

# 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

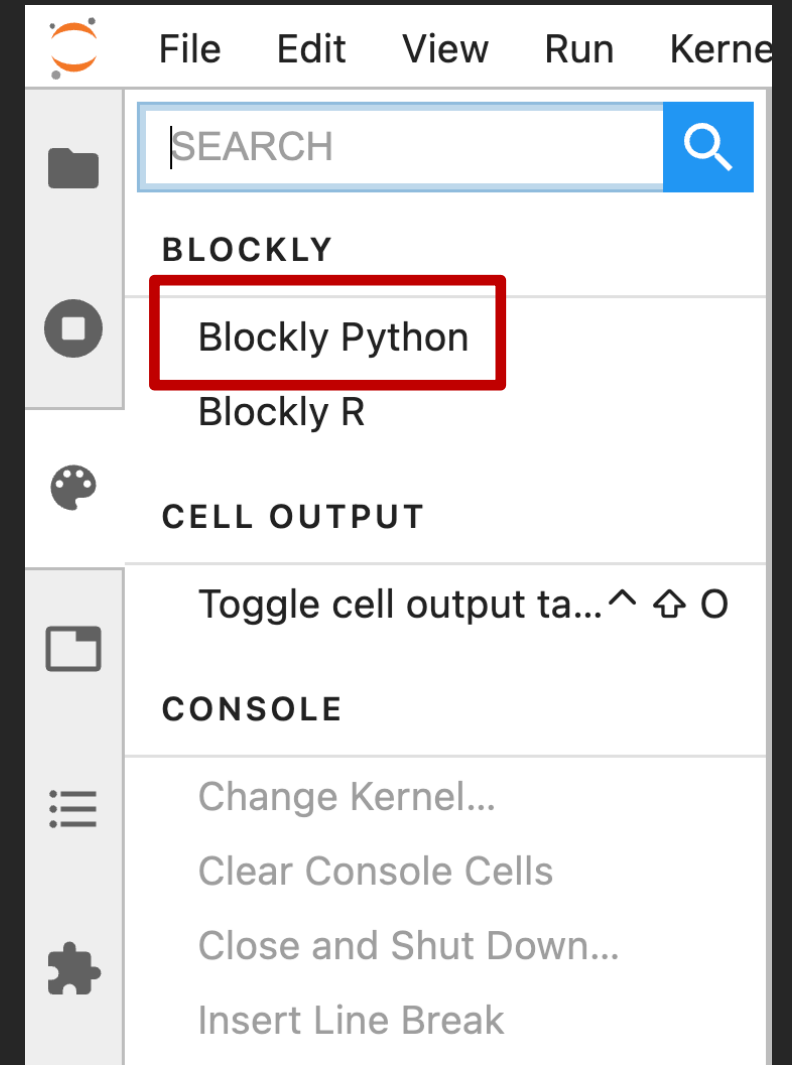


```
print('Hi')
```

# How-to: Open Blockly Interface (1)

1. Open Commands
2. Click Blockly Python

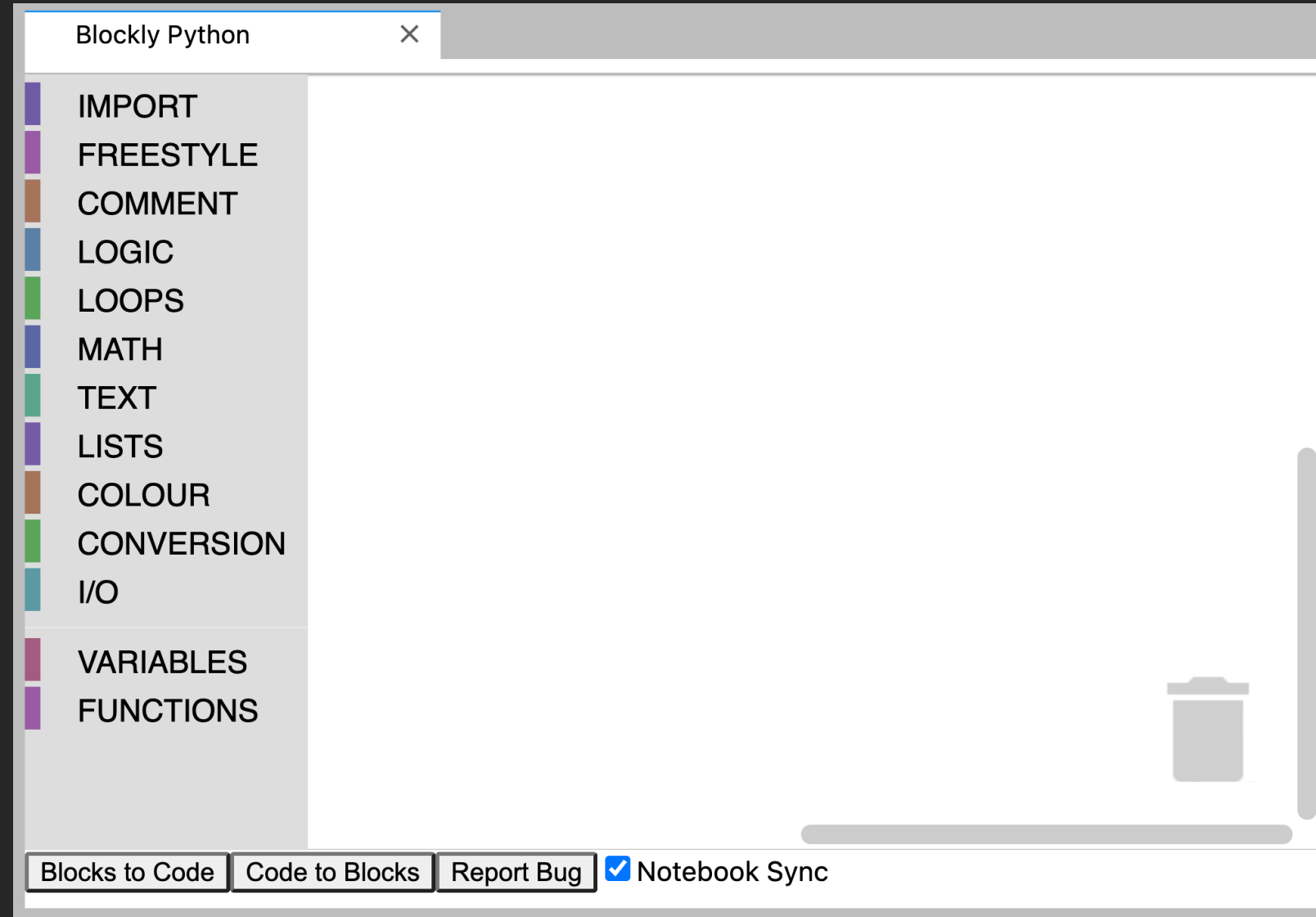
This will open a Blockly tab.



# How-to: Open Blockly Interface (2)

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

**But** sometimes it may not render correctly...

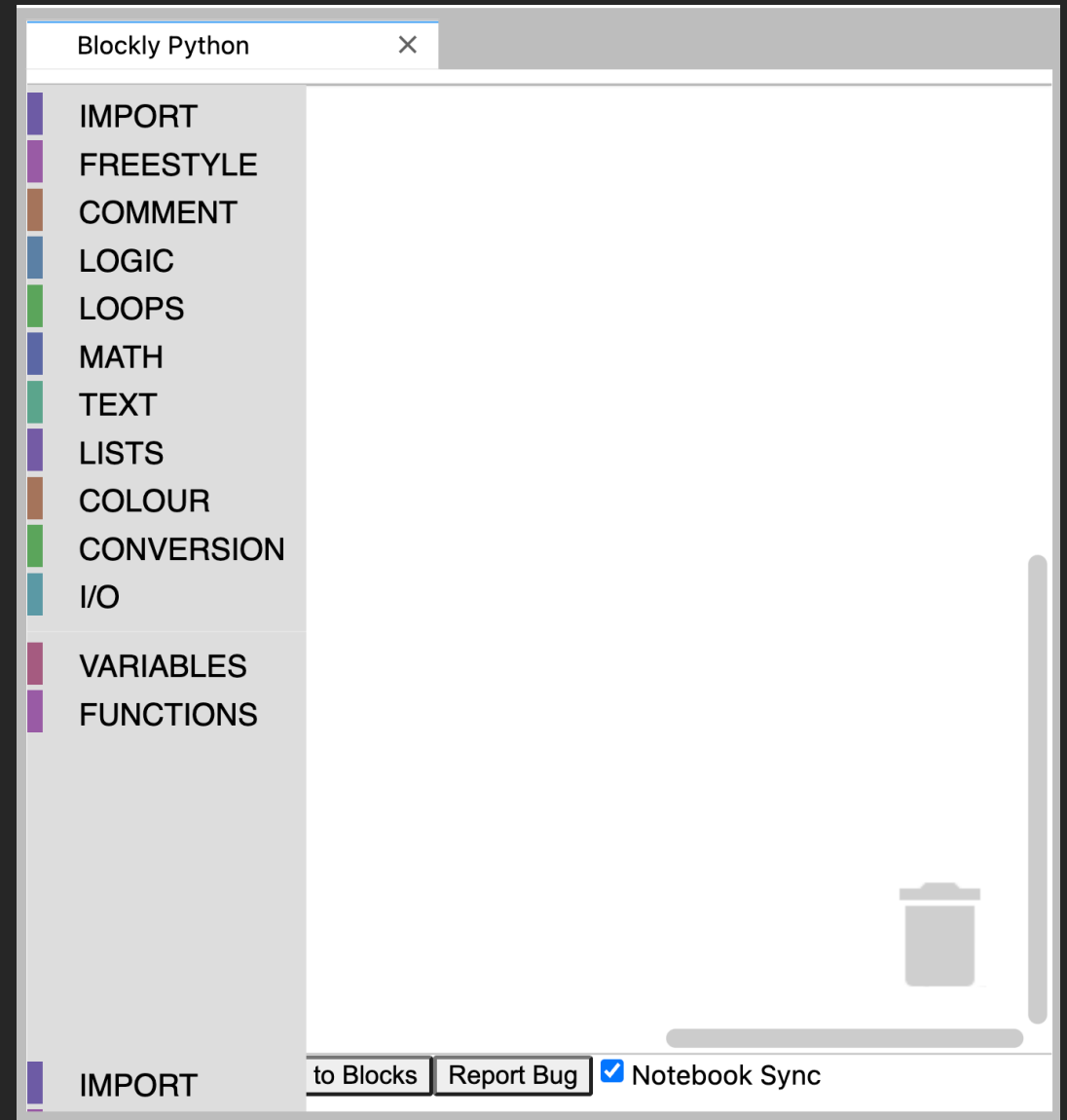


# How-to: Fix Blockly Interface Render Issues

**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

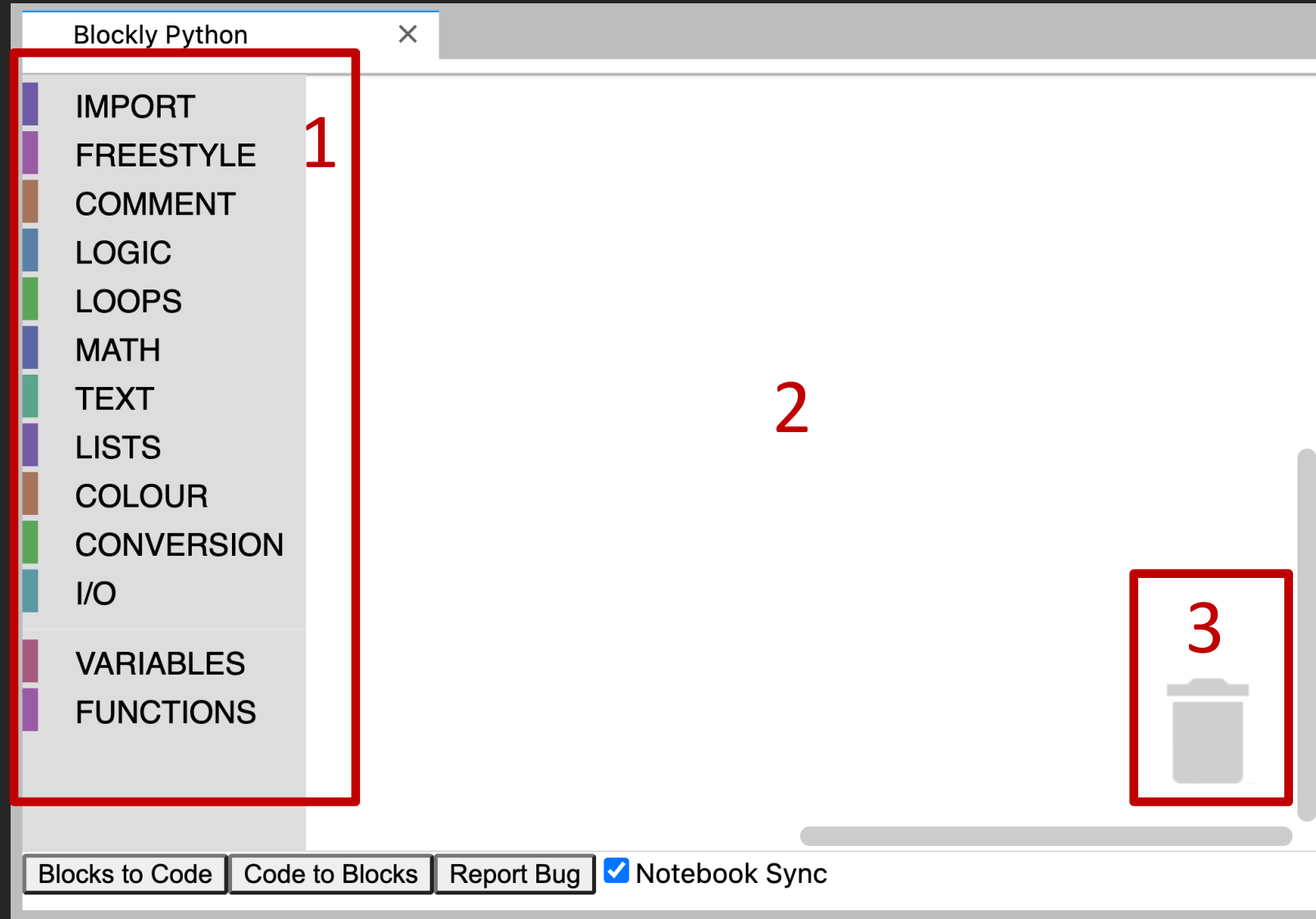


# Activity: Open Blockly Editor

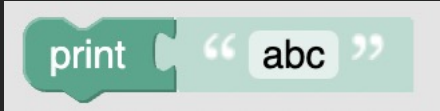
1. Close all open tabs (except pdf)
2. Reload the page
3. Open Blockly Editor & confirm rendered **correctly**
4. Close Blockly Editor
5. Open Blockly Editor & confirm rendered **incorrectly**
6. Reload the page
7. Confirm Blockly Editor rendered **correctly**

# Blockly Interface

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



# Activity: Playtime with Blocks

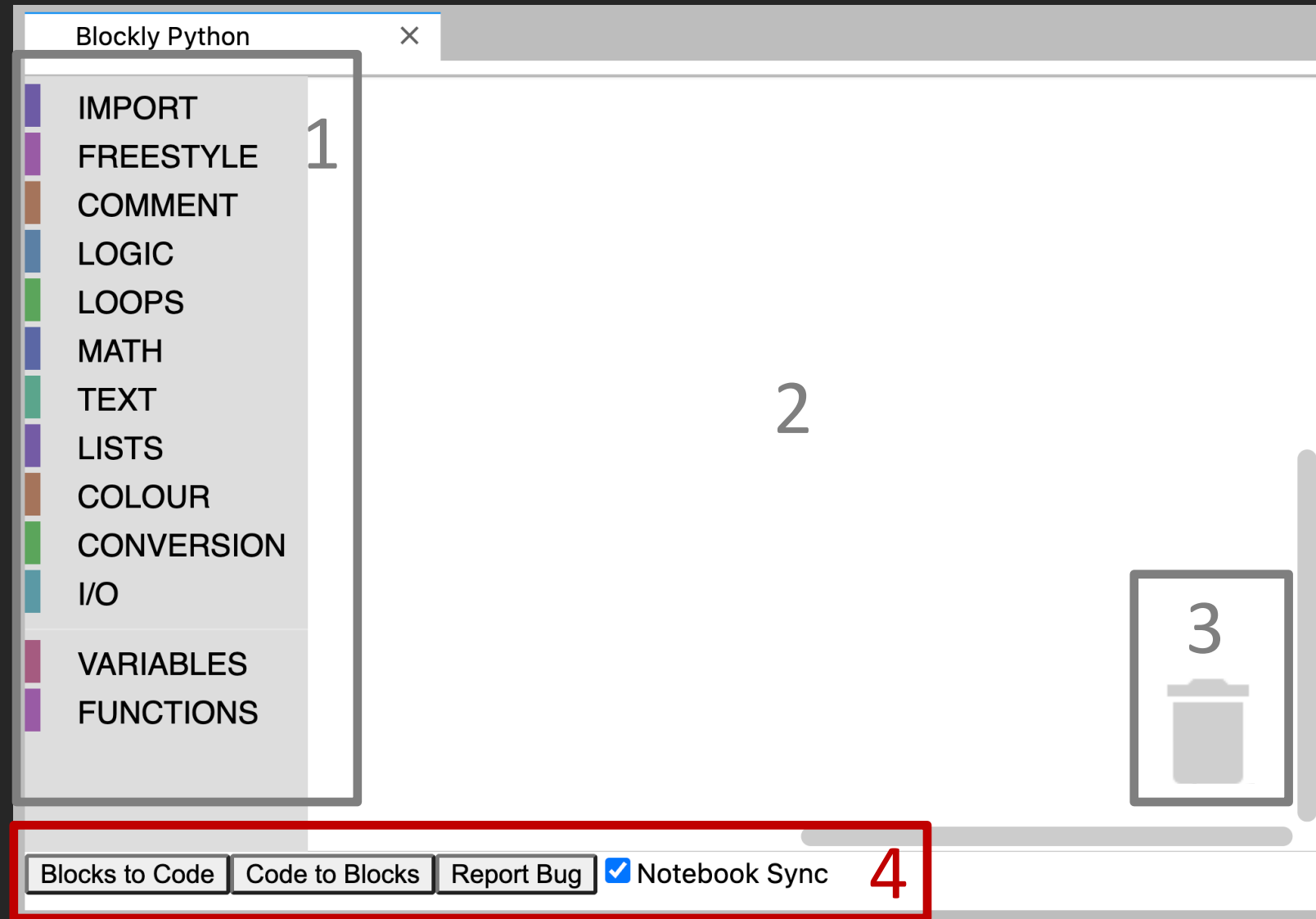
1. Open Blockly Editor (if not already open)
2. Click the Text section of Blockly Toolbox
3. Find the “print text” block The image shows a Blockly 'print text' block. It is a light blue block with a tab on the left side. The word 'print' is written in a dark blue font on the tab. To the right of the tab, the text '“ abc ”' is displayed in a dark blue font, indicating the text to be printed.
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 with two main panes. The left pane, titled 'Blockly Python', contains a Blockly workspace with a single block: `import pandas as pd`. The right pane, titled 'Data-science-and-the-nat', shows a Jupyter notebook with the following content:

### Load the data into a dataframe

Use [Data Science and the Nature of Data](#) or the [Reference](#) if you've forgotten any of these steps.

Import the `pandas` library, which lets us work with dataframes:

- `import pandas as pd`

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

Load a dataframe with the data in "datasets/flowers.csv" and display it:

- Set `dataframe` to with `pd` do `read_csv` using `"datasets/flowers.csv"`
- `dataframe` (to display)

```
[4]: dataframe = pd.read_csv('datasets/flowers.csv')
```

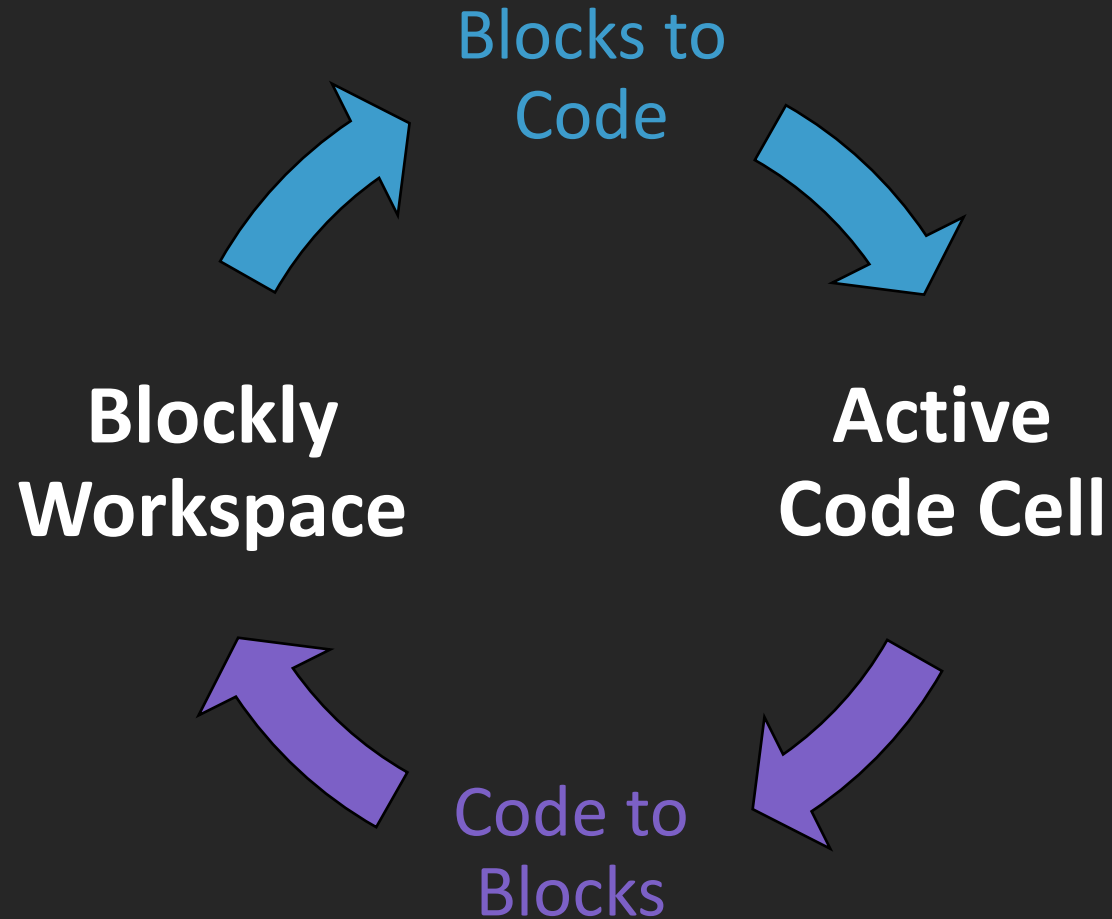
```
dataframe
```

```
[4]:
```

	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

At the bottom of the interface, the status bar shows '1 10 xpython | Idle' on the left and 'Mode: Edit Ln 2, Col 1 Data-science-and-the-nature-of-data-PS.ipynb' on the right.

# JupyterLab Blockly Integration



# Blocks to Code

Blocks

Blocks to  
Code

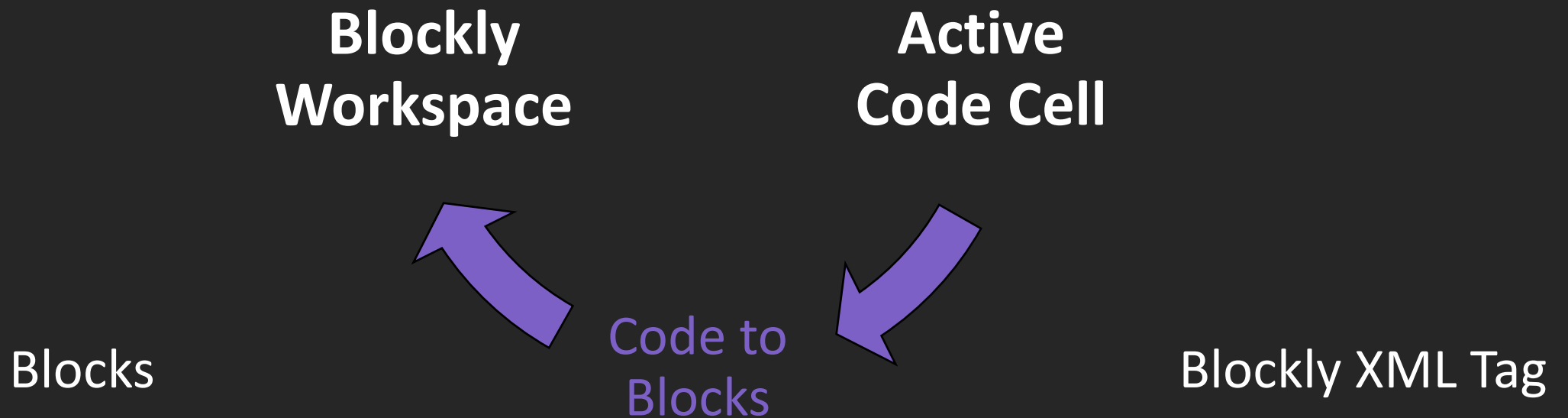
Python code  
Blockly XML Tag

**Blockly  
Workspace**

**Active  
Code Cell**



# Code to Blocks



# Activity: 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 upload to Discord

# Summary

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