Student Handbook

Assignment Submission HB

As you progress through L2D, you'll complete one assignment per fortnightly lesson. Assignments are Jupyter Notebook templates in your Assignments folder and must be submitted before the next lesson release (typically before 00:00, 14 days after release). Timely submissions are graded within 14 days; late submissions may delay feedback and certificates. For extensions due to extenuating circumstances, please contact *admin@learntodiscover.ai*.

Below is a step-by-step instruction on how to submit an assignment through GitHub Codespaces. Codespaces allow yo to create a remote, online instance of Microsoft Visual Studio Code. This means that any issues with hardware, software, package dependencies should be taken care of.

While they are mostly great, some students do sometimes run into issues with Codespaces, when submitting assignments. If you are having any difficulties, please contact us at *admin@learntodiscover.ai*.

TIP

We recommend Google Chrome as the best browser for stability and ease, however it should work with the other mainstream browsers.

Assignment Workflow

1. Understanding your assignment folder:

- Navigate to your lesson repository and open the Assignments folder
- Items within: **'XX_Assignment.ipynb'** - a Jupyter notebook containing the assignment questions and spaces to answer them. Please fill in this file.
- Feedback folder - where your feedback will be uploaded within 14 days
- More information on feedback is provided within this folder
- Any data needed for assignment will be in the Data folder located in the

home page/directory

2. Creating a Codespace:

1. Click the green Code button and select the Codespaces tab from your home screen.
2. Click the 'Create codespace on main' button.
3. add new information.

NOTE

Codespace setup may take several minutes and can be affected
by your internet connection. Click Building codespace in the bottom
right to see the your progress. It will indicate, in this log, when the
Codespace creation is complete.

3. Open & set up your assignment:

1. Find your assignment notebook using the file Explorer tab (on the left - an icon of two overlaid pages).

2. In the top right, if you don’t already see L2D_VE displayed, please click Select Kernel (top right) → Select Python Environments.

3. Choose L2D_VE (our pre-configured environment)

NOTE

If properly selected, the name L2D_VE will be displayed in the top right where you selected the kernel originally alongside the Python version. Your Notebooks will then run code, as expected.

4. Complete your assignment:

- Cells are the basis of Jupyter notebooks, in brief they are a distinct container

in a Jupyter Notebook where you write and execute individual pieces of
Python code.

- Cells allow you to break your work into logical segments that can be run

independently.

- Add code cells using the + Code button

- Run cells with the **play button** (a little triangle) or **Ctrl + Enter**

NOTE

Your work autosaves while your internet connection remains active. But we do recommend that, before closing a Codespace tab or window, you save your work manually (in the ‘File’ menu) - just in case.

Submission Workflow

1. Submit your completed assignment:

Only complete these next steps when you are ready to submit your assignment. If you are pausing your work simply exit the browser as normal and begin again with step 2 where under the **Codespaces tab** there will be your old workspace. Your work should be saved within.

1. Click the Source Control tab (fork symbol) on the far left. This opens a VSCode interface for Git control

2. Click the + symbol next to your file to stage changes

3. Enter a commit message (e.g., "Assignment completed”)

4. Click the green Commit button, then press the Sync Changes button.

NOTE

This is the typical workflow of git: Save changes → Stage changes → Commit (with message) → Sync (push) to the main repo (typically in the cloud).

2. Delete your Codespace after use (mandatory):

After submitting your assignment, you **must delete your codespace**. This step is essential, as active codespaces consume computational resources on GitHub's servers. Leaving too many codespaces running may result in GitHub restricting your access to this feature. Each codespace operates as a dedicated virtual computer instance that requires resources to maintain.

1. Return to your repository, click **Code → Codespaces** tab

2. Find your Codespace

3. Click the three dots (...) beside it

4. Select Delete from the dropdown menu

While this workflow might initially seem complex, it familiarises you with GitHub's infrastructure, and enables you to easily share and collaborate on code, should you continue coding in the future.

To help illustrate the workflow, watch the video below:

▷

Video unavailable:

No URL provided

Local Assignment Submission

For those who prefer to work offline or bypass the Codespace method, you can complete assignments on your local machine using Microsoft Visual Studio Code (VS Code). This approach allows you to work without an active internet connection and submit your work when you reconnect by cloning the repository on GitHub to your local repository. From which any changes can be pushed back to main.

Setup requirements

To install all the necessary tools needed to work locally and push back, see our **Python Fundamentals: Induction** tha contains the necessary information to install and setup Python, VS Code, and Git.

Previous Next