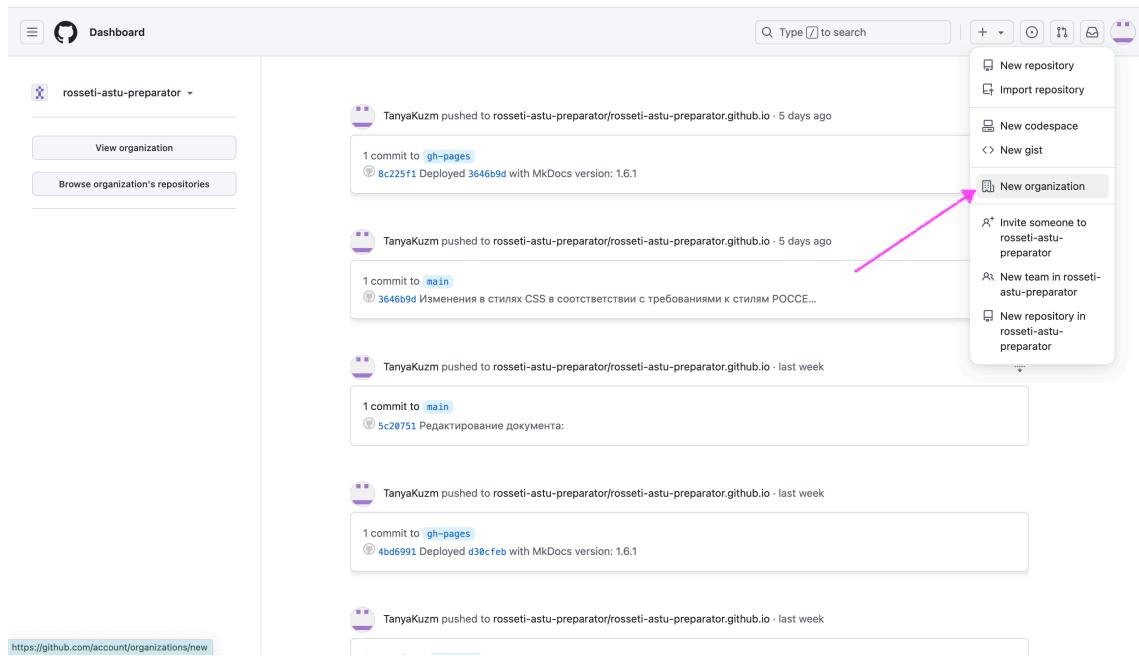


# Step 1: Creating an organization in GitHub

## Create an organization:

- Log in to GitHub.
- Select the "Create an organization" menu.



## Select a pricing plan:

- Choose an appropriate pricing plan for your organization.

The screenshot shows the GitHub pricing page titled "Choose a plan" and "Pick a plan for your organization". It displays three plans:

- Free**: \$0 USD per user/month. Includes: Unlimited public/private repositories, Automatic security and version updates, 2,000 CI/CD minutes/month (Free for public repositories), 500MB of Packages storage (Free for public repositories), Issues & Projects, Community support, and GitHub Copilot Access.
- MOST POPULAR Team**: \$4 USD per user/month. Includes all features of the Free plan plus: Access to GitHub Codespaces, Protected branches, Multiple reviewers in pull requests, Draft pull requests, Code owners, Required reviewers, Pages and Wikis, and Environment deployment branches and secrets.
- Enterprise**: Starting at \$21 USD per user/month. Includes all features of the Team plan plus: Data residency, Enterprise Managed Users, User provisioning through SCIM, Enterprise Account to centrally manage multiple organizations, Environment protection rules, Repository rules, Audit Log API, SOC1, SOC2, type 2 reports annually, and FedRAMP Tailored Authority to Operate (ATO).

At the bottom of the page is a URL: [https://github.com/account/organizations/new?plan=free&ref\\_cta=Create%2520a%2520free%2520organization&ref\\_loc=cards&ref\\_page=%2Forgанизations%2Fplan](https://github.com/account/organizations/new?plan=free&ref_cta=Create%2520a%2520free%2520organization&ref_loc=cards&ref_page=%2Forgанизations%2Fplan)

## Set up the organization:

- Fill in the necessary settings.

The screenshot shows the "Set up your organization" form. It includes fields for:
 

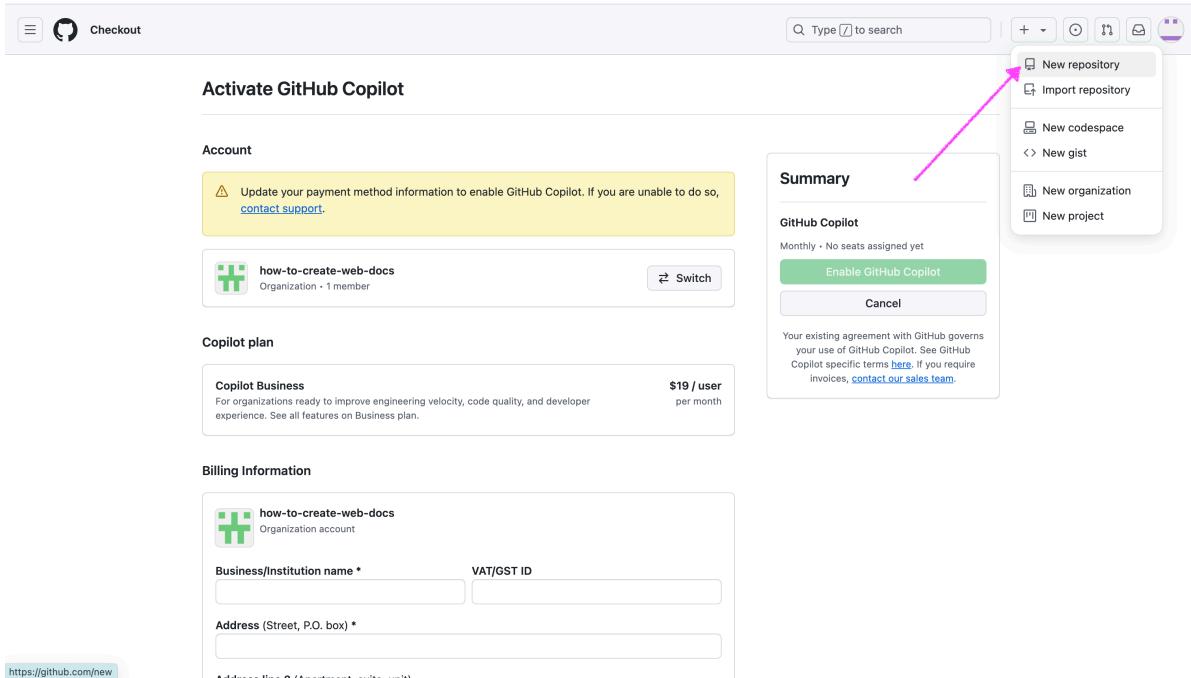
- Organization name \***: how-to-create-web-docs. A note says: "This will be the name of your account on GitHub. Your URL will be: https://github.com/how-to-create-web-docs."
- Contact email \***: tedykuzmina@gmail.com
- This organization belongs to:**
  - My personal account (i.e., TanyaKuzm)
  - A business or institution (For example: GitHub, Inc., Example Institute, American Red Cross)
- Name of business or institution this organization belongs to \***: web-docs. A note says: "This business or institution — not TanyaKuzm (your personal account) — will control this organization."
- Verify your account**: A large empty text area for a verification code.

- Confirm the creation.

► The URL of your future site will be generated from the "Organization name" field and will follow this structure: [https://\[site-name\].github.io/](https://[site-name].github.io/).

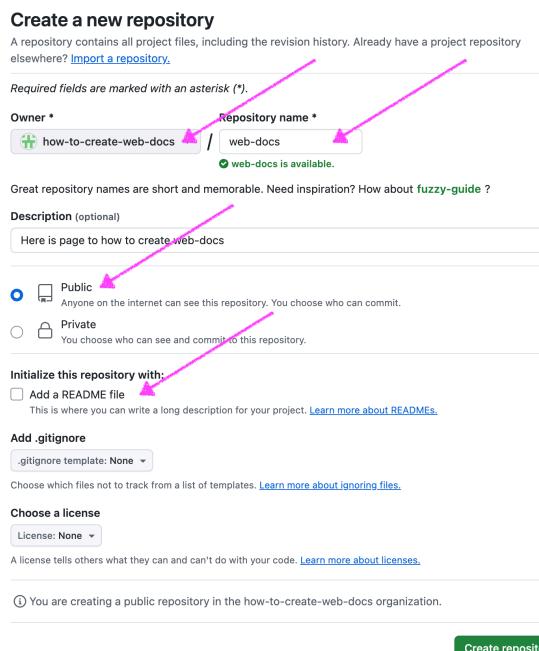
# Step 2: Create a new repository

## Create a repository:



The screenshot shows the 'Activate GitHub Copilot' page. On the right, a sidebar menu is open with several options: 'New repository' (highlighted with a pink arrow), 'Import repository', 'New codespace', 'New gist', 'New organization', and 'New project'. The main area displays account information, a copilot plan (Copilot Business), and billing details for an organization account.

- Create a repository on behalf of the organization.
- Select public/empty (the repository should not contain, for example, README.md).



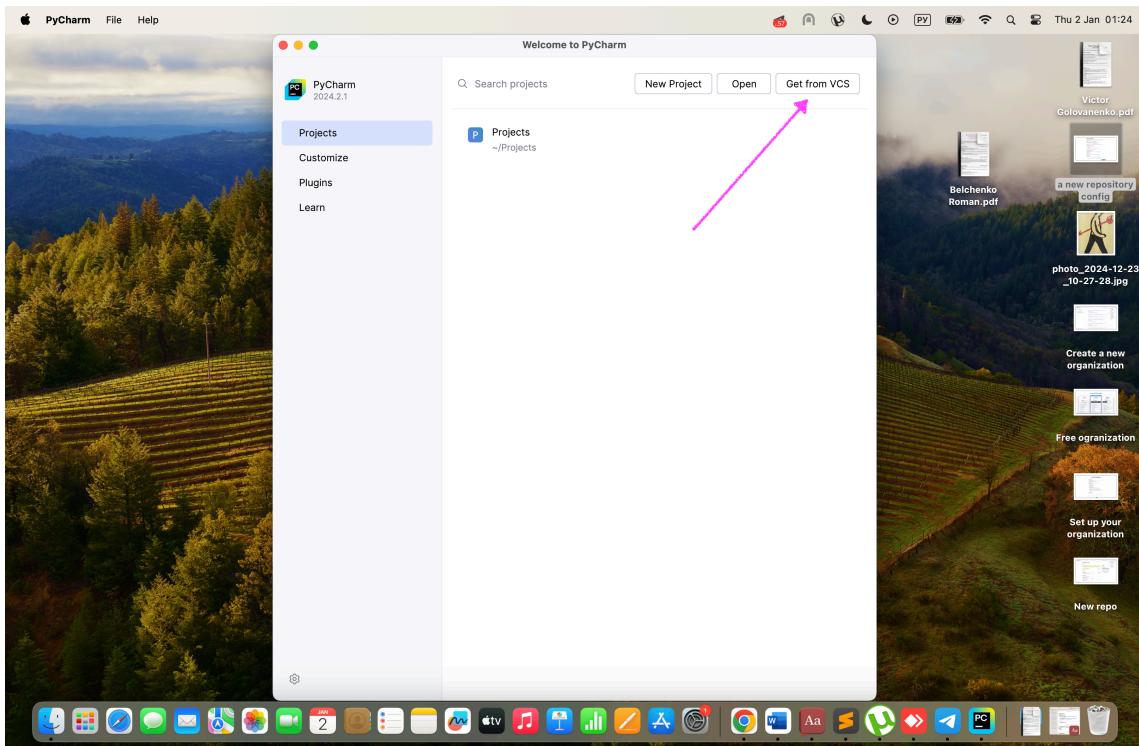
The screenshot shows the 'Create a new repository' form. Key fields highlighted with pink arrows include:

- Owner: 'how-to-create-web-docs'
- Repository name: 'web-docs'
- Visibility: 'Public' (selected)
- Initialization: 'Add a README file' (unchecked)

The form also includes sections for description, .gitignore, license selection, and a note about creating a public repository in the organization.

Clone the repository in PyCharm:

- Copy the repository URL (either HTTPS or SSH).
- In PyCharm, select "Clone Repository" or "Get from VCS".



- Copy the URL from your organization's page in GitHub (I choose SSH).

A screenshot of a GitHub repository page for "how-to-create-web-docs/web-docs". The page has a header with tabs: "Code" (highlighted with a pink arrow), "Issues", "Pull requests", "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". Below the header, there's a section for "GitHub Copilot" and another for "Give access to the people you work with". A large blue box highlights the "Code" tab, the URL input field ("git@github.com:how-to-create-web-docs/web-docs.git"), and the command-line cloning instructions. The URL input field is also highlighted with a pink box. The cloning instructions include both HTTPS and SSH options, along with terminal commands for both.

- Paste the URL and click "Clone".

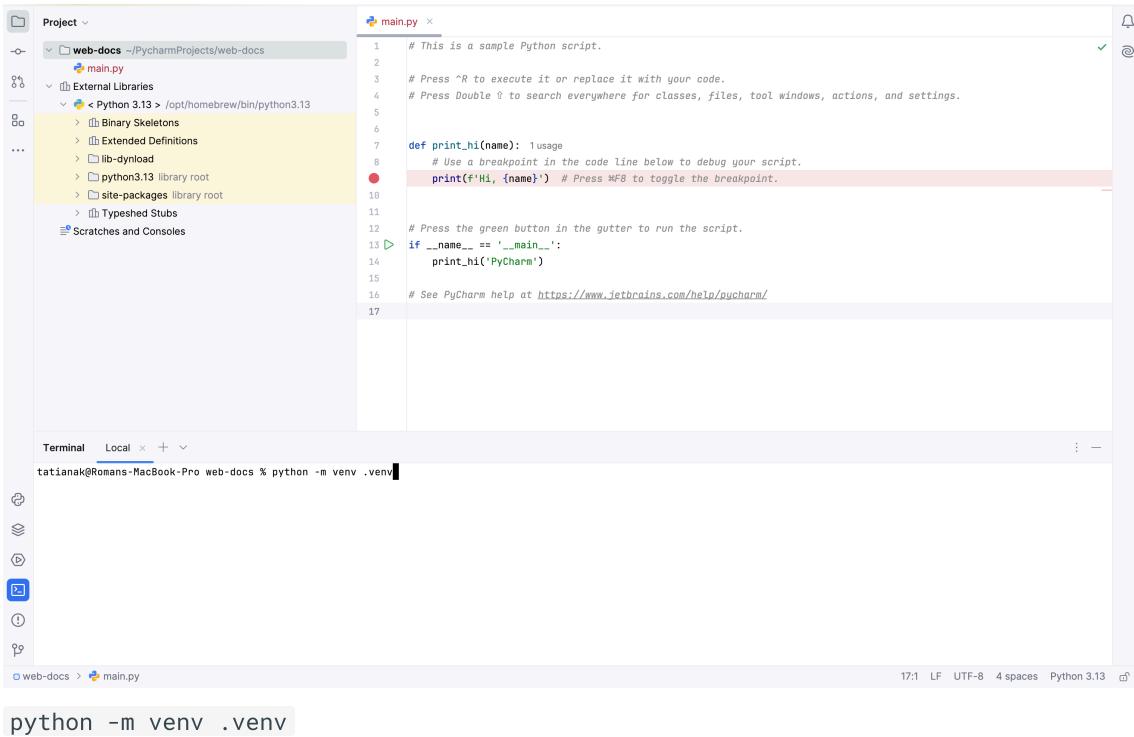
- If you choose **SSH**, make sure you have **SSH** settings configured. We'll discuss why this is more convenient in later posts.

- Click "Trust Project" in the window that opens.

## Step 3: Setting up a virtual environment

### Create a virtual environment:

- Run the following command in the **PyCharm terminal**:



```
# This is a sample Python script.

# Press ^R to execute it or replace it with your code.
# Press Double @ to search everywhere for classes, files, tool windows, actions, and settings.

def print_hi(name): usage
    # Use a breakpoint in the code line below to debug your script.

    print(f'Hi, {name}') # Press ⌘F8 to toggle the breakpoint.

# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/

```

Terminal Local + ×  
tatinak@Romans-MacBook-Pro web-docs % python -m venv .venv  
python -m venv .venv

- If you get a "command not found" error, check your installed Python version and adjust the command accordingly.

```

1 # This is a sample Python script.
2
3 # Press ^R to execute it or replace it with your code.
4 # Press Double ⇑ to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7 def print_hi(name):  usage
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}')  # Press ⌘F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17

```

Terminal Local x + v  
tatianak@Romans-MacBook-Pro web-docs % python -m venv .venv  
zsh: command not found: python  
tatianak@Romans-MacBook-Pro web-docs %

## Your Python version:

```

1 # This is a sample Python script.
2
3 # Press ^R to execute it or replace it with your code.
4 # Press Double ⇑ to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7 def print_hi(name):  usage
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}')  # Press ⌘F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17

```

Terminal Local x + v  
mkdocs-material  
Successfully installed babel-2.16.0 certifi-2024.12.14 charset-normalizer-3.4.1 click-8.1.8 colorama-0.4.6 ghp-import-2.1.0 idna-3.10 jinja2-3.1.5 markdown-3.4.4 mergedep-1.3.4 mkdocs-1.6.1 mkdocs-get-deps-0.2.0 mkdocs-material-9.5.49 mkdocs-material-extensions-1.3.1 packaging-24.2 paginate-0.5.7 pathspec-0.1x.1 plasmashell-2.18.0 pydownExtensions-10.13 python-dateutil-2.9.0.post0 pyyaml-6.0.2 pyyaml-env-tag-0.1 regex-2024.11.6 requests-2.32.3 six-1.17.0 urllib3-2.3.0 wa...

(.venv) tatianak@Romans-MacBook-Pro web-docs % pip3.13 install mkdocs mkdocs-material

Python Console.py

Python Interpreter  
Python 3.13 (Projects)  
Python 3.13  
Python 3.12  
Add New Interpreter...  
Interpreter Settings...  
Manage Packages...

For example, I used: `python3.13 -m venv .venv`

- You can see the **PyCharm** version in the lower right corner of the screen.
- Setting up a virtual environment helps avoid conflicts between packages in different projects.

- It's not possible to install two or more different versions of the same package in one environment. The solution is to create another environment.

**Activate the environment:**

- In the PyCharm terminal, run: `source .venv/bin/activate`

## Step 4: Install MkDocs and the necessary packages

**Install via pip:**

- Run the following in the terminal: `pip install mkdocs mkdocs-material`
- If needed, specify the version with: `pip3.13 install mkdocs mkdocs-material`
- Alternatively, use the "Packages" toolbar in PyCharm.