

Using Blockly in Jupyter Notebooks

How to Write Code Without Writing Code

What is Blockly?

- Block-based programming editor
- Users place interlocking, graphical blocks representing code concepts
- Blockly generates equivalent Python code

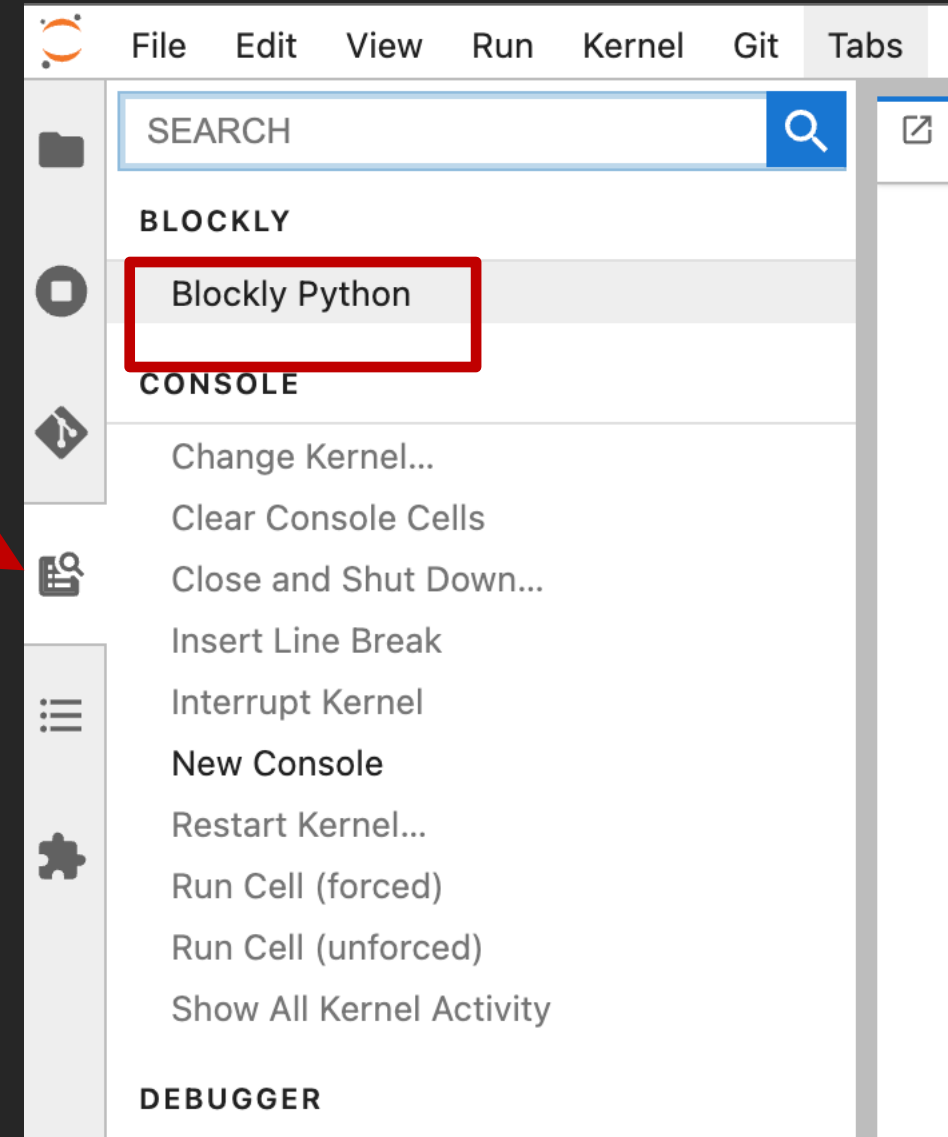


```
print('Hi')
```

How-to: Open Blockly Interface (1)

1. Open Commands
2. Click Blockly Python

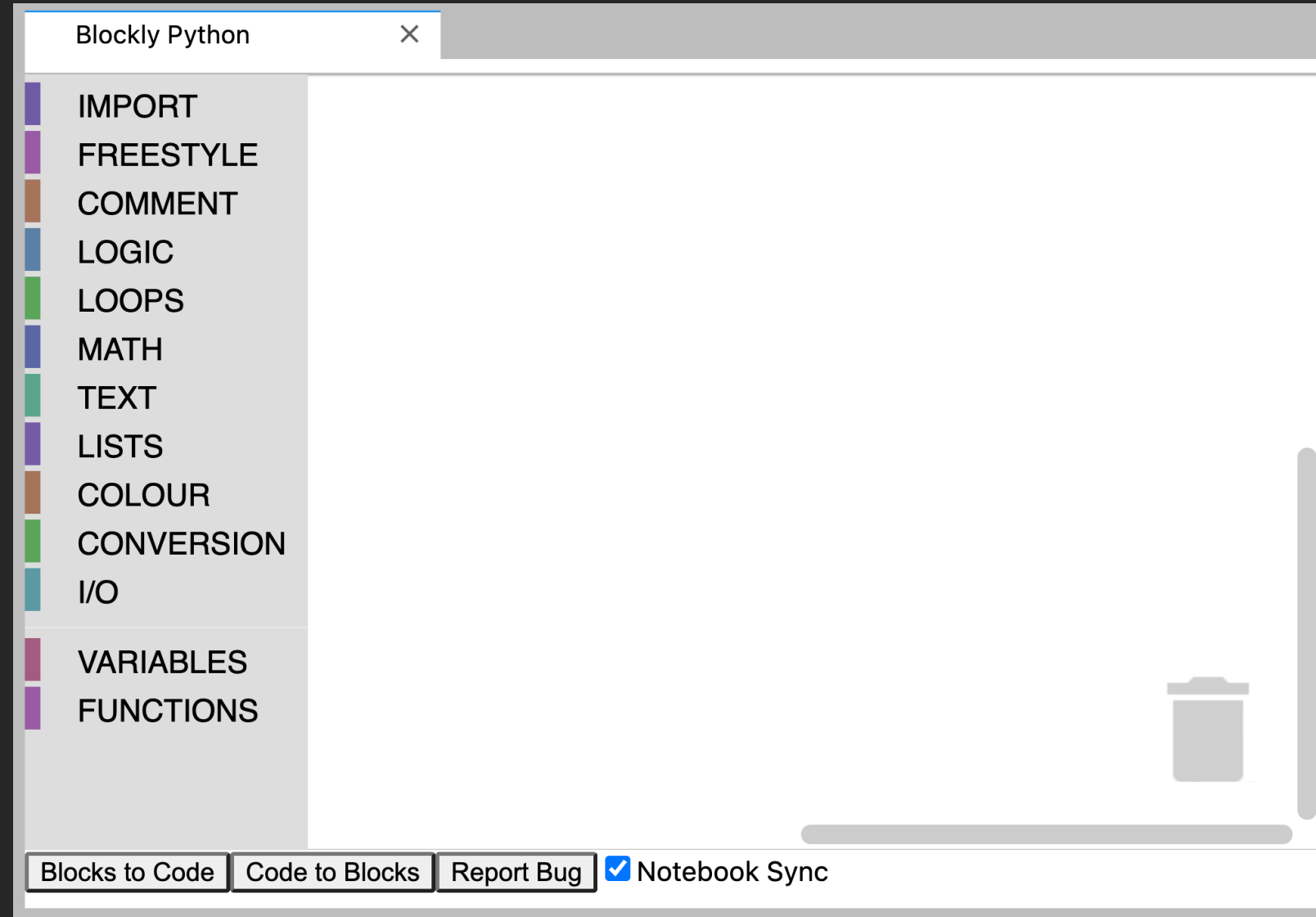
This will open a Blockly tab.



How-to: Open Blockly Interface (2)

This is how the Blockly Editor **should look**...

But sometimes it may not render correctly...

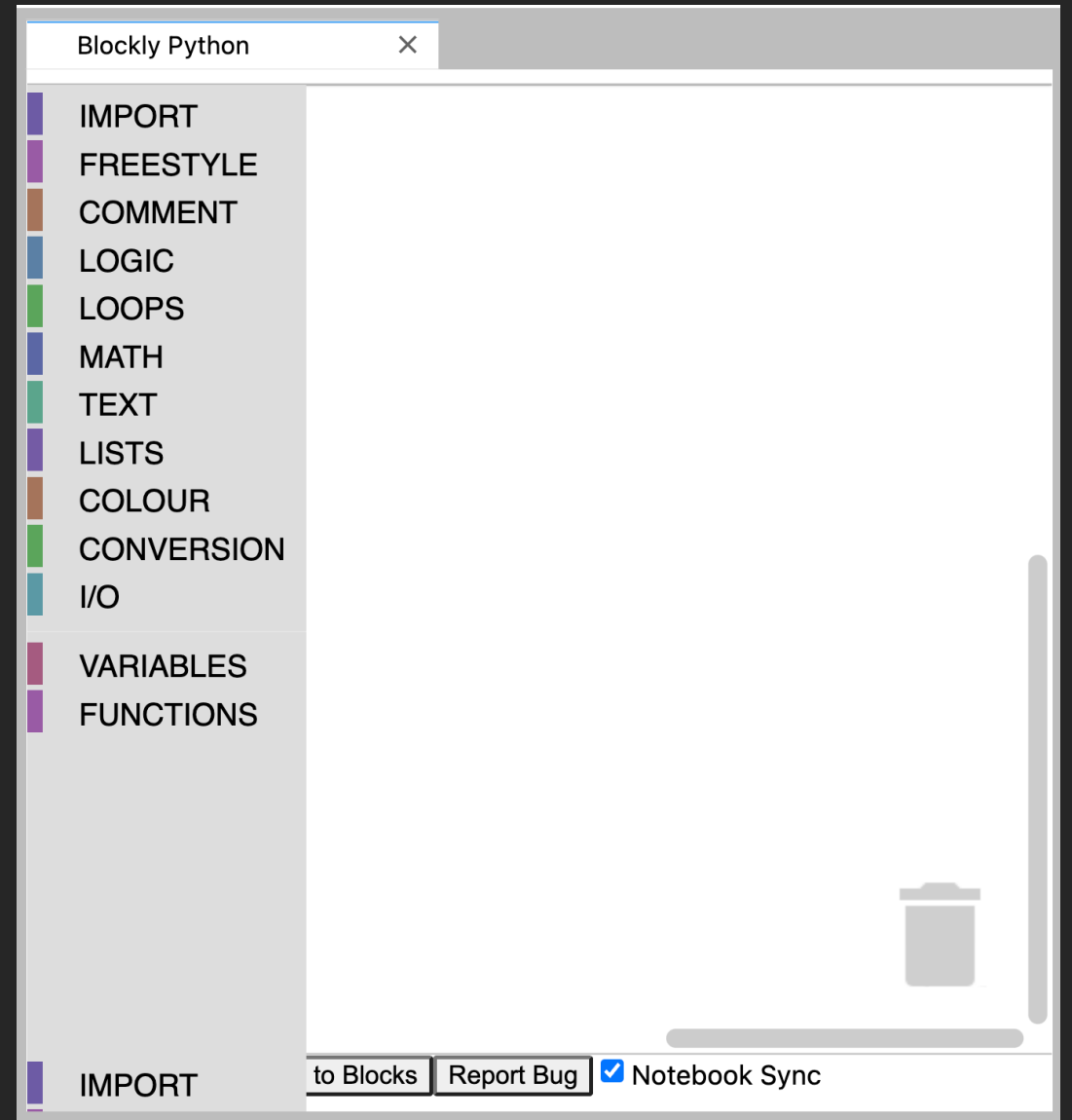


How-to: Fix Blockly Interface Render Issues

Solution: Reload the page.

To avoid render errors:

1. Open the Blockly Editor
2. Close the Blockly Editor
- 3. Reload this page**
4. Open the Blockly Editor

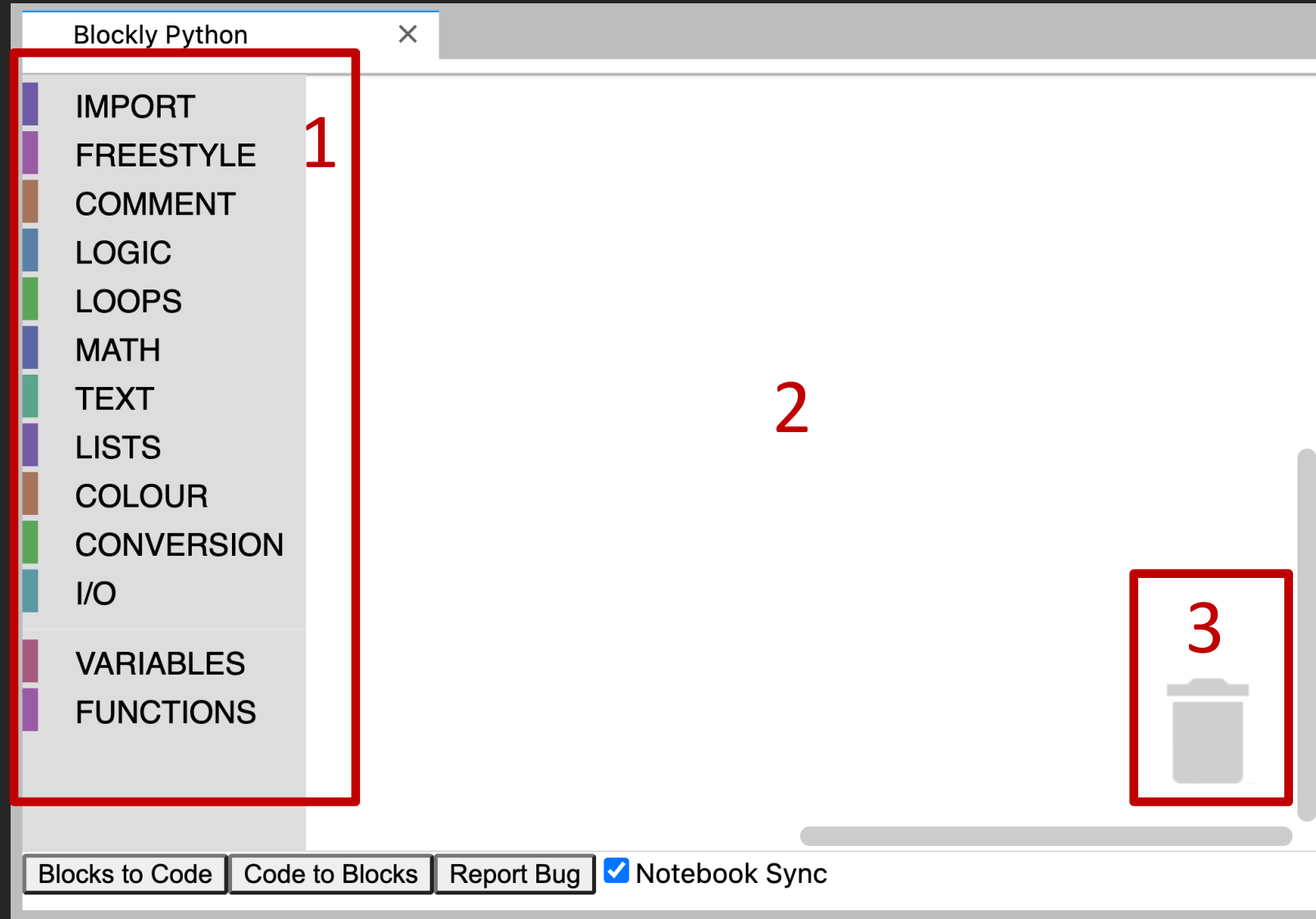


Activity 1: Open Blockly Editor

1. Close all open tabs (except pdf)
2. Reload the page
3. Open Blockly Editor & confirm rendered **correctly**
4. Close Blockly Editor
5. Open Blockly Editor & confirm rendered **incorrectly**
6. Reload the page
7. Confirm Blockly Editor rendered **correctly**

Blockly Interface

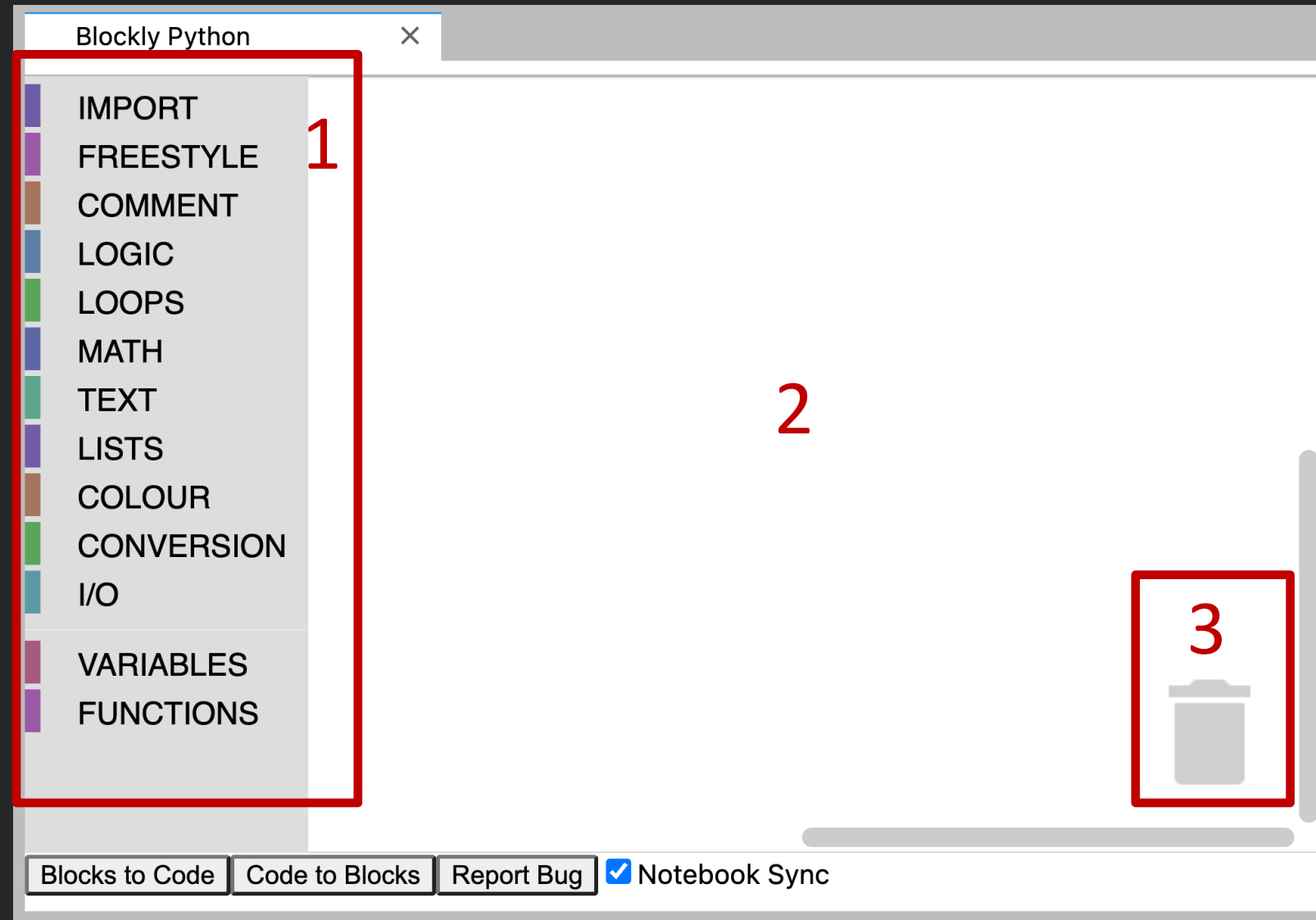
1. Blockly Toolbox
2. Blockly Workspace
3. Blockly Trash Can




Blockly Interface

1. Blockly Toolbox
2. Blockly Workspace
3. Blockly Trash Can

Demo!



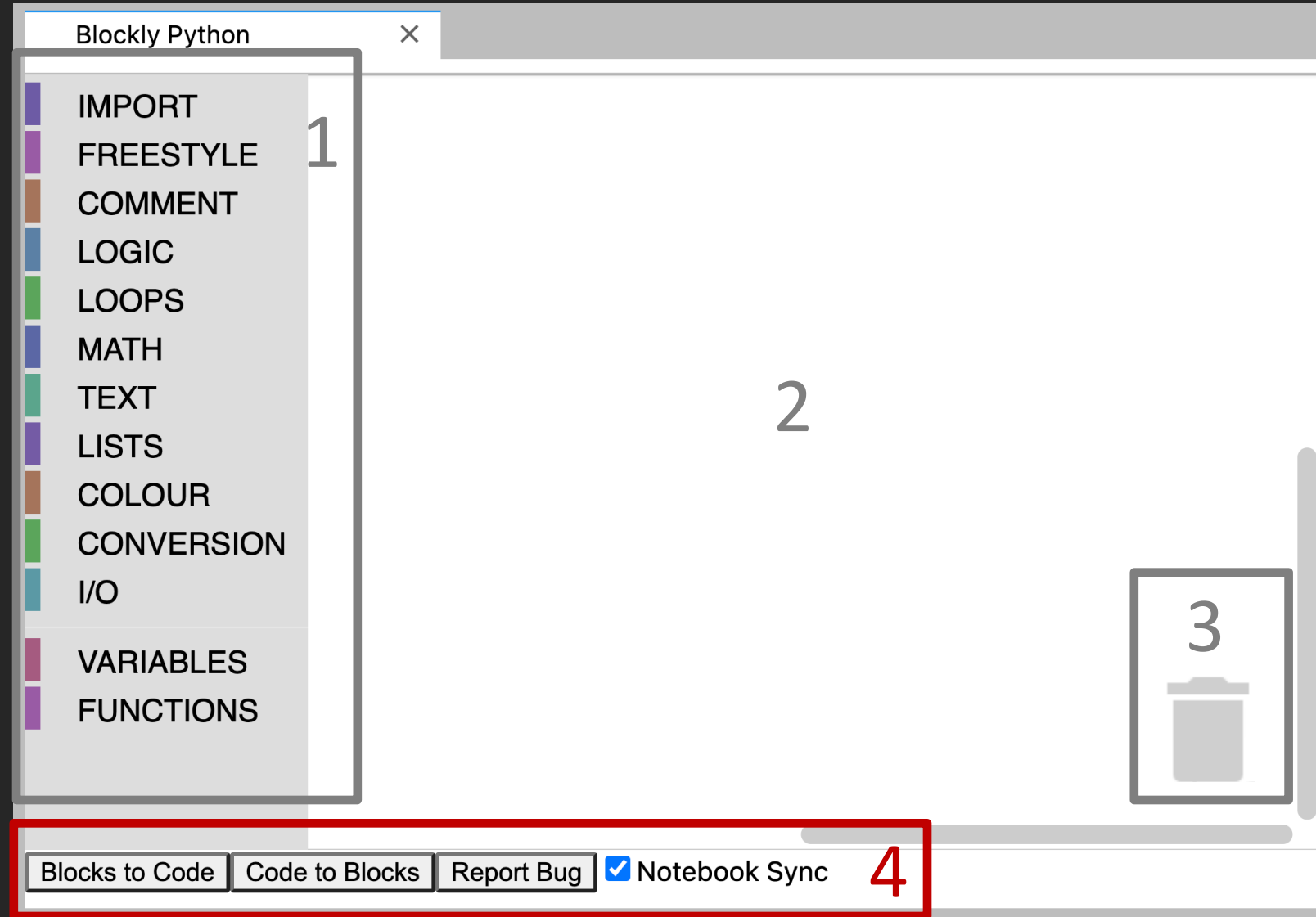
Activity 2: Playtime with Blocks

1. Open Blockly Editor (if not already open)
2. Click the Text section of Blockly Toolbox
3. Find the “print text” block The image shows a Blockly 'print text' block. It is a light blue block with a darker blue tab on the left that says 'print'. To the right of the tab is a text input field containing the text 'abc'.
4. Hover over the block to see a description
5. Click or drag the block into the Blockly Workspace
6. Replace “abc” with your name
7. Take a screenshot of your block and upload to Discord
8. Drag the block to the Blockly Trash Can

Blockly Interface

1. Blockly Toolbox
2. Blockly Workspace
3. Blockly Trash Can

4. JupyterLab Blockly Integration Toolbar



Using JupyterLab with Blockly

The screenshot displays the JupyterLab interface. On the left, the 'Blockly Python' workspace is active, showing a sidebar with categories like IMPORT, FREESTYLE, COMMENT, LOGIC, LOOPS, MATH, TEXT, LISTS, COLOUR, CONVERSION, I/O, VARIABLES, and FUNCTIONS. A block for 'import pandas as pd' is visible in the workspace. On the right, a Jupyter notebook titled 'Data-science-and-the-nat' is open. The notebook content includes a title 'Load the data into a dataframe', a paragraph explaining the goal, and a code cell with the following code:

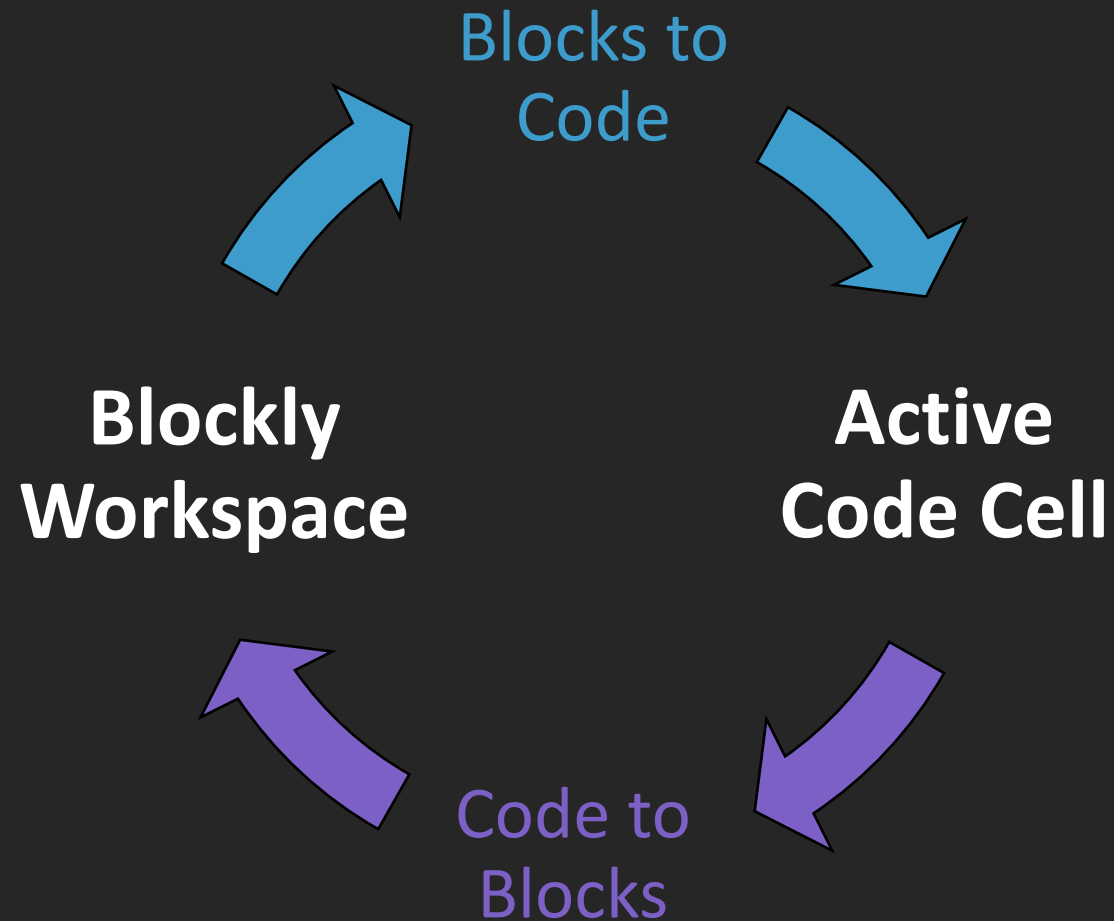
```
[3]: import pandas as pd
```

Below the code cell, the notebook displays the output of the code, which is a dataframe with 5 rows and 4 columns: File, PetalColor, PetalShape, and Size.

	File	PetalColor	PetalShape	Size
0	0001.png	multicolor	rounded	medium
1	0002.png	unicolor	rounded	medium
2	0003.png	unicolor	unrounded	large
3	0004.png	multicolor	rounded	medium
4	0005.png	multicolor	rounded	small

The bottom status bar shows 'Mode: Edit', 'Ln 2, Col 1', and the file name 'Data-science-and-the-nature-of-data-PS.ipynb'.

JupyterLab Blockly Integration



JupyterLab Blockly Integration

Blocks

Python code
Blockly XML Tag

**Blockly
Workspace**

**Active
Code Cell**

Blocks

Blockly XML Tag

Activity 3: Code to Blocks, Blocks to Code

1. Open `blockly_integration.ipynb`
2. Copy/paste the provided **Blockly XML tag** into the indicated empty code cell
3. Click **Code to Blocks** & observe the **Blockly Workspace**
4. Click **Blocks to Code** & observe the **code input** of the **active cell**
5. Run the **active cell**
6. Take a screenshot that includes both the **code input** & **code output** and save and show to the mentor

Summary

- Blockly Interface
- Blocks to Code and Code to Blocks
- How to add code to notebook cell with Blockly