the commit snapshots. They serve as unique commit IDs.

What does `master` represent?

The latest commit on the master branch.

What does `HEAD ->` tell you?

Which branch the developer is currently on. Also, which version is in the working directory.

Consider this Git log graph (log messages omitted).

```
* a3b25e7 (HEAD -> master)
* e41c5b6
* 9d63832
* 40f26d8
```

Which commit is the oldest?

40f26d8

Which commit is the newest?

a3b25e7

Which commits make up the version history of the `master` branch?

a3b25e7, e41c5b6, 9d63832, 40f26d8

Which version of the code is currently in the working directory?

a3b25e7

Does the repo currently have a detached HEAD? Explain your answer.

No. The HEAD is attached to the master branch.

Consider this Git log graph (log messages omitted).

```
* a3b25e7 (master)
* e41c5b6
* 9d63832 (HEAD)
* 40f26d8
```

Which version of the code is currently in the working directory?

9d63832

Does the repo currently have a detached HEAD? Explain your answer.

Yes. The HEAD is not attached to any branch.
Here, the HEAD points at commit 9d63832.

Each of the following problems presents a Git log graph (log messages omitted) of a local repo and a scenario. Write the log graph that would result from the scenario.

- If you need to add a commit, use the hash `c1c1c1c`.
- If a command would result in an error, write "ERROR" and explain why the error occurred.

Here is an example problem and solution to help clarify what's expected.

Example Problem

Scenario: Developer makes changes to the code, stages the changes, and commits.

```
* 86b8116 (HEAD -> master)
* dc003f8
* 026c6cf
```

Example Solution

```
* c1c1c1 (HEAD -> master)
* 86b8116
* dc003f8
* 026c6cf
```

Scenario: Developer makes changes to the code, stages the changes, and commits.

```
* a3b25e7 (HEAD -> master)
* e41c5b6
* 9d63832
* 40f26d8
```

* c1c1c1c (HEAD -> master)
* a3b25e7
* e41c5b6
* 9d63832
* 40f26d8

Scenario: Developer runs `git checkout 9d63832`.

```
* a3b25e7 (HEAD -> master)
* e41c5b6
* 9d63832
* 40f26d8
```

---

* a3b25e7 (master)
* e41c5b6
* 9d63832 (HEAD)
* 40f26d8

Scenario: Developer makes changes to the code, stages the changes, and commits.

```
* a3b25e7 (master)
* e41c5b6
* 9d63832 (HEAD)
* 40f26d8
```

ERROR. Commits are not allowed when the HEAD is detached.

Scenario: Developer runs `git checkout master`.

```
* a3b25e7 (master)
* e41c5b6
* 9d63832 (HEAD)
* 40f26d8
```

---

* a3b25e7 (HEAD -> master)
* e41c5b6
* 9d63832
* 40f26d8

Part 2: Repos with Multiple Branches

Consider this Git log graph (log messages omitted).

```
*   a3469b1 (HEAD -> master)
|\
| * befebda (iss3)
| * 5fc2fa3
* |   279f80f
|\ \
| * | c53d97e (iss1)
| * | 9ebb260
| |/
* | 4c61006 (iss2)
* | de9bef7
* | 1381360
|/
* 894a0c1
* c395543
```

What does `(iss1)` denote?

The latest commit on the iss1 branch.

What does `(iss3)` denote?

The latest commit on the iss3 branch.

Which branch is the developer currently on? That is, which branch is currently checked out?

The master branch (as per the HEAD pointer).

Consider this Git log graph (log messages omitted).

```
*   a3469b1 (HEAD -> master)
|\
| * befebda (iss3)
| * 5fc2fa3
* |   279f80f
|\ \
| * | c53d97e (iss1)
| * | 9ebb260
| |/
* | 4c61006 (iss2)
* | de9bef7
* | 1381360
|/
* 894a0c1
* c395543
```

List all the commits that make up the version history of the `iss2` branch.

4c61006, de9bef7, 1381360, 894a0c1, c395543

List all the commits that make up the version history of the `iss1` branch.

c53d97e, 9ebb260, 894a0c1, c395543

List all the commits that make up the version history of the `master` branch.

All the commits are part of the history of the master branch.

Consider this Git log graph (log messages omitted).

```
*   a3469b1 (HEAD -> master)
|\
| * befebda (iss3)
| * 5fc2fa3
* |   279f80f
|\ \
| * | c53d97e (iss1)
| * | 9ebb260
| |/
* | 4c61006 (iss2)
* | de9bef7
* | 1381360
|/
* 894a0c1
* c395543
```

Working from top to bottom, list each commit in which one branch was merged into another branch. Tell which branch was merged into which other branch. Tell how you can tell that a merge happened. Tell which two versions of the code were merged together. (Don't bother attempting to guess fast-forward merges for this question, because no definitive indicators for such merges are stored in the log.)

a3469b1 - iss3 was merged into master. Version 279f80f was combined with version befebda. I can tell these versions were merged because each has an edge that leads into a3469b1.

279f80f - iss1 was merged into master. Version 4c61006 was combined with version c53d97e. I can tell these versions were merged because each has an edge that leads into 279f80f.

Which of the following commands would show you which files in your working directory are untracked, which changes are staged, which branch you're on, and whether your working directory is clean?

a) `git init`

b) `git log`

c) `git status` ✓

Each of the following problems presents a Git log graph (log messages omitted) of a local repo and a scenario. Write the log graph that would result from the scenario.

- If you need to add a commit, use the hash `c1c1c1c`.
- If a command would result in an error, write "ERROR" and explain why the error occurred.

Here is an example problem and solution to help clarify what's expected.

Example Problem

Scenario: Developer makes changes to the code, stages the changes, and commits.

```
* 86b8116 (HEAD -> master)
* dc003f8
* 026c6cf
```

Example Solution

```
* c1c1c1 (HEAD -> master)
* 86b8116
* dc003f8
* 026c6cf
```

Scenario: Developer runs `git checkout -b iss2`.

```
* 97c9227 (HEAD -> master)
* ed11409
* 137d7d0
```

```
* 97c9227 (HEAD -> iss2, master)
* ed11409
* 137d7d0
```

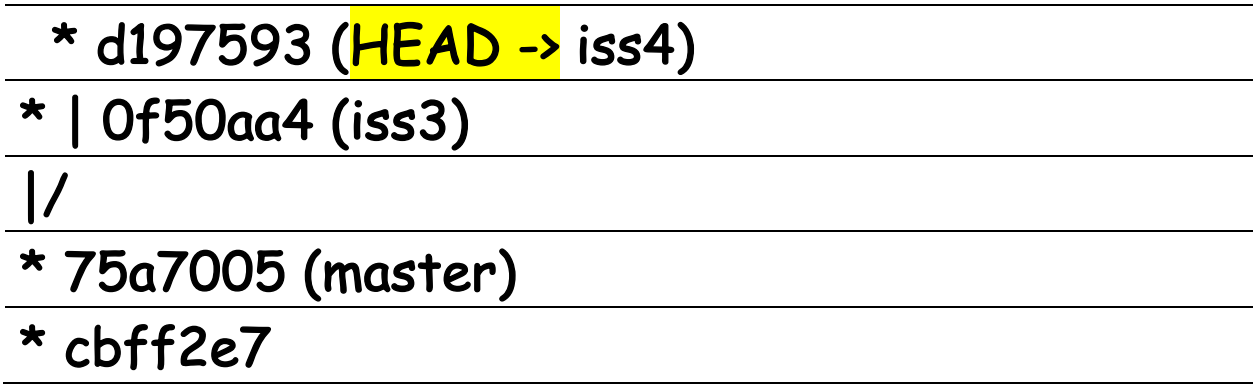
Scenario: Developer runs `git checkout -b iss12`.

```
  * 04ca8c6 (master)
* | 3283dd7 (HEAD -> iss11)
|/
* a8da338
* fe2251a
```

```
  * 04ca8c6 (master)
* | 3283dd7 (HEAD -> iss12, iss11)
|/
* a8da338
* fe2251a
```

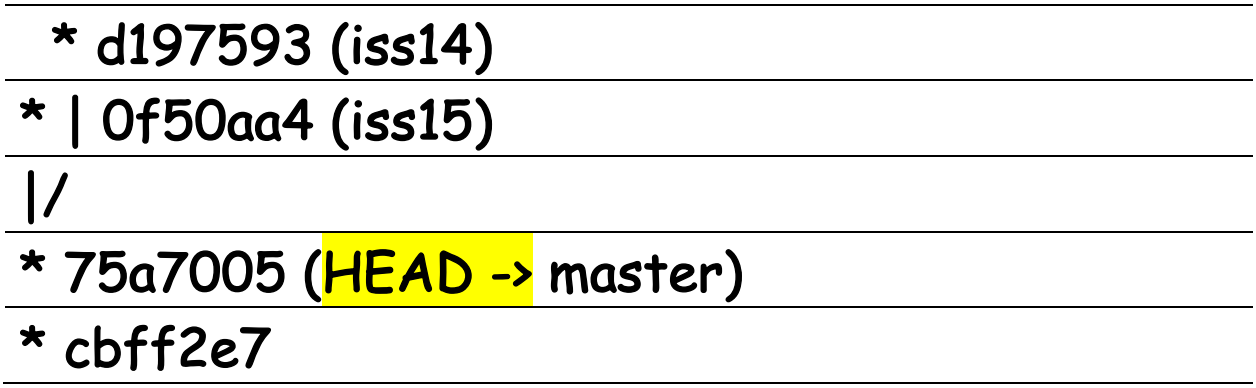
Scenario: Developer runs `git checkout iss4`.

```
  * d197593 (iss4)
* | 0f50aa4 (iss3)
|/
* 75a7005 (HEAD -> master)
* cbff2e7
```



Scenario: Developer runs `git checkout master`.

```
  * d197593 (HEAD -> iss14)
* | 0f50aa4 (iss15)
|/
* 75a7005 (master)
* cbff2e7
```



Scenario: Developer makes changes to the code, stages the changes, and commits.

```
  * 04ca8c6 (master)
* | 3283dd7 (HEAD -> iss2)
|/
* a8da338
* fe2251a
```

```
* c1c1c1c (HEAD -> iss2)
| * 04ca8c6 (master)
* | 3283dd7
|/
* a8da338
* fe2251a
```

Scenario: Developer runs `git merge iss5`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.

```
  * d3994b3 (iss6)
* | 0152ac4 (iss5)
|/
* 77a9025 (HEAD -> master)
* cdf1207
```

* d3994b3 (iss6)
* | 0152ac4 (iss5, HEAD -> master)
|/
* 77a9025
* cdf1207

Would this be a fast-forward merge? Explain your answer.

Yes, because all the commits in the version history of the master branch are also in the version history of the iss5 branch.

Scenario: Developer runs `git merge iss7`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.

```
  * g4ca8c6 (iss7)
* | 3283dd7 (HEAD -> master)
|/
* a8da338
* fe2251a
```

* c1c1c1c (HEAD -> master)
|\
| * g4ca8c6 (iss7)
* | 3283dd7
|/
* a8da338
* fe2251a

Would this be a fast-forward merge? Explain your answer.

No, because the version history of the master branch contains a commit (3283dd7) that is not also in the version history of the iss7 branch.

Scenario: Developer runs `git merge master`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.

```
  * ff72baa (HEAD -> iss16)
* | 7b6c2b2 (master)
|/
* 6c61cdb
* 3e27f99
```

* c1c1c1c (HEAD -> iss16)
|\
| * ff72baa
* | 7b6c2b2 (master)
|/
* 6c61cdb
* 3e27f99

Would this be a fast-forward merge? Explain your answer.

No, because the version history of the iss16 branch contains a commit (ff72baa) that is not also in the version history of the master branch.