Working with Google Colab and GitHub

Ali Etemadi

Tehran University of Medical Sciences

Introduction to Google Colab

- Colaboratory (Colab) is a free, cloud-based service by Google.
- Replicates Jupyter Notebook functionalities in the cloud.
- Accessible via: https://colab.research.google.com/notebooks/welcome.ipynb
- No installation required.

Using Google Colab

- Functions similarly to desktop Jupyter Notebook.
- Requires a free Google account for full feature access.
- Utilizes a cell-oriented paradigm for tasks.
- Example source code may vary due to version differences and hardware.

Features of Google Colab

- Write and run code.
- Create associated documentation.
- Display graphics.
- Works with .ipynb files, similar to Jupyter Notebook.
- Best used with Chrome or Firefox.
- Suitable for data science and Python projects.

- **Executing Code in Colab**
- Select the cell.
- Click the Run button (right-facing arrow).
- Current cell remains selected after execution.

Managing Cell Outputs

- Clear output of individual cells using the block next to the output.
- Hovering over the block shows execution details (e.g., who executed the content).

Additional Cell Options

- Click the vertical ellipsis on the right side of the cell.
- Access a menu of options specific to that cell.

- **Uploading and Saving Code**
- Upload code from your local drive.
- Save to Google Drive or GitHub.
- Access code from any device via these sources.
- **Syncing Across Devices**
- Use Chrome and sync settings across devices.
- Code is available on desktop, tablet, and smartphone.
- Same repository and Chrome setup, different devices.
- **Flexibility vs. Performance**
- Colab offers flexibility but may trade off speed and ergonomics.
- Local Notebook generally executes code faster.
- Viewing and editing code on smaller devices can be challenging.
- **File Download Options**
- Colab supports .ipynb and .py file downloads.
- Notebook offers additional formats: HTML, LaTeX, PDF.
- Limited presentation creation options in Colab.
-

Using Local Runtime Support

- **Benefits of Local Runtime**
- Improved speed and resource access.
- Access to local files on your machine.
- Control over the Notebook version used for execution.
- **Setting Up Local Runtime**
- Colab connects to a local copy of Notebook.
- Requires installation of Notebook on your local system.
- Better speed compared to cloud-based execution.
- **Requirements for Local Runtime**
- Compatible with Windows, Linux, or OS X.
- Requires an appropriate browser (not Internet Explorer).
- **Security Considerations**
- Risk of infection from Notebook code.
- Trust the source of the code.
- Your machine is not open to others sharing code; they use their own runtimes.

Working with Notebooks

New Notebook

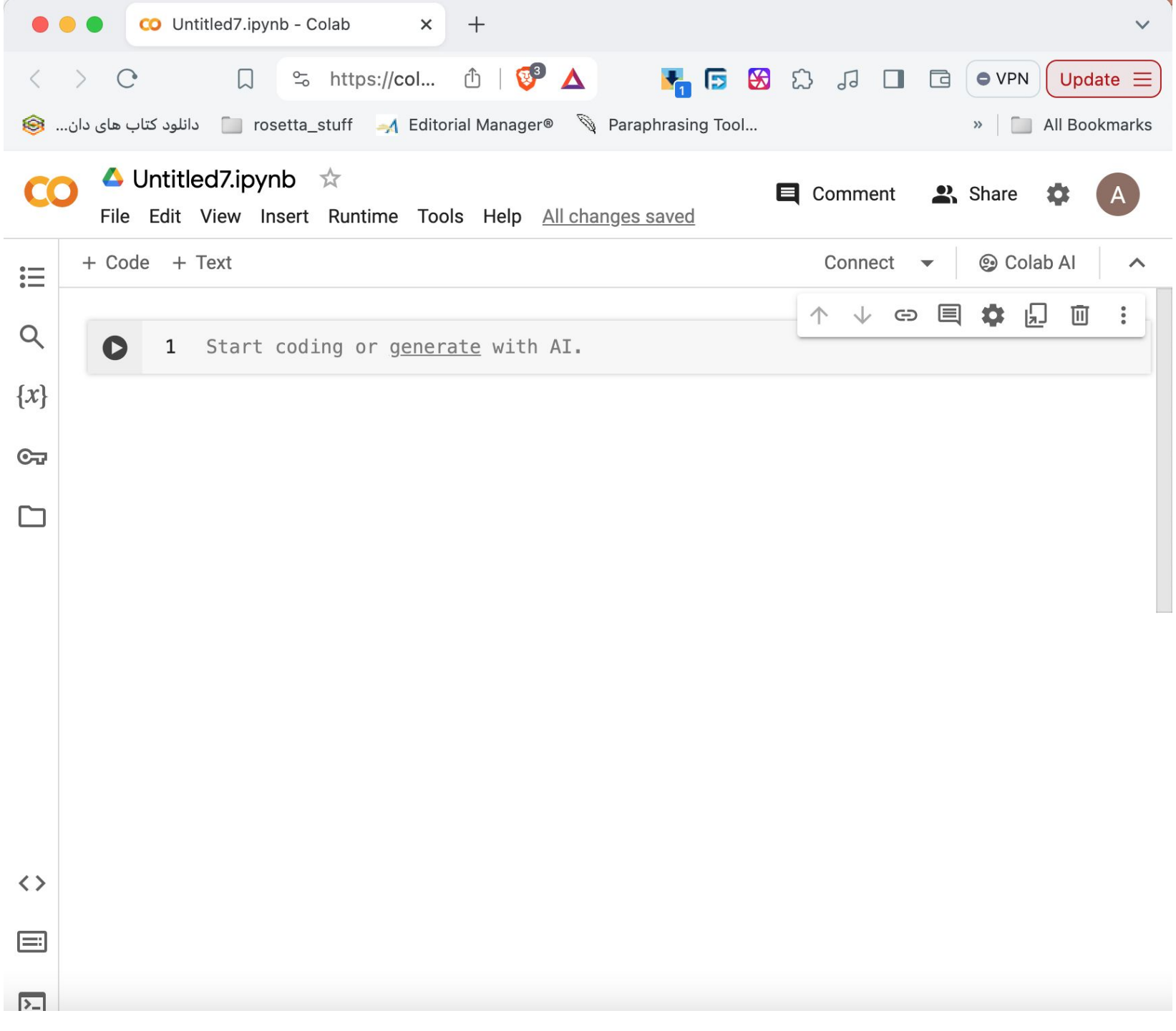
Creating a New Notebook

Go to : https://colab.research.google.com/notebooks/welcome.ipynb

- - Choose `File ➪ New Notebook`.
- - A new Python 3 notebook will open.
- - Click on the filename to change it.

Running Code in Colab

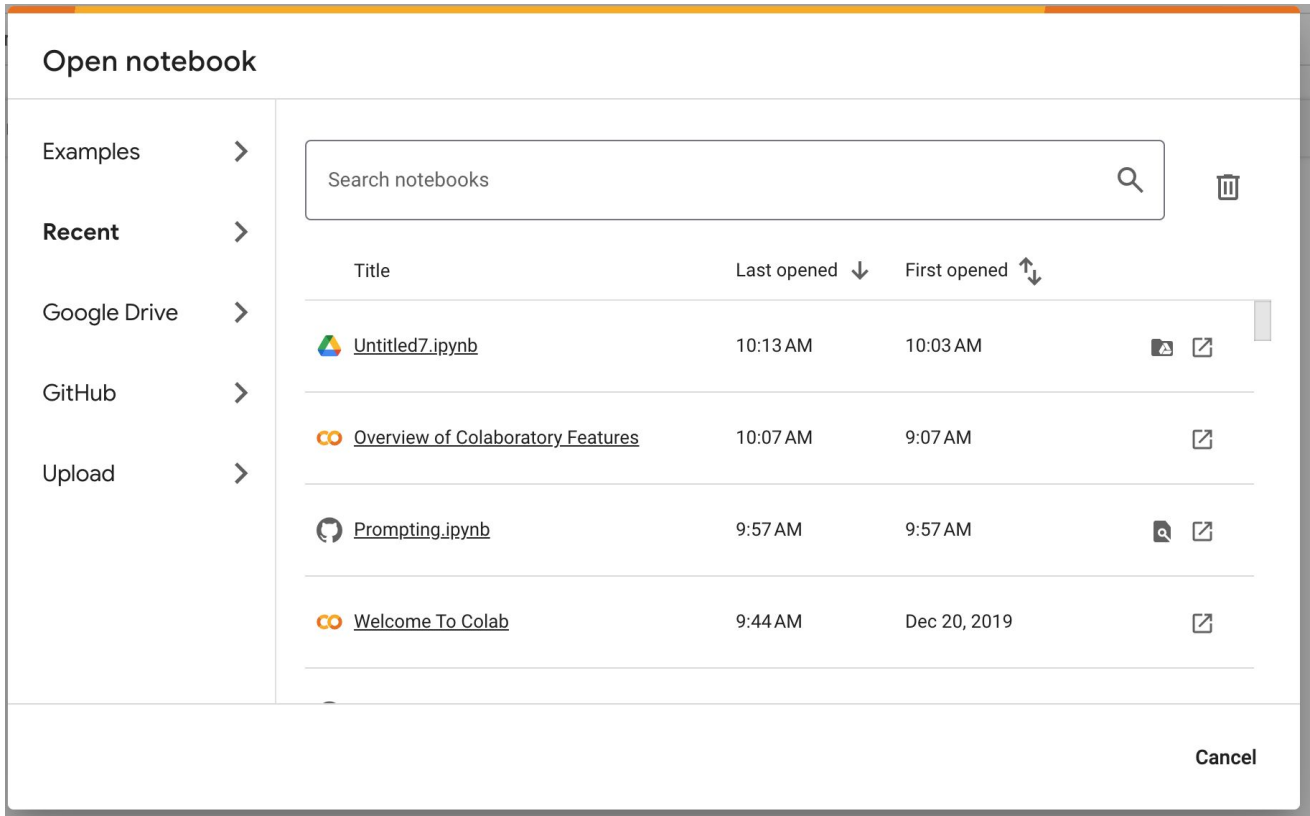
- - Click the right-pointing arrow on the left side of the cell.
- - Cell focus does not change to the next cell automatically.


Untitled7.ipynb - Colab
https://col...
VPN
Update
دانلود کتاب های دان...
rosetta_stuff
Editorial Manager®
Paraphrasing Tool...
All Bookmarks
Untitled7.ipynb
Comment
Share
A
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Connect
Colab AI
1 Start coding or generate with AI.

Opening Existing Notebooks

Steps to Open Notebooks

- - Choose `File ➪ Open Notebook`.
- - A dialog box appears (as shown in Figure)



Sources for Existing Notebooks

- - Local storage
- - Google Drive
- - GitHub
- - Colab examples

Saving Notebooks

- Cloud-Based Saving Options**
- - Colab offers various cloud-based saving options.
- - Local drive saving requires downloading the file.

- Using Google Drive**
- - Default storage location: Google Drive (https://drive.google.com/).
- - Choose `File ➪ Save` to save to the root directory.
- - Select a different folder in Google Drive if needed.

Using GitHub to Save Notebooks

Introduction to GitHub

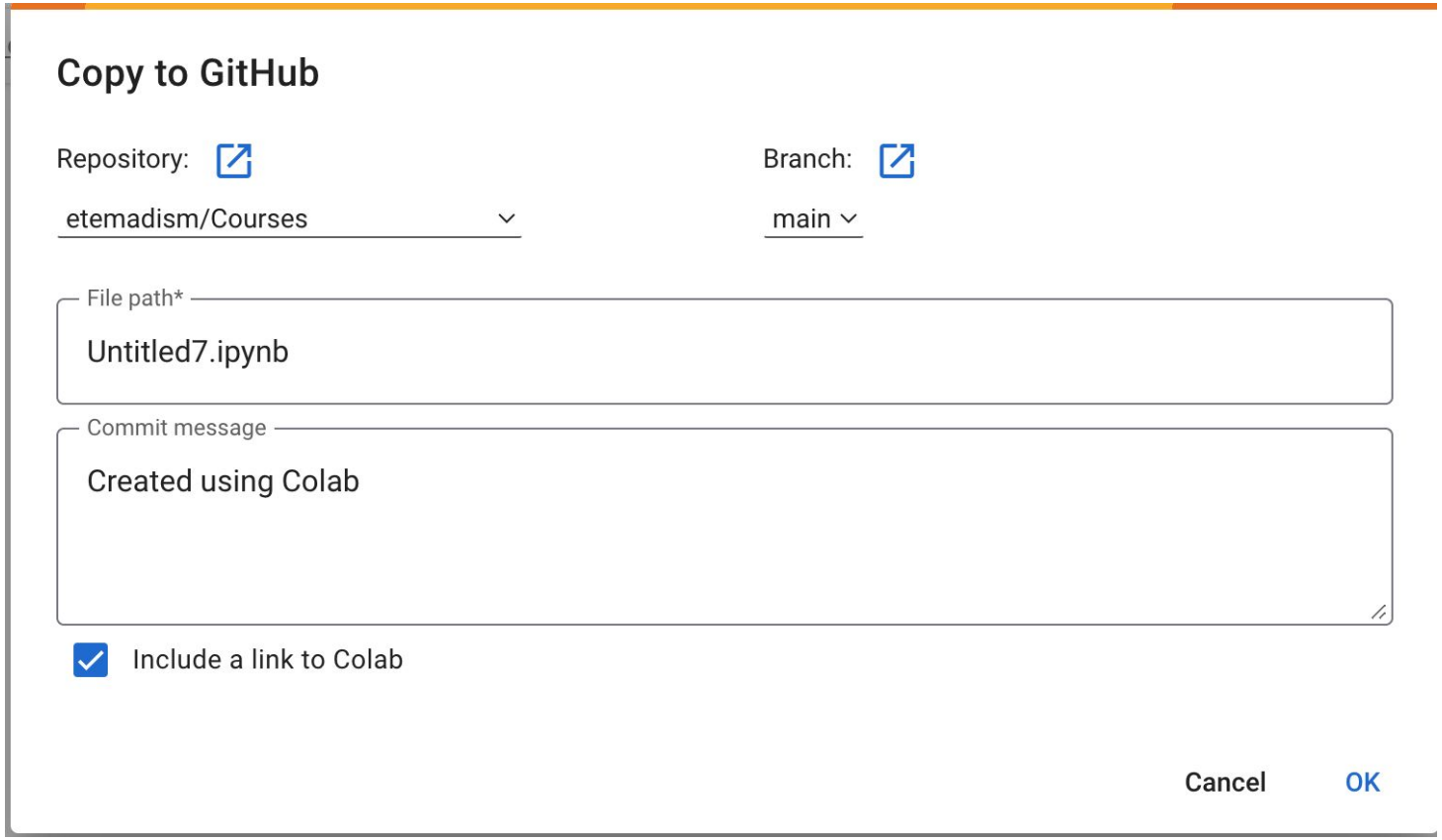
- Alternative to Google Drive for saving content.
- Organized method for sharing code for discussion, review, and distribution.
- GitHub link: https://github.com/

Saving a File to GitHub

- Choose File ➪ Save a Copy in GitHub.
- Sign in to GitHub if prompted.

GitHub Save Dialog Box

- After signing in, a dialog box appears (as shown in Figure).
- Select repository and specify details for saving the file.

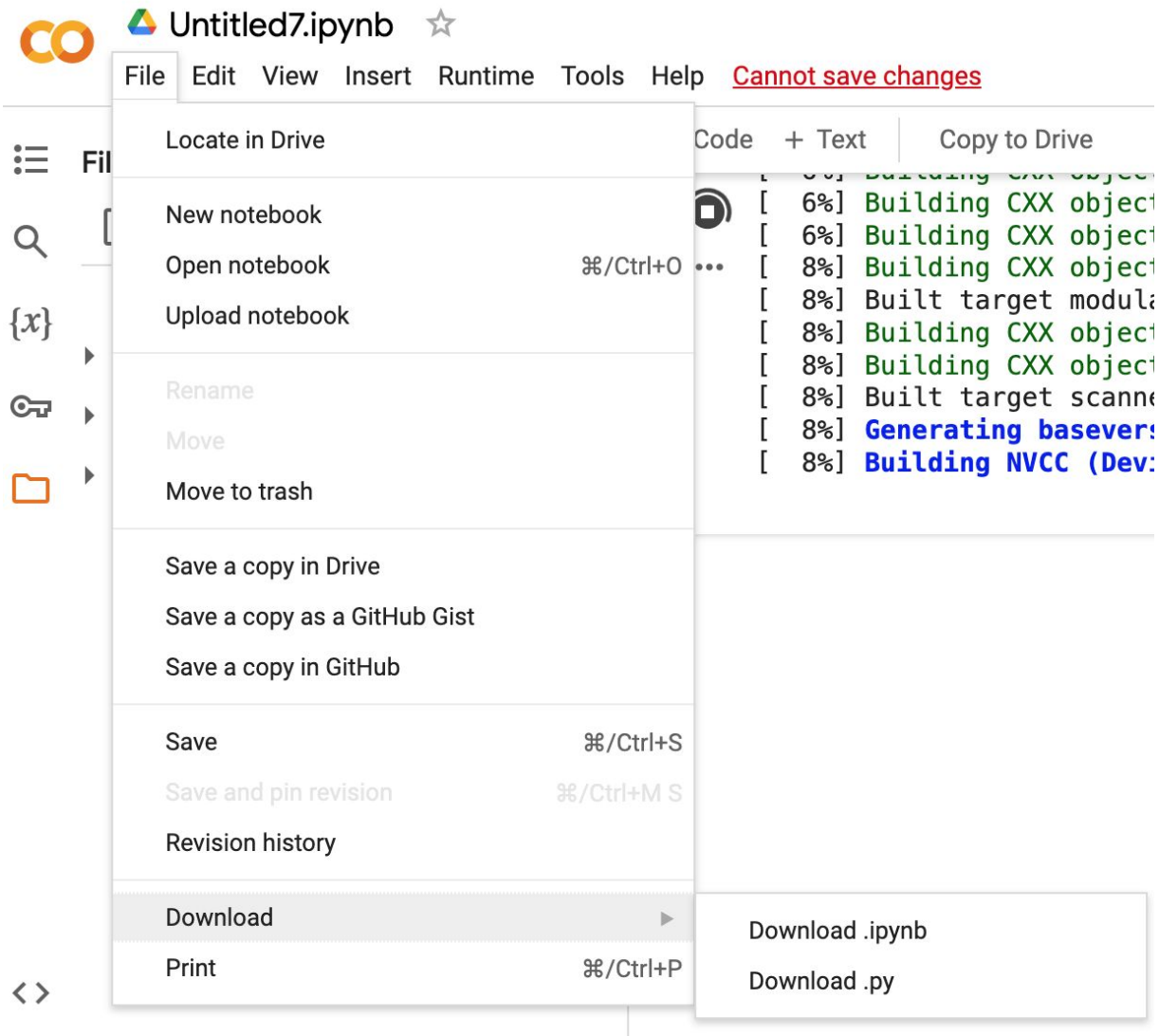


Downloading notebooks

Colab supports two methods for downloading notebooks to your local drive:

- **.ipynb** files (using File➪Download .ipynb)
- and **.py** files (using File➪Download .py).

- In both cases, the file appears in the default download directory for your browser; Colab doesn't offer a method for downloading the file to a specific directory.



Performing Common Tasks

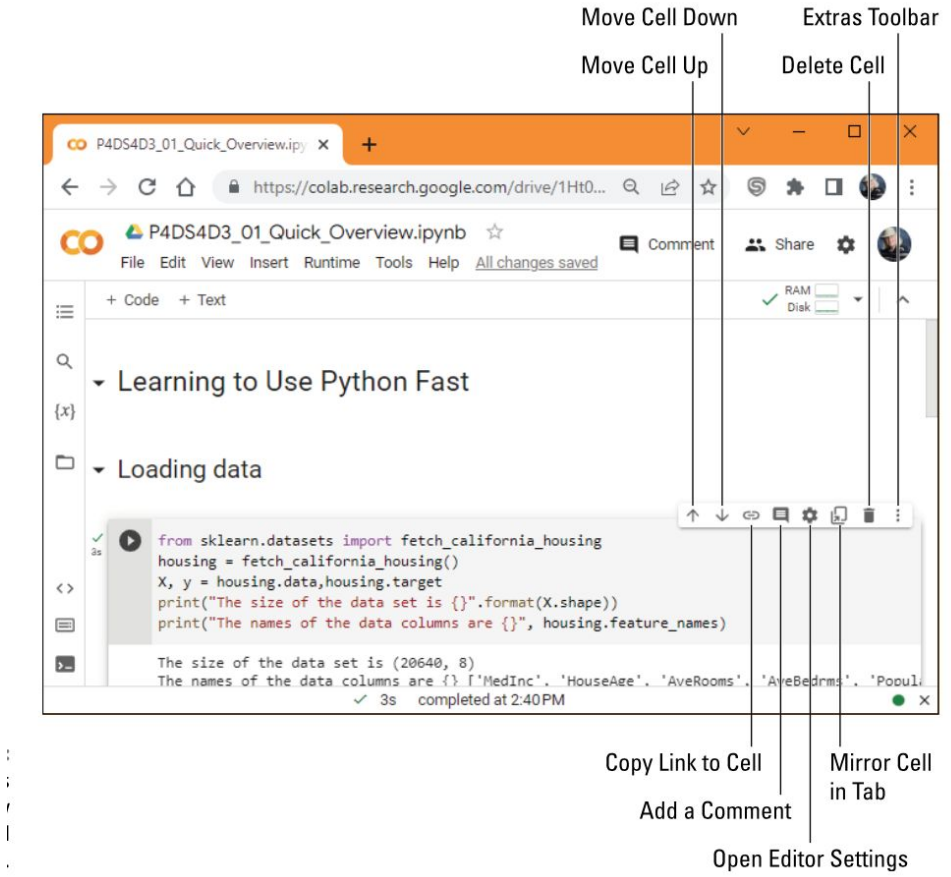
Creating Code Cells

Initial Code Cell**
- First cell in Colab is a code cell.

Accessing Extra Options
- Vertical ellipsis at the rightmost end of the toolbar.

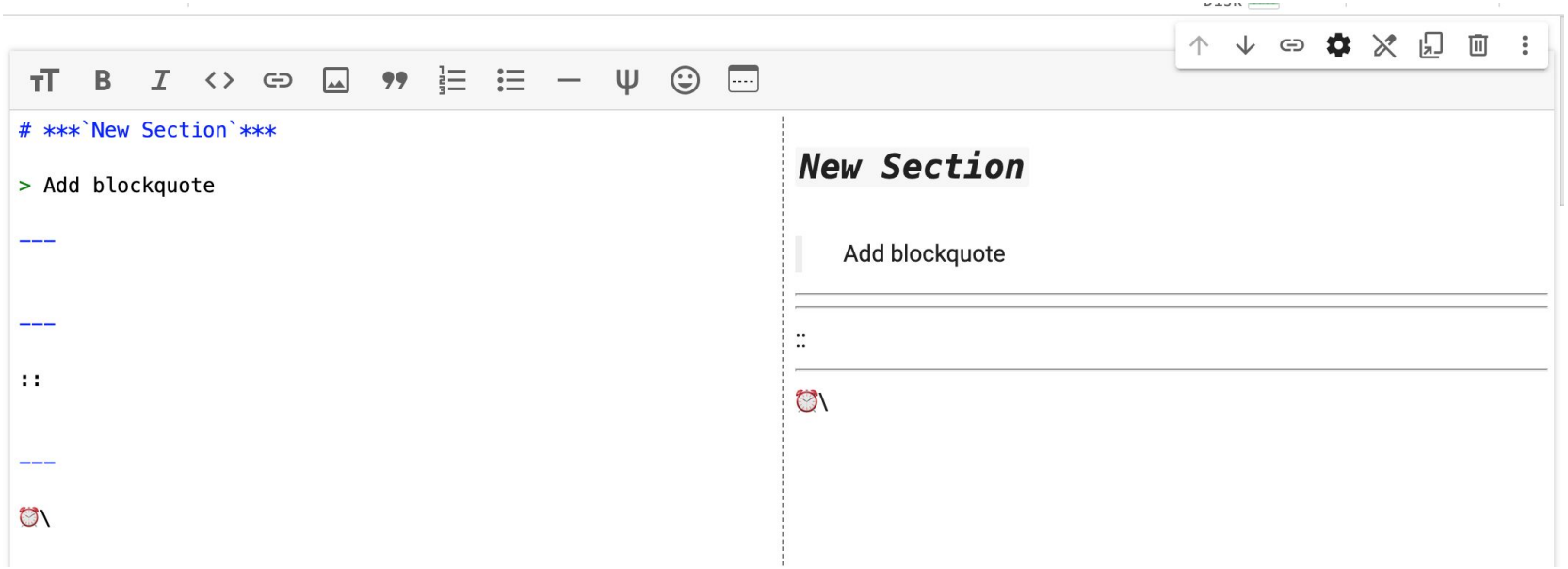
1. **Move Cell Up**: Moves the cell up by one position.
2. **Move Cell Down**: Moves the cell down by one position.
3. **Copy Link to Cell**: Places a link to the cell on the Clipboard.
4. **Add a Comment**: Creates a comment balloon to the right of the cell.
5. **Open Editor Settings**: Modify Colab's behavior.

Slide 69: More Extra Features
6. **Mirror Cell in Tab**: Creates a mirror view of the cell in a side window.
7. **Delete Cell**: Removes the cell from the notebook.



Creating Text Cells

New section
New subsection
New sub-subsection



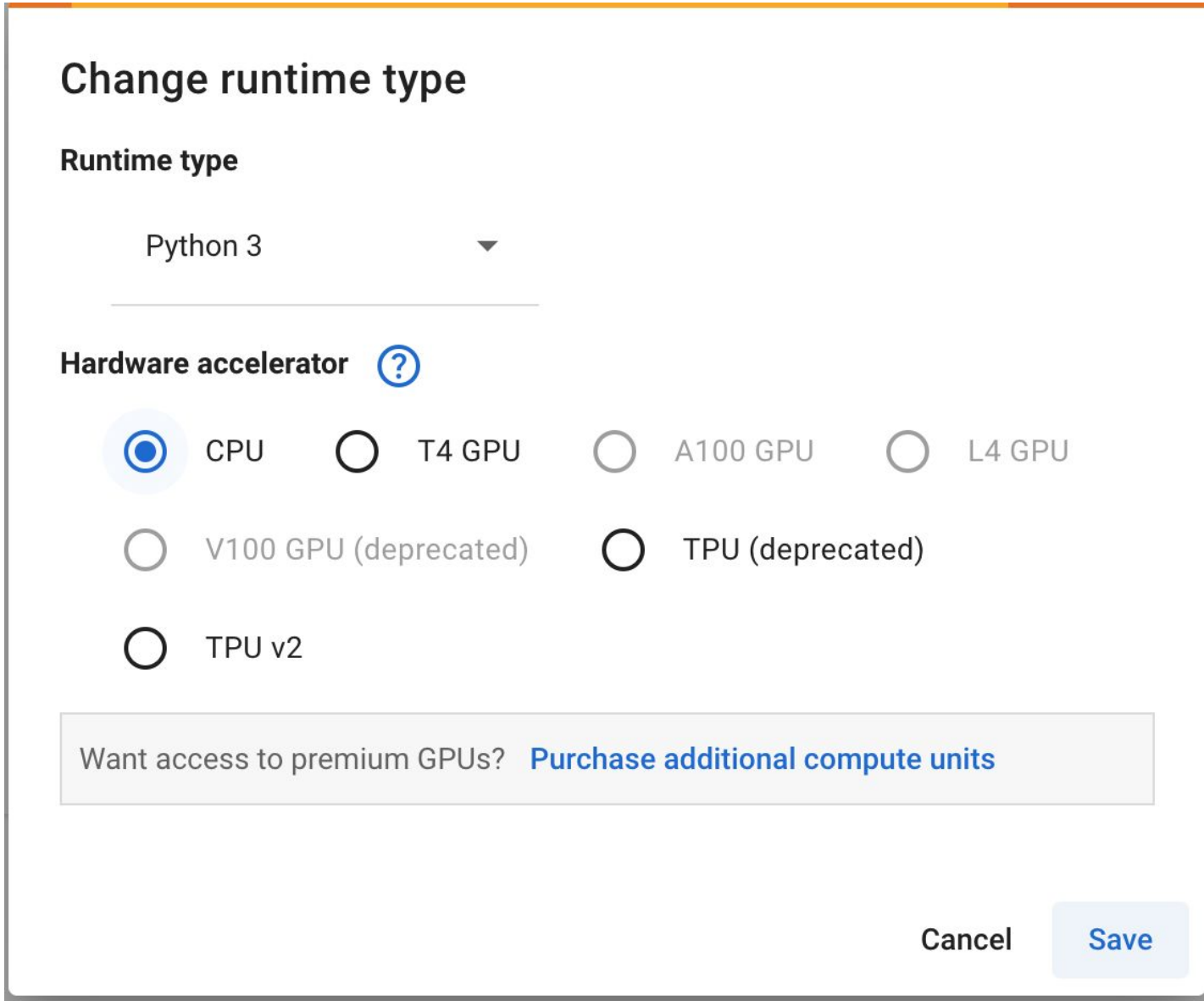
* * Italic
** ** Bold
*** *** Italic an Bold

Reference

Markdown	Preview
`**bold text**`	**bold text**
`*italicized text*` or `_italicized text_`	*italicized text*
`` `Monospace` ``	`Monospace`
`~~strikethrough~~`	~~strikethrough~~
`[A link](https://www.google.com)`	A link
`![An image](https://www.google.com/images/rss.png)`	

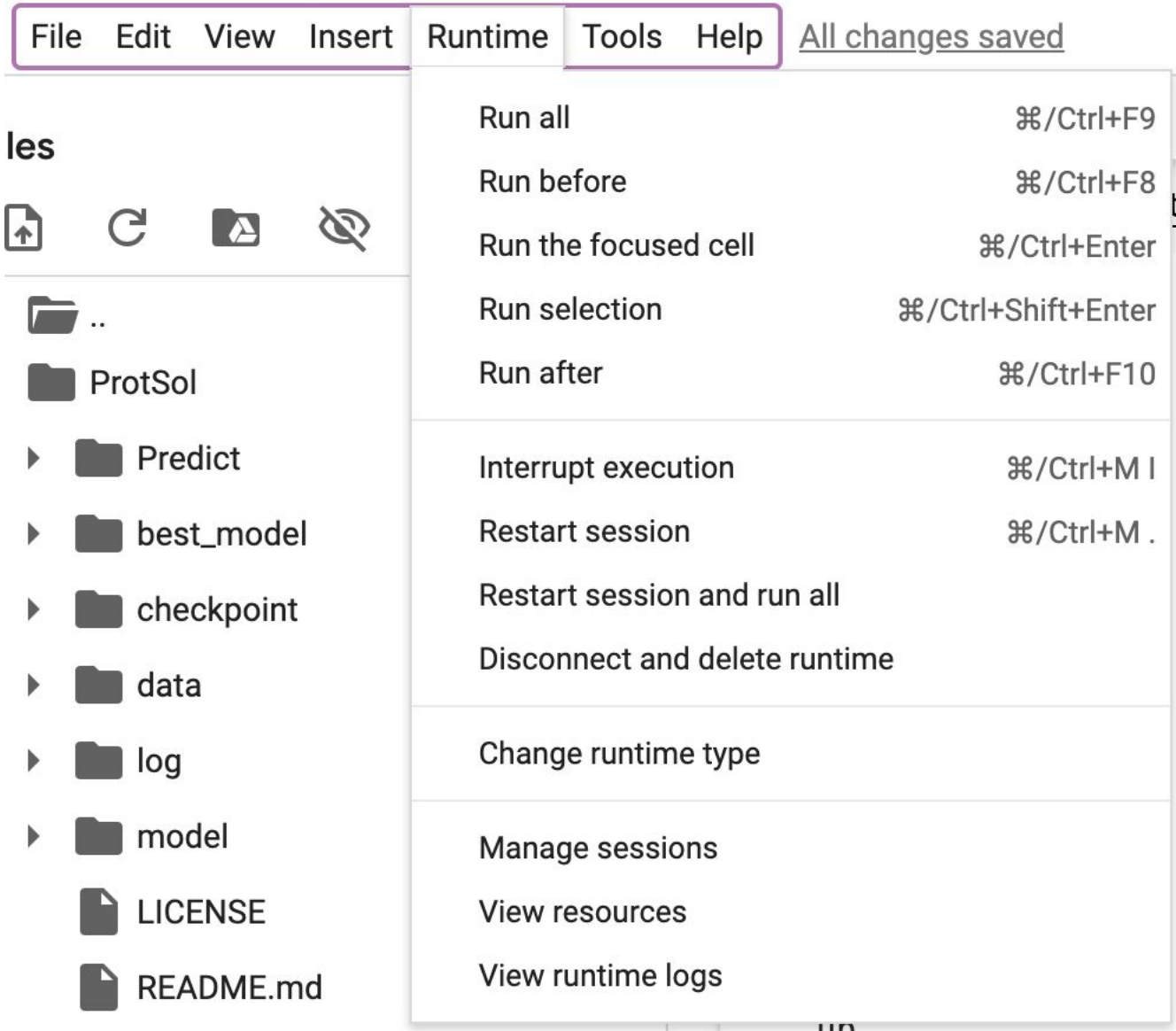
Enabling Hardware Acceleration

- Go to Edit ➪ Notebook Settings to open the Notebook Settings dialog box.
- Options to add GPU and TPU for enhanced performance.



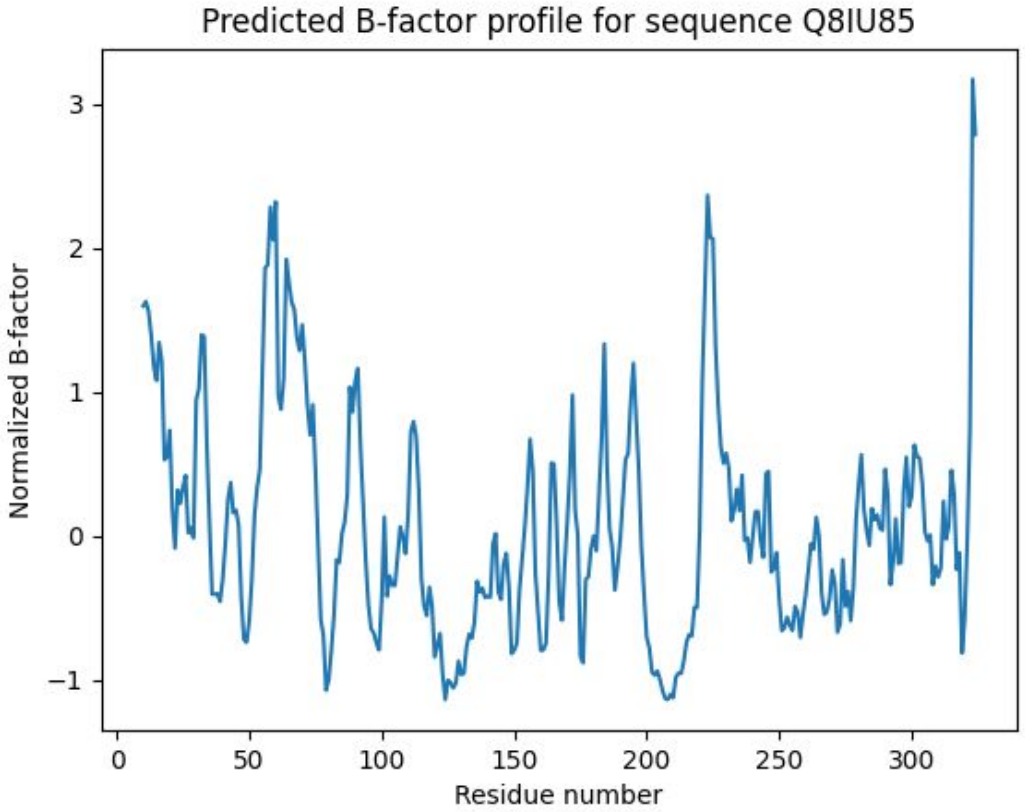
Executing the Code

- **Run Current Cell**:
  - Click the right-pointing arrow or choose Runtime ➪ Run the Focused Cell.
- **Run Other Cells**:
  - Options to run next, previous, or selected cells from the Runtime menu.
- **Run All Cells**:
  - Choose Runtime ➪ Run All to execute all code from the top to the bottom of the notebook.
  - Interrupt execution anytime with Runtime ➪ Interrupt Execution.



Make a colab for running FlexAmino

- Go to https://github.com/upcmarina/flexamino
- And install it on Colab
- Make a run test on this protein: https://www.uniprot.org/uniprotkb/Q8IU85/entry
- You must make a result out of it like the figure in write side.



GitHub

Introduction

- GitHub is a web-based platform that uses Git for version control, allowing multiple people to work on projects simultaneously. It is widely used for software development and version control, providing a range of tools for collaboration, code review, and project management.

Cloning a Repository

1. **Copy the Repository URL:**
   - Go to the GitHub repository page you want to clone, like:

   https://github.com/upcmarina/flexamino.git
   - Click the "Code" button and copy the URL.
2. **Clone the Repository Locally:**
   1. In your terminal, navigate to the directory where you want to store the cloned repository.
   2. Use the **git clone** command followed by the URL you copied:

   git clone https://github.com/upcmarina/flexamino.git

1. **Sign Up for GitHub:**
    1. Go to GitHub and sign up for an account.
2. **Install Git:**
    1. Download and install Git from Git's official site.

Linux: `apt-get install git`

Mac: `brew install git`

1. After installation, configure your Git with your GitHub credentials using the following commands in your terminal:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Create a New Repository

1. Log in to your GitHub account.
2. Click the "+" icon in the top right corner and select "New repository".
3. Name your repository, add a description (optional), choose public or private, and click "Create repository".

Locally on Your Machine:

1. Open your terminal (in VS code) or Git Bash.
2. Navigate to the directory where you want to create your project.
3. Initialize a new Git repository with:

```
git init
```

Making Changes and Committing

1. **Make Changes:**
   - Open your project in VS code and make changes to your files.
2. **Stage Changes:**
   1. In your terminal, stage the changes using:

   git remote add origin
   https://github.com/your-username/your-repository.git

Push to GitHub

- Push your code to the remote repository

  git push -u origin main

In nutshell

Navigate to your project directory

cd path/to/your/project

configure your Git with your GitHub credentials using the following commands in your terminal:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

Initialize Git

git init

Add GitHub repository as remote

git remote add origin https://github.com/your-username/your-repository.git

Add files to staging

git add .

Commit the changes

git commit -m "Initial commit"

Push to GitHub

git push -u origin main

GitHub Codespaces

- **GitHub Codespaces** is a cloud-based development environment provided by GitHub, allowing developers to write, run, and debug their code directly from a web browser. It is built on top of Visual Studio Code and integrates seamlessly with GitHub repositories, making it an excellent tool for collaborative development and contributing to open-source projects.

Enable GitHub Codespaces for Your Repository

1. **Navigate to Your Repository**:
    1. Go to GitHub and navigate to the repository where you want to use Codespaces. https://github.com/upcmarina/flexamino
2. **Enable Codespaces**:
    1. Click the "Code" button on the repository page.
    2. In the dropdown menu, select the "Codespaces" tab.
    3. Click "New codespace" to create a new Codespace.

- **Step 2: Creating and Launching a Codespace**

1. **Create a Codespace**:
    2. After clicking "New codespace," GitHub will start creating your Codespace environment. This may take a few moments.
2. **Launch the Codespace**:
    2. Once the environment is ready, you will be redirected to a web-based Visual Studio Code interface.