
An Introduction to VS Code

Peng Lian

2023-06-28

Table of contents

- Overview
- User Interface
- The Editor
- Source control
- Work with Python
- Acknowledgement
- Questions?
- Thanks for your attention!

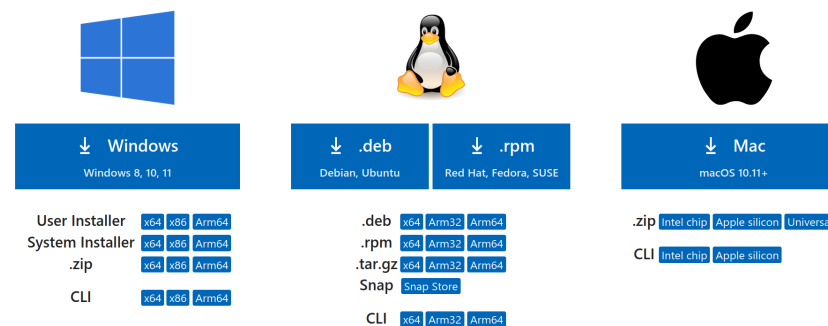
Overview

What is VS Code

Visual Studio Code (VS Code) is a free and open-source code editor developed by Microsoft. It is available for Windows, macOS, and Linux. VS Code is a powerful code editor that can be used for a variety of tasks, including:

- Programming
- Editing text
- Debugging
- Building and running applications
- Extending with extensions

VS Code is a great choice for both beginners and experienced developers. In the Stack Overflow 2022 Developer Survey, VS Code was ranked **the most popular** developer environment tool among 71,010 respondents, with 74.48% reporting that they use it.



Note: [Visual Studio Code](#) is a totally separate product from [Visual Studio](#) the IDE.

Source code: <https://github.com/Microsoft/vscode/>

Documentations: <https://code.visualstudio.com/docs>

VS Code on BioHPC

1. Go to portal.biohpc.swmed.edu -> Cloud Services -> [Web Visualization](#)
2. Open a terminal
3. `module load vscode 1.66.1`
4. `code`

User Interface

Basic layout

The screenshot illustrates the Visual Studio Code interface with five key components labeled A through E:

- A Activity Bar:** Located on the left side, it contains icons for switching between views: Explorer, Search, Source Control, Run and Debug, and Extensions.
- B Side Bar:** Located on the left side, it displays the file explorer for the current workspace, showing a list of files and folders such as `.mention-bot`, `.travis.yml`, `.yarnrc`, `appveyor.yml`, `CODE_OF_CONDUCT.md`, `CONTRIBUTING.md`, `gulpfile.js`, `LICENSE.txt`, `npm-debug.log`, `OSSREADME.json`, `package.json`, `product.json`, `README.md`, `ThirdPartyNotices.txt`, `tsfmt.json`, `tslint.json`, `yarn.lock`, `vscode-docs`, `.vscode`, and `blogs`.
- C Editor Groups:** The central area where code is edited. It shows multiple open files: `findModel.ts`, `findOptionsWidget.ts`, `CONTRIBUTING.md`, `contextmenu.ts`, and `package.json`. The `findOptionsWidget.ts` file is currently selected and shows TypeScript code with imports and class definitions.
- D Panel:** Located at the bottom, it displays the `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, and `TERMINAL` views. The `TERMINAL` view is active, showing a list of files and their sizes, such as `1133 LICENSE.txt`, `607796 npm-debug.log`, `42422 OSSREADME.json`, `3699 package.json`, `683 product.json`, `3732 README.md`, `103675 ThirdPartyNotices.txt`, `729 tsfmt.json`, `11050 tslint.json`, and `203283 yarn.lock`.
- E Status Bar:** Located at the bottom, it displays the current file's name (`master`), line and column numbers (`Ln 1, Col 1`), the number of spaces (`Spaces: 2`), and the current encoding (`UTF-8`).

Side by side editing

The screenshot displays the Visual Studio Code interface with two files open in a split editor:

- extHostApiCommands.ts:** Contains a class `ExtHostApiCommands` with methods `registerCommands()` and `registerCommands()` that register various commands like `vscode.executeWorkspace`, `vscode.executeDefinition`, `vscode.executeHoverPr`, and `vscode.executeDocumen`.
- buildfile.js:** Contains a function `createModuleDescription` and an array of module descriptions for various VS Code views like `vs/workbench`.

On the right side of the image, there is a smaller screenshot of the Visual Studio Code interface showing a terminal window with the following text:

Visual Studio Code - Open Source

build passing build passing

VS Code is a new type of tool that combines the simplicity of a code editor with what developers need for their core edit-build-debug cycle. Code provides comprehensive editing and debugging support, an extensibility model, and lightweight integration with existing tools.

The vscode repository is where we do development and there are many ways you can participate in the project, for example:

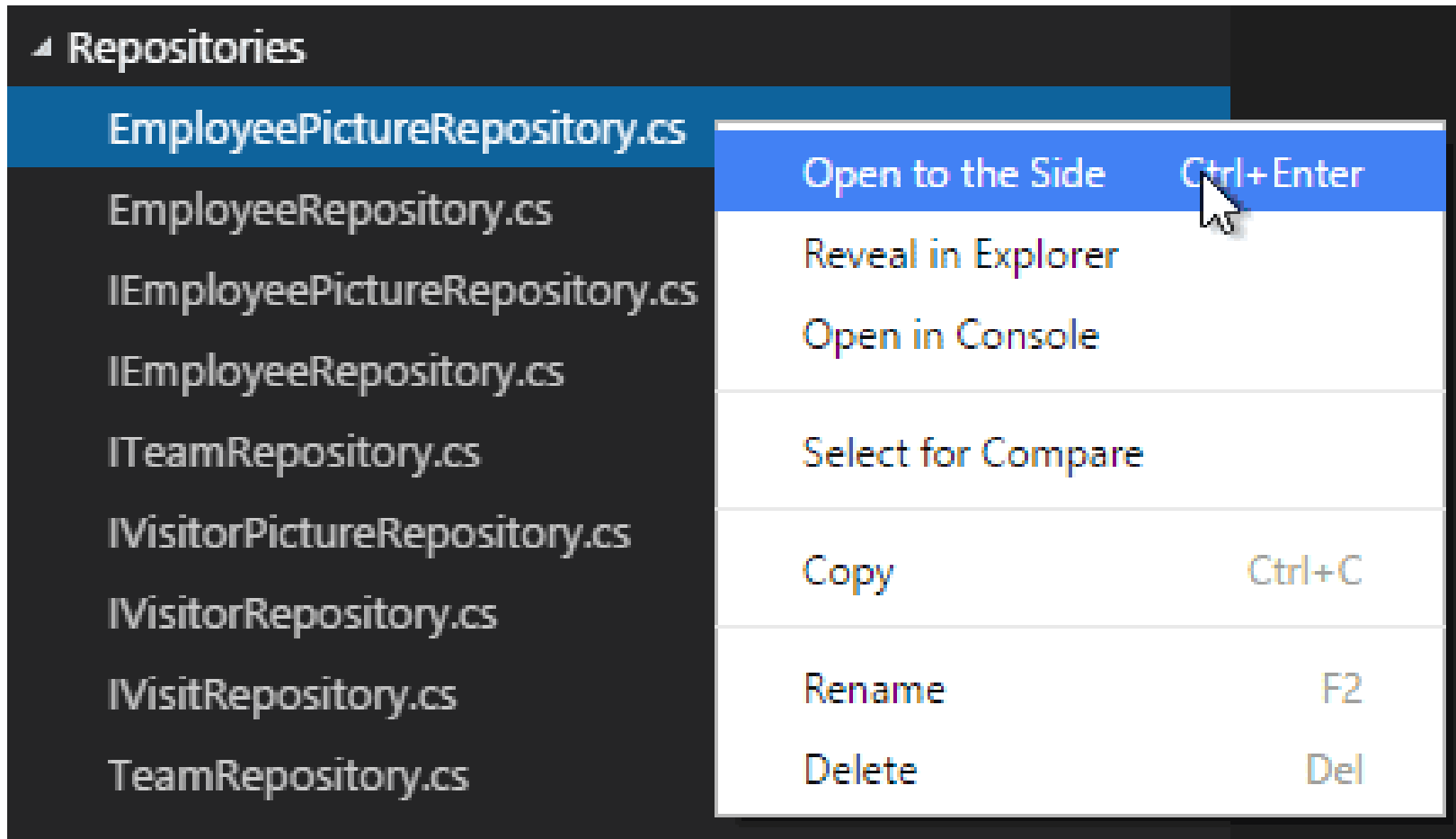
- Submit bugs and feature requests and help us verify as they are checked in
- Review source code changes
- Review the documentation and make pull requests for anything from typos to new content

- Click the **Split Editor** button
- Or **Alt+Click** on a file in the Explorer
- Or drag and drop a tab

Minimap

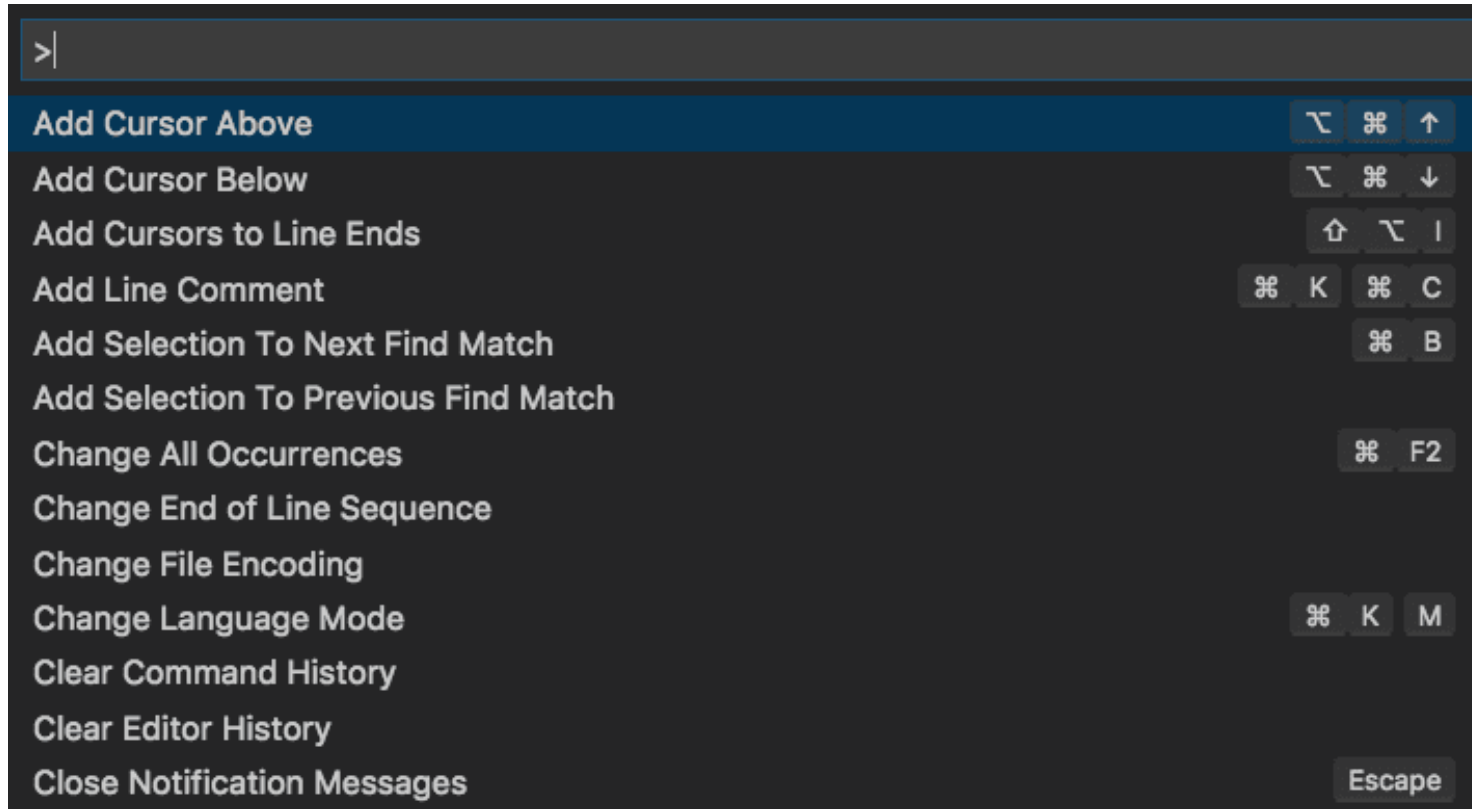
```
JS gulpfile.js x
25  gulp.task('clean-out-folder', common.rimraf('out'));
26
27  gulp.task('clone-vscode-website', ['clean-out-folder'], function (cb) {
28    fs.mkdir('out');
29    process.chdir('./out');
30    git.clone(URL, function (err) {
31      if (err) {
32        console.log('could not clone vscode-website')
33        console.log(err);
34        cb(err);
35      } else {
36        process.chdir('./vscode-website');
37        git.checkout(BRANCH, function (error) {
38          console.log('checked out branch:', BRANCH);
39          process.chdir('../..');
40          cb(error);
41        });
42      }
43    });
44  });
45
46  gulp.task('commit', function () {
47    process.chdir('./out/vscode-website');
```

Explorer



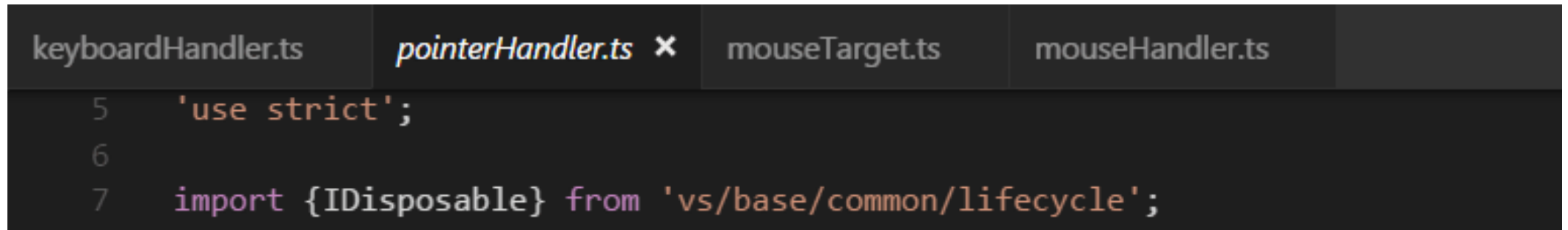
A high-level overview of the source code, which is useful for quick navigation and code understanding.

Command Palette



The Command Palette provides access to many commands. You can execute editor commands, open files, search for symbols, and see a quick outline of a file.

Preview mode

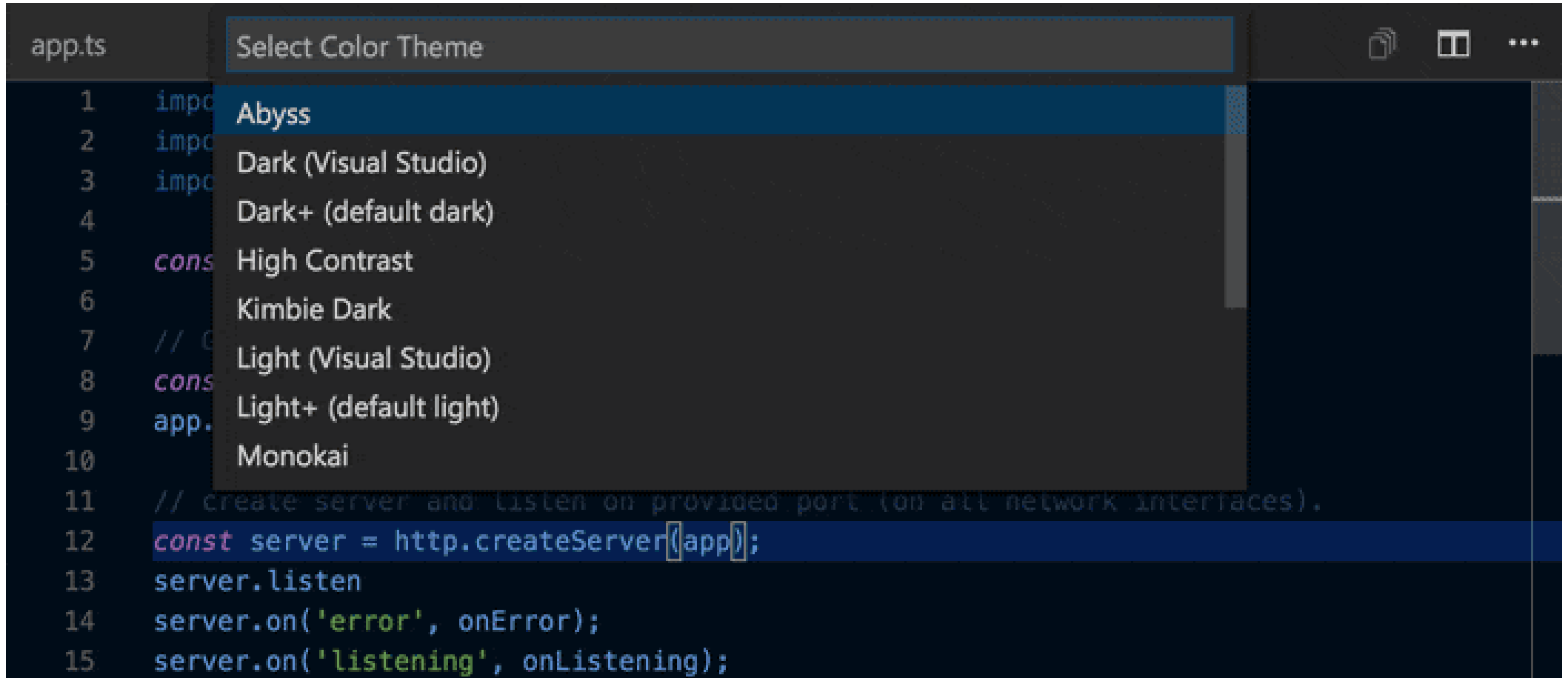
A screenshot of a code editor interface. At the top, there are four tabs: 'keyboardHandler.ts', 'pointerHandler.ts' (which is italicized to indicate preview mode), 'mouseTarget.ts', and 'mouseHandler.ts'. Below the tabs, the code editor shows three lines of TypeScript code: line 5: `'use strict';`, line 6: (empty), and line 7: `import {IDisposable} from 'vs/base/common/lifecycle';`

```
keyboardHandler.ts  pointerHandler.ts ✕  mouseTarget.ts  mouseHandler.ts
5  'use strict';
6
7  import {IDisposable} from 'vs/base/common/lifecycle';
```

When you single-click or select a file in the Explorer, it is shown in a preview mode and reuses an existing Tab. This is useful if you are quickly browsing files and don't want every visited file to have its own Tab. When you start editing the file or use double-click to open the file from the Explorer, a new Tab is dedicated to that file.

Preview mode is indicated by *italics* in the Tab heading.

Selecting the Color Theme

A screenshot of a code editor window. The title bar shows 'app.ts' and a 'Select Color Theme' dropdown menu is open. The menu lists several themes: Abyss, Dark (Visual Studio), Dark+ (default dark), High Contrast, Kimbie Dark, Light (Visual Studio), Light+ (default light), and Monokai. The 'Abyss' theme is currently selected and highlighted in a darker blue. The background of the editor shows a dark theme with some code visible, including a line with 'const server = http.createServer(app);' which is highlighted in blue.

```
app.ts
1  impo
2  impo
3  impo
4
5  cons
6
7  // C
8  cons
9  app.
10
11 // create server and listen on provided port (on all network interfaces).
12 const server = http.createServer(app);
13 server.listen
14 server.on('error', onError);
15 server.on('listening', onListening);
```

Open the Color Theme picker:

- File → Preferences → Theme → Color Theme
- Or **Ctrl+K Ctrl+T**

The Editor

Settings

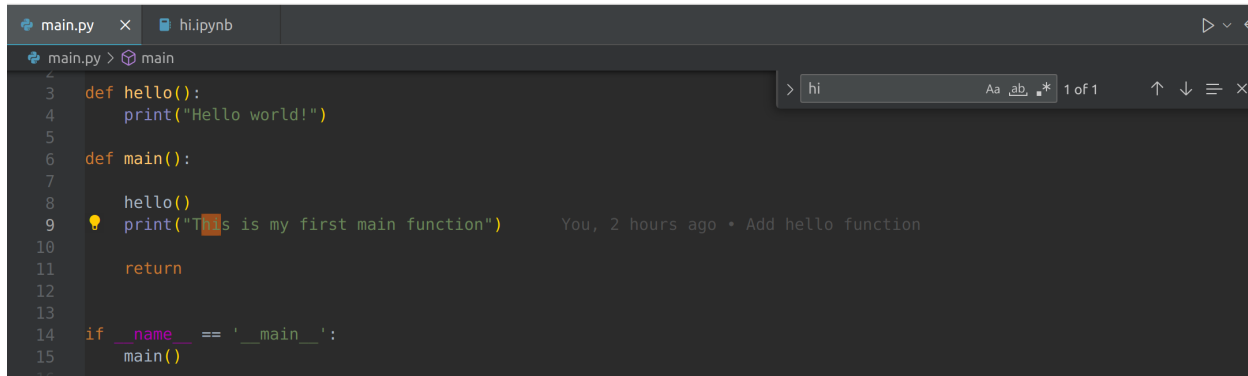
- Settings
 - File -> Preferences -> Settings
 - Or `Ctrl+,`
 - Global settings: `~/ .config/Code/User/settings.json`
 - Project settings: `.vscode/settings.json`
- Settings Sync
 - File -> Preferences -> Settings Sync
- Zen mode Zen Mode lets you focus on your code by hiding all UI except the editor, going to full screen and centering the editor layout.
 - View -> Appearance -> Zen mode
 - `Ctrl+K Z`
 - **Double Esc** exits Zen Mode

Multiple selections

```
31 .global-message-list.transition {  
32 →   -webkit-transition: top 200ms linear;  
33 →   -ms-transition:      top 200ms linear;  
34 →   -moz-transition:     top 200ms linear;  
35 →   -khtml-transition:   top 200ms linear;  
36 →   -o-transition:       top 200ms linear;  
37 →   transition:          top 200ms linear;  
38 }
```


Find and replace

- **Ctrl+F** ~ Find string in editor or find file in explorer
- **Enter** or **Shift+Enter** to navigate



The screenshot shows a code editor with two tabs: 'main.py' and 'hi.ipynb'. The 'main.py' tab is active, displaying the following Python code:

```
3 def hello():
4     print("Hello world!")
5
6 def main():
7
8     hello()
9     print("This is my first main function")
10
11     return
12
13
14 if __name__ == '__main__':
15     main()
```

A search bar is open in the top right corner, containing the text 'hi'. The search results show '1 of 1' matches. The first match is highlighted in the code at line 9: 'This is my first main function'. A tooltip for this match reads: 'You, 2 hours ago • Add hello function'.

- **Ctrl+Shift+F** ~ Find string across files

Folding and unfolding

```
364  ▣    private onCursorSelectionChanged(): void {
365  ▣        if (this.state === State.Hidden) {
366          return;
367        }
368
369        this.editor.layoutContentWidget(this);
370    }
371
372  ⊕    private onEditorBlur(): void { ...
373  }
374
375
376
377
378
379
380  ▣    private onListSelection(e: ISelectionChangeEvent<CompletionItem>): void {
381  ▣        if (!e.elements.length) {
382          return;
383        }
384    }
```

Indentation & file encoding

VS Code analyzes your open file and determines the indentation used in the document. The auto-detected indentation overrides your default indentation settings. The detected setting is displayed on the right side of the Status Bar.

Ln 1, Col 1

Tab Size: 4

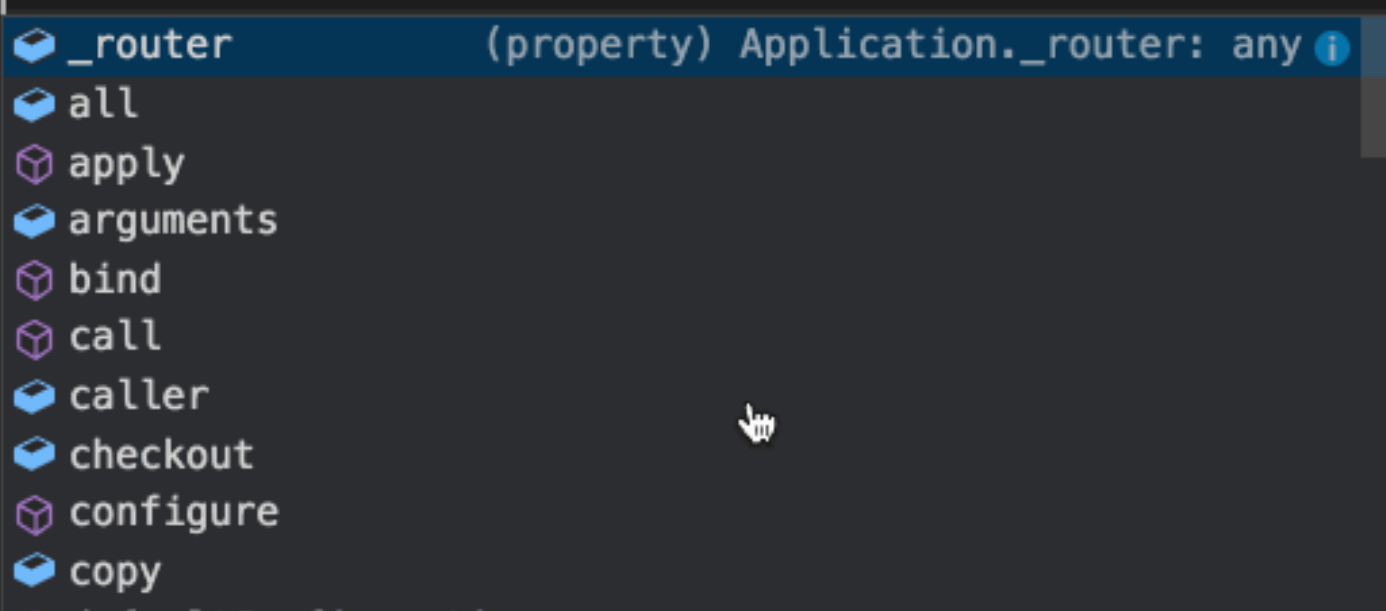
UTF-8

CRLF

IntelliSense

IntelliSense is a general term for various code editing features including: code completion, parameter info, quick info, and member lists. IntelliSense features are sometimes called by other names such as “code completion”, “content assist”, and “code hinting.”

```
1  const express = require('express')
2  const app = express()
3
4  app.
```



- _router** (property) Application._router: any ⓘ
- all
- apply
- arguments
- bind
- call
- caller
- checkout
- configure
- copy

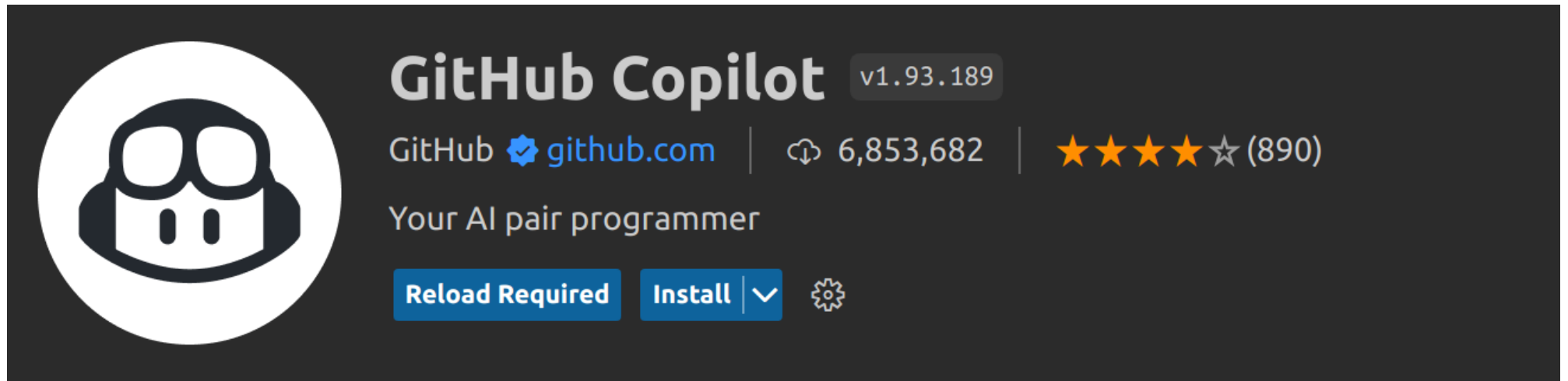
Snippets

Code snippets are templates that make it easier to enter repeating code patterns, such as loops or conditional-statements.

In Visual Studio Code, snippets appear in IntelliSense `Ctrl+Space` mixed with other suggestions, as well as in a dedicated snippet picker (Insert Snippet in the Command Palette).

AI tools in VS Code

The GitHub Copilot extension is an AI pair programmer tool that helps you write code faster and smarter. You can use the Copilot extension in VS Code to generate code, learn from the code it generates, and even configure your editor.



The screenshot shows the GitHub Copilot extension interface in VS Code. On the left is the Copilot logo, a stylized face with glasses. To the right, the text reads "GitHub Copilot" with the version "v1.93.189" in a grey box. Below this, it says "GitHub" with a blue checkmark and "github.com". To the right of that is a cloud icon and the number "6,853,682". Further right are five stars, with the first four filled and the fifth empty, followed by "(890)". Below the stars is the text "Your AI pair programmer". At the bottom, there are three buttons: "Reload Required" in a blue box, "Install" in a blue box with a dropdown arrow, and a gear icon for settings.

Source control

Using Git in VS Code

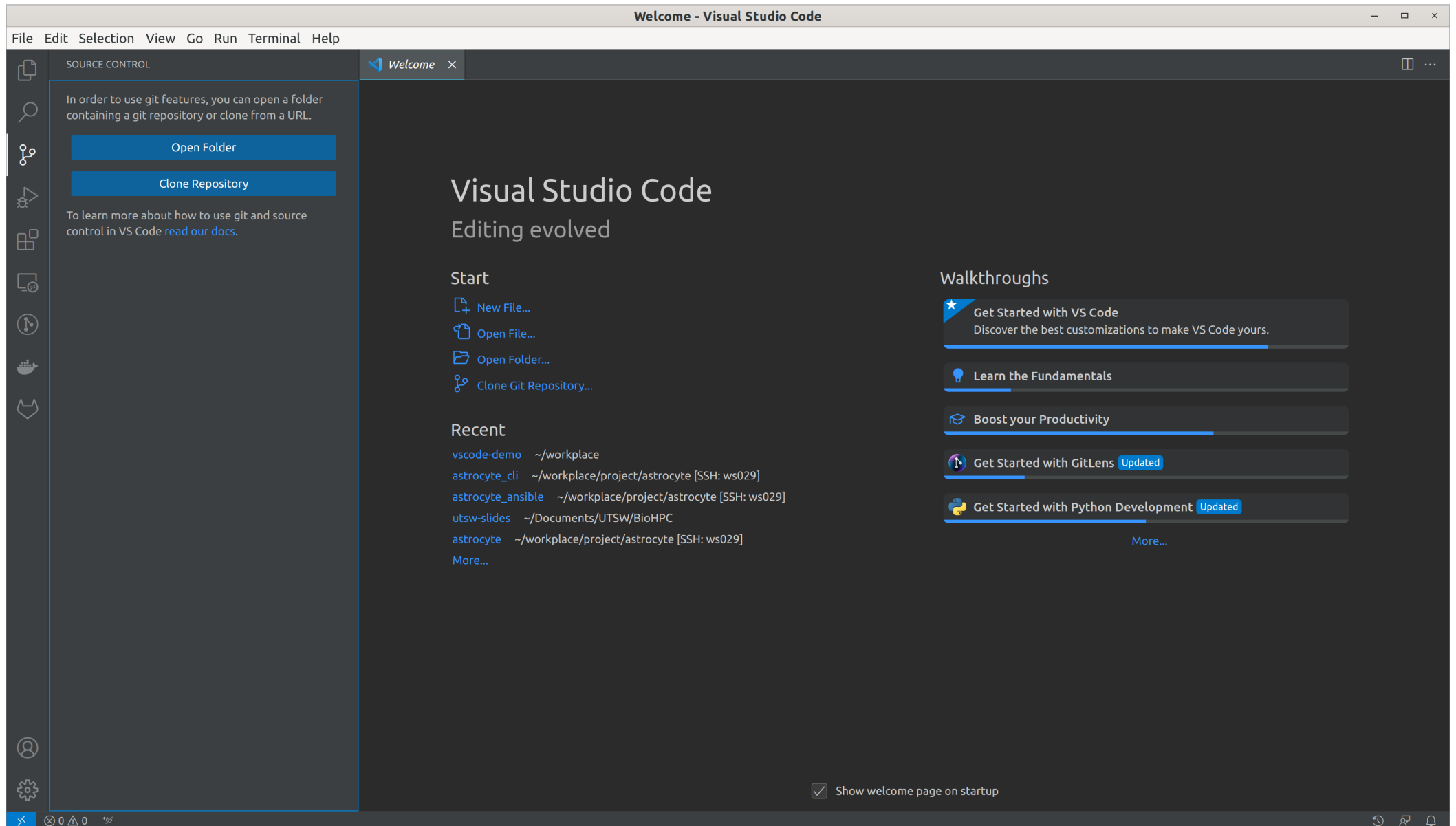
Visual Studio Code has integrated source control management (SCM) and includes Git support out-of-the-box. Many other source control providers are available through extensions.

Git documentations:

- [Cheat Sheet](#)
- [Git-SCM website](#)
- [Manual](#)
- [Book](#)
- [Videos](#)

Note: make sure you have Git installed on your computer.

Open folder or clone repo locally



The screenshot shows the Visual Studio Code interface. The title bar reads "Welcome - Visual Studio Code". The menu bar includes "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". The left sidebar is the "SOURCE CONTROL" view, which contains the following text: "In order to use git features, you can open a folder containing a git repository or clone from a URL." Below this text are two blue buttons: "Open Folder" and "Clone Repository". Further down, it says "To learn more about how to use git and source control in VS Code [read our docs.](#)". The main editor area displays the "Welcome" page with the following content:

Visual Studio Code

Editing evolved

Start

- New File...
- Open File...
- Open Folder...
- Clone Git Repository...

Recent

- vscode-demo ~/workplace
- astrocyte_cli ~/workplace/project/astrocyte [SSH: ws029]
- astrocyte_ansible ~/workplace/project/astrocyte [SSH: ws029]
- utsw-slides ~/Documents/UTSW/BioHPC
- astrocyte ~/workplace/project/astrocyte [SSH: ws029]
- More...

Walkthroughs

- Get Started with VS Code
Discover the best customizations to make VS Code yours.
- Learn the Fundamentals
- Boost your Productivity
- Get Started with GitLens **Updated**
- Get Started with Python Development **Updated**
- More...

At the bottom right of the main editor area, there is a checkbox labeled "Show welcome page on startup" which is checked.

Initialize the local repo

The screenshot shows the Visual Studio Code interface. The title bar reads "Welcome - vscode-demo - Visual Studio Code". The menu bar includes "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help".

SOURCE CONTROL

The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.

[Initialize Repository](#)

To learn more about how to use git and source control in VS Code [read our docs](#).

You can directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

[Publish to GitHub](#)

Visual Studio Code
Editing evolved

Start

- [New File...](#)
- [Open File...](#)
- [Open Folder...](#)
- [Clone Git Repository...](#)

Recent

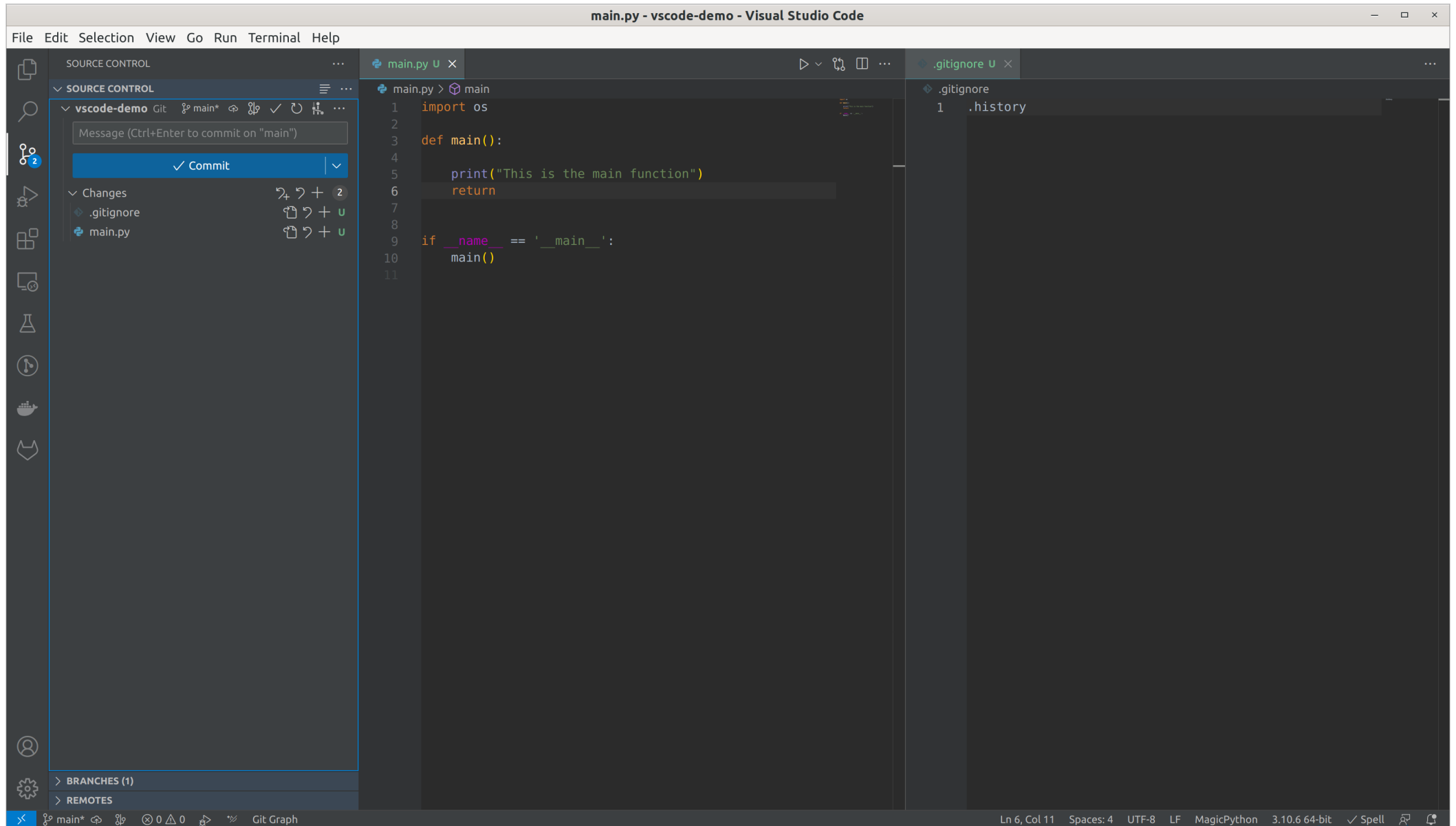
- workplace ~
- astrocyte_cli ~/workplace/project/astrocyte [SSH: ws029]
- astrocyte_ansible ~/workplace/project/astrocyte [SSH: ws029]
- utsw-slides ~/Documents/UTSW/BioHPC
- astrocyte ~/workplace/project/astrocyte [SSH: ws029]
- More...

Walkthroughs

- [Get Started with VS Code](#)
Discover the best customizations to make VS Code yours.
- [Learn the Fundamentals](#)
- [Boost your Productivity](#)
- [Get Started with GitLens](#) Updated
- [Get Started with Python Development](#) Updated
- [More...](#)

Show welcome page on startup

Create new files



The screenshot shows the Visual Studio Code interface with the following components:

- Window Title:** main.py - vscode-demo - Visual Studio Code
- Menu Bar:** File Edit Selection View Go Run Terminal Help
- Left Panel (Source Control):**
 - Expanded view of the repository 'vscode-demo' on the 'main' branch.
 - A commit message input field: "Message (Ctrl+Enter to commit on 'main')".
 - A blue "Commit" button.
 - Changes section showing two files: ".gitignore" and "main.py", both with a plus icon and a 'u' icon, indicating they are new files.
- Editor Area:**
 - main.py:** A Python script with the following code:

```
1 import os
2
3 def main():
4
5     print("This is the main function")
6     return
7
8
9 if __name__ == '__main__':
10     main()
11
```
 - .gitignore:** A file containing the text:

```
1 .history
```
- Bottom Panel:**
 - STATUS BAR: main* (branch), 0 Δ 0 (diff), Git Graph (icon).
 - DEBUG CONSOLE: Ln 6, Col 11 | Spaces: 4 | UTF-8 | LF | MagicPython | 3.10.6 64-bit | Spell (checked) | (notification icons).

Stage changes & commit

The screenshot displays the Visual Studio Code interface for a project named 'vscode-demo'. The 'SOURCE CONTROL' view on the left shows the following structure:

- vscode-demo (Git main*)
 - Staged Changes (1)
 - main.py
 - Changes (1)
 - .gitignore

The 'Commit' button is highlighted in blue. The main editor shows the code for 'main.py':1 import os
2
3 def main():
4
5 print("This is the main function")
6 return
7
8
9 if __name__ == '__main__':
10 main()
11

The right editor shows the content of '.gitignore':1 .history

The status bar at the bottom indicates the current file is 'main.py' on the 'main*' branch, with 0 changes staged and 0 changes. The status bar also shows the Python interpreter path: 'MagicPython 3.10.6 64-bit'.

Stage changes & commit

The screenshot displays the Visual Studio Code interface with the following components:

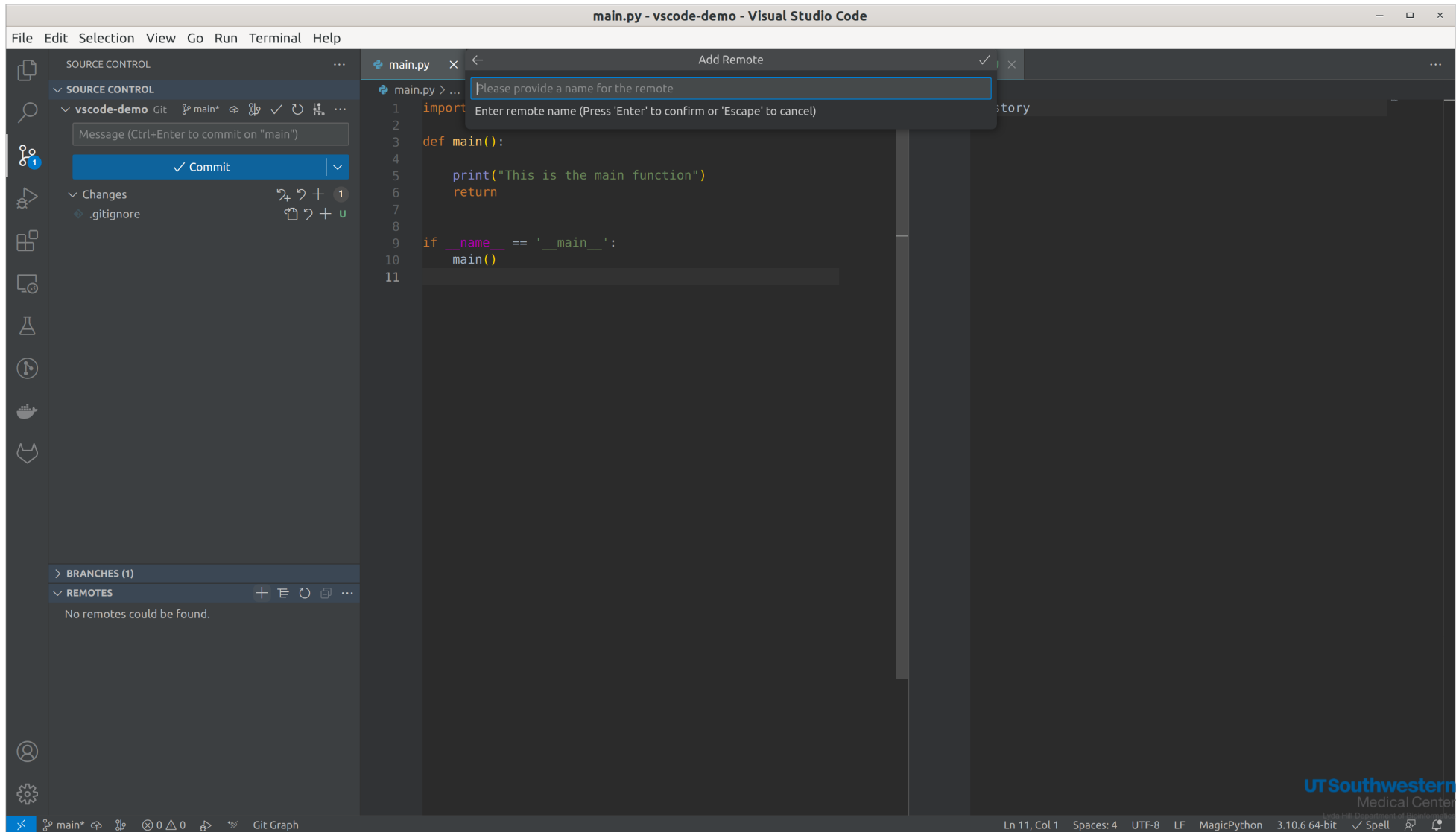
- Window Title:** main.py - vscode-demo - Visual Studio Code
- Menu Bar:** File Edit Selection View Go Run Terminal Help
- Source Control Panel (Left):**
 - Shows the repository path: `vscode-demo` on the `main` branch.
 - Includes a commit message input field: "Message (Ctrl+Enter to commit on 'main')".
 - Features a prominent blue **Commit** button.
 - Lists **Staged Changes**: `main.py` (1 file).
 - Lists **Changes**: `.gitignore` (1 file).
- Editor (Center):** Displays the content of `main.py`:

```
1 import os
2
3 def main():
4
5     print("This is the main function")
6     return
7
8
9 if __name__ == '__main__':
10     main()
11
```
- Editor (Right):** Displays the content of `.gitignore`:

```
1 .history
```
- Bottom Panel:** Shows **BRANCHES (1)** and **REMITES**.
- Status Bar (Bottom):** Displays the current branch (`main*`), file encoding (UTF-8), and other settings.

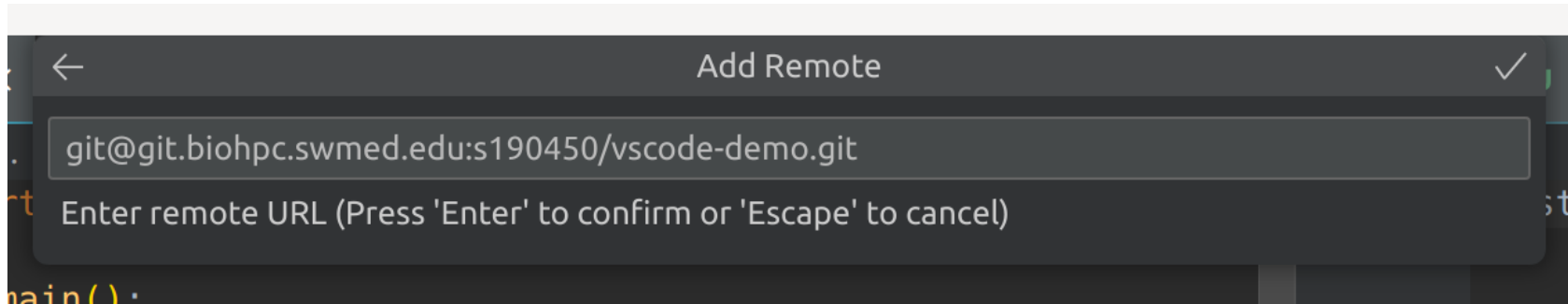
Add remote

- Add remote name



Add remote URL and confirm

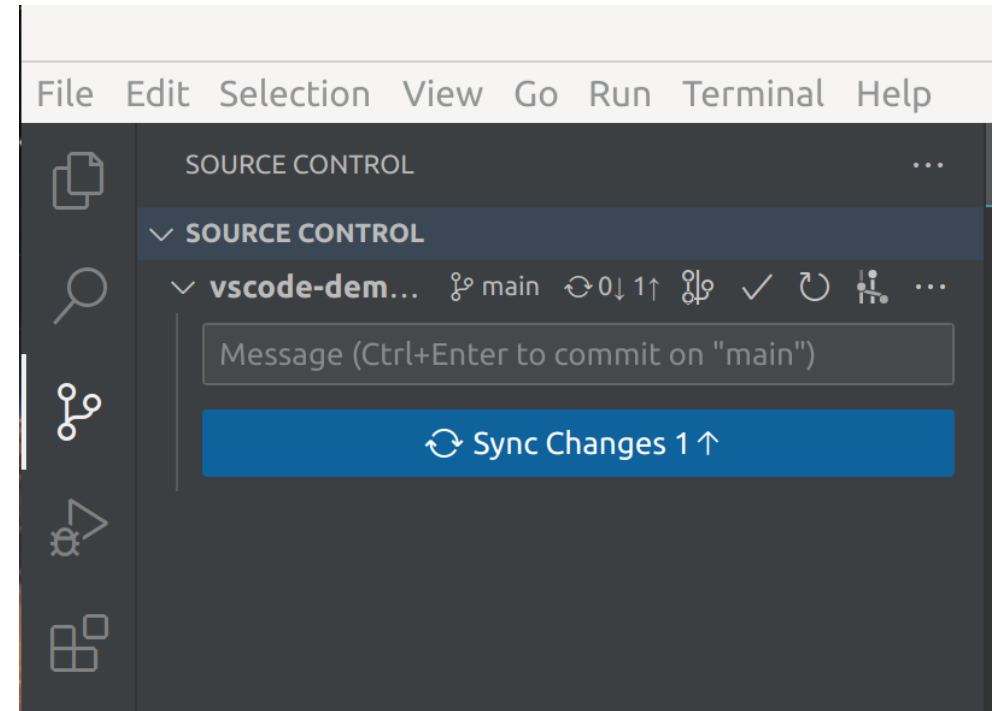
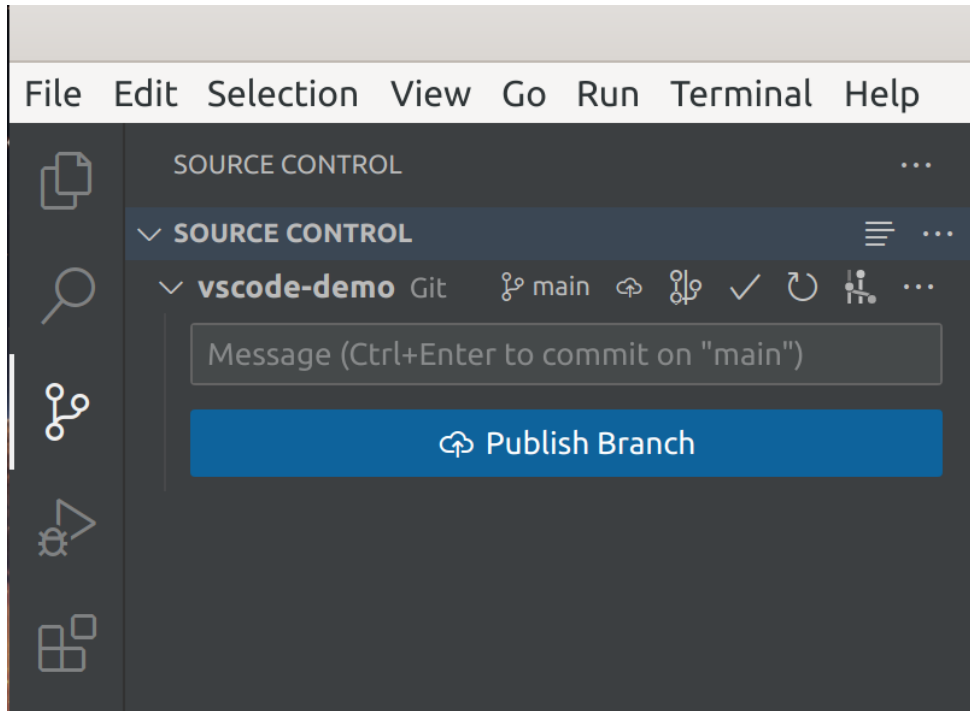
- Add remote URL



- Confirm add remote

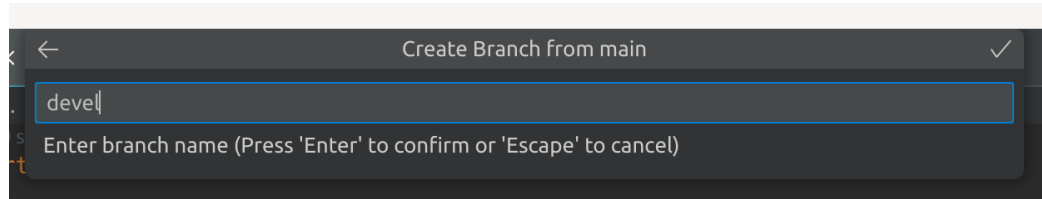
Push/Sync changes

- Push branch to remote
- Sync changes

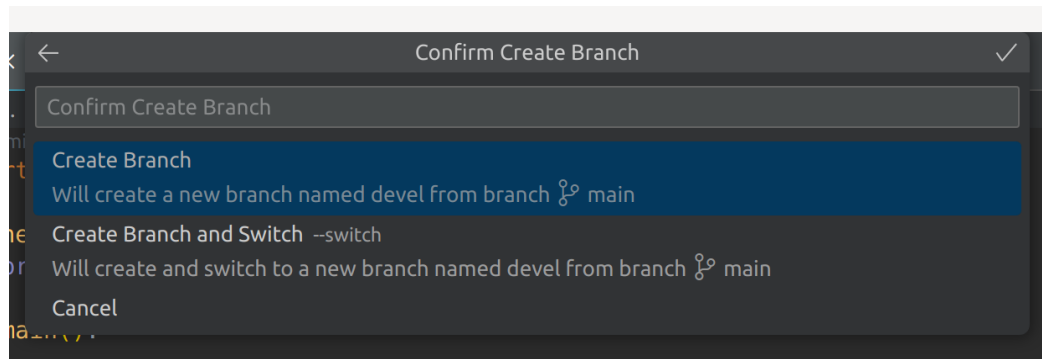


Working with branch

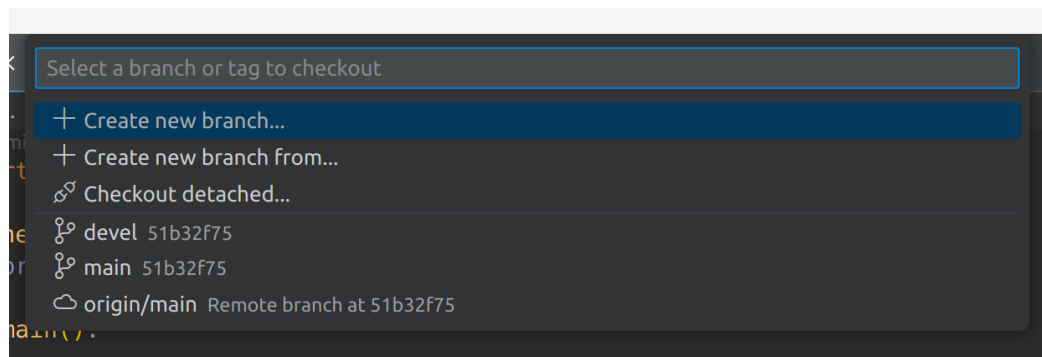
- Add a branch name



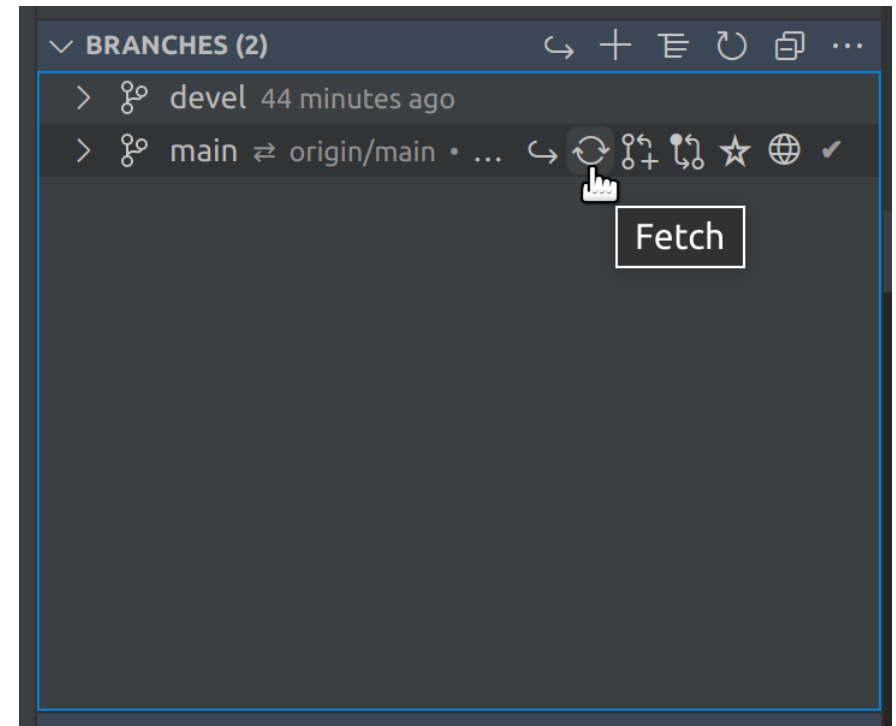
- Create branch



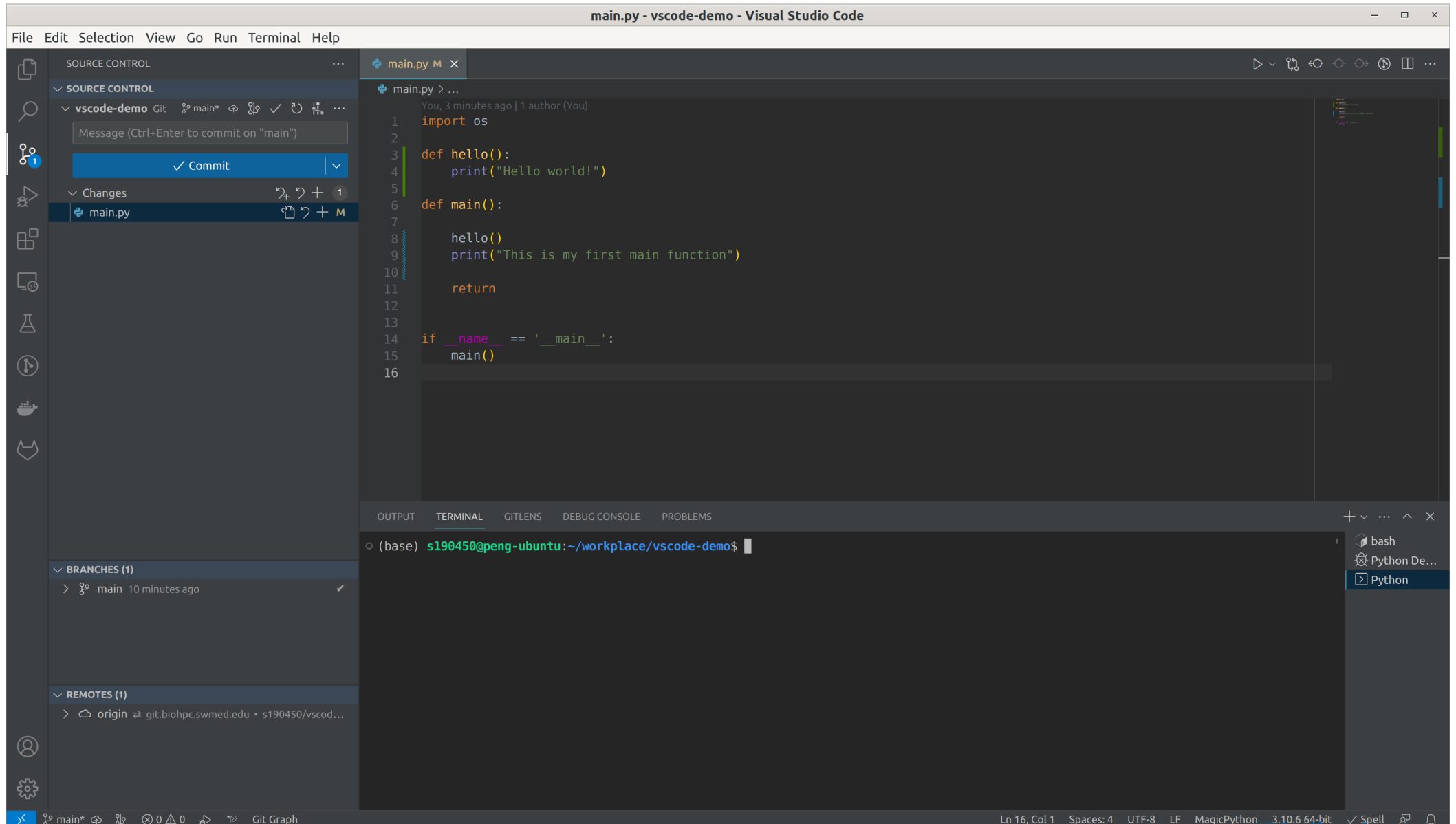
- Switch branch



- Fetch branch



Add changes and commit to devel



The screenshot shows the Visual Studio Code interface with a Python file named `main.py` open. The file contains the following code:

```
1 import os
2
3 def hello():
4     print("Hello world!")
5
6 def main():
7
8     hello()
9     print("This is my first main function")
10
11     return
12
13
14 if __name__ == '__main__':
15     main()
16
```

The SOURCE CONTROL view on the left shows the current state of the repository:

- SOURCE CONTROL**
 - Repository: `vscode-demo` (Git)
 - Current branch: `main*`
 - Message: `Message (Ctrl+Enter to commit on "main")`
 - Commit** button is visible.
 - Changes**: `main.py` (1 change)
- BRANCHES (1)**: `main` (10 minutes ago)
- REMOTES (1)**: `origin` (git.biohpc.swmed.edu + s190450/vscod...)

The TERMINAL view at the bottom shows the current shell session:

```
(base) s190450@peng-ubuntu: ~/workplace/vscode-demo$
```

The status bar at the bottom indicates the current file is `main.py` on the `main` branch, with 0 errors and 0 warnings. The status bar also shows the current encoding (UTF-8) and the Python interpreter (3.10.6 64-bit).

Compare branches

The screenshot shows the Visual Studio Code interface with a file named `main.py` open. A dialog box titled "Compare devel with" is displayed, prompting the user to "Choose a reference to compare with (or enter a reference using #)". The dialog lists several references:

- `devel` (checked) with commit hash `5c850ee` and a timestamp of "14 minutes ago".
- `main` with commit hash `51b32f7` and a timestamp of "30 minutes ago".
- `origin/main` (remote branch) with commit hash `51b32f7` and a timestamp of "30 minutes ago".

The background editor shows the following Python code in `main.py`:

```

1 import sys
2
3 def hello():
4     print("Hello world!")
5
6 def main():
7
8     hello()
9     print("This is my first main function")
10
11     return
12
13
14 if __name__ == '__main__':
15     main()
16
  
```

The left sidebar shows the "SOURCE CONTROL" view with the following structure:

- SOURCE CONTROL**
 - `vscode-demo` (Git)
 - devel (4 minutes ago)
 - Compare devel with <branch, tag, or ref>
 - Publish devel to a remote
 - ↑ Add hello function You, 4 minutes ago
 - ← main, ... > Add .gitignore You, 20 minutes ...
 - Add main.py You, 37 minutes ago
 - main (20 minutes ago)
- BRANCHES (2)**
 - devel (4 minutes ago)
 - Compare devel with <branch, tag, or ref>
 - Publish devel to a remote
- REMOTES (1)**
 - origin (git.biohpc.swmed.edu • s190450/vscod...)

The bottom status bar shows the current branch is `devel` and the file is `main.py`. The terminal at the bottom shows a shell prompt: `(base) s190450@peng-ubuntu: ~/workspace/vscode-demo$`.

File differences

The screenshot displays the Visual Studio Code interface with a file diff view for `main.py`. The window title is `main.py (c4bc725) ↔ main.py (5c850ee) - vscode-demo - Visual Studio Code`.

Source Control Sidebar:

- Branches: `devel` (4 minutes ago), `main` (20 minutes ago)
- Commits: `main, ...` (Add .gitignore, 20 minutes ago), `main` (Add main.py, 37 minutes ago)
- Remotes: `origin` (git.biohpc.swmed.edu • s190450/vscod...)

Code Diff:

```

1 import os
2
3 def main():
4     print("This is the main function")
5
6     return
7
8
9 if __name__ == '__main__':
10     main()
11
12
13
14 def hello():
15     print("Hello world!")
16
17 def main():
18     hello()
19     print("This is my first main function")
20
21     return
22
23 if __name__ == '__main__':
24     main()
25
26

```

Terminal:

```

(base) s190450@peng-ubuntu: ~/workspace/vscode-demo$

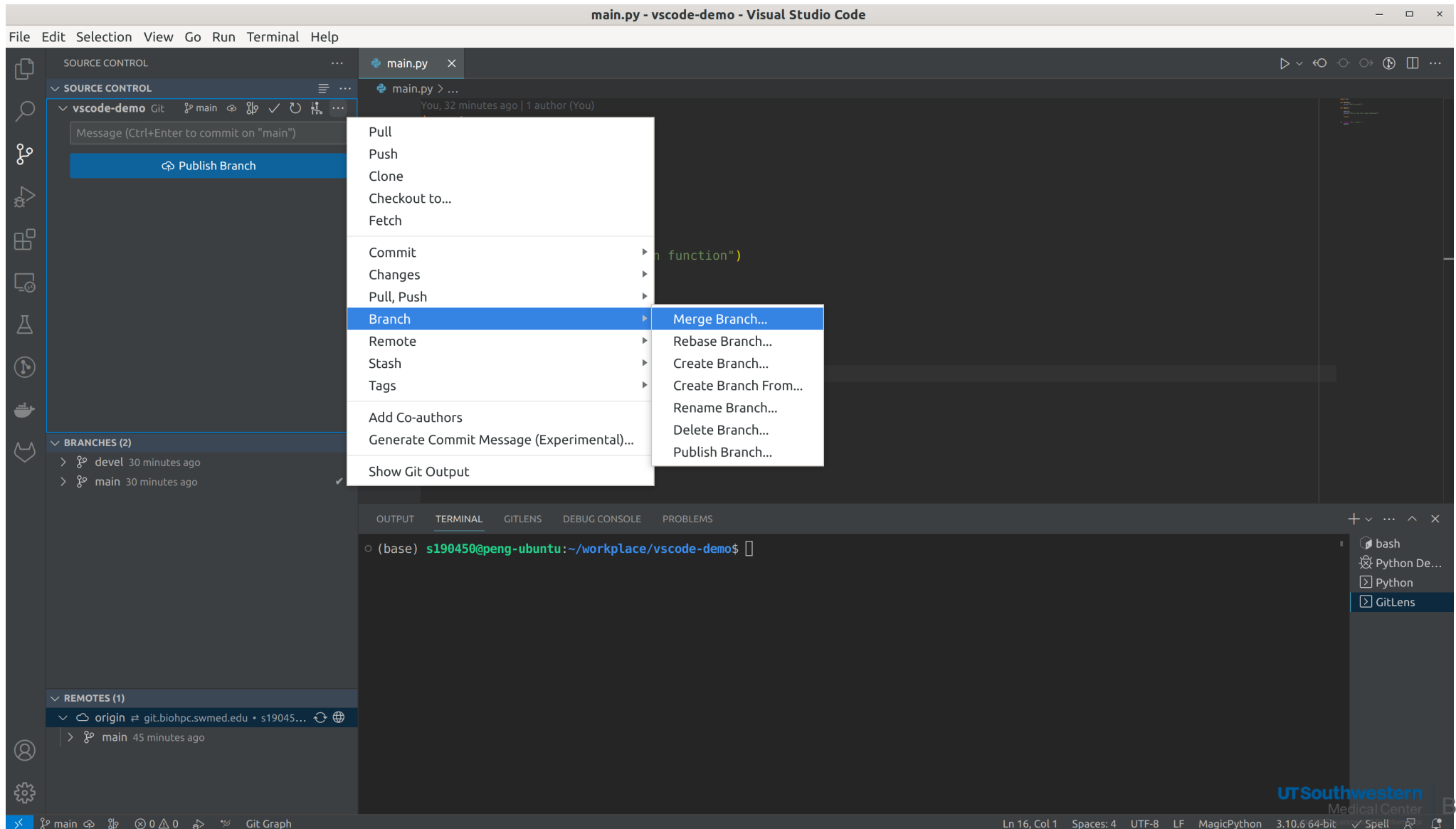
```

Status Bar:

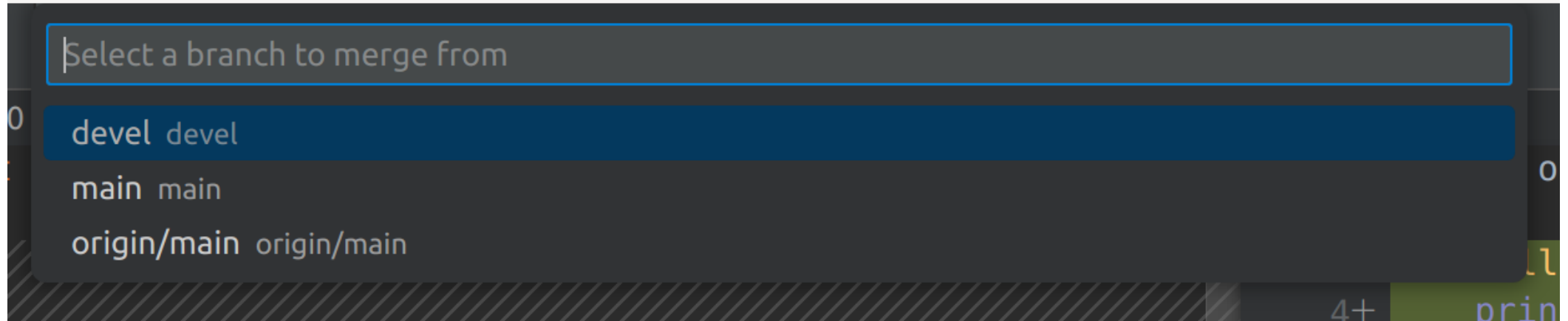
- Branch: `devel`
- Git Graph: 0 additions, 0 deletions
- File: `main.py` (Ln 3, Col 1)
- Encoding: UTF-8
- Python: 3.10.6 64-bit
- Spell: ✓

Merge branches

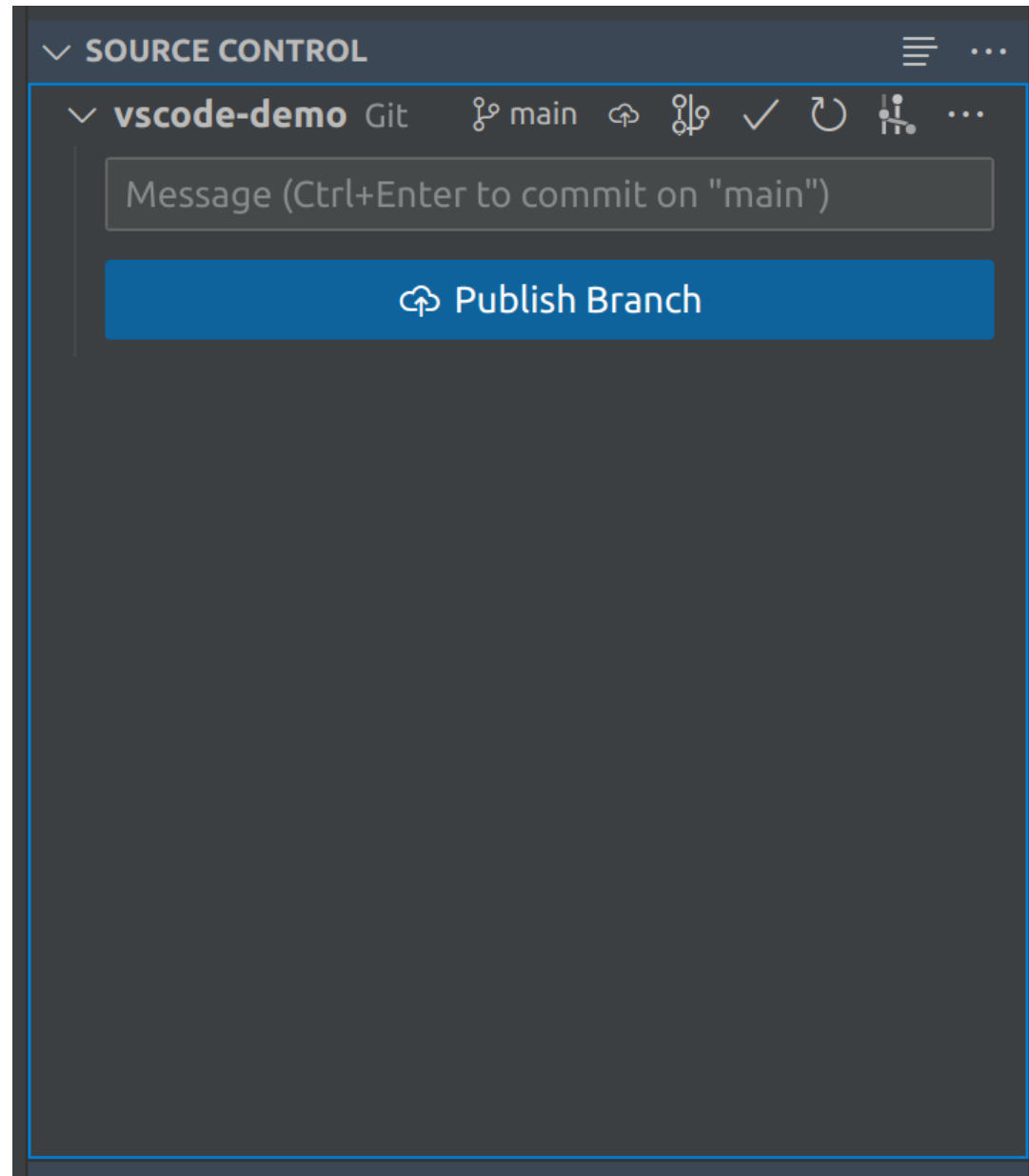
Note: Switch to the **target branch** before starting to merge



Select a branch to merge from



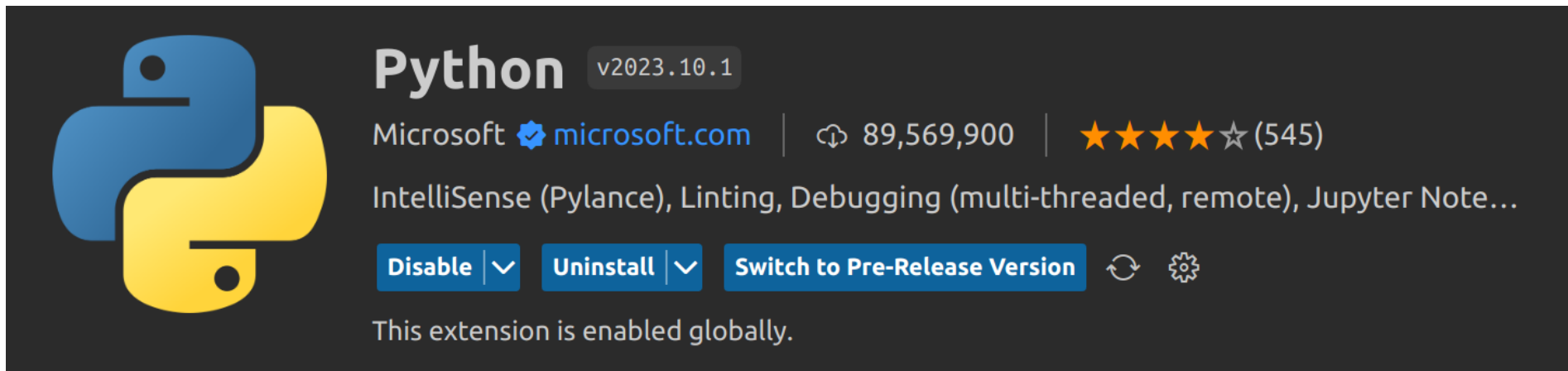
Publish merged branch



Work with Python

Prerequisites

- Python 3
- VS Code Python extension






The screenshot shows the Python extension page in the Visual Studio Code marketplace. On the left is the Python logo. To its right, the word "Python" is displayed in a large font, with the version "v2023.10.1" in a smaller font to its right. Below this, the publisher "Microsoft" is listed with a verified badge and the website "microsoft.com". To the right of this, the number of downloads "89,569,900" is shown, followed by a star rating of four stars and a half star, with "(545)" reviews. Below the star rating, a snippet of the extension's features is visible: "IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Note...". At the bottom of the extension card, there are three buttons: "Disable" with a dropdown arrow, "Uninstall" with a dropdown arrow, and "Switch to Pre-Release Version". To the right of these buttons are icons for refresh and settings. Below the buttons, a status message reads "This extension is enabled globally."

- Jupyter extension (optional)



Jupyter v2023.5.1101742258

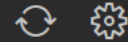
Microsoft  microsoft.com |  67,741,555 |  (288)

Jupyter notebook support, interactive programming and computing that supports...

Disable | 

Uninstall | 

Switch to Pre-Release Version



This extension is enabled globally.

Create a virtual environment

- Command Palette (`Ctrl+Shift+P`) → Python: Create Environment

Select an environment type

Venv Creates a `.venv` virtual environment in the current workspace

Conda Creates a `.conda` Conda environment in the current workspace

Select a interpreter

- Command Palette (**Ctrl+Shift+P**) -> **Python: Select Interpreter**

Select Interpreter

Selected Interpreter: `.\.venv\Scripts\python.exe`

+ Enter interpreter path...

⚙ Use Python from `python.defaultInterpreterPath` setting `~\AppData\Local\Microsoft\WindowsAp...`

★ Python 3.10.5 (.venv: venv) <code>.\.venv\Scripts\python.exe</code>	Recommended
Python 3.9.12 ('base') <code>~\Anaconda3\python.exe</code>	Conda
Python 3.9.12 ('base') <code>~\Downloads\Anaconda\python.exe</code>	
Python 3.9.13 64-bit (microsoft store) <code>~\AppData\Local\Microsoft\WindowsApps\python3.9.exe</code>	Global
Python 3.10.5 64-bit <code>~\AppData\Local\Programs\Python\Python310\python.exe</code>	
Python 3.9.13 64-bit <code>C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.9_3.9.3568.0_x6...</code>	
Python 3.10.5 64-bit (system) <code>~\OneDrive - Microsoft\Documents\Projects\Unstructured Data Analysis\.</code>	

Create debugging configuration

The screenshot shows the Visual Studio Code interface with a Python file named 'main.py' open. A 'Select a debug configuration' dialog box is displayed over the code editor. The dialog lists several debug configurations:

- Python File: Debug the currently active Python file
- Module: Debug a Python module by invoking it with '-m'
- Remote Attach: Attach to a remote debug server
- Attach using Process ID: Attach to a local process
- Django: Launch and debug a Django web application
- FastAPI: Launch and debug a FastAPI web application
- Flask: Launch and debug a Flask web application
- Pyramid: Launch and debug a Pyramid web application

The code in the editor is as follows:

```

1
2
3 def hello():
4     print('Hello World!')
5
6 def main():
7     hello()
8
9
10
11
12
13
14 if __name__ == '__main__':
15     main()
16

```

The terminal at the bottom shows the command prompt:

```

(base) s190450@peng-ubuntu: ~/workspace/vscode-demo$

```

The interface also shows the 'RUN AND DEBUG' sidebar on the left with a 'Run and Debug' button and instructions to create a launch.json file. The 'BREAKPOINTS' sidebar at the bottom left shows options for 'Raised Exceptions', 'Uncaught Exceptions' (checked), and 'User Uncaught Exceptions'.

Debugging

The screenshot shows the Visual Studio Code interface with a Python file named `main.py` open. The code is as follows:

```

1
2
3 def hello():
4     print("Hello world!")
5
6 def main():
7
8     hello()
9     print("This is my first main function")
10
11     return
12
13
14 if __name__ == '__main__':
15     main()
16

```

An exception has occurred: `NameError`. The error message is:

```

Exception has occurred: NameError
name '__name__' is not defined

File "/home/s190450/workplace/vscode-demo/main.py", line 14, in <module>
    if __name__ == '__main__':
NameError: name '__name__' is not defined

```

The call stack shows the error occurred in the `<module>` at line 14:

```

CALL STACK
<module> main.py 14:1

```

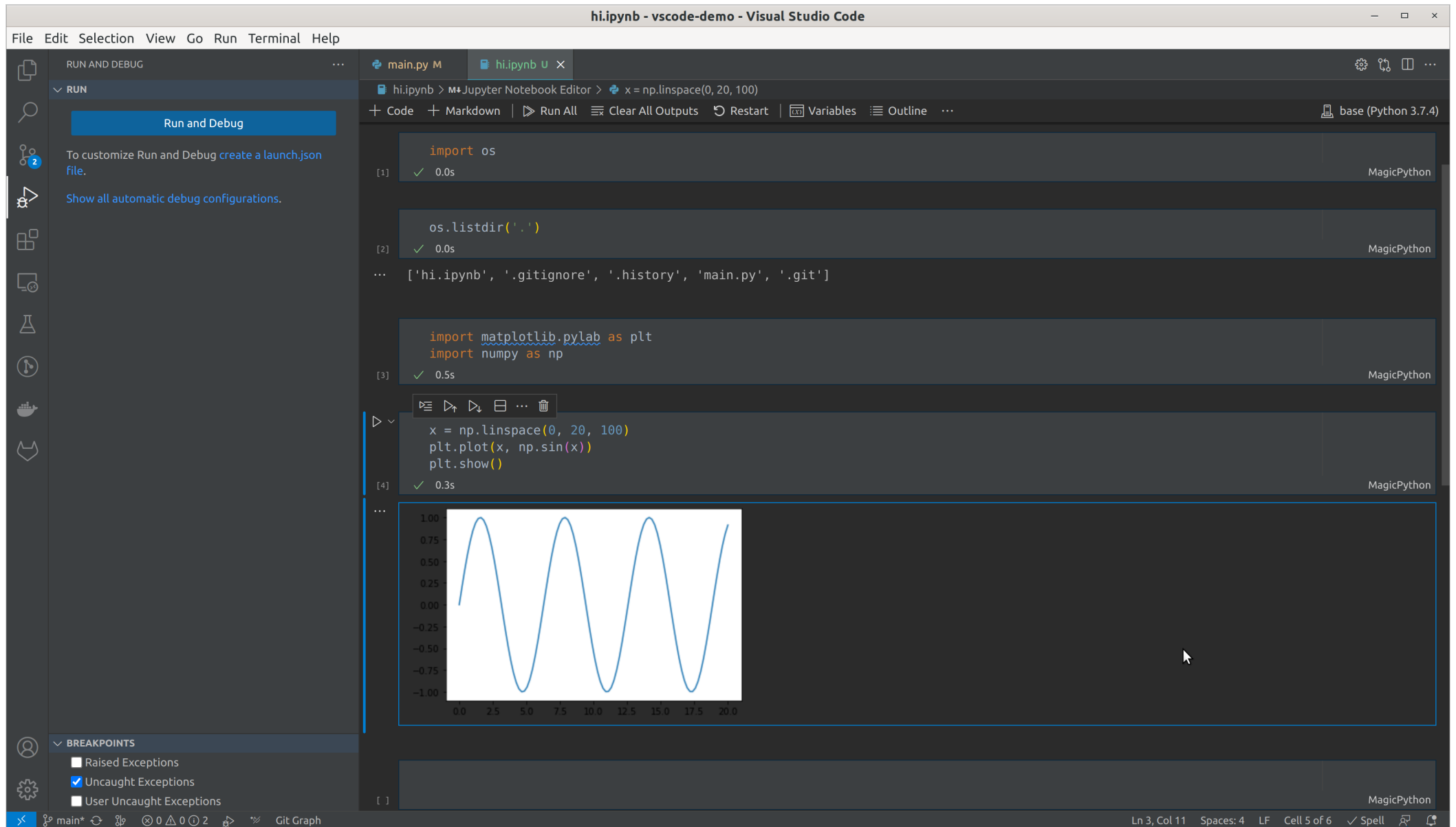
The terminal shows the command `python3 main.py` and the output of the script:

```

./debugpy/./debugpy/server/cli.py", line 284, in run_file
    runpy.run_path(target, run_name="__main__")
File "/home/s190450/.vscode/extensions/ms-python.python-2023.10.1/pythonFiles/lib/python/debugpy/_vendored/pydevd/_pydevd_bundle/pydevd_runpy.py", line 321, in run_path
    return run_module(code, init_globals, run_name,
File "/home/s190450/.vscode/extensions/ms-python.python-2023.10.1/pythonFiles/lib/python/debugpy/_vendored/pydevd/_pydevd_bundle/pydevd_runpy.py", line 135, in _run_module_code
    _run_code(code, mod_globals, init_globals,
File "/home/s190450/.vscode/extensions/ms-python.python-2023.10.1/pythonFiles/lib/python/debugpy/_vendored/pydevd/_pydevd_bundle/pydevd_runpy.py", line 124, in _run_code
    exec(code, run_globals)
File "/home/s190450/workplace/vscode-demo/main.py", line 14, in <module>
    if __name__ == '__main__':
NameError: name '__name__' is not defined. Did you mean: '_name_'?
(base) s190450@peng-ubuntu:~/workplace/vscode-demo$ cd /home/s190450/workplace/vscode-demo ; /usr/bin/env /bin/python3 /home/s190450/.vscode/extensions/ms-python.python-2023.10.1/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 52789 -- /home/s190450/workplace/vscode-demo/main.py

```

Jupyter in VS Code



The screenshot displays the Visual Studio Code interface with a Jupyter Notebook open. The notebook contains the following code cells:

```
[1] import os
    ✓ 0.0s
    MagicPython
```

```
[2] os.listdir('.')
    ✓ 0.0s
    MagicPython
```

```
['hi.ipynb', '.gitignore', '.history', 'main.py', '.git']
```

```
[3] import matplotlib.pyplot as plt
    import numpy as np
    ✓ 0.5s
    MagicPython
```

```
[4] x = np.linspace(0, 20, 100)
    plt.plot(x, np.sin(x))
    plt.show()
    ✓ 0.3s
    MagicPython
```

The output of the fourth cell is a plot of a sine wave. The x-axis ranges from 0.0 to 20.0, and the y-axis ranges from -1.00 to 1.00. The plot shows a smooth, periodic wave oscillating between approximately -0.9 and 0.9.

The interface also shows the 'RUN AND DEBUG' sidebar on the left, the 'BREAKPOINTS' section at the bottom left, and the status bar at the bottom right indicating the current line and column (Ln 3, Col 11).

Acknowledgement

- Thank all BioHPC team members for their support.
- Please acknowledge our contribution by adding the following sentence to your paper:

This research was supported in part by the computational resources provided by the BioHPC supercomputing facility located in the Lyda Hill Department of Bioinformatics, UT Southwestern Medical Center.

Questions?

**Thanks for your
attention!**