

Lesson

Weekend

Intermediate JavaScript (/intermediate-javascript)

/ Object-Oriented JavaScript (/intermediate-javascript/object-oriented-javascript)

/ Accessing Code From Different Branches

Text

In this section, we will be providing a repository that you can use as a reference as you work through the lessons. There will be links to this repository throughout the section. However, each link will be to a **different branch** in the repository.

Going forward, it will be common for a section to include a reference repository. Some links will point to a specific branch in a repository (which is the case in this section). Other links will point to a specific commit in a repository (which will be the case in the reference repository for other sections). This lesson will focus on working with a reference repository using branches because that is directly relevant to this section.

A link to a reference repository in Learn How to Program will always have this format:

📁 Example GitHub Repo for the Address Book

(https://github.com/epicodus-lessons/oop-address-book-v2/tree/3_unique_ids)

If you click on this link, it will take you to a specific branch in the Address Book project called `3_unique_ids`. In GitHub, we can browse through the code in this specific branch. However, if we were to click on `epicodus-lessons/oop-address-book` (<https://github.com/epicodus-lessons/oop-address-book-v2>) at the top of the page (specifically the `oop-address-book` part), it will take us to the *default* branch. This is usually going to be `main`, but in this case the main branch has been removed and the default branch is `1_address_book_constructors_and_prototype_methods`.

If you navigate around the code of a specific branch or commit on GitHub, make sure you don't accidentally navigate to another commit, branch, or even the most recent code. Otherwise, the code will be different from the reference code you are supposed to be looking at for the lesson.

If you want the reference code locally, you have two choices:

- *Clone* the project and switch between branches. You can easily navigate between branches this way.
- *Download* the code at that particular point in time. You won't need to switch branches to access the reference code, but you also will not be able to access other branches or commits, either.

Accessing Different Branches in a Cloned Project

If you *clone* the project (which is what we usually do), it will automatically be set to the default branch (usually `main`). We can verify this by running the `$ git branch` command in the root directory of the project:

```
$ git branch
1_address_book_constructors_and_prototype_methods
```

Note that none of the other branches are showing. `$ git branch` only shows local branches by default, not remote branches. To see all of the branches, we can do the following:

```
$ git branch -a
* 1_address_book_constructor_and_prototype_methods
  remotes/origin/1_address_book_constructor_and_prototype_m
  ethods
  remotes/origin/2_objects_within_objects
  remotes/origin/3_unique_ids
  remotes/origin/4_finding_and_deleting_contacts
  remotes/origin/5_address_book_user_interface
  remotes/origin/6_adding_interactivity
  remotes/origin/7_event_delegation
  remotes/origin/8_event_delegation_2
  remotes/origin/HEAD -> origin/1_address_book_constructor_
  and_prototype_methods
  remotes/origin/main
```

It's not showing in the example above, but the `remotes/origin` branches will probably be highlighted in red in your terminal.

We can access any of these branches with `$ git checkout` :

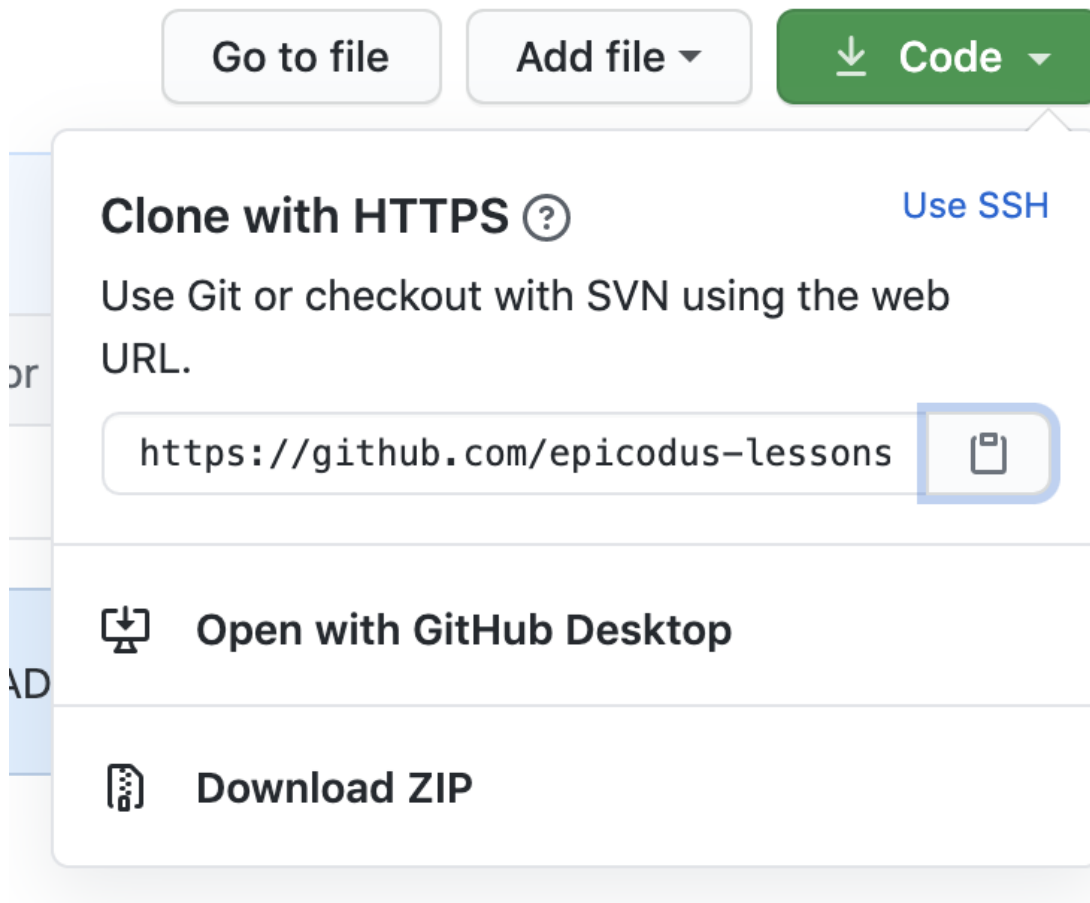
```
$ git checkout 3_unique_ids
Switched to branch '3_unique_ids'
Your branch is up to date with 'origin/3_unique_ids'.
$ git branch
  1_address_book_constructor_and_prototype_methods
* 3_unique_ids
```

As we can see here, we checked out the remote branch `3_unique_ids`. If we run `$ git branch` again, we'll see we are on the new branch — and that it is now a local branch as well.

We can easily navigate to any branch in the project this way.

Accessing A Single Branch in a Downloaded Project

We can also download the project directly by selecting the *Download ZIP* options, as the image below shows.



When we download a project at a specific reference point, whether that's from the main branch, or a different branch, the code from that reference point will be downloaded. This is also the case if you download a project from a commit other than the most recent one.

There's no need to switch branches or commits — you've already got the code from that reference point. While this can be convenient, **the downloaded code is not linked to the Git repository**. If you run `git status` on it, you'll get the error `fatal: not a git repository (or any of the parent directories): .git`. So while you have a reference from that moment in time, you can't switch branches or commits because it's just the code without any git information.

Generally, it's a better idea to clone the code and use `$ git checkout` to navigate to the branch or commit that you want to access. However, it's important to know how the download option is different — and there may be times when you prefer to download the project instead. For instance, if you want to start your own project from a specific commit, but don't want to be tied to the previous commit history, downloading the project is a better option.

[Previous \(/intermediate-javascript/object-oriented-javascript/constructor-and-prototype-methods\)](#)

[Next \(/intermediate-javascript/object-oriented-javascript/objects-within-objects\)](#)

Lesson 8 of 33

Last updated March 23, 2023

[disable dark mode](#)



Epicodus (<http://www.epicodus.com>)

© 2023 Epicodus (<http://www.epicodus.com/>), Inc.