

PYTHON

# FLOW CONTROL

Tien-Lam Pham  
Anh-Tuan Nguyen  
Phenikaa School of Computing

# OUTLINE

---

- if
- if-else
- if-elif-else
- for loop
- while loop
- assignment

# IF CONDITION

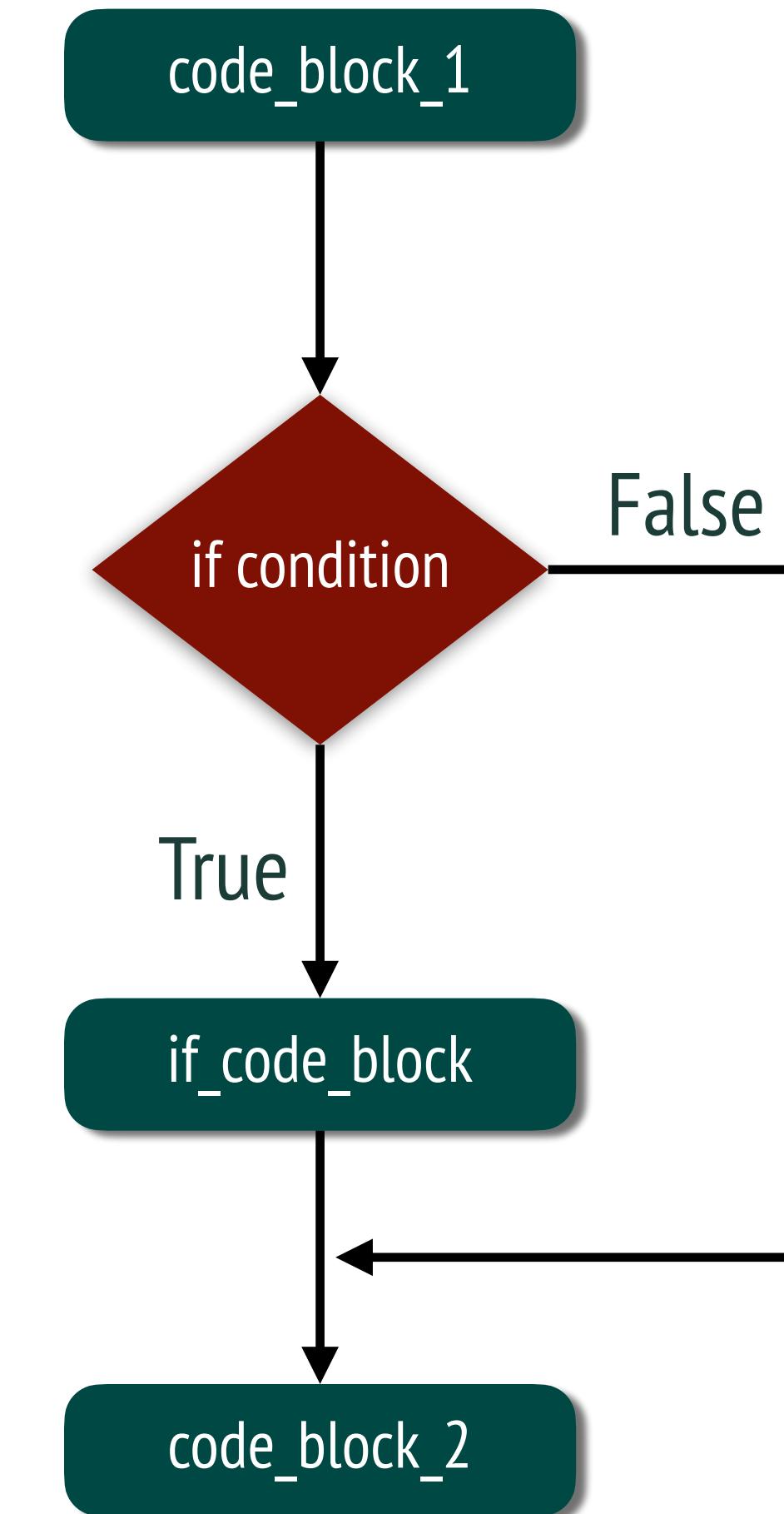
A screenshot of a code editor window titled "if.py". The code is as follows:

```
1 # code_block_1
2 if condition:
3     # if_code_block
4
5 # code_block_2
```

The code editor has three callout boxes with arrows pointing to specific parts of the code:

- A green box labeled "keyword" points to the word "if" in line 2.
- A green box labeled "colon" points to the colon ":" in line 2.
- A green box labeled "indentation" points to the four spaces under the "if" keyword in line 3.

The status bar at the bottom shows "Line 7, Column 1" and "Tab Size: 4".



# IF CONDITION

The screenshot shows a Jupyter Notebook interface on a web browser. The notebook title is "Python (unsaved changes)". The code in cell In [11] is:

```
# code_block_1
print("Start")

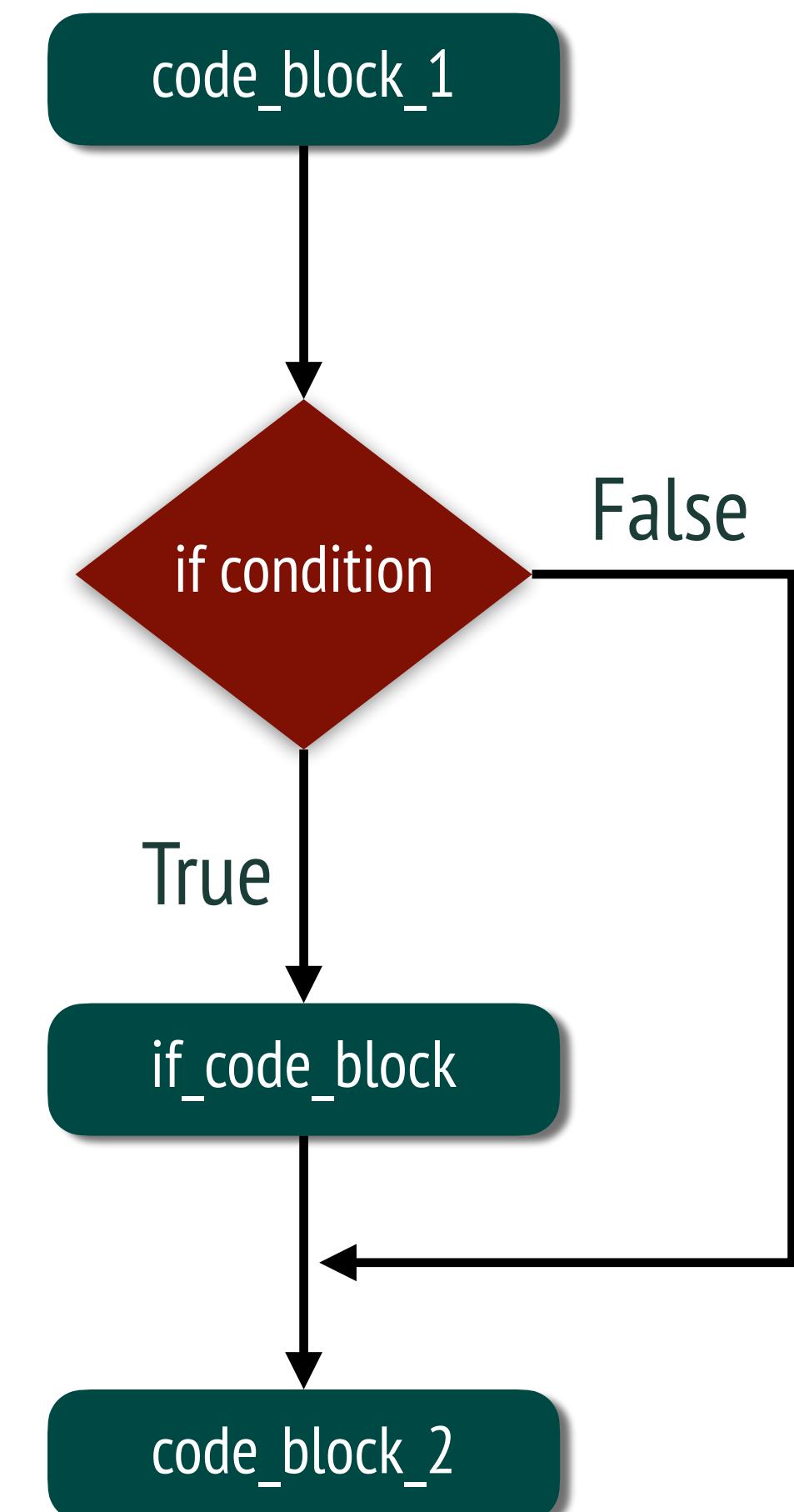
if 10>5:
    # if_code_block
    print("True")

# code_block_2
print("Stop")
```

The output of the code is:

```
Start
True
Stop
```

The browser's address bar shows "localhost". The top navigation bar includes links for WiDs, PhD, DeadLine, scrap, Environments, MIE, My-SQL, Jp, Street style, Passion, Scholarship, Machine learning, Bike & Accessory, Your Projects, LaTeX Editor, and equipments.

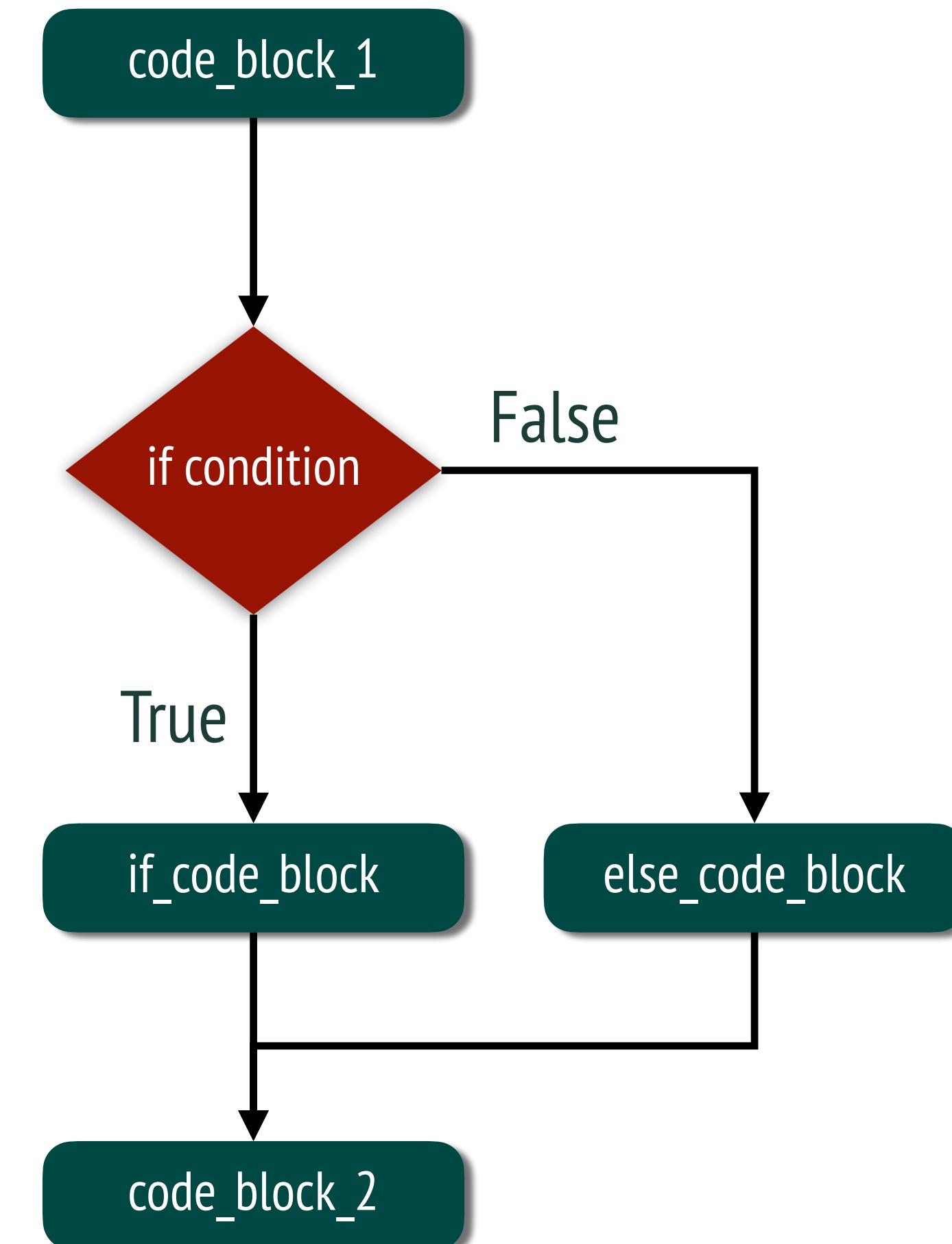


# IF-ELSE CONDITION

A screenshot of a code editor window titled "if.py". The code is as follows:

```
1 # code_block_1
2 if condition:
3     # if_code_block
4 else:
5     # else_code_block
6
7 # code_block_2
8
9
```

The cursor is at line 9, column 1. The status bar shows "Tab Size: 4".



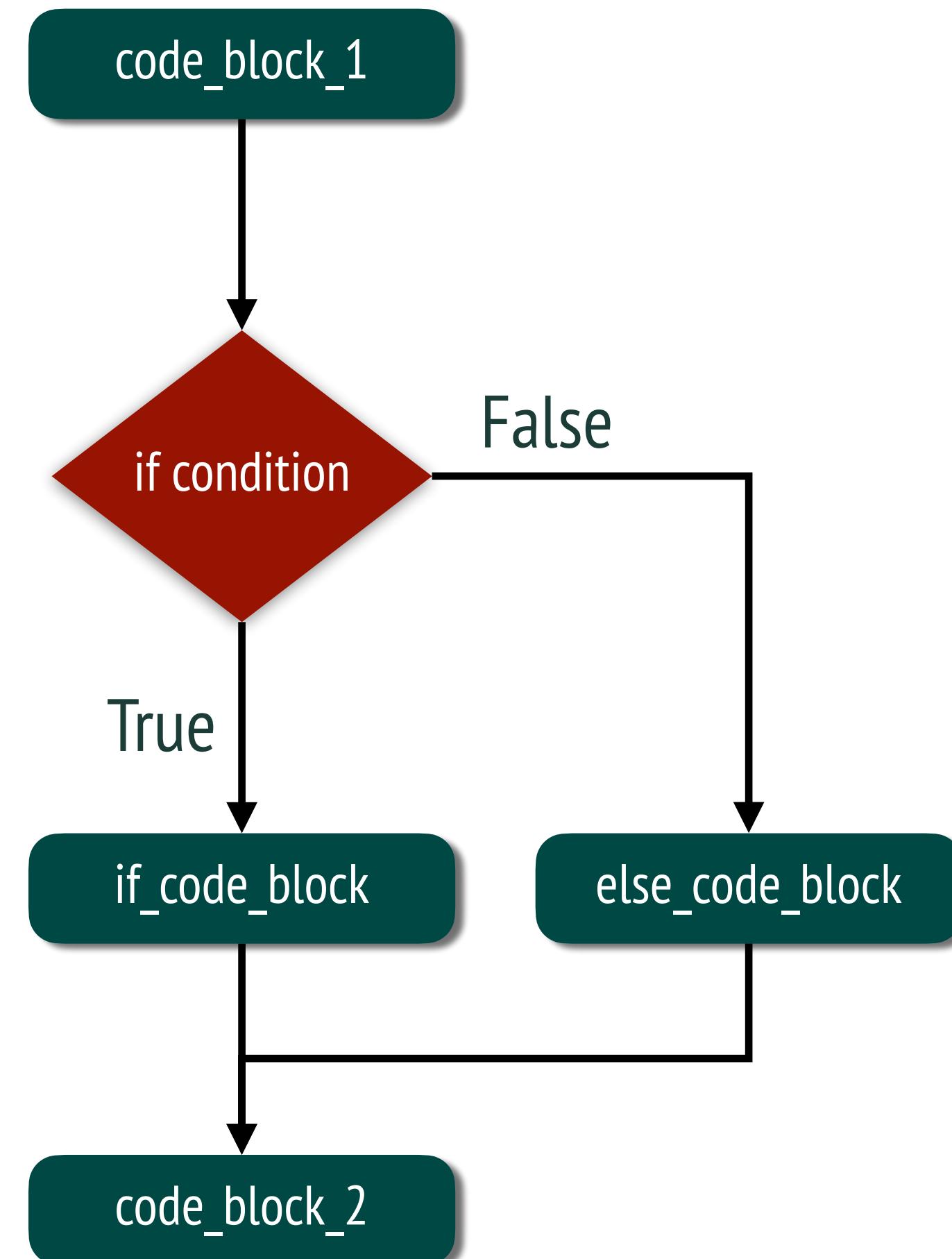
# IF-ELSE CONDITION

The screenshot shows a Jupyter Notebook interface. In the code cell (In [14]), the following Python code is written:

```
# code_block_1
print("Start")
a = 10
b = 5
if b>a:
    # if_code_block
    print("b > a")
else:
    # else_code_block
    print("a >= b")
# code_block_2
print("Stop")
```

The output cell (In [ ]:) shows the execution results:

```
Start
a >= b
Stop
```

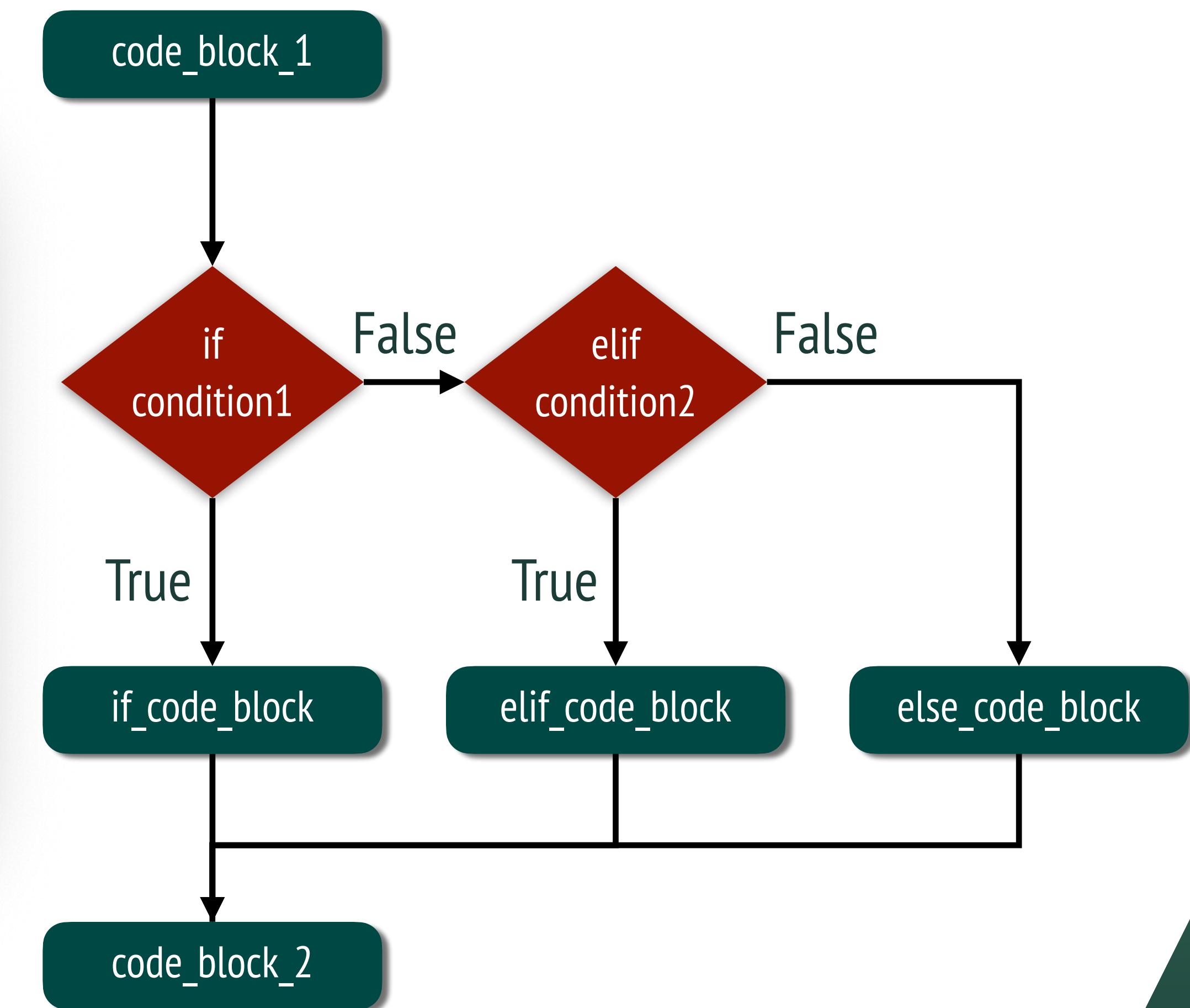


# IF-ELIF-ELSE CONDITION

A screenshot of a code editor window titled "if.py". The code is as follows:

```
1 # code_block_1
2 if condition1:
3     # if_code_block
4 elif condition2:
5     # elif_code_block
6 else:
7     # else_code_block
8 # code_block_2
```

The status bar at the bottom indicates "Line 9, Column 1; Saved ~/Dropbox/Lam-Quyen/HUS/python\_programm".



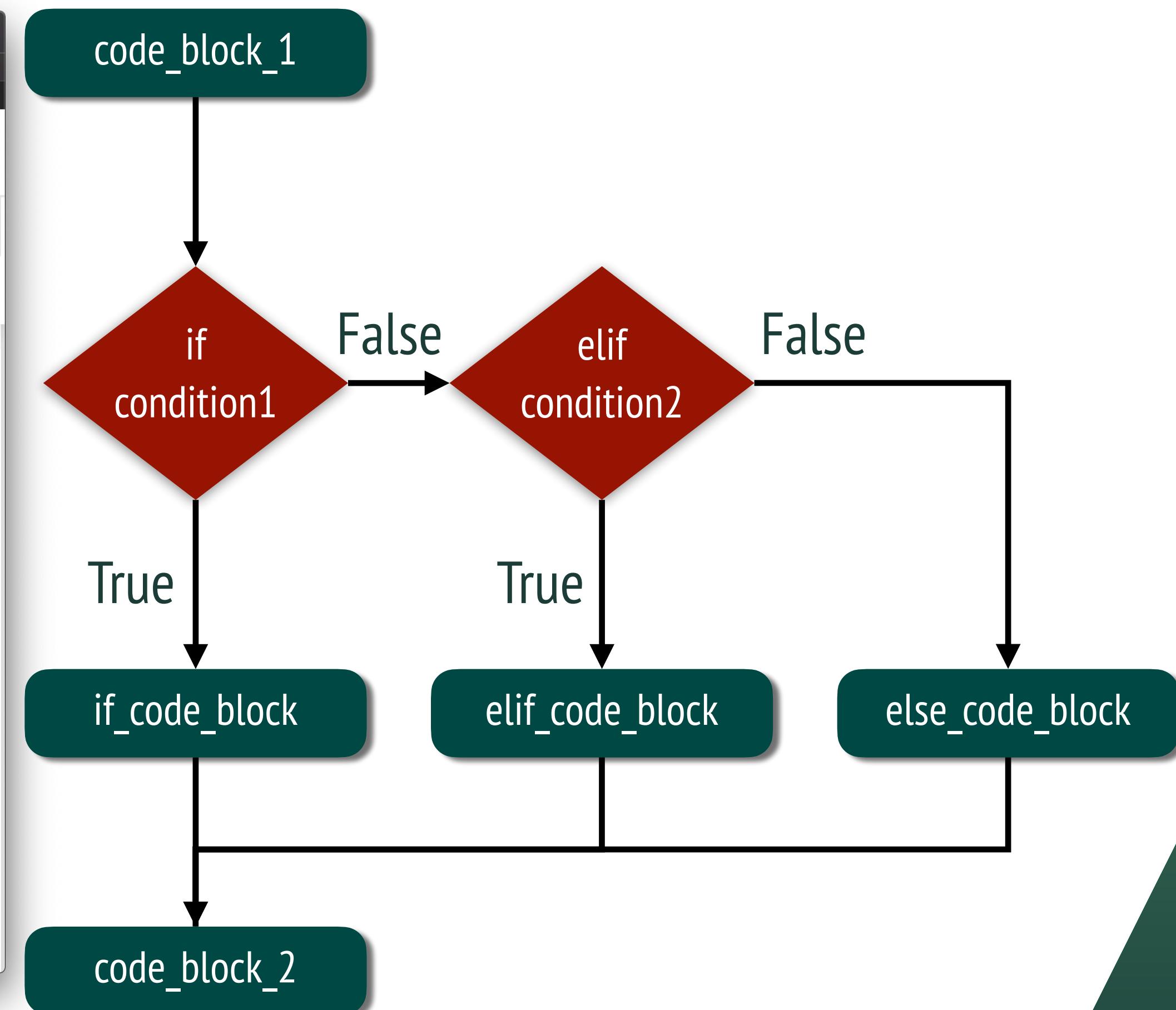
# IF-ELIF-ELSE CONDITION

The screenshot shows a Jupyter Notebook interface on a Mac OS X system. The title bar says "localhost" and "Python - Jupyter Notebook". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The toolbar has icons for file operations and cell execution. In the code cell (In [13]), the following Python code is written:

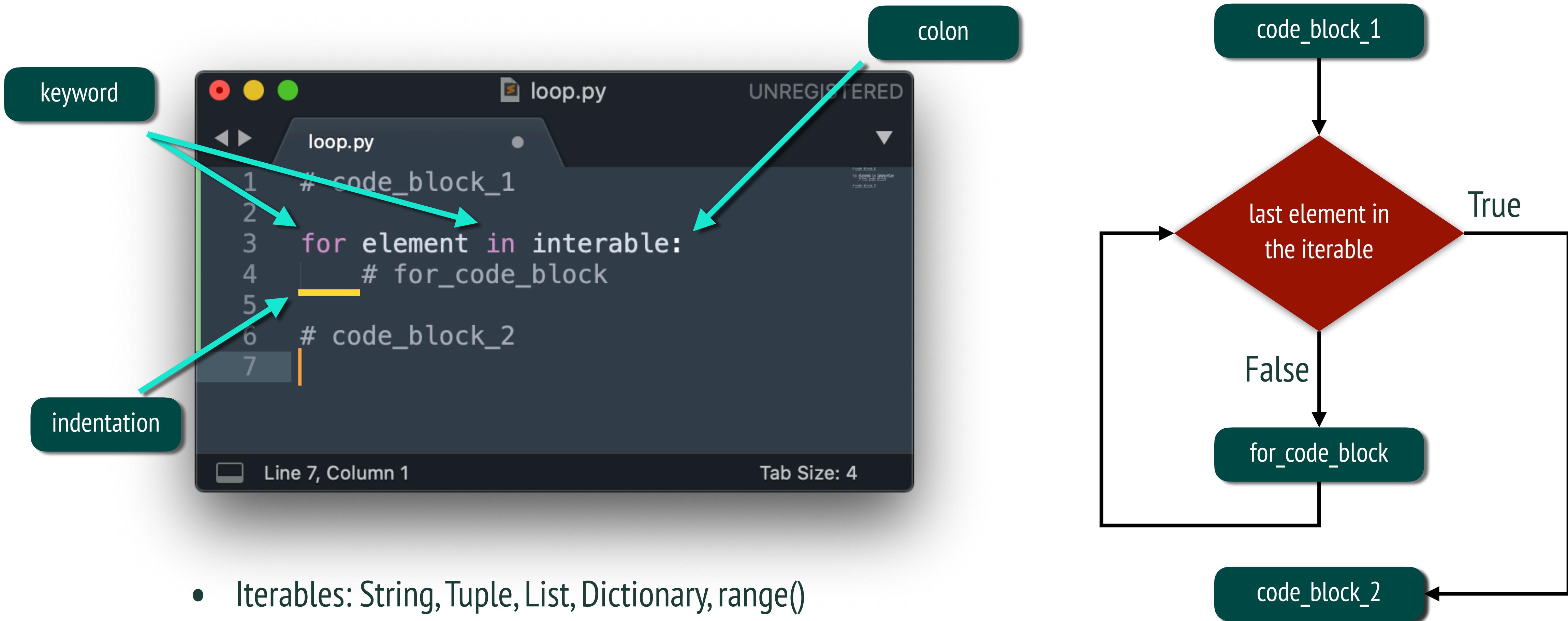
```
# code_block_1
print("Start")
a = 10
b = 5
if b>a:
    # if_code_block
    print("b > a")
elif a>b:
    # elif_code_block
    print("a > b")
else:
    # else_code_block
    print("a = b")
# code_block_2
print("Stop")
```

The output cell shows the results of the code execution:

```
Start
a > b
Stop
```

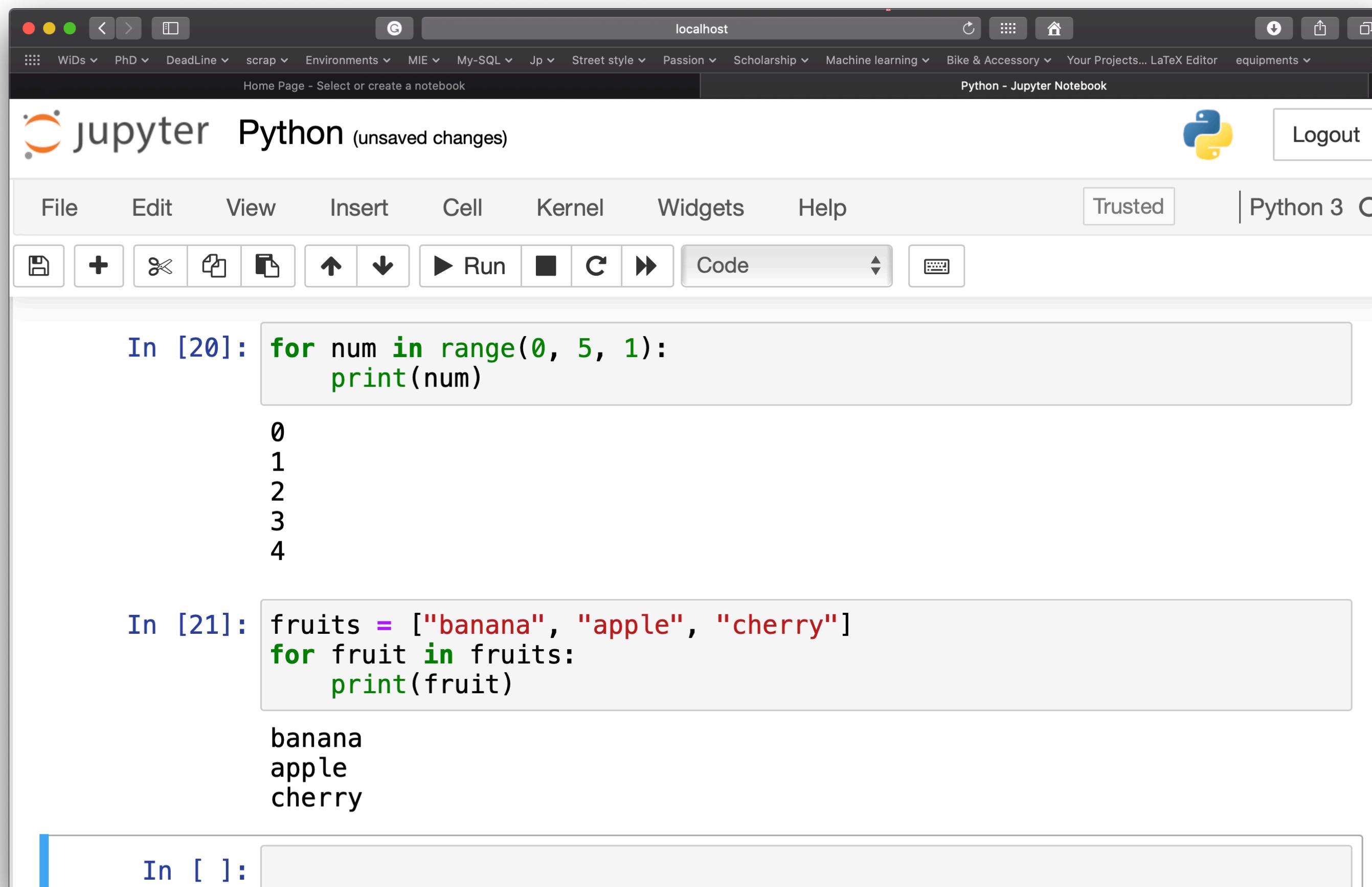


# FOR LOOP



# FOR LOOP

- Iterables: String, Tuple, List, Dictionary, range()

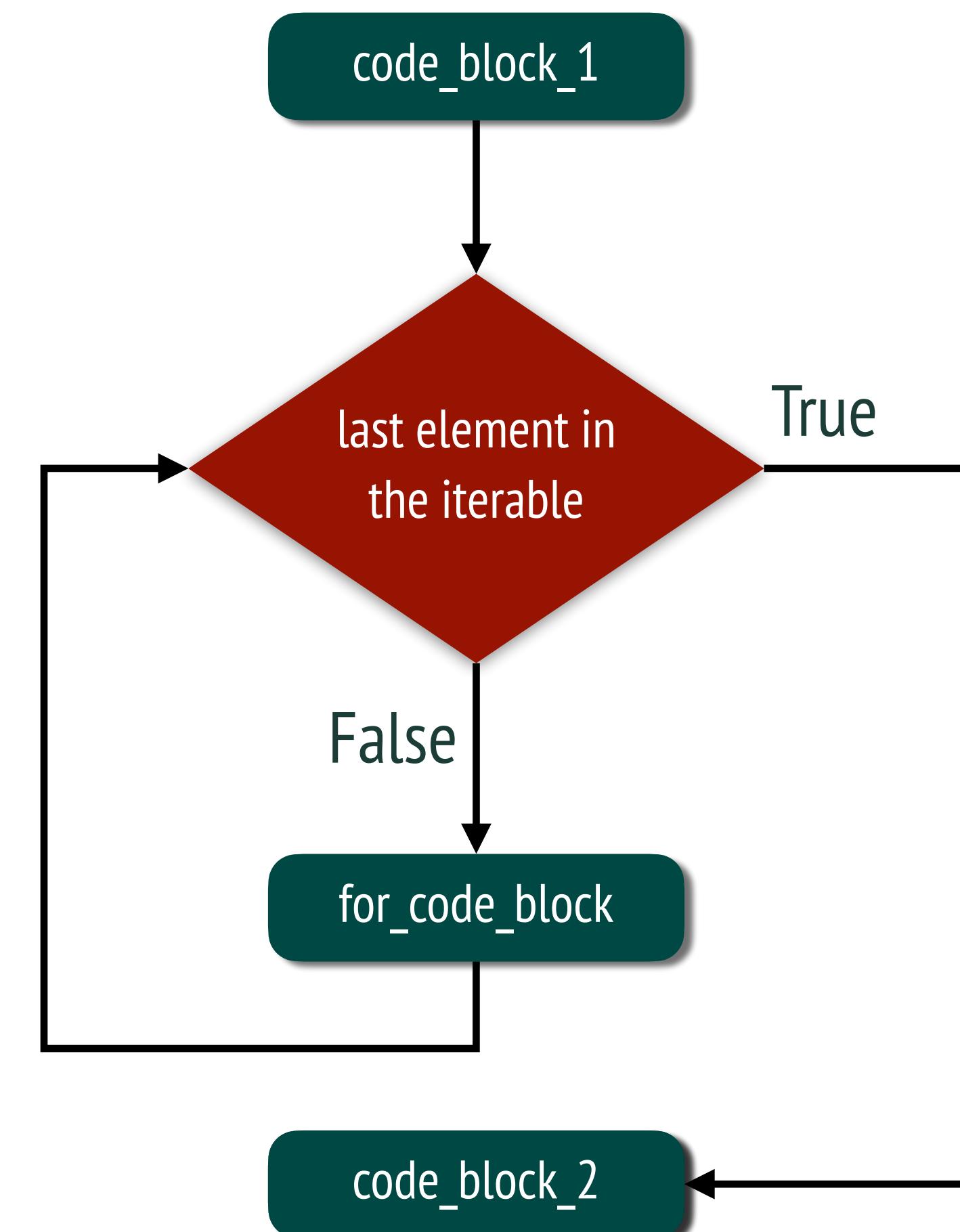


```
In [20]: for num in range(0, 5, 1):
    print(num)

0
1
2
3
4

In [21]: fruits = ["banana", "apple", "cherry"]
for fruit in fruits:
    print(fruit)

banana
apple
cherry
```



# FOR LOOP

- break keyword

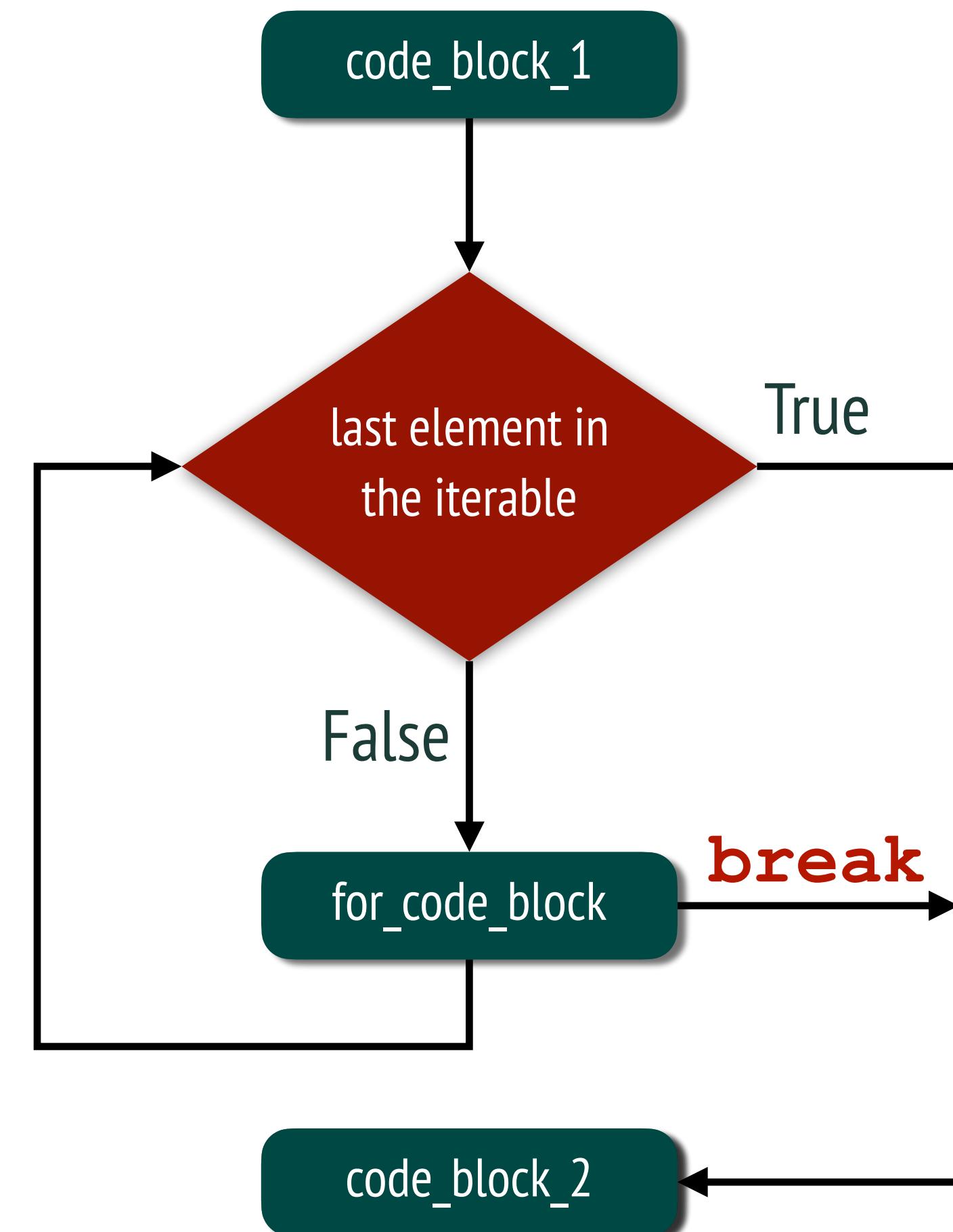
A screenshot of a Jupyter Notebook interface. The title bar shows 'localhost' and 'Python - Jupyter Notebook'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. Below the menu is a toolbar with icons for file operations and code execution. In the main area, cell [22] contains the following Python code:

```
In [22]: for num in range(0, 5, 1):
    if num == 3:
        break
    print(num)
```

The output of this cell is:

```
0
1
2
```

Below the code cell, there are four empty input fields labeled 'In [ ]:'.



# FOR LOOP

- continue keyword

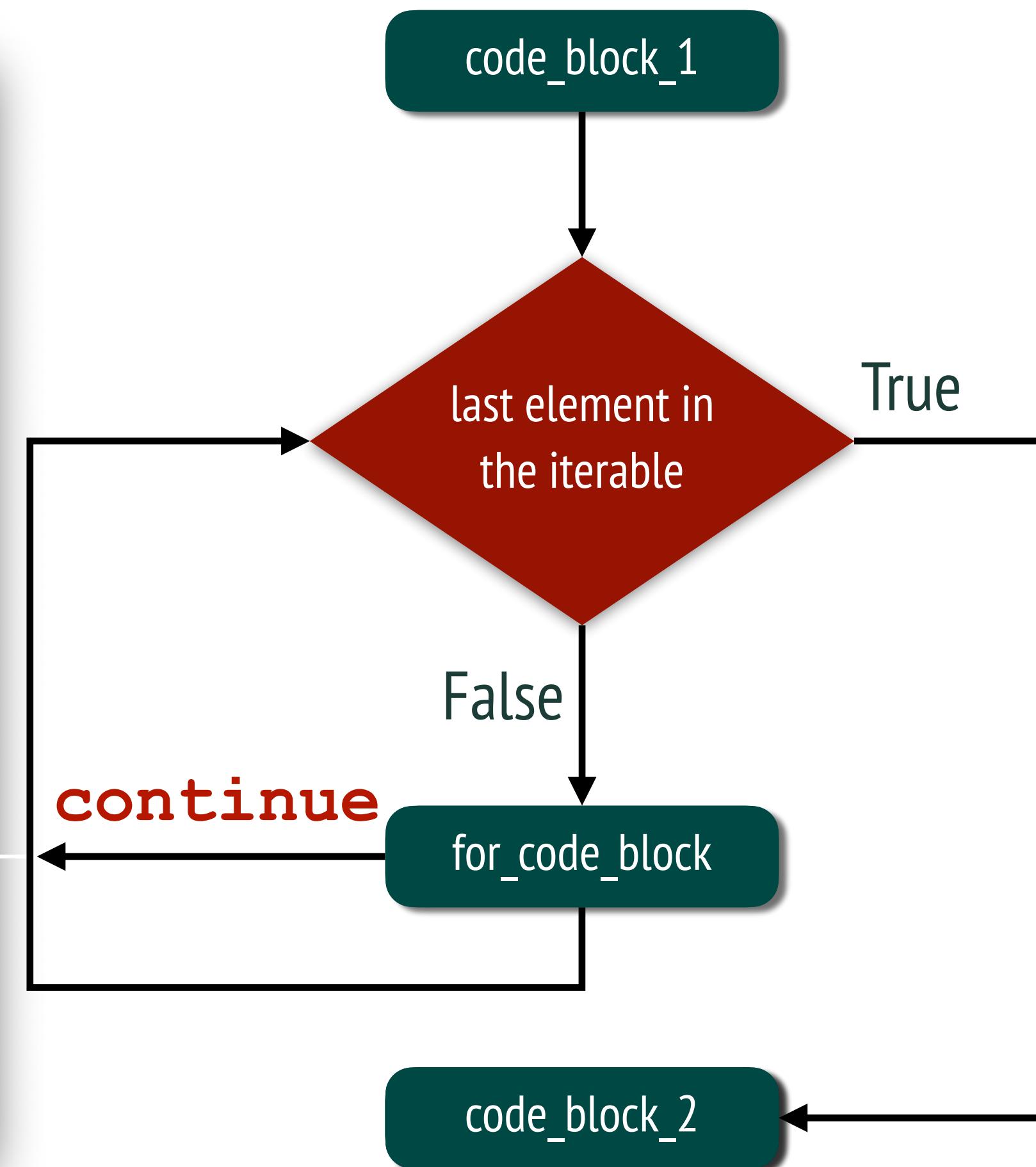
A screenshot of a Jupyter Notebook interface. The top bar shows the URL as localhost and the notebook title as Python - Jupyter Notebook. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. Below the menu is a toolbar with various icons. The code cell In [23] contains the following Python code:

```
In [23]: for num in range(0, 5, 1):
    if num == 3:
        continue
    print(num)
```

The output cell shows the results of the execution:

```
0
1
2
4
```

The input cell In [ ]: is currently active.



# WHILE LOOP

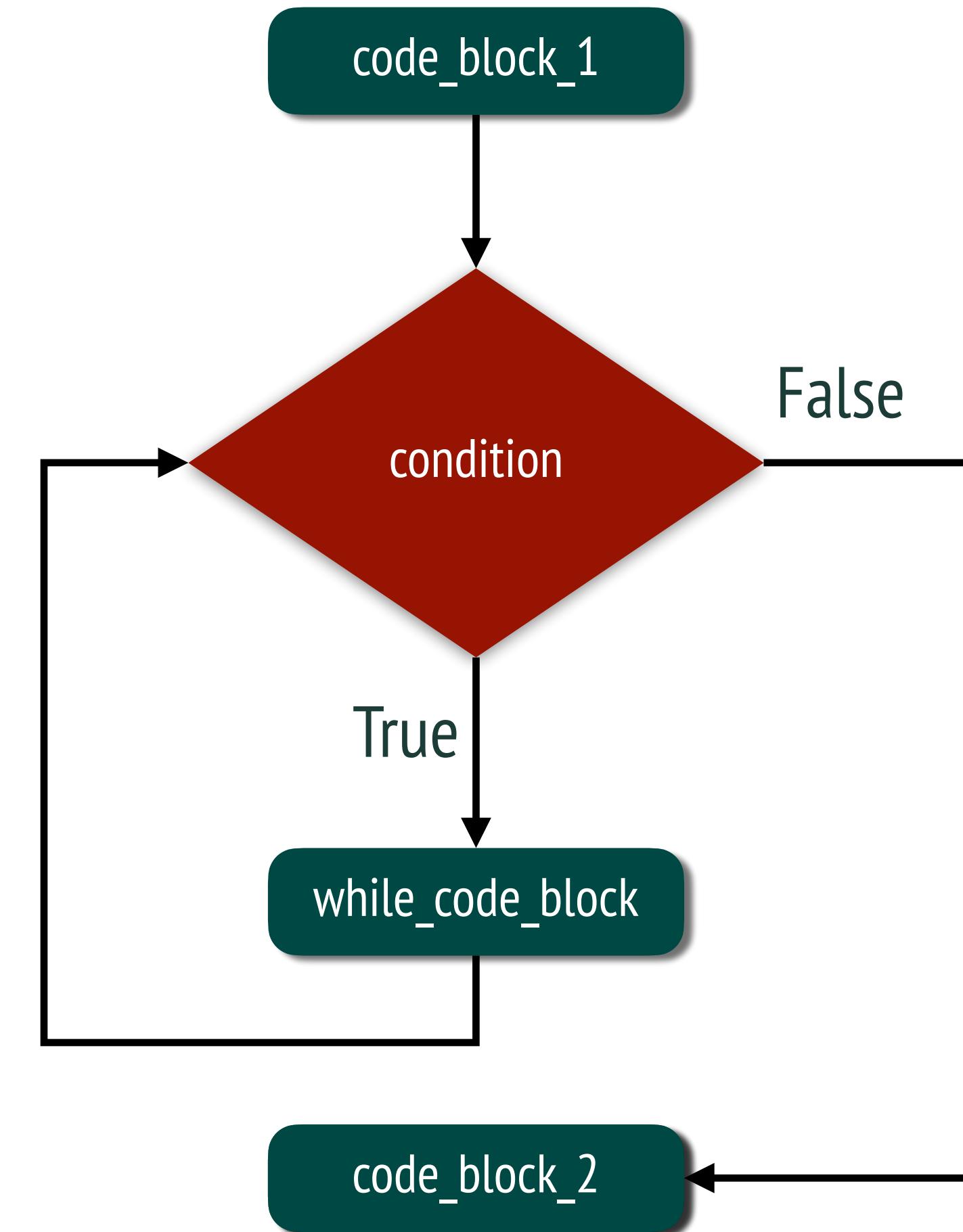
A screenshot of a code editor window titled "loop.py". The code is as follows:

```
1 # code_block_1
2
3 while condition:
4     # while code block
5
6     # code_block_2
7
```

The code editor has a dark theme. Arrows from callout boxes point to specific parts of the code:

- A green arrow labeled "keyword" points to the word "while".
- A green arrow labeled "colon" points to the colon character ":" at the end of line 3.
- A green arrow labeled "indentation" points to the four spaces under the "# while code block" comment on line 4.

At the bottom of the editor, it says "Line 7, Column 1" and "Tab Size: 4".



# WHILE LOOP

A screenshot of a Jupyter Notebook interface. The title bar shows 'localhost' and 'Python - Jupyter Notebook'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The toolbar has icons for file operations, run cell, and code editor. In cell [24], the code is:

```
In [24]: print("Start")
num = 0

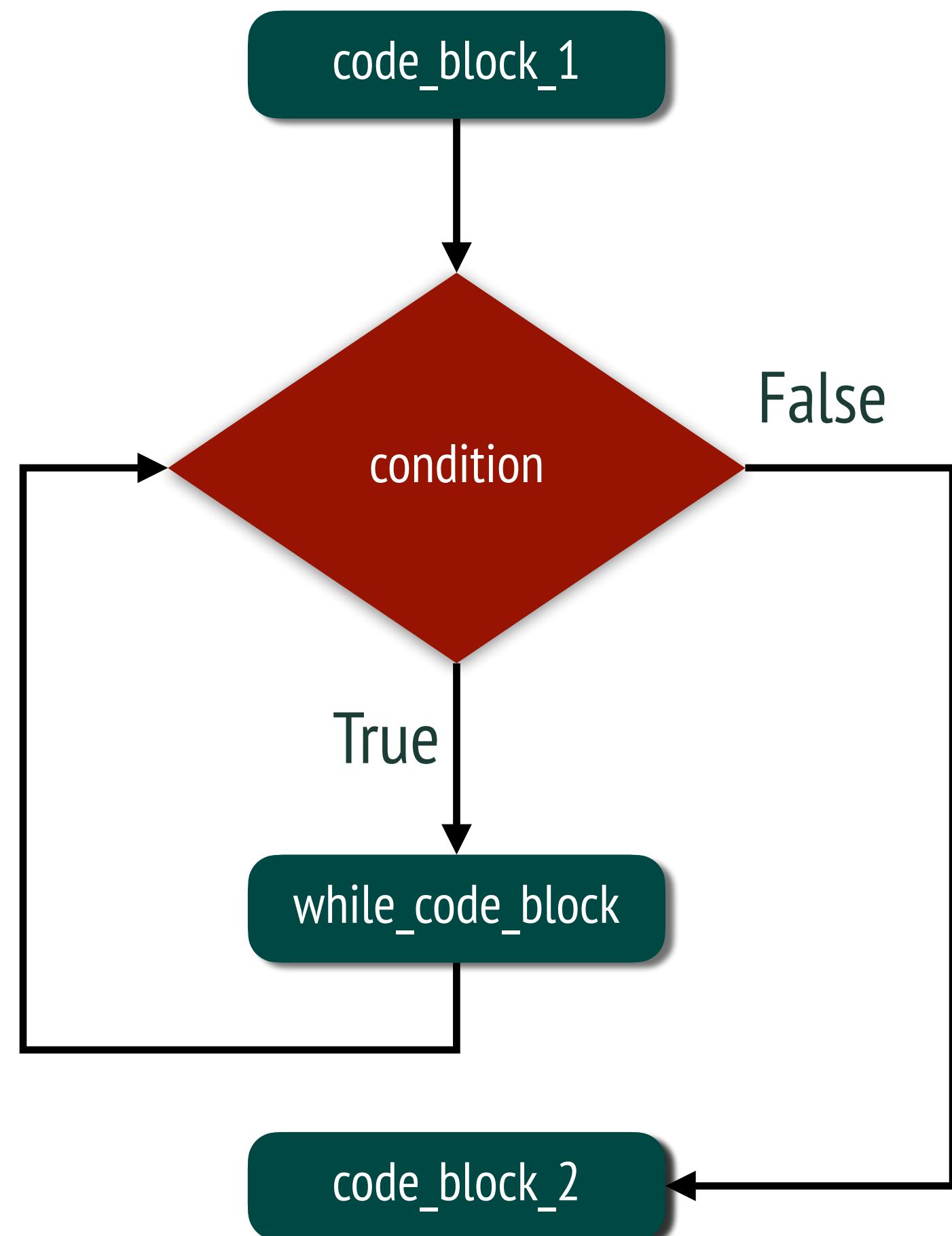
while num < 5:

    print(num)
    num += 1

print("Stop")
```

The output pane shows the execution results:

```
Start
0
1
2
3
4
Stop
```



# ASSIGNMENT

---