☰ ⬛ (/learn)   ⚙ 📋

A Simple Cherry-Pick

Learn how you can apply the concept of "cherry-picking" with the help of an example.

We'll cover the following   ⌃

- Port changes from one branch to another
  - Setup
  - Branch off
  - Cherry-picking a commit

Port changes from one branch to another

Since every commit in Git is a change set with a reference ID, you can easily port changes from one branch to another.

Setup

To demonstrate this, create a simple repository with two changes:

```
1    mkdir lgthw_cherry_pick
2    cd lgthw_cherry_pick
3    git init
4    echo change1 > file1
5    git add file1
6    git commit -am change1
7    echo change2 >> file1
8    git commit -am change2
9    git log --all --oneline --decorate --graph
```

⟳

Terminal 1

Terminal

Click to Connect...

Branch off

At this point, branch off into two branches: `master` and `experimental`.
Type the following commands in the terminal given above:

```
10  git branch experimental
11  git checkout experimental
12  echo crazy change >> file1
13  cat file1
14  git commit -am crazy
15  echo more sensible change >> file1
16  cat file1
17  git commit -am sensible
```
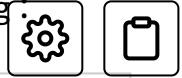
Terminal 1

Terminal

Cherry-picking a commit

You decide that the "sensible" change is the one you want to keep.

First, get the reference ID for that "sensible" commit with a `git log`:

```
18   git log --all --oneline --decorate --graph
```

Then, checkout the master and run a cherry-pick command using the identifier (replace "ID" with the below).

```
19   git checkout master
20   git cherry-pick ID
```


Commit ID to be cherry-picked

Remember to replace the ID below with the commit identifier you found in the `git log` command.

Terminal 1

Terminal

Sometimes a cherry-pick might fail because the `diff` cannot be applied easily, as in this case.

← **Back**

Introduction: Cherry-Picking

Next →

When Cherry-Picking Might Fail

✔ Mark as Completed

Report an Issue

[?] Ask a Question
(https://discuss.educative.io/tag/a-simple-cherry-pick__cherry-picking__learn-git-the-hard-way)