

1. To get all patches between your current branch and the master branch, do:

1 / 1 point

☐ **git format-patch --all master**

☒ **git format-patch master**

✓ **Correct**

This will generate patches since last synced with master

2. The best way to form patches for use with git is to:

0 / 1 point

☐ Use **diff -Nur**

☒ Work in a text editor while comparing sources

✗ **This should not be selected**

This is a bad procedure and error prone

☒ Use **git-format-patch**

✓ **Correct**

This ensures all formatting is done correctly for git use

3. In modern advanced distributed projects, submitting patches through email is:

1 / 1 point

☒ Something only used for relatively simple submissions, often to subsystem maintainers

☐ The best procedure, lowest common denominator, since not everyone has git

☐ Prohibited

✓ **Correct**

It is just much easier and less error prone to use git itself for larger submissions

4. When applying patches:

1 / 1 point

- ☐ It is good to use the **--interactive** option
- ☒ It is a good idea to test first with the **--dry-run** option
- ☐ It is good to use the **--force** option

☒ **Correct**

It is sensible to test first rather than have to back out a partial success

5. Select the true statement:

1 / 1 point

- ☐ Most modern email clients have built-in support for git, so one need not worry about whitespace, line-wrapping and other potential problems
- ☒ Most modern email clients do not have built-in support for git, so one needs to worry about whitespace, line-wrapping and other potential problems

☒ **Correct**

Email clients are git-ignorant