(Advanced) A Jupyter notebook is not a python program

Saturday, May 4, 2019 4:42 PM

It might seem that a Jupyter notebook is the same thing as a python program consisting of the concatenation of its cells.

Unfortunately, this is not true.

The difference is subtle and concerns how a Jupyter notebook is executed.

- One cell is executed at a time.
- Cells can be executed in any order.
- Cells can be changed after they are executed, and not executed again.
- During this, the state of the iPython interpreter is cumulative over what is executed, not what is written.

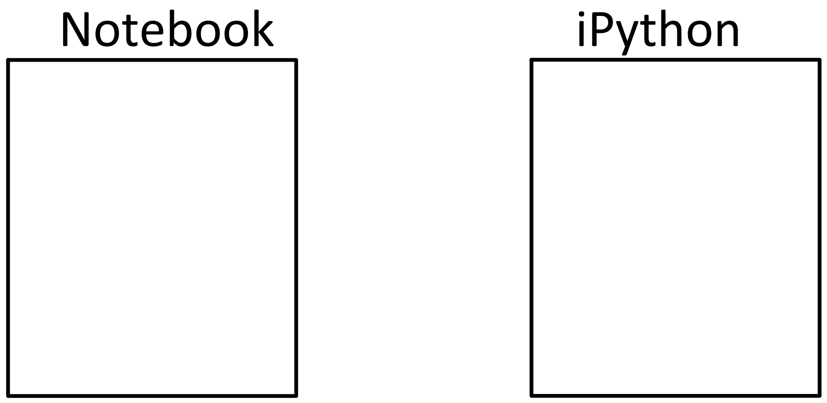
A better model of execution

- There are two components involved in executing a Jupyter notebook.
    a. A Notebook.
    b. An iPython interpreter.
- At any time, some cells have been executed and some cells have not been executed.
- Thus, the iPython state can be inconsistent with what the notebook seems to compute.

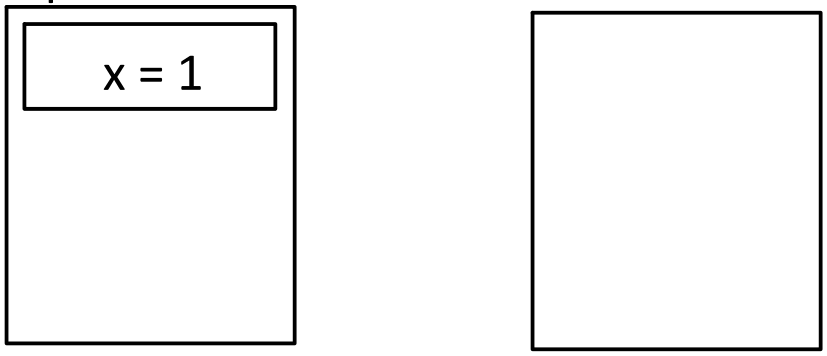
Consider the following quandary

- Put `x = 1` in a cell
- Execute that cell.
- Change the cell to contain `x = 2` but don't execute it.
- In another cell, type `print(x)` and execute it.
- The result will be 1!

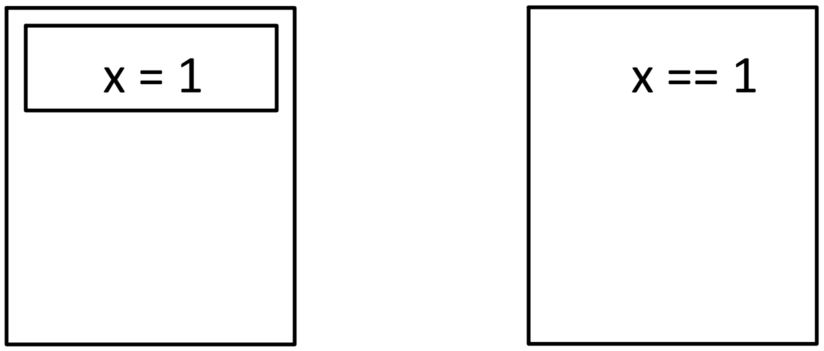
Demonstration



We put x = 1 into a cell

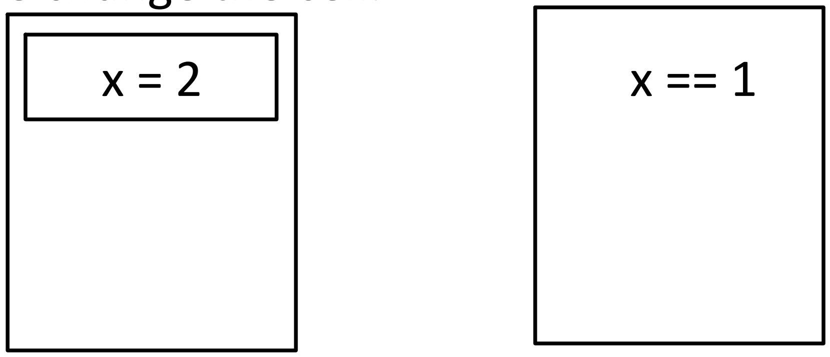


We execute the cell.

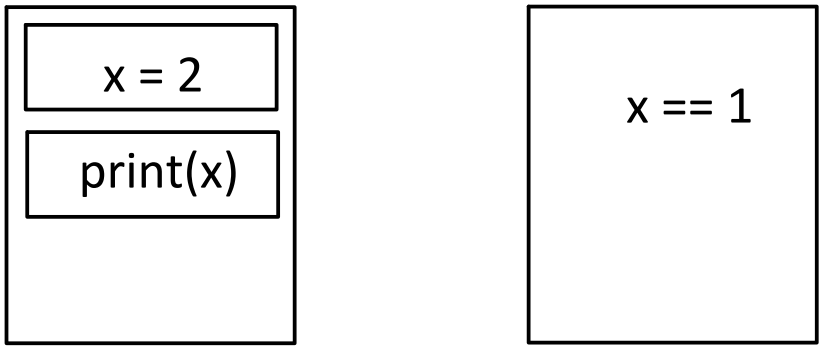


We change the cell.

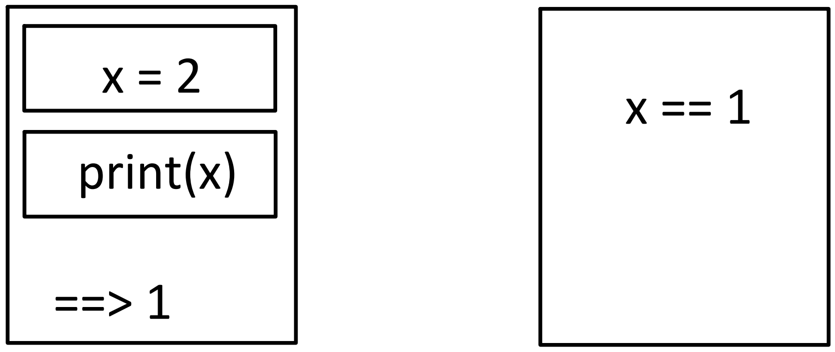
We change the cell.



We add another cell.



We execute the new cell.



So, it seems that we have shown that 2 == 1.
But we have not.

The problem is that **the state of iPython**, on right, **is different than the state of the notebook**, on the left.

"Doctor, it hurts if I do this!"

Doctor: "Then don't do that!"

This is one of many cases in which mysterious behavior can be eliminated by saving and reloading the notebook and executing all cells.

You can save yourself a lot of time by detecting this situation and rectifying it before it confuses you!