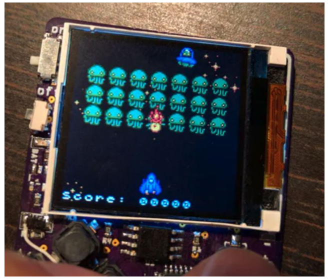
How to Program a Drone to Fly with Python.

Did you know that instead of using a remote controller or mobile app, that you can write a computer program to fly a drone?

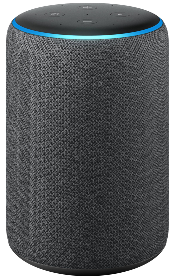
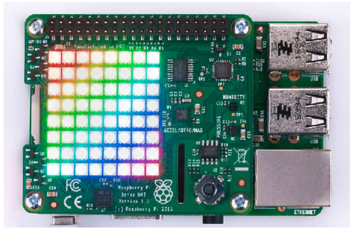
Many electronic devices are controlled by computer programs (also called software).

These devices use an Application Programming Interface or API to send commands to devices. An API translates computer programs into actions on a digital device. An API allows you to write computer programs for lots of different devices using the same programming language.

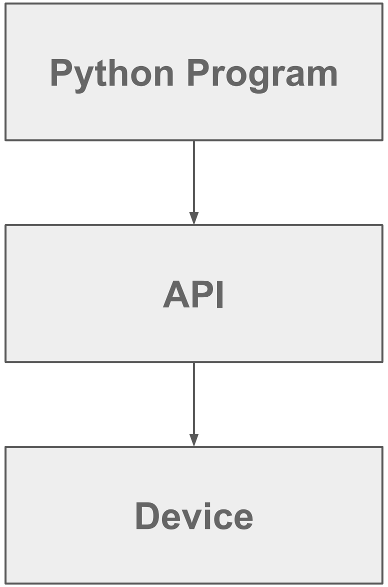
If you know the Python Programming language, you can write programs for hundreds of different devices like those on the next page which all have Python APIs.


Score: 00000

iRobot

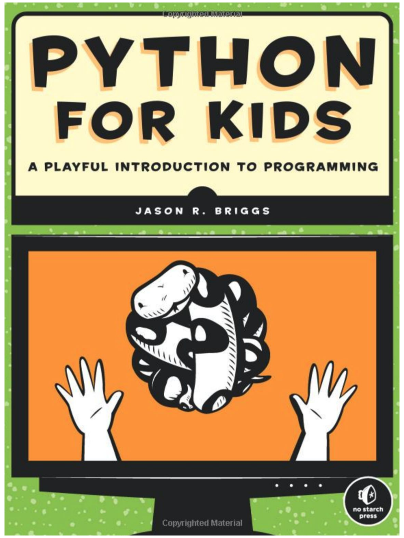
For example, if you wanted to write a program for a drone to fly a precise route around your neighborhood to take video, you could use Python to do that. Once you write the program, all you have to do is run it. The drone could do hundreds of different commands from your program each time in exactly the same way.

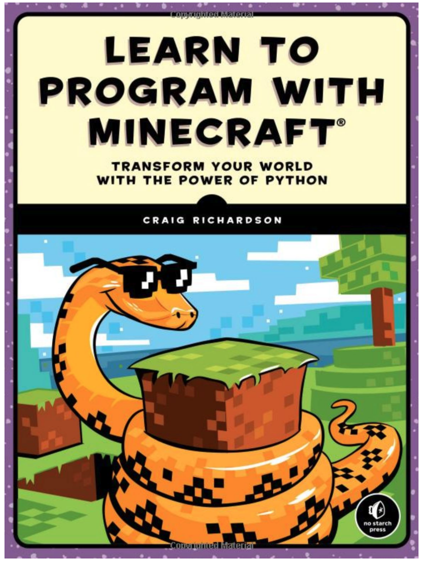


“API, tell the drone to fly forward”

“OK, I’ll send your command to the drone”

“OK, API, I will fly forward”




LEARN TO PROGRAM WITH MINECRAFT®
TRANSFORM YOUR WORLD WITH THE POWER OF PYTHON
CRAIG RICHARDSON
no starch press

The Python programming language is one of the easiest, but also most powerful languages to learn.

So what does a real Python program look like?

On the next page, can you guess what these real Python commands are telling the drone to do?

```
drone.takeoff()

drone.forward(50)

drone.cw(90)

drone.flip(“f”)

drone.land()
```

This is the name of your drone in the program which is known as an “object”. We named the drone “drone”, but we can use any name.

After the period, we have our command which is known as a “method” or “function”. This method tells the drone to fly forward.

```
drone.forward(50)
```

The period separates our object from the command we want to send to the drone.

Inside the parentheses, we are telling the drone how far forward to go. In this case, the “forward” method needs to know how far to go forward in centimeters. This is called a “argument” and can be any valid number for this method.

This is the name of your drone in the program which is known as an “object”. We named the drone “drone”, but we can use any name.

This method tells the drone to rotate clockwise.

```
drone.cw(90)
```

The period separates our object from the command we want to send to the drone.

In this case, the method needs to know the number of degrees to rotate. You can use any number to tell the drone how much to rotate.

This is the name of your drone in the program which is known as an “object”. We named the drone “drone”, but we can use any name.

This method tells the drone to flip.

```
drone.flip(“f”)
```

The period separates our object from the command we want to send to the drone.

Inside the parentheses, notice the “f” is inside quotation marks. Valid values here include the following:

```
“l” - left
“r” - right
“f” - forward
“b” - back
```

This is the name of your drone in the program which is known as an “object”. We named the drone “drone”, but we can use any name.

This method tells the drone to takeoff and must be run before any other commands.

```
drone.takeoff()
```

The period separates our object from the command we want to send to the drone.

In this case, because there is only one way to takeoff, there are no arguments required within the parentheses.

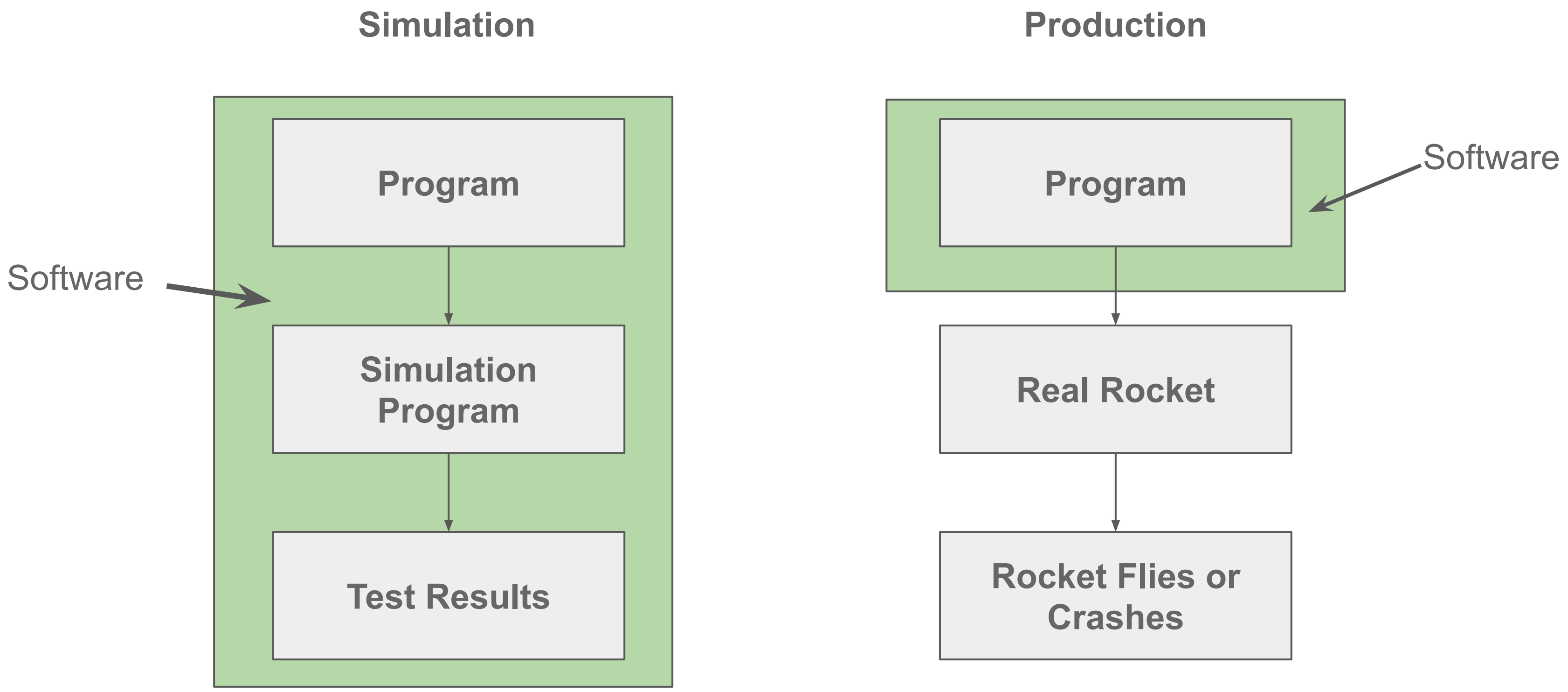
Python Command	English Translation
`drone.takeoff()`	Drone, takeoff.
`drone.land()`	Drone, land.
`drone.up(25)`	Drone, fly up 25 centimeters.
`drone.down(25)`	Drone, fly down 25 centimeters.
`drone.left(25)`	Drone, fly left 25 centimeters.
`drone.right(25)`	Drone, fly right 25 centimeters.
`drone.forward(50)`	Drone, fly forward 50 centimeters.
`drone.back(50)`	Drone, fly backward 50 centimeters.
`drone.cw(90)`	Drone, rotate in place 90 degree clockwise.
`drone.ccw(90)`	Drone, rotate in place 90 degree counterclockwise.
`drone.flip(“f”)`	Drone, do a forward flip. Valid params are “f”, “b”, “l” or “r”.

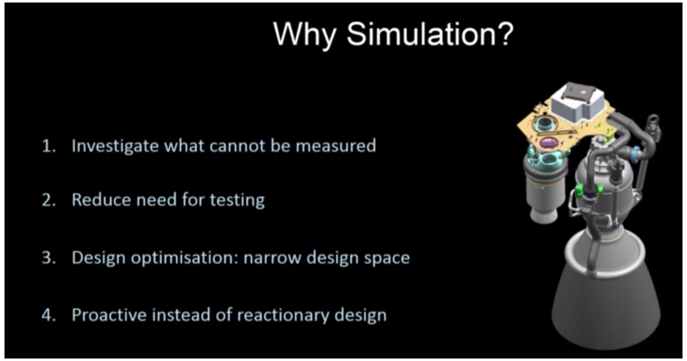
How do you think computer programmers test their programs for computer controlled machines like self-driving cars, rockets, and flying drones?

What happens when the computer program has mistakes in it?

python™

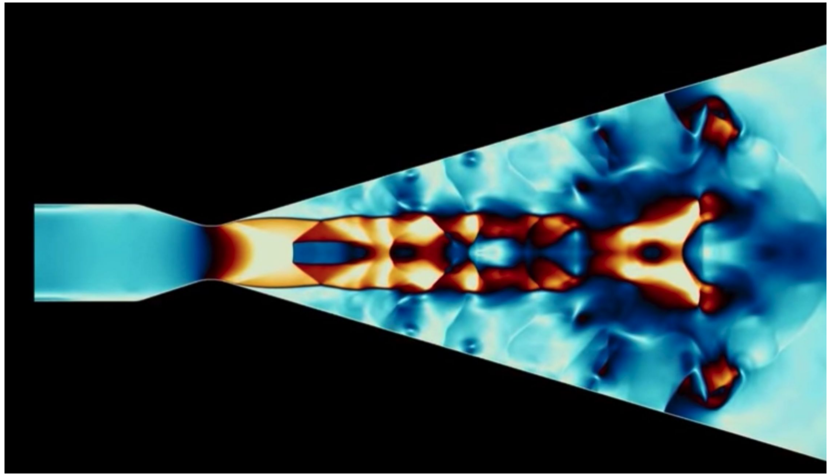
When mistakes in a computer program can cause a machine like a rocket or drone to crash, programmers often use a computer simulation to test the software before using it in the real world.


Simulation
Production
Program
Simulation Program
Test Results
Software
Program
Real Rocket
Rocket Flies or Crashes
Software



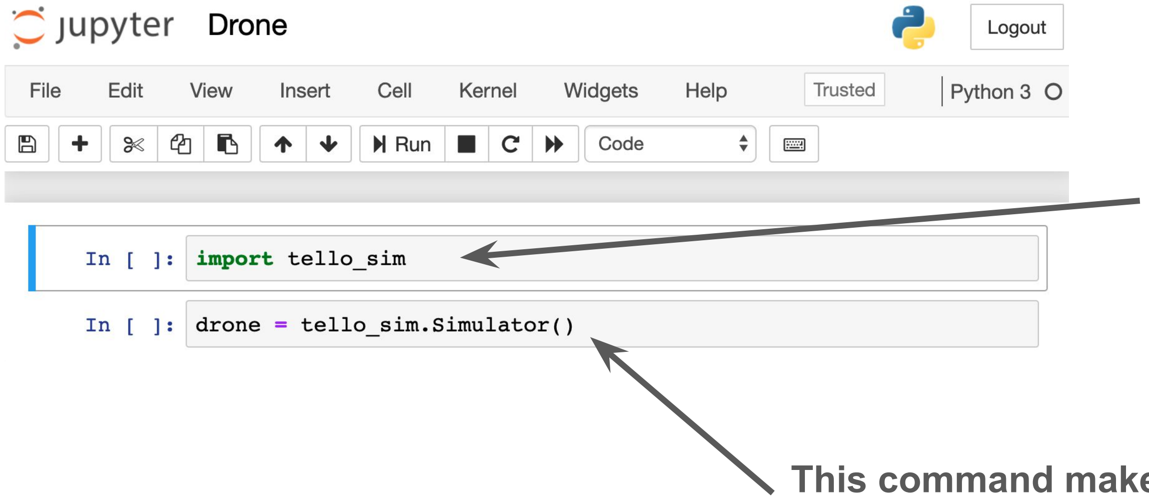
Why Simulation?

1. Investigate what cannot be measured
2. Reduce need for testing
3. Design optimisation: narrow design space
4. Proactive instead of reactionary design



So, we are also going to use simulation to program our drone flights. This will allow us to test our program before we send it to a real drone.

Once we open up Jupyter in our web browser, we can start writing our drone simulation program. The two commands shown below get us started.



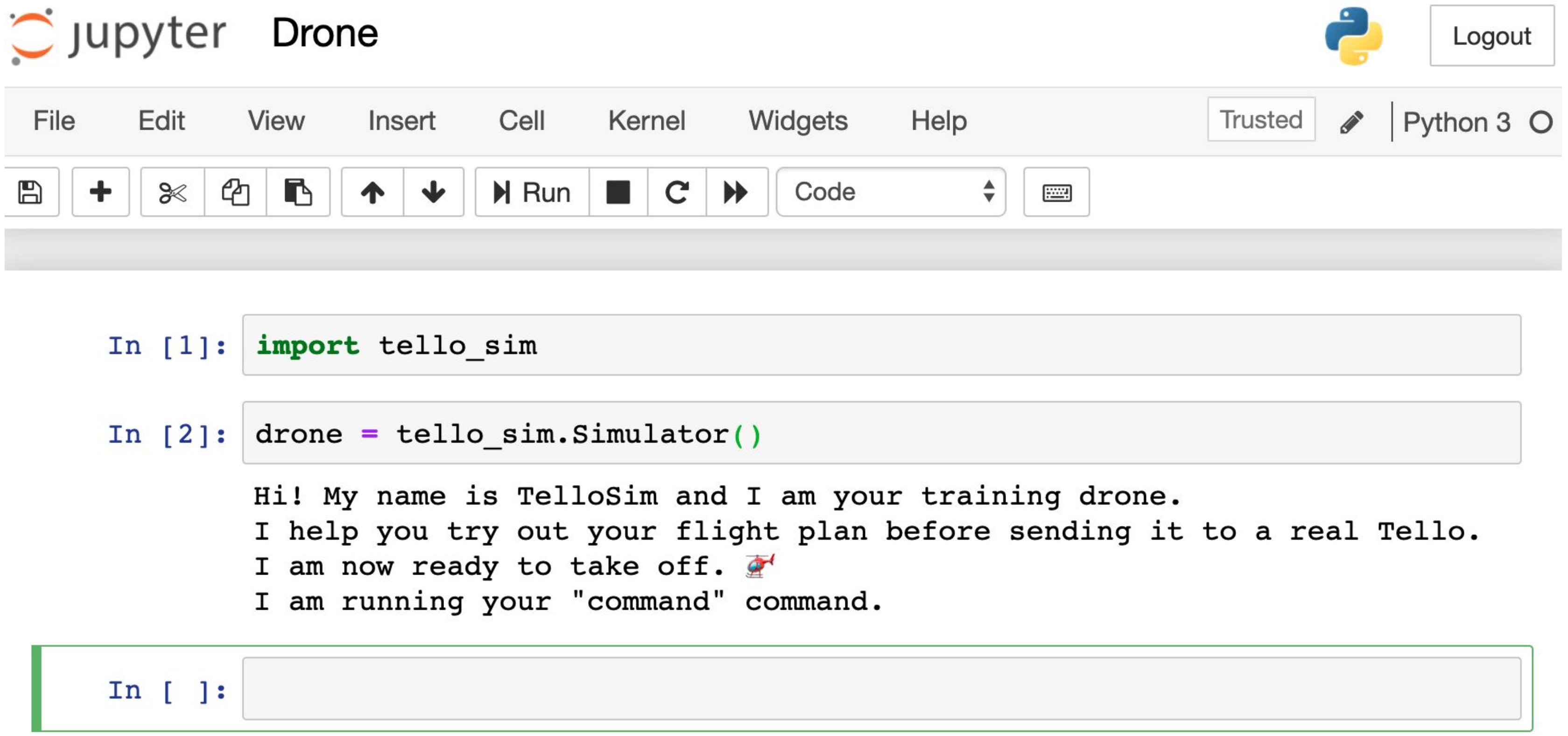
```
In [ ]: import tello_sim
```

```
In [ ]: drone = tello_sim.Simulator()
```

This command loads our simulation software.

This command makes a Python object called “drone”.

If our commands are run correctly, we should see the following output.



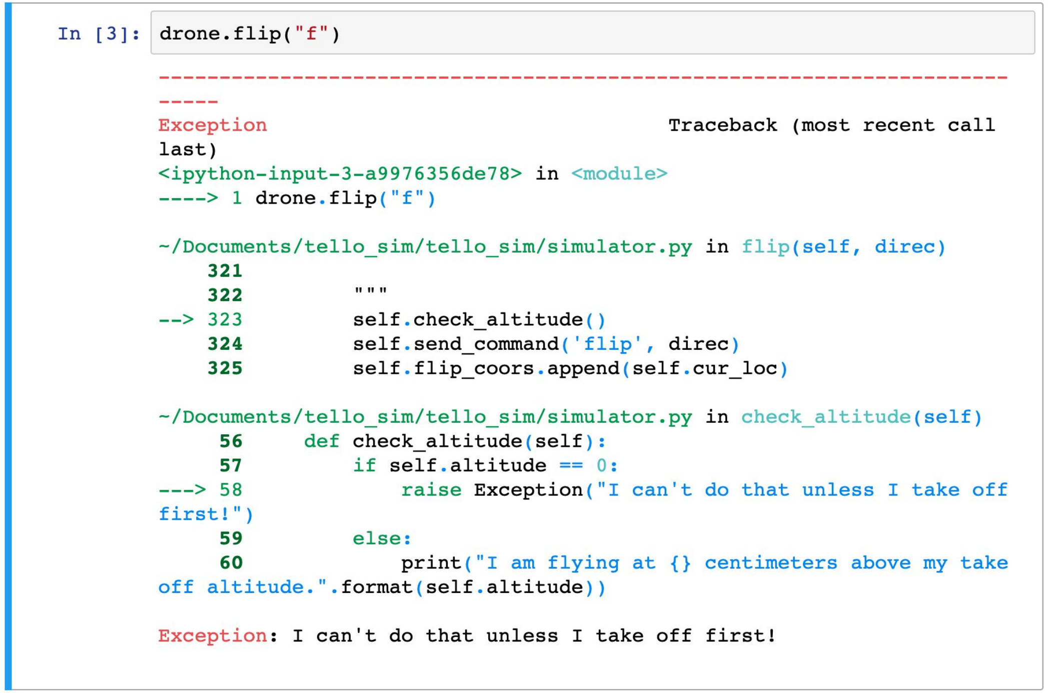
```
In [1]: import tello_sim

In [2]: drone = tello_sim.Simulator()

Hi! My name is TelloSim and I am your training drone.
I help you try out your flight plan before sending it to a real Tello.
I am now ready to take off. 🚁
I am running your "command" command.

In [ ]:
```

What do you think our simulation will do if we make a mistake in our program? For example, what it we try to flip the drone before we take off?



```
In [3]: drone.flip("f")

        ---------------------------------------------------------------------------
        Exception                                 Traceback (most recent call last)
        <ipython-input-3-a9976356de78> in <module>
        ----> 1 drone.flip("f")

        ~/Documents/tello_sim/tello_sim/simulator.py in flip(self, direc)
            321
            322         """
        --> 323         self.check_altitude()
            324         self.send_command('flip', direc)
            325         self.flip_coors.append(self.cur_loc)

        ~/Documents/tello_sim/tello_sim/simulator.py in check_altitude(self)
             56     def check_altitude(self):
             57         if self.altitude == 0:
        ---> 58             raise Exception("I can't do that unless I take off first!")
             59         else:
             60             print("I am flying at {} centimeters above my take off altitude.".format(self.altitude))

        Exception: I can't do that unless I take off first!
```

This is what an error looks like. We can see at the bottom, that the error tells us what we did wrong.

Parentheses around parameter.

```
drone.forward(50)
```

Lower case. No spaces.

Whole number without quotation marks.

✓

Parentheses around parameter.

```
drone.flip(“f”)
```

Lower case. No spaces.

Letter inside quotation marks.

✓

```
drone .forward(50)
```

Extra space.

✗

```
Drone.Forward(50)
```

Not lower case.

❌

Quotation marks around number.

```
drone.forward(“50”)
```

Letter parameter needs quotation marks.

```
drone.flip(f)
```

❌

Now, let’s look at what the simulator does when we have a flight program with no errors. For this flight, we will:

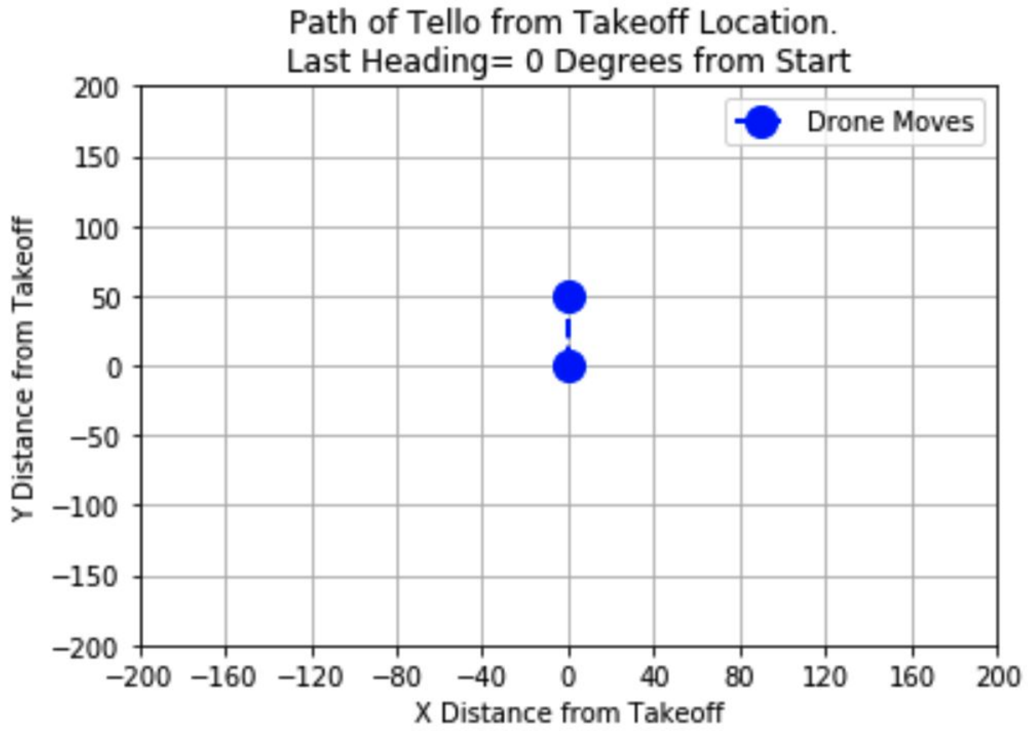
- Take off
- Fly forward 50 centimeters
- Turn right 90 degrees
- Fly forward 100 centimeters
- Land

```python
drone.takeoff()
```

```
Get ready for takeoff!
I am running your "takeoff" command.
My estimated takeoff altitude is 81 centimeters
```

```
drone.forward(50)
```

```
I am flying at 81 centimeters above my takeoff altitude.
My current bearing is 0 degrees.
I am running your "forward 50" command.
```

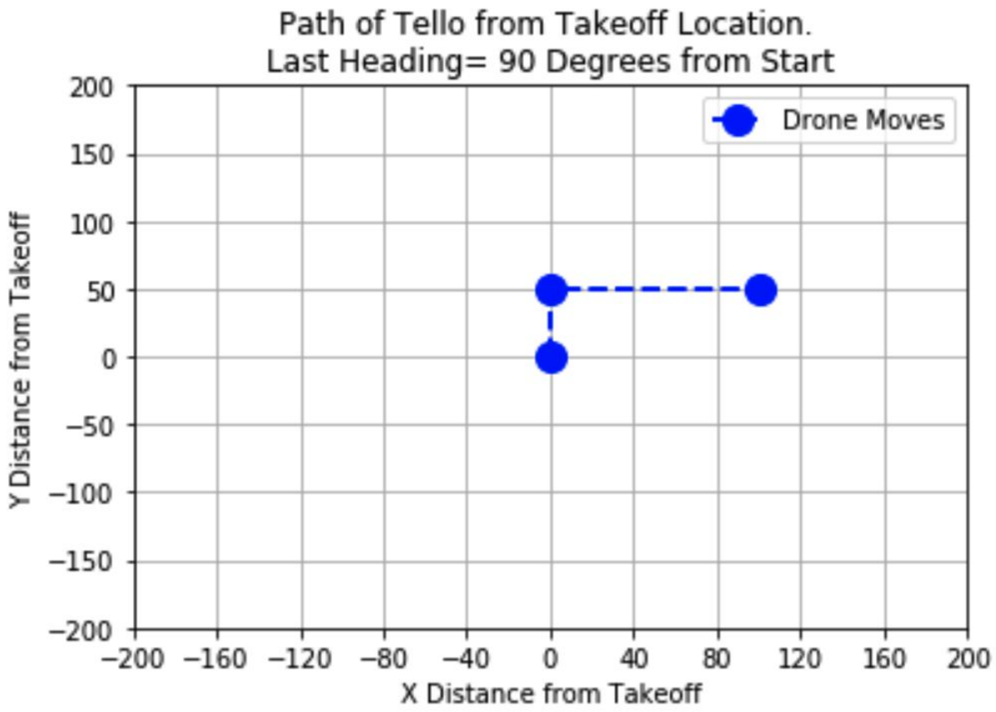


```python
drone.cw(90)
```

```
I am flying at 81 centimeters above my takeoff altitude.
My current bearing is 0 degrees.
I am running your "cw 90" command.
My new bearing is 90 degrees.
```

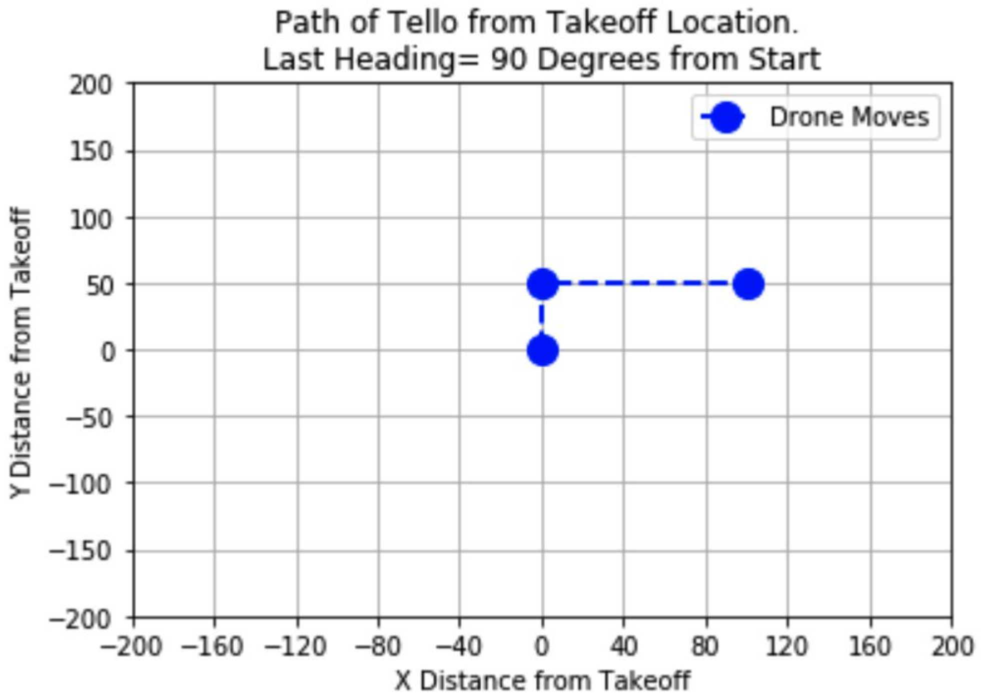
```
drone.forward(100)
```

```
I am flying at 81 centimeters above my takeoff altitude.
My current bearing is 90 degrees.
I am running your "forward 100" command.
```



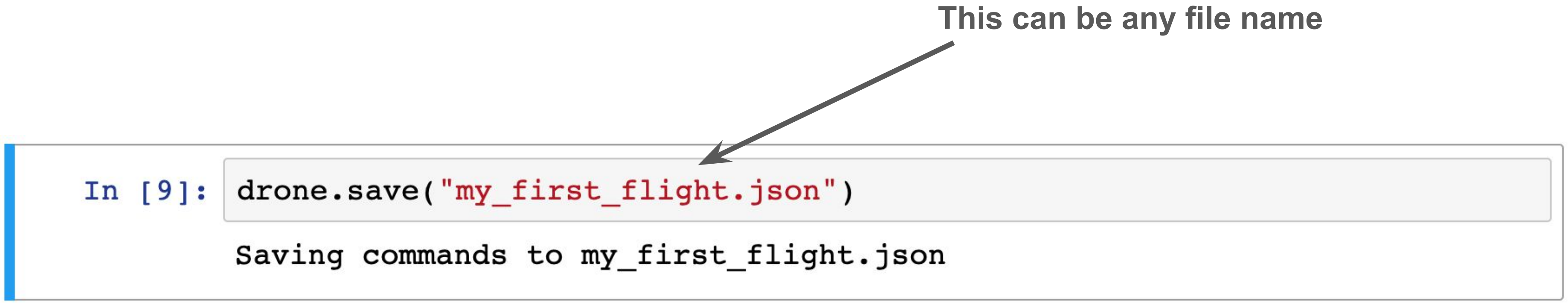
```
drone.land()
```

```
Get ready for landing!
I am flying at 81 centimeters above my takeoff altitude.
I am running your "land" command.
Here are the graphs of your flight! I can't wait to try this for real.
```



There are two different ways to send our program to a real drone to test it in the real world.

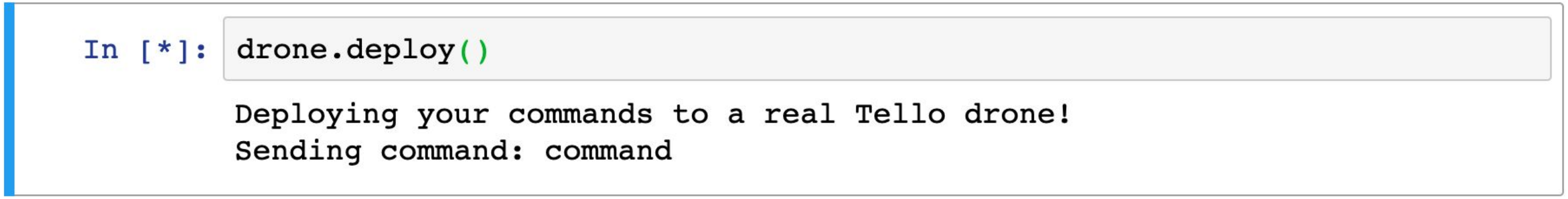
1. Save our program and send it to someone else to run.



```
In [9]: drone.save("my_first_flight.json")

        Saving commands to my_first_flight.json
```

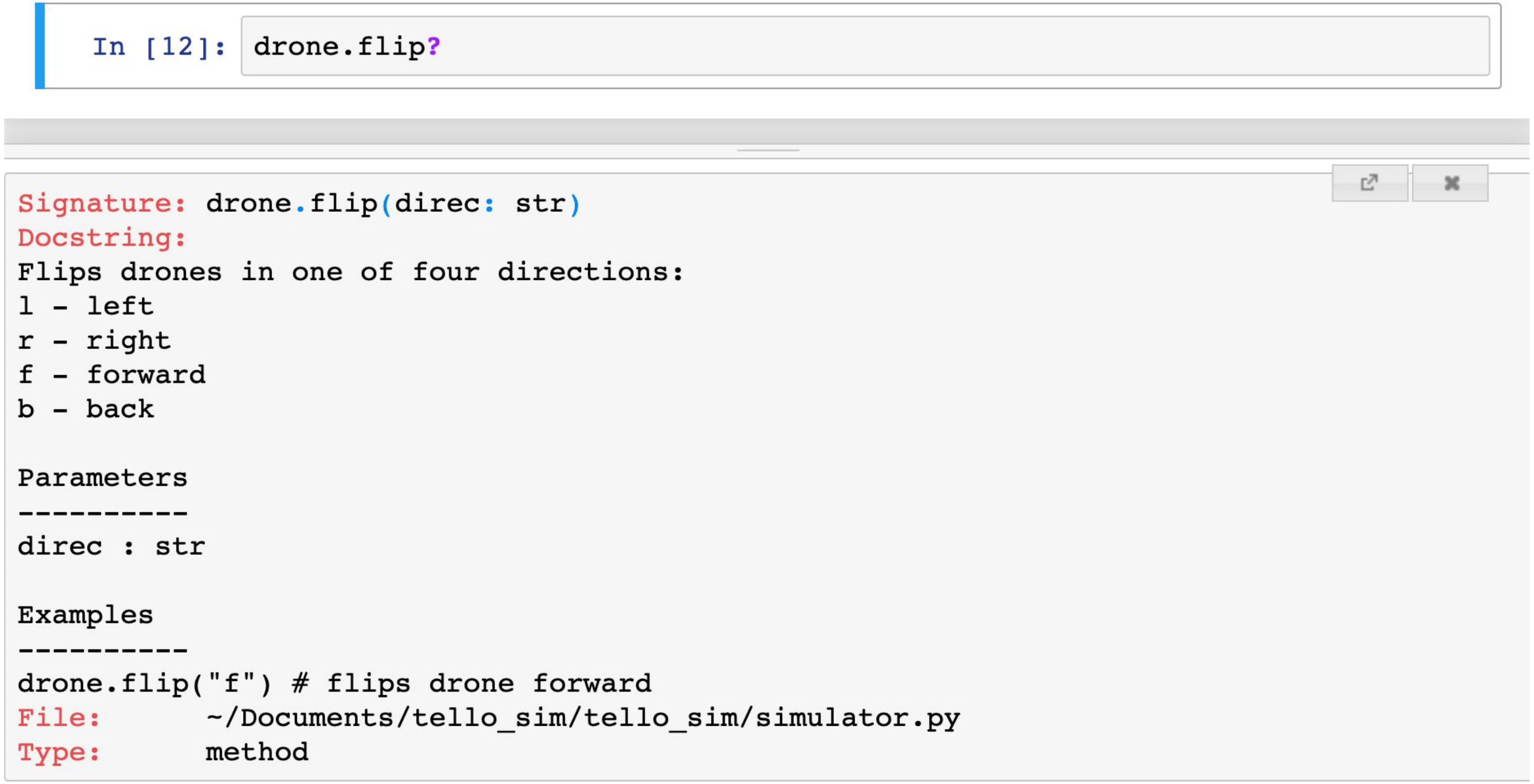
2. Connect to the Tello drone using your WiFi connection and send your program to the drone.



```
drone.deploy()
```

```
Deploying your commands to a real Tello drone!
Sending command: command
```

You can get help in your Jupyter Notebook by running a command with a question mark after it.



```
In [12]: drone.flip?
```

```
Signature: drone.flip(direc: str)
Docstring:
Flips drones in one of four directions:
l - left
r - right
f - forward
b - back

Parameters
----------
direc : str

Examples
----------
drone.flip("f") # flips drone forward
File:      ~/Documents/tello_sim/tello_sim/simulator.py
Type:      method
```

Word or Term	Definition
Computer Program	A computer program is a collection of instructions that performs a specific task when executed by a computer.
Computer Programming Language	A specific set of instructions that can be used to program a computer. This includes rules for what is correct and how to accomplish tasks.
Python	A computer programming language that is easy to learn, but also very powerful.
Computer Software	Computer software is a collection of instructions or programs that performs a specific task when executed by a computer.
API	A application programming interface is a program that translates instructions from a programming language to another system to make controlling a system easier.
Computer Simulation	A computer program that acts like a real device to test programs to see how they work in a safe environment.