

# Git Command Semantics

## Practice Problems

Each of the following problems presents a Git log graph (log messages omitted) of a local repo and a scenario. Update the graph by crossing out and/or adding appropriate text.

- You may or may not need to use the two blank lines above the graph.
- If you need to add a commit, use the hash `c1c1c1c`.
- If a command would be rejected by GitHub (e.g., because the remote contains work that you do not have locally), write “REJECTED” on the top line.
- Assume that all remote bookmarks depicted are up to date.

I have included an example problem and solution below to help clarify what’s expected.

### Example Problem

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

```
* 86b8116 (HEAD -> master, iss1)
* dc003f8
* 026c6cf
```

### Example Solution

```
* c1c1c1 (HEAD -> master)


---


* 86b8116 (iss1)


---


* dc003f8


---


* 026c6cf


---


```



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

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

---

---

**\* 04ca8c6 (master)**

---

**\* | 3283dd7 (iss11, HEAD -> iss12)**

---

**|/**

---

**\* a8da338**

---

**\* fe2251a**

---

3. 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**

---

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

```
* d3994b3 (HEAD -> iss13)
* | 0152ac4 (origin/master, master)
|/
* 77a9025 (origin/iss13)
```

---

**\* c1c1c1c (HEAD -> iss13)**

---

**\* d3994b3**

---

**\* | 0152ac4 (origin/master, master)**

---

**|/**

---

**\* 77a9025 (origin/iss13)**

---

5. Scenario: Developer runs `git checkout iss4`.

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

---

---

**\* d197593 (HEAD -> iss4)**

---

**\* | 0f50aa4 (iss3)**

---

**|/**

---

**\* 75a7005 (master)**

---

**\* cbff2e7**

---

6. Scenario: Developer runs `git checkout master`.

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

---

---

**\* d197593 (iss14)**

---

**\* | 0f50aa4 (iss15)**

---

**|/**

---

**\* 75a7005 (HEAD -> master)**

---

**\* cbff2e7**

---

7. 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
```

---

8. Would this be a fast-forward merge?

- a. Yes ✓
- b. No



9. 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**

---

10. Would this be a fast-forward merge?

a. Yes

b. No ✓

11. 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**

12. Would this be a fast-forward merge?

a. Yes

b. No ✓

13. Scenario: Developer runs `git merge iss18`. Assume that auto-merge, if used, would complete successfully with no merge conflicts.

```
* d3994b3 (iss17)
* | 0152ac4 (iss18)
|/
* 77a9025 (HEAD -> master)
* cdf1207
```

---

---

```
* d3994b3 (iss17)
* | 0152ac4 (iss18, HEAD -> master)
|/
* 77a9025
* cdf1207
```

---

14. Would this be a fast-forward merge?

- a. Yes ✓
- b. No

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

```
* a3b25e7 (origin/master, master)
* e41c5b6
* 9d63832 (HEAD -> iss8, origin/iss8)
* 40f26d8
```

---

---

---

```
* a3b25e7 (HEAD -> iss8, origin/master, master)
* e41c5b6
* 9d63832 (origin/iss8)
* 40f26d8
```

---

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

```
* d3994b3 (HEAD -> iss19, origin/iss19)
* | 0152ac4 (origin/master, master)
|/
* 77a9025
```

---

**\* c1c1c1c (HEAD -> iss19)**

---

**| \**

---

**| \* d3994b3 (origin/iss19)**

---

**\* | 0152ac4 (origin/master, master)**

---

**| /**

---

**\* 77a9025**

---

17. Scenario: Developer runs `git push`. Assume that auto-merge, if used, would complete successfully with no merge conflicts. Assume that all issue branches are tracking with their corresponding branches on the remote.

```
* d3994b3 (HEAD -> iss9)
* | 0152ac4 (origin/master, master)
|/
* 77a9025 (origin/iss9)
```

---

---

---

---

```
* d3994b3 (HEAD -> iss9, origin/iss9)
* | 0152ac4 (origin/master, master)
|/
* 77a9025
```

18. Scenario: Developer runs `git push`. Assume that auto-merge, if used, would complete successfully with no merge conflicts. Assume that all issue branches are tracking with their corresponding branches on the remote.

```
* 79bb885 (HEAD -> iss20)
* | 33a99ac (origin/iss20, master)
|/
* d23531d (origin/master)
```

**REJECTED**

---

---

---

---

---

---

---

---

---

---

19. Scenario: Developer runs `git push origin master`. Assume that auto-merge, if used, would complete successfully with no merge conflicts. Assume that all issue branches are tracking with their corresponding branches on the remote.

```
* e28a3c2 (HEAD -> iss10, origin/iss10)
* f40dd3b
* | 1061bb5 (origin/master, master)
|/
* 86b8116
```

# REJECTED

---

---

---

---

---

---

---

---

---

---



20. Scenario: Developer runs `git push`. Assume that auto-merge, if used, would complete successfully with no merge conflicts. Assume that all issue branches are tracking with their corresponding branches on the remote.

```
* e28a3c2 (HEAD -> iss21)
* f40dd3b (origin/iss21)
* | 1061bb5 (master)
|/
* 86b8116 (origin/master)
```

---

---

**\* e28a3c2 (HEAD -> iss21, origin/iss21)**

**\* f40dd3b**

**\* | 1061bb5 (master)**

**|/**

**\* 86b8116 (origin/master)**

---