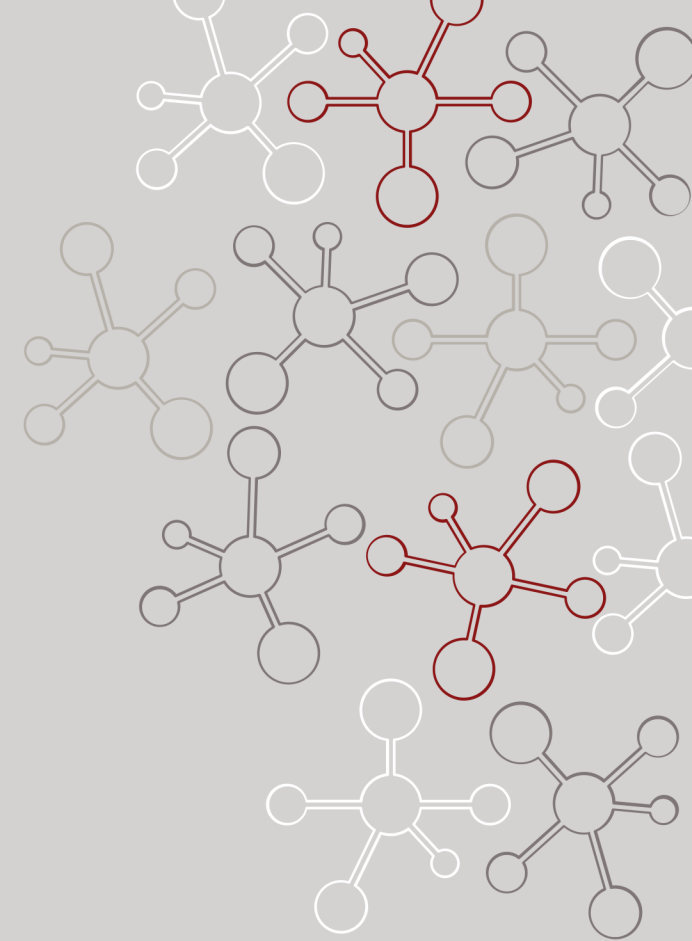


# Getting Started

Michael Kagan  
SLAC

Slides adapted from: J. Ngadiuba



# Github

# Github Basics

3

We will use Github for accessing the project code. For a full tutorial, see [here](#)

To get started:

1. Create Github account: <https://github.com/>
2. Got to the [project repository](#) and fork it
  - A fork is a copy of the repository (repo) in your own account, allowing you to edit and experiment without changing original repo.

Click Here

The screenshot shows the GitHub interface for the repository 'makagan / SSI\_Projects'. The repository is marked as 'Private'. At the top right, there are buttons for 'Unwatch' (1), 'Fork' (0), and 'Star' (0). The 'Fork' button is circled in red, and the text 'Click Here' is written next to it. Below the repository name, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. On the left side, there is a sidebar with a file explorer showing the repository structure: 'jet\_notebooks', 'python\_advanced', 'python\_basics', 'pytorch\_basics', 'slides', and 'LICENSE'. The main content area shows the 'README.md' file. The file is titled 'Machine Learning For Colliders Project for SSI 2023' and contains the text: 'This project will explore LHC Jet data sets and enable participants to develop ML models for jets, such as for classification or generative modeling. More specifically, the data includes various boosted jets, including high-level jet features, jet-images, and per-particle features.'

# Github Basics

4

We will use Github for accessing the project code. For a full tutorial, see [here](#)



To get started:

1. Create Github account: <https://github.com/>
2. Got to the [project repository](#) and fork it

## Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Owner \* Repository name \*

 ml-slac / SSI\_Projects 

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

☒ Copy the `main` branch only

Contribute back to makagan/SSI\_Projects by adding your own branch. [Learn more.](#)

 You are creating a fork in the ml-slac organization.

Create fork

Must be your username →

Click Here

# Github Basics

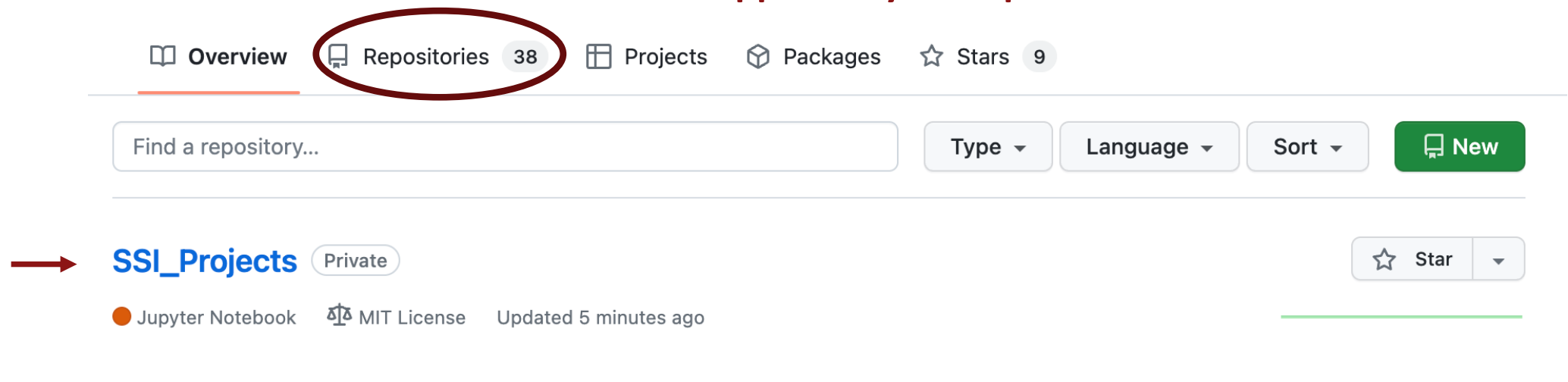
5

We will use Github for accessing the project code. For a full tutorial, see [here](#)

To get started:

1. Create Github account: <https://github.com/>
2. Got to the [project repository](#) and fork it

**Should now appear in your repositories**



# Github Commands I

---

6

Before and throughout the course there will be changes to the [original course repository](#) and **you will need to keep your own fork of the repo up-to-date**

You first need to install git on your computer (if it's not already installed) following [these](#) instructions

Open a terminal, go to your preferred folder, and type the commands:

```
git clone https://github.com/your-username/SSI_Projects.git
cd SSI_Projects
git remote add course https://github.com/makagan/SSI_Projects.git
git fetch course
git merge course/main
git push
```

# Github Commands II

---

7

After the git merge command you will see a list of files that got changed in your local folder with respect to your remote (on github) repository

The last git push command just pushes these changes to your remote repository

At this point your local folder together with your remote repository should be fully synch with the original repository

# Github Commands III

---

8

Whenever you change or add a new file in your local `SSI_Projects` folder and you want to save these local changes to your remote github repository (highly recommended) you go again to the terminal and type:

```
cd SSI_Projects
git status
git add fileX folderX ...
git commit -m "whatever message explaining changes"
git push origin main
```

With the `git status` command you can see the list of changes — be sure you add them all to the commit when using `git add`

Keep in mind that your local/remote changes will merge with the changes in the original course repo when following the steps in previous slide 4 — this might raise conflicts and/ or out-of-synch issues



# Google Colab

We will be using **Colab** to run the hands-on part

- **Colab** is a free platform developed by Google to execute code on the cloud: **you will need a google account**

The project materials are served with Python notebooks through **jupyter**

If you're new to jupyter notebooks, select a cell and hit “shift + enter” to execute the code

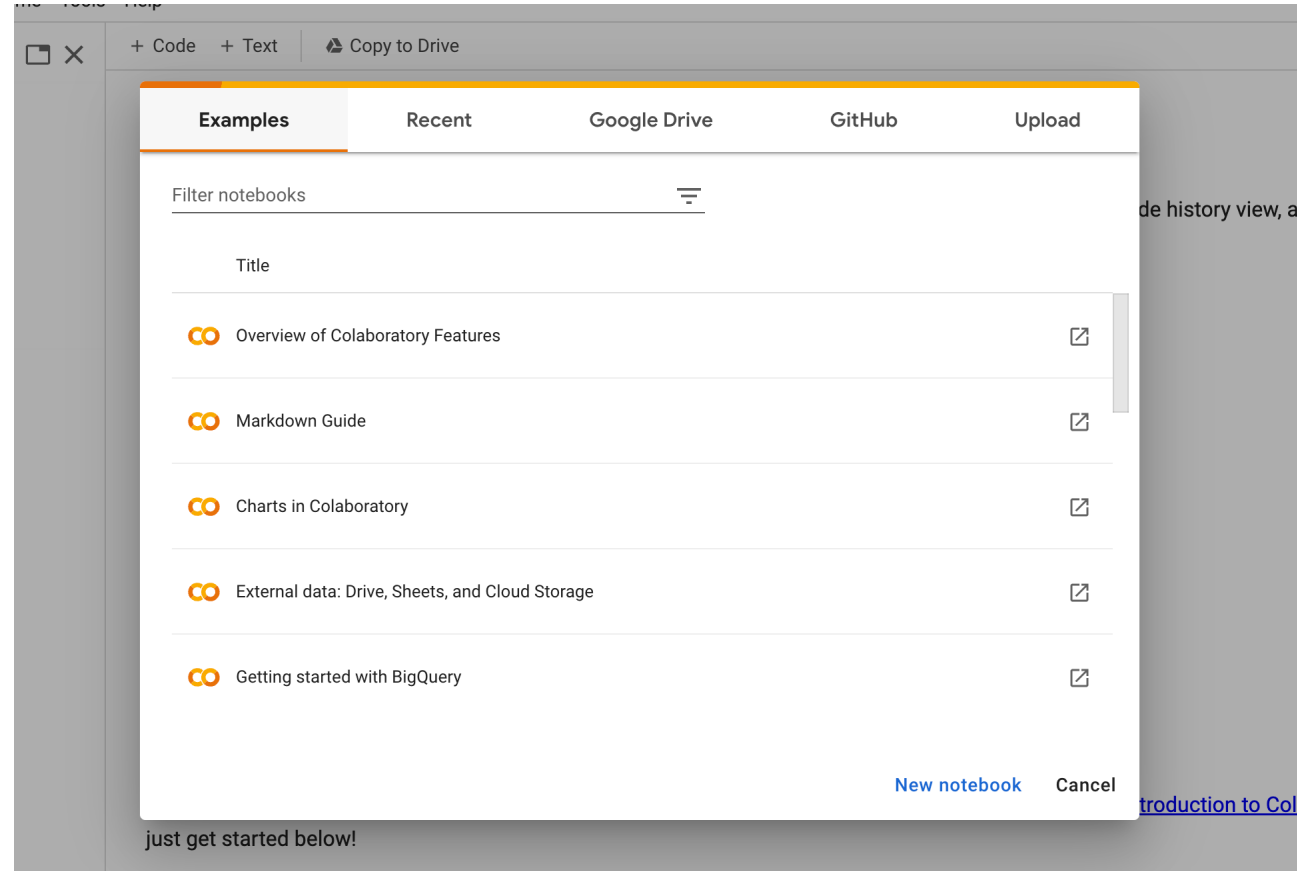
jupyter is rather intuitive but for a full tutorial see [here](#)

# Step 1

11

Make sure you have a Google account

Go to: <https://colab.research.google.com/>



## Step 2: Import from Github

---

12

Click on the `GitHub` tab

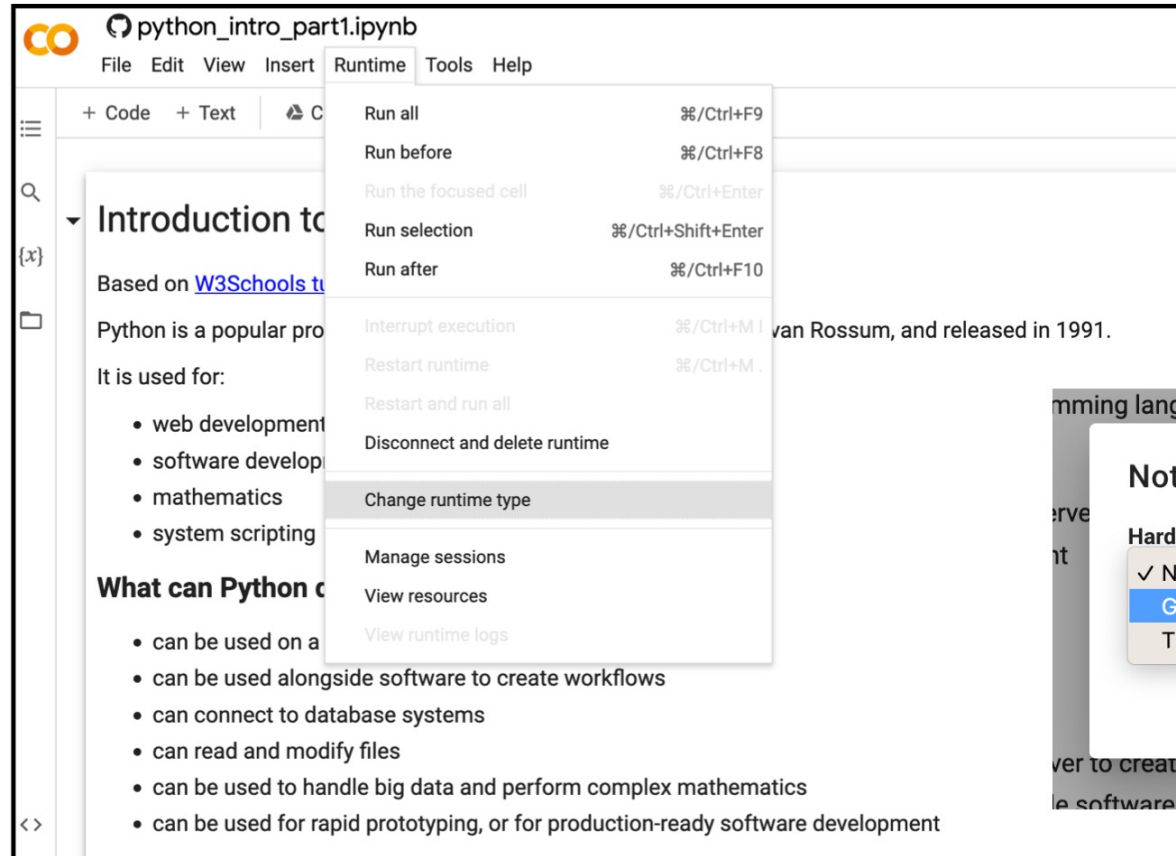
Specify the repo: `makagan/SSI_Projects`

- NOTE: you can specify your own fork of the repo, so you can save changes

Click on one of the `.ipynb` notebooks

# Step 3: Use GPU

13



The screenshot shows a Jupyter Notebook titled 'python\_intro\_part1.ipynb'. The 'Runtime' menu is open, displaying various options. The option 'Change runtime type' is highlighted. The notebook content includes a title 'Introduction to Python', a link to 'W3Schools', and a list of uses for Python: web development, software development, mathematics, and system scripting. Below this, a section titled 'What can Python do?' lists several capabilities, including using Python on a server, creating workflows, connecting to databases, and handling big data.

python\_intro\_part1.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Introduction to Python

Based on [W3Schools tutorial](#)

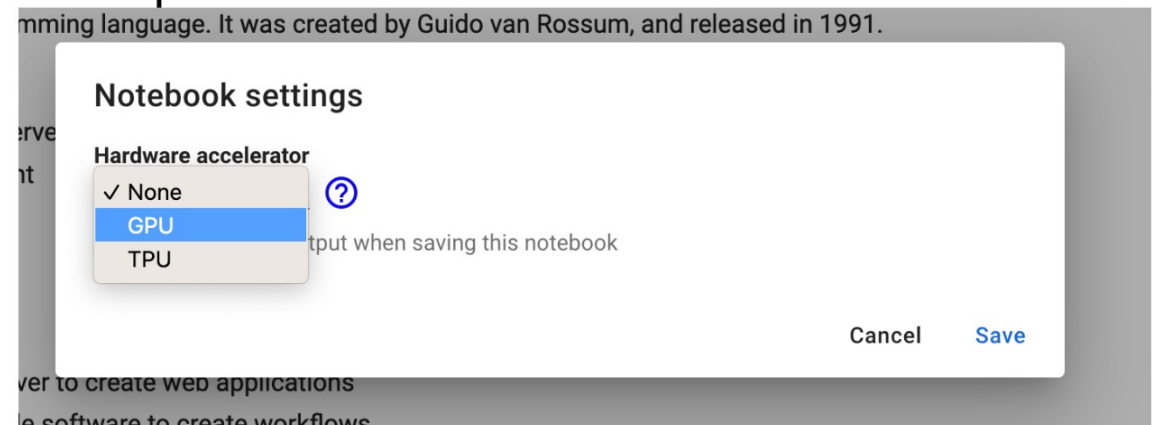
Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development
- software development
- mathematics
- system scripting

**What can Python do?**

- can be used on a server
- can be used alongside software to create workflows
- can connect to database systems
- can read and modify files
- can be used to handle big data and perform complex mathematics
- can be used for rapid prototyping, or for production-ready software development



The screenshot shows the 'Notebook settings' dialog box. The 'Hardware accelerator' section is expanded, showing three options: 'None' (selected with a checkmark), 'GPU' (highlighted in blue), and 'TPU'. A question mark icon is next to the 'GPU' option. The dialog box also includes 'Cancel' and 'Save' buttons.

Notebook settings

Hardware accelerator

- ✓ None
- GPU
- TPU

Cancel Save

# Prerequisites

We will use python for the projects

If you are not familiar with it you must go through some of the basics functionalities by running these two notebooks in Colab (see previous slides) before the course:

- `python_basics/python_intro_part1.ipynb`
- `python_basics/python_intro_part2.ipynb`

The notebooks also contain a few optional simple exercises to help you getting more familiar with it (let me know if you have issues)