

# Using Blockly in Jupyter Notebooks

# What is Blockly?

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

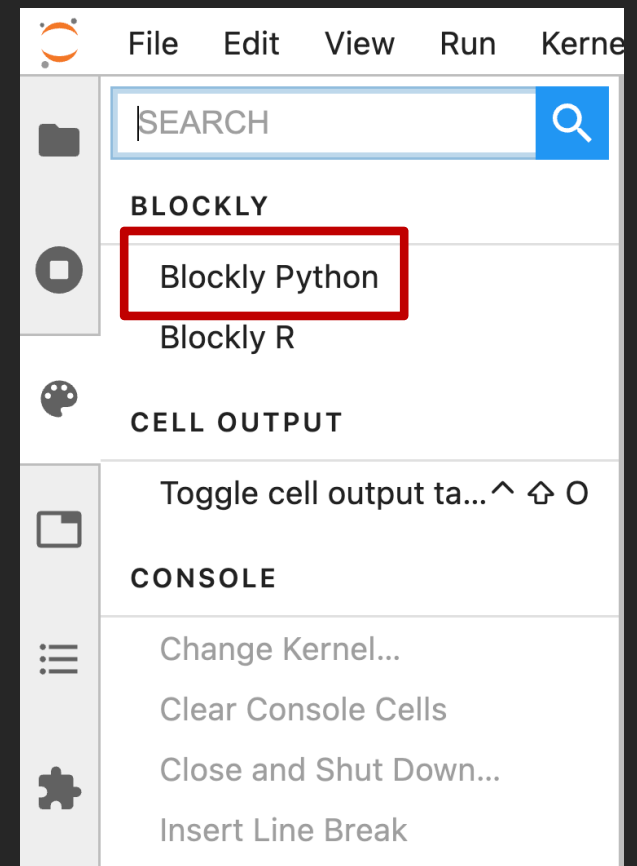


```
print('Hi')
```

# How-to: Open Blockly Interface (1)

1. Open Commands
2. Click Blockly Python

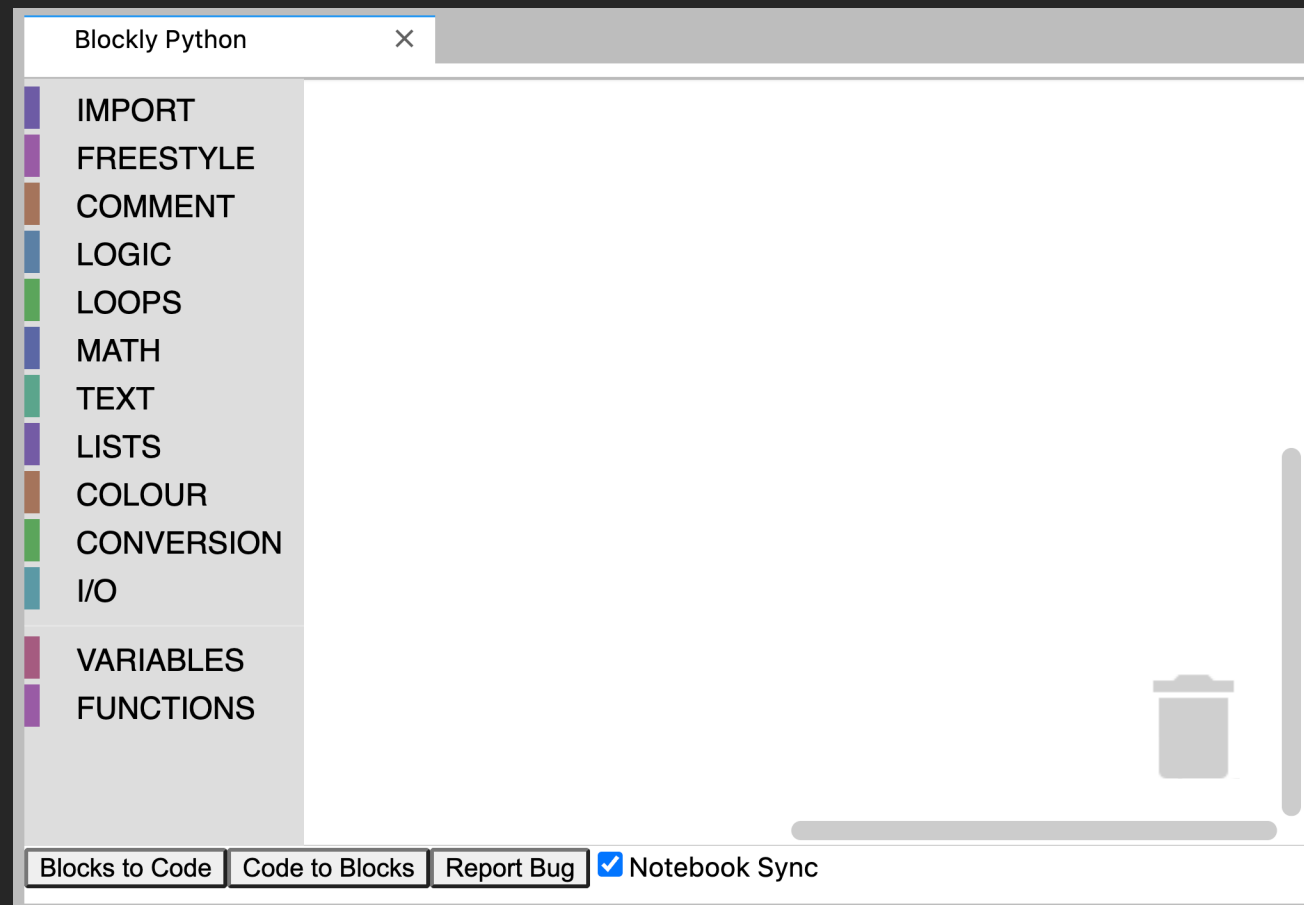
This will open a Blockly tab.



# How-to: Open Blockly Interface (2)

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

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

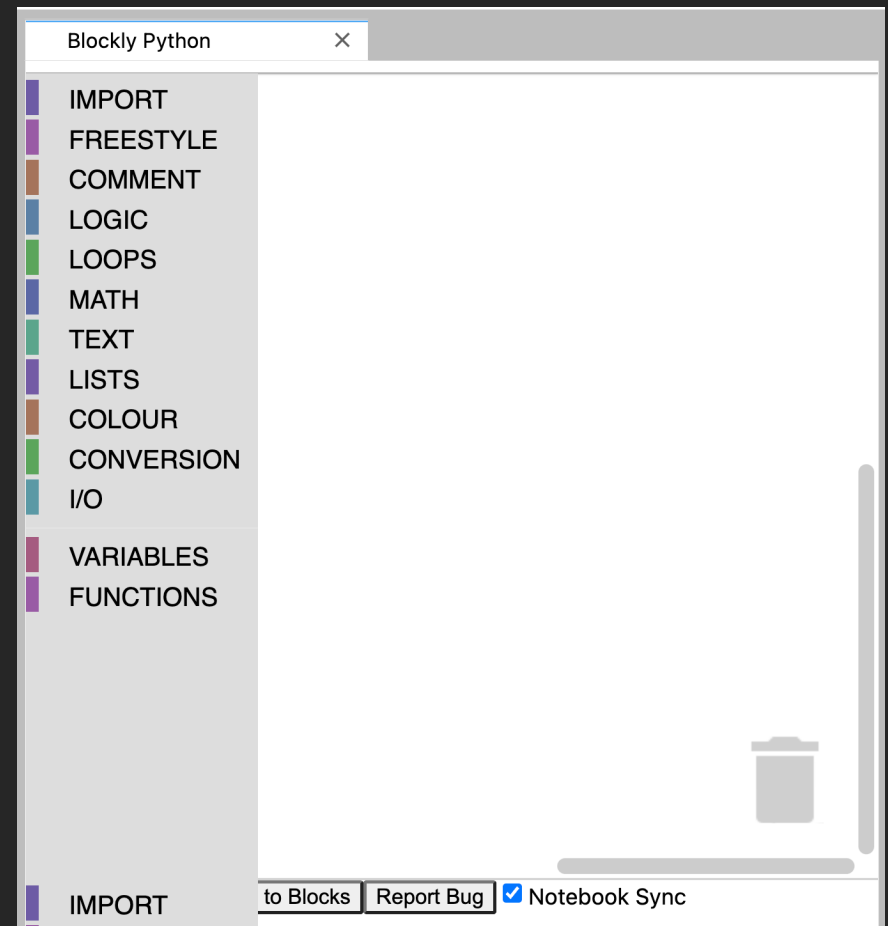


# How-to: Fix Blockly Interface Render Issues

**Solution:** Reload the page.

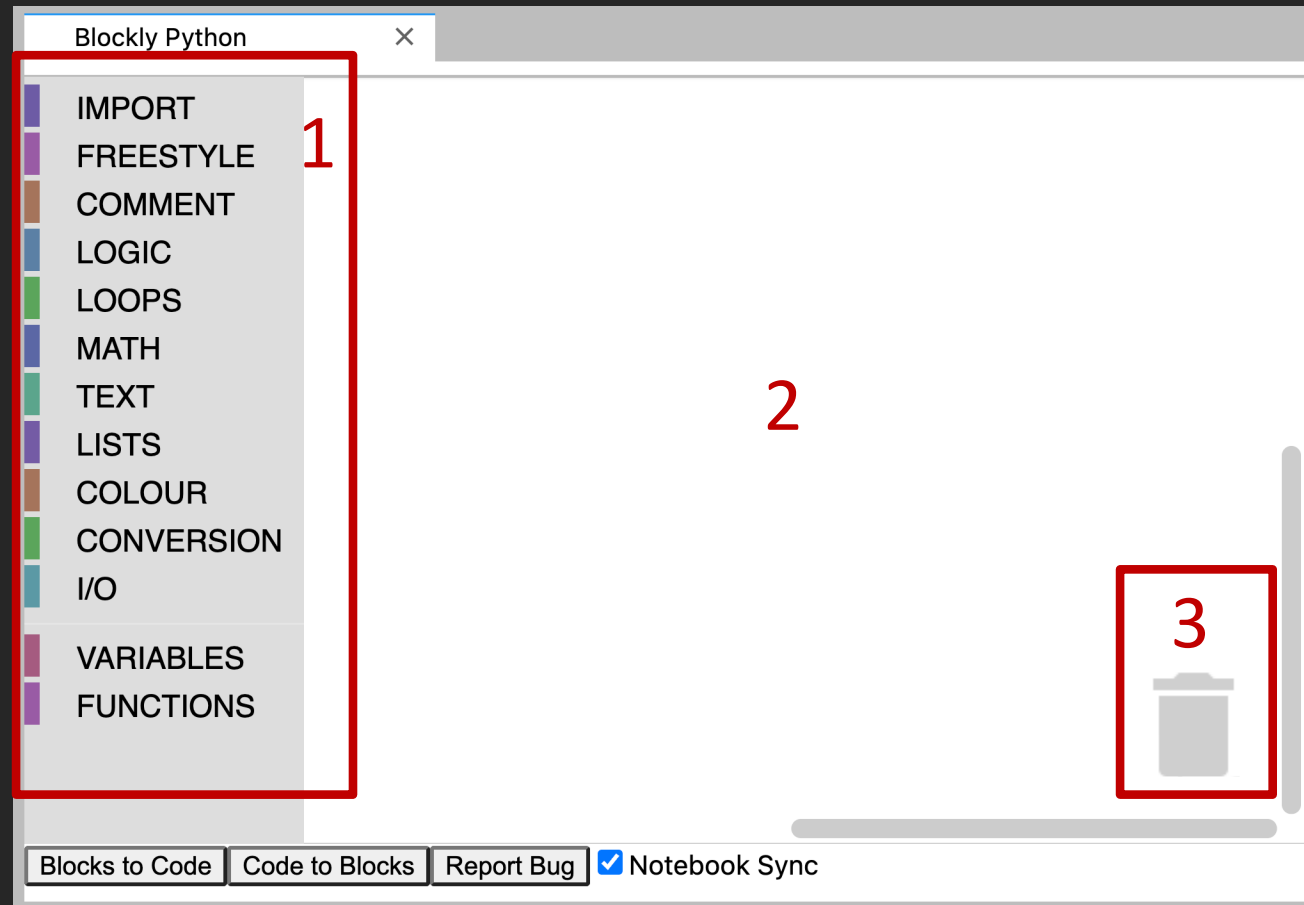
To avoid render errors:

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



# Blockly Interface

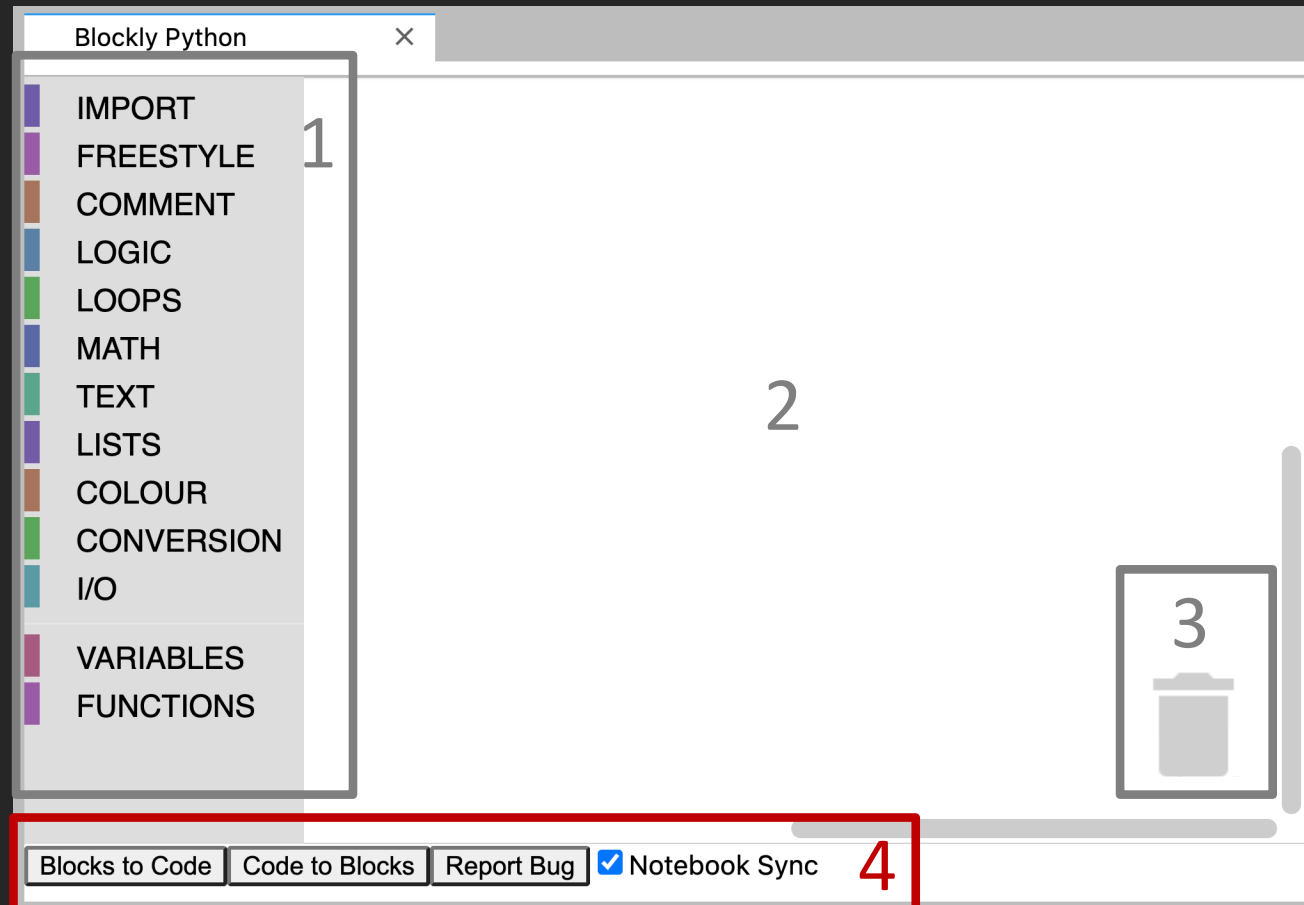
1. Blockly Toolbox
2. Blockly Workspace
3. Blockly Trash Can



# 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 components: a Blockly editor on the left and a Code editor on the right.

**Blockly Editor (Left):**

- Menu: File, Edit, View, Run, Kernel, Tabs, Settings, Help
- Panel: Blockly Python
- Left sidebar: A list of block categories including IMPORT, FREESTYLE, COMMENT, LOGIC, LOOPS, MATH, TEXT, LISTS, COLOUR, CONVERSION, I/O, VARIABLES, and FUNCTIONS.
- Workspace: A single block is present with the code `import pandas as pd`.
- Bottom bar: Buttons for "Blocks to Code", "Code to Blocks", "Report Bug", and "Notebook Sync" (checked).

**Code Editor (Right):**

- Tab: Data-science-and-the-nat
- Language: xpython
- Section: Load the data into a dataframe
- Text: Use [Data Science and the Nature of Data](#) or the [Reference](#) if you've forgotten any of these steps.
- Text: Import the `pandas` library, which lets us work with dataframes:
- List:
  - `import pandas as pd`
- Code cell [3]:

```
[3]: import pandas as pd
#<xml xmlns="https://developers.google.com/blockly/xml"><variables><vari
```
- Text: Load a dataframe with the data in "datasets/flowers.csv" and display it:
- List:
  - Set dataframe to with `pd` do `read_csv` using "datasets/flowers.csv"
  - dataframe (to display)
- Code cell [4]:

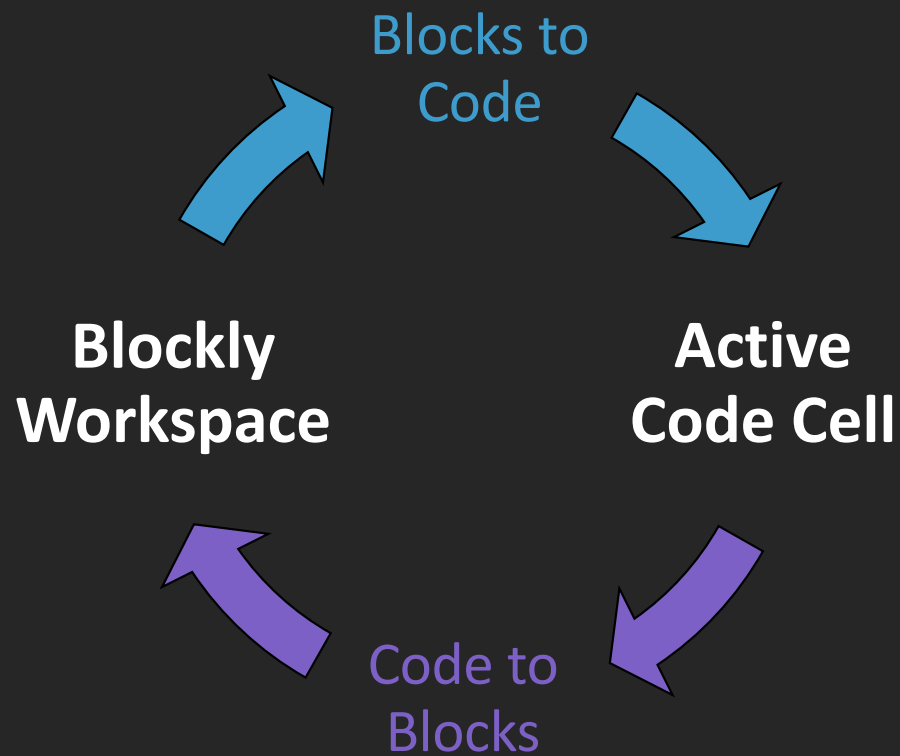
```
[4]: dataframe = pd.read_csv('datasets/flowers.csv')
dataframe
#<xml xmlns="https://developers.google.com/blockly/xml"><variables><vari
```
- Table output:

	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
- Code cell [4] (continued):

```
#<xml xmlns="https://developers.google.com/blockly/xml"><variables><vari
```
- Bottom bar: Mode: Edit, Ln 2, Col 1, Data-science-and-the-nature-of-data-PS.ipynb



# JupyterLab Blockly Integration



# Blocks to Code

Blocks

Blocks to  
Code

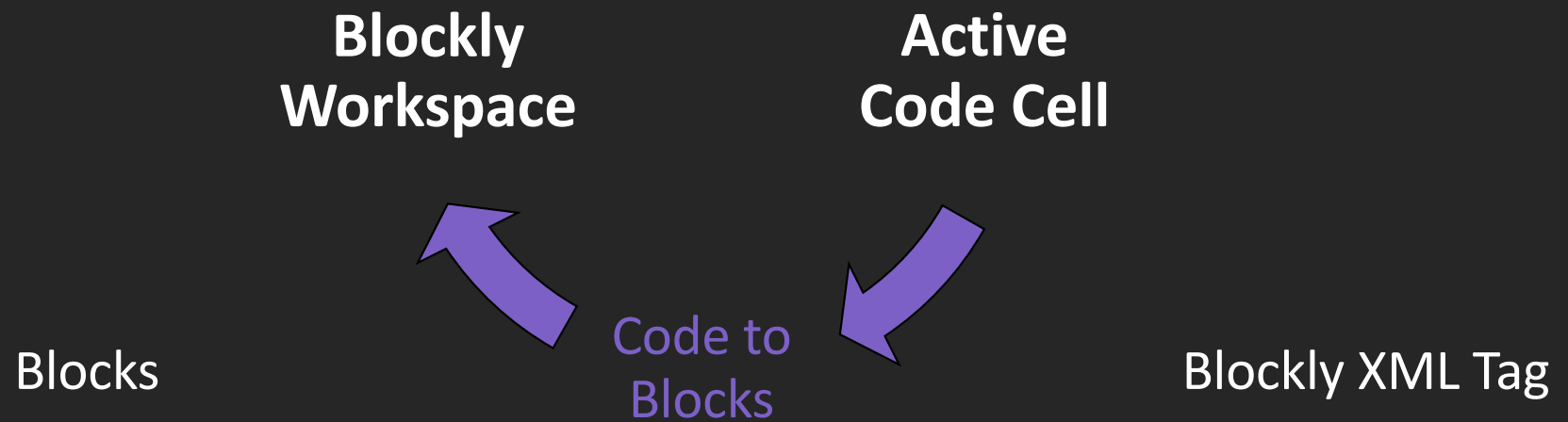
Python code  
Blockly XML Tag

**Blockly  
Workspace**

**Active  
Code Cell**



# Code to Blocks



# Summary

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