

## COLAB 2: Connecting Github to Google Colab

To connect Github to Google Colab, you need to do the following steps.

1. Follow the instructions to [create a Github account](#) available on Canvas and fork the lectures, but skip the instructions related to downloading "Github Desktop". We will do this in Google Drive instead.
2. [Create a personal access token](#): This gives third-party websites (like Google Drive), access to your Github files.

3. [Upload](#) the following file to Google Drive.

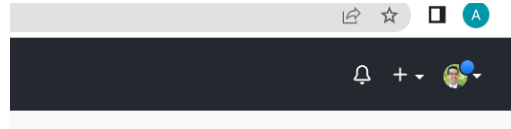
"GITHUB 4\_importGITHUB\_RunOnce.ipynb"

"GITHUB 5\_syncGITHUB\_RunFrequently.ipynb"

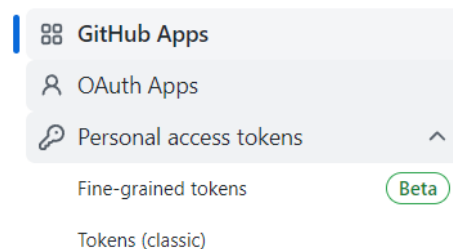
- They don't have to be in any specific location, but make sure to keep them somewhere handy (you will need them again).
  - These files substitute the role of Github Desktop: they will allow you to manually synchronize Drive with Github.
4. [Open these files in Google Colab](#) and follow the instructions. You will have to enter the Personal Token you obtained from Github.
    - As the title suggests, the "import" file copies a repository from your Github account to Google Drive. Bear in mind that this will NOT automatically synchronize. Think of it more as a one-time download.
    - The "sync" file tells you how to sync with the cloud, and to check for updates to the Lecture Notes. You need to run this every time you want to sync.
      - This makes "Github" different from other cloud storage services and it's done that way by design.
      - Programmers typically sync their work with colleagues or team members once they're ready to share what they've been doing.

## Appendix: Creating a Personal Access Token

1. Sign-in to "github.com". Click on the top-right corner with your profile picture.



2. In the drop-down menu select "Settings"
3. Scroll down and search for an option called "Developer settings". You might see this in the panel on the left.
4. Go to personal access-tokens and click on the option "Tokens (classic)".



5. Click on the option "Generate new token" and select the (classic) option.
6. You should see a menu like this:

### New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

What's this token for?

**Expiration \***

30 days

The token will expire on Tue, Feb 7 2023

- Add any name that you like
- Be careful to set the expiration date. By default it's only 30 days but that would be too short for our course.
  - I would set it for six months or "without expiration".

- If you set it for a shorter time, don't worry. If it expires, just generate a new one. In the other instructions we explain how to work with a new token.
  - Remember to never share your Github token with other people, since this is the access to your account information. Only you will have access to the token and any websites you give permission to it.
- Select the following Scopes (these are the permissions). Basically, we're choosing permissions to read and write. This is important to synchronize the data with Google Drive.

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input type="checkbox"/> <b>admin:public_key</b>	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> <b>admin:repo_hook</b>	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input checked="" type="checkbox"/> <b>admin:org_hook</b>	Full control of organization hooks

7. Generate the token and write it down in a safe location.