

Python

PyCharm IDE
© Jeff Parker
Fall 2019

*What is written without effort is in general
read without pleasure.*

-Samuel Johnson

PyCharm IDE

We run through steps to

Create a project

Run the project

Find and fix errors

Debug the project

2

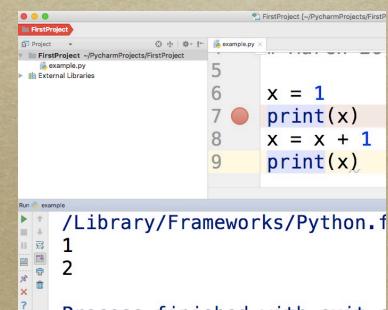
Walk Through example

We will create a simple program using PyCharm, an
Integrated Development Environment (IDE)

Download from <https://www.jetbrains.com/pycharm>

Community version is free

Current version is 2019.2.1



A screenshot of the PyCharm IDE interface. The code editor shows a file named 'example.py' with the following content:

```
5
6     x = 1
7     print(x)
8     x = x + 1
9     print(x)
```

The terminal window at the bottom shows the output of running the script:

```
1
2
```

3

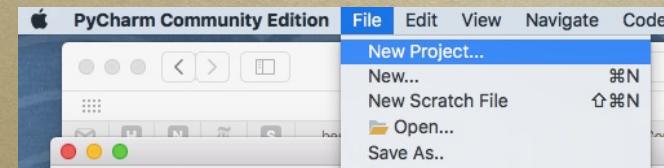
Create a new project

Download and install PyCharm

Run the application.

Under the **File** menu, select **New Project**

Programs live in a Project, which can hold multiple files



4

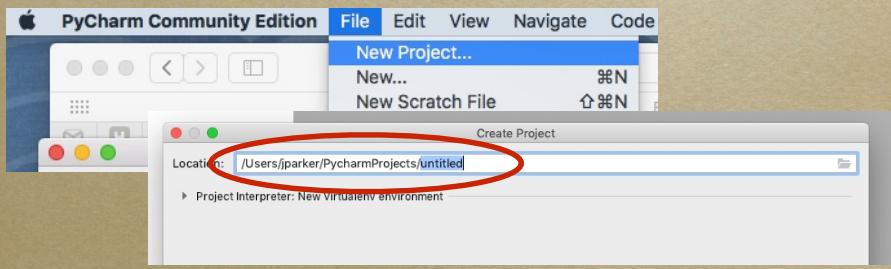
Create a new project

Download and install PyCharm

Run the application.

Under the **File** menu, select **New Project**

Programs live in a Project, which can hold multiple files

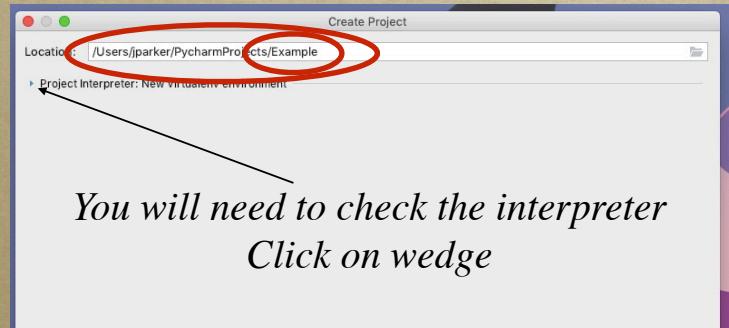


Where do we store it?

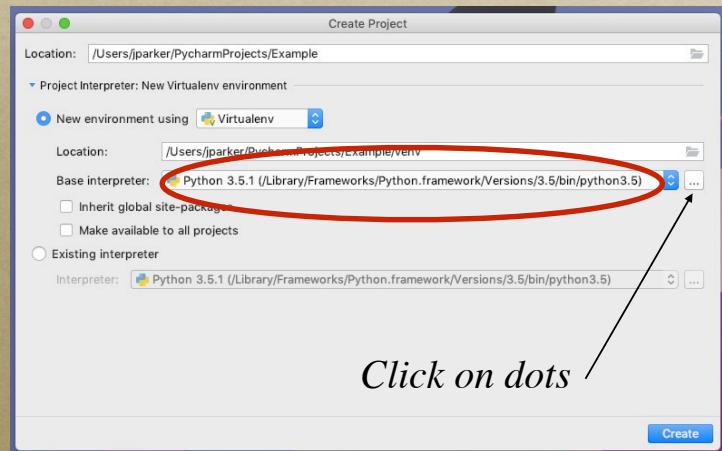
Under the File menu, select New Project

We enter the name for the project: **FirstProject**

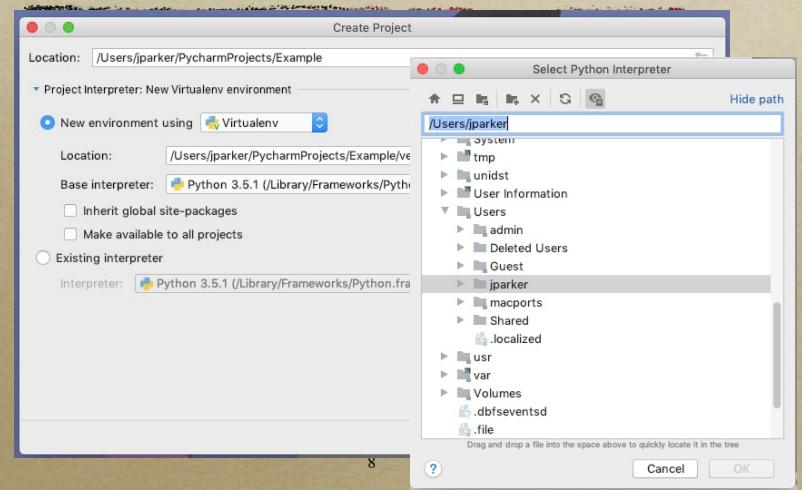
Note **location** of project and **change name**



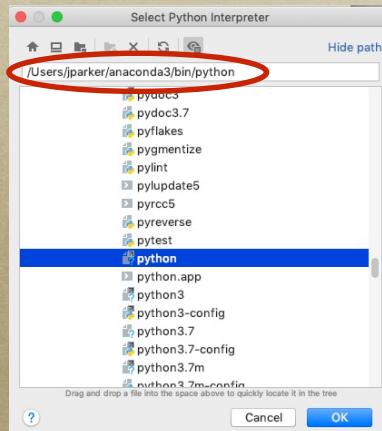
May need to change version



Navigate to python

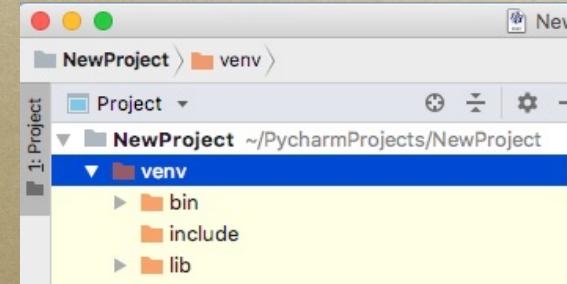


Navigate to python



Now add a Program File

Your project will need a file to hold your program
Open the virtual environment folder

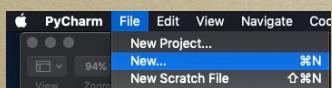


10

Now add a Program File

Your project will need a file to hold your program

Ask for a new...

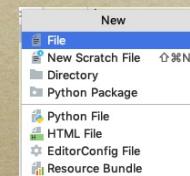
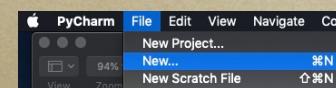


11

Now add a Program File

Your project will need a file to hold your program

Ask for a new... File..

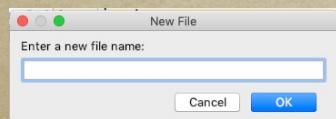


12

Give the file a name

We will call this file **example.py**

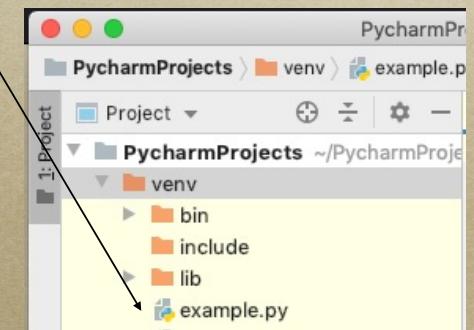
Always use .py as the file extension



13

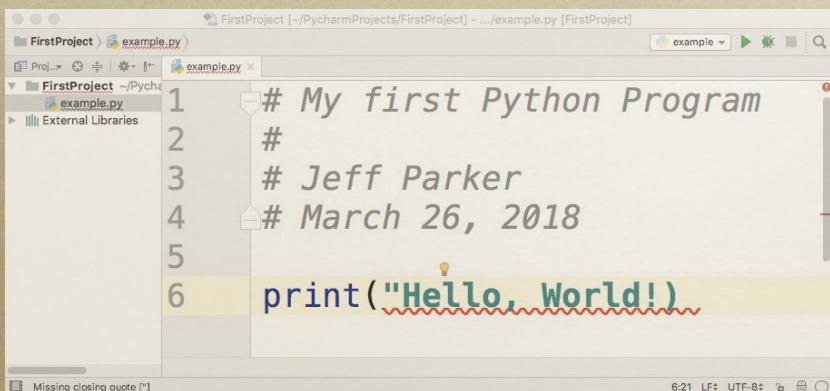
Write your program

See the file name: **example.py**



14

Sample first program



```
# My first Python Program
#
# Jeff Parker
# March 26, 2018
print("Hello, World!")
```

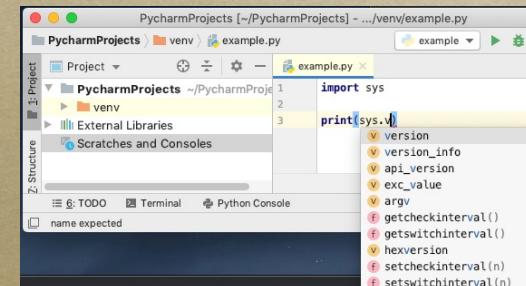
15

More useful program

What can I type after sys. ?

I want **sys.version**

PyCharm knows the API



Which version?

A screenshot of the PyCharm IDE interface. The top bar shows 'PycharmProjects [-/PycharmProjects] - .../venv/example.py'. The main area displays a code editor with the following content:

```
1 import sys
2
3 print(sys.version)
```

The output window below shows the command run and its output:

```
Process finished with exit code 0
```

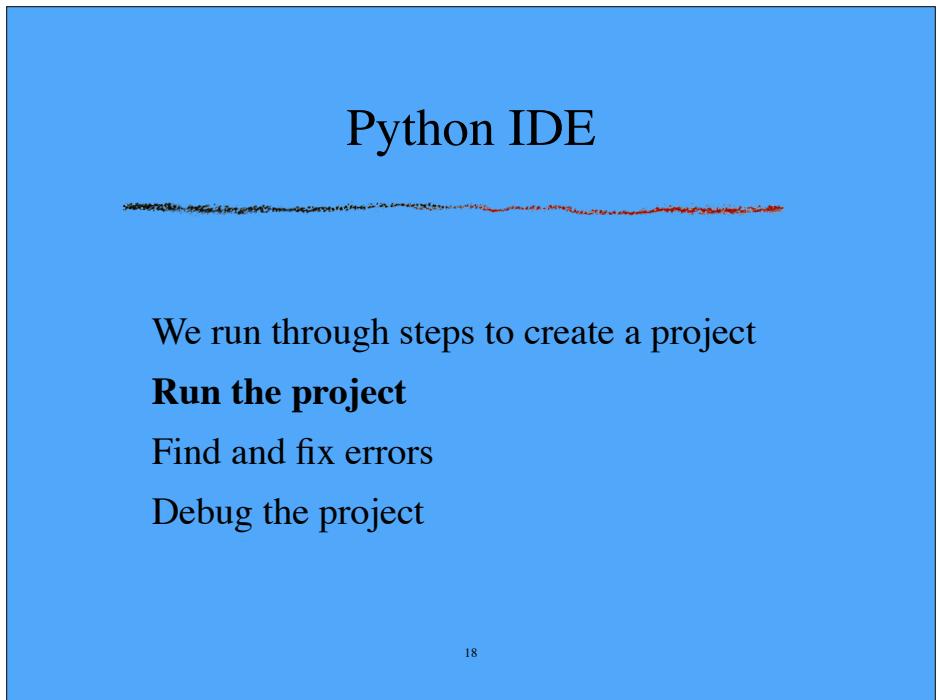
The terminal output shows:

```
/Users/jparker/PycharmProjects/venv/bin/python /Users/jparker/PycharmProjects/venv/ex
3.7.3 (default, Mar 27 2019, 16:54:48)
[GCC 4.2.1 Compatible Apple LLVM 10.0.0 (clang-1000.11.46.5)]
17
```

A red circle highlights the terminal window.

17

Python IDE



18

Run your program

A screenshot of the PyCharm IDE interface. The top bar shows 'Run' as the active tab. The main area displays a code editor with the following content:

```
3 # Jeff Parker
4 # March 26, 2018
5
6 print("Hello, World!")
```

The Run menu is open, showing the following options:

- Run 'example'
- Debug 'example'
- Run...
- Debug...
- Attach to Local Process...
- Edit Configurations...
- Import Test Results
- Stop

A red arrow points from the 'Run' menu to the 'Run' tab in the top bar.

19

Command Line Parameters?

Edit Configuration

A screenshot of the PyCharm IDE interface. The top bar shows 'Run' as the active tab. The main area displays a code editor with the following content:

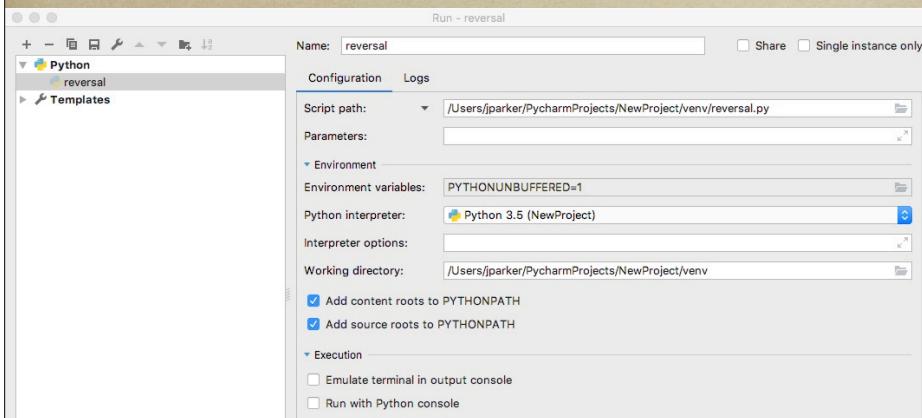
```
1. reversal
```

The Run menu is open, showing the following options:

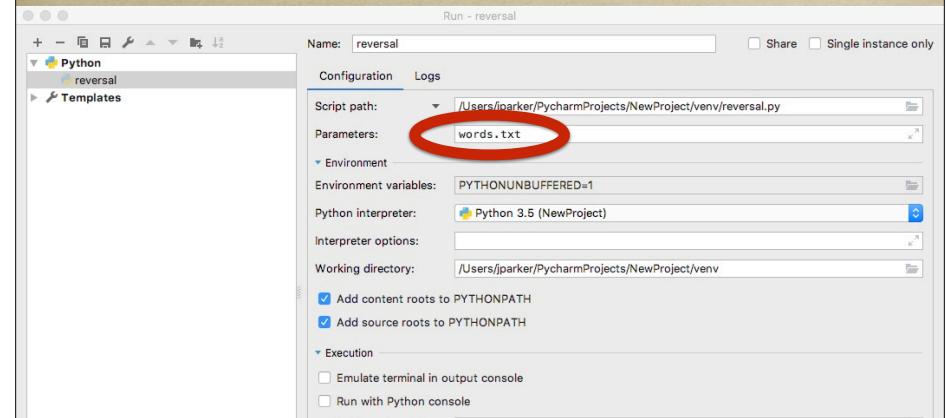
- Run 'reversal'
- Debug 'reversal'
- Run...
- Debug...
- Edit Configurations...
- Import Test Results
- Show Running List
- Show Background Processes...
- Step Over
- Force Step Over
- Step Into
- Force Step Into
- Smart Step Into
- Step Out
- Run to Cursor
- Force Run to Cursor
- Resume Program
- Evaluate Expression...
- Quick Evaluate Expression
- Show Execution Point

A red circle highlights the 'Edit Configurations...' option in the Run menu.

Brings up Config Pane

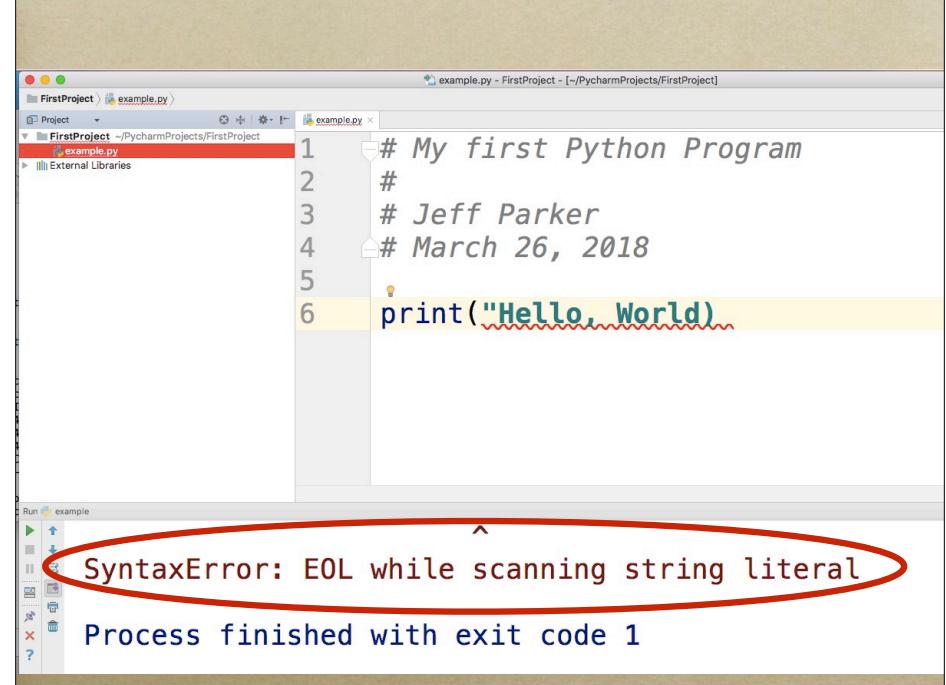


Brings up Config Pane



Python IDE

We run through steps to create a project
Run the project
Find and fix errors
Debug the project



Fix your program

A screenshot of the PyCharm IDE. The project is named 'FirstProject' and contains a single file 'example.py'. The code in 'example.py' is as follows:

```
2 #  
3 # Jeff Parker  
4 # March 26, 2018  
5  
6 print("Hello, World")
```

The line 'print("Hello, World")' is highlighted in yellow and has a red circle around it, indicating a syntax error.

25

Run your program

A screenshot of the PyCharm IDE showing the run output. The 'Run' menu is open, and 'Run example' is selected. The output window shows the following:

```
6 print("Hello, World")  
Hello, World  
Process finished with exit code 0
```

The output 'Hello, World' is circled in red.

Python IDE

- We run through steps to create a project
- Run the project
- Find and fix errors
- Debug the project**

27

Debug a simple program

Clicked to the left of line 7 to set a **breakpoint** before line 7

A screenshot of the PyCharm IDE showing a breakpoint. The code in 'example.py' is:

```
4 # March 26, 2018  
5  
6 x = 1  
7 print(x)  
8 x = x + 1  
9 print(x)
```

A red circle is placed on the line 'print(x)' at position 7, indicating a breakpoint. This line is also highlighted in pink.

Debug new program

A screenshot of the PyCharm IDE interface. The window title is "FirstProject > example.py". The "Run" menu is open, showing options like "Run 'example'", "Debug 'example'", and "Step...". The "Debug 'example'" option is highlighted with a blue selection bar. Below the menu, the code editor shows a script named "example.py" with the following content:

```
# Je
# March 26, 2018
x = 1
print(x)
x = x + 1
print(x)
```

A screenshot of the PyCharm IDE interface. The window title is "FirstProject > example.py". The code editor shows the same script as the previous screenshot. A tooltip is displayed over the line "print(x)" at line 7, containing the text "Tooltip: wait over icon". The "Run" menu is also visible at the top.

The code in the editor is:

```
x = 1 x: 1
print(x)
x = x + 1
print(x)
```

First Step

A screenshot of the PyCharm IDE interface. The window title is "FirstProject > example". The "Debug" tab is selected. The status bar at the bottom left says "Connected to pydev debugger". The code editor shows the same script as before, with the first line "x = 1" visible.

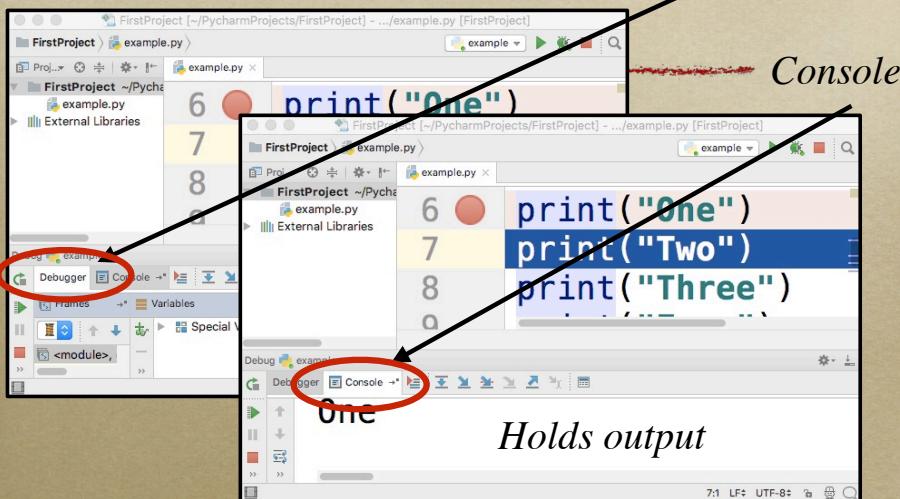
31

Final Result

A screenshot of the PyCharm IDE interface. The window title is "FirstProject > example". The "Debug" tab is selected. The status bar at the bottom left says "Connected to pydev debugger". The code editor shows the same script as before, with the output "1" and "2" printed to the console.

32

Debugger Windows



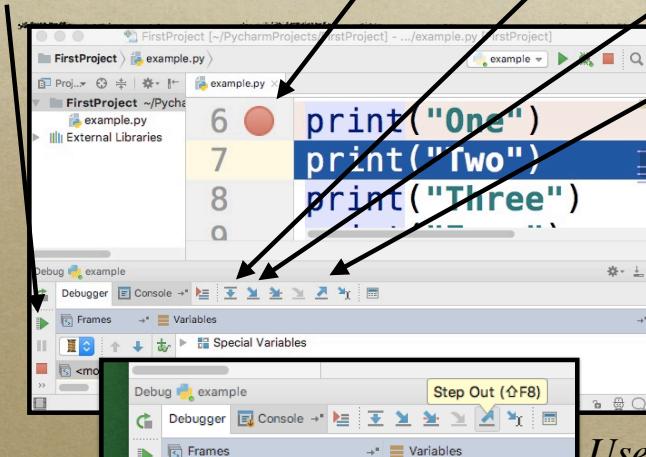
33

Resume: aka
Run to Breakpoint

Controls
Breakpoint

Step Over
Step Into

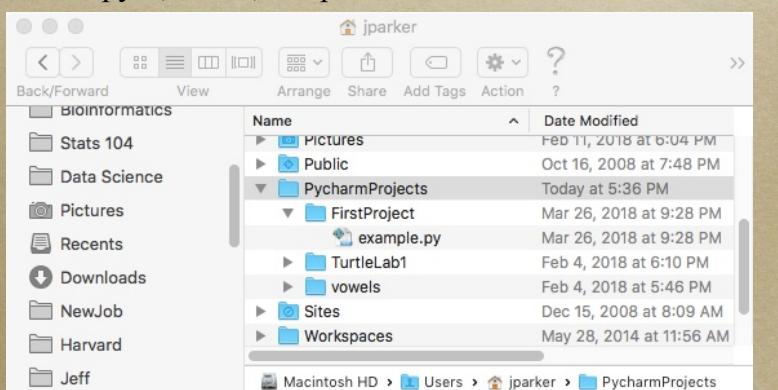
Step Out



34

Find your file

Remember where you left the file
You can copy it, edit it, or upload it



Command Line

Remember where you left the file
You can copy it, edit it, or upload it

A screenshot of a terminal window titled 'FirstProject — -bash — 60x15'. The session shows the following command-line interaction:dcem78:PycharmProjects jparker\$ pwd
/Users/jparker/PycharmProjects
dcem78:PycharmProjects jparker\$ cd FirstProject/
dcem78:FirstProject jparker\$ ls
example.py
dcem78:FirstProject jparker\$ more example.py
My first Python Program
#
Jeff Parker
March 26, 2018
dcem78:PycharmProjects jparker\$ cat example.py
x = 1
print(x)
x = x + 1
print(x)

Warning

TabError: inconsistent use of tabs and spaces in indentation

Don't mix tabs and spaces:

Don't use tabs, or

Use them, and then replace them

Some editors can de-tab

You do it by hand: search for tabs, replace

37

Python IDE

We run through steps to create a project

Run the project

Find and fix errors

Debug the project

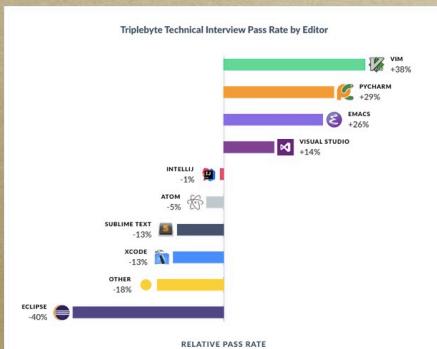
Please take the time to play with this now

You will be handsomely repaid later

38

Datapoint about Editors

“Vim and Emacs users are twice as likely to pass our technical interview as Eclipse users” - Triplebyte



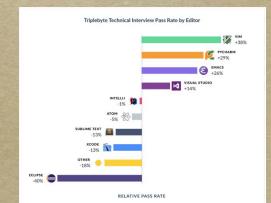
Datapoint about Editors

I don't think that using vi makes you smarter.
vi does some things very well, but requires effort to learn
Those who master it have patience

Consider Walter Mischel's two marshmallow experiment

<https://www.newyorker.com/science/maria-konnikova/struggles-psychologist-studying-self-control>

40



Example: Binary Search

Run through a small example

See the code in the Notebook

Sample code in today's examples

binary_search.py - buggy function

binary_search_test.py - some tests of the function

41

Example: Binary Search

```
def binary_search(list_of_numbers: list, number: int) -> int:
    """Takes a list of numbers and a target value
    Returns the position of the target in the list"""

    # Set boundaries
    low = 0
    high = len(list_of_numbers) - 1

    # Check subarray
    while (low <= high):
        mid = (low + high) // 2
        if number == list_of_numbers[mid]:
            return mid
        elif number < list_of_numbers[mid]:
            high = mid
        else:
            low = mid + 1

    # Not found
    return -1
```

42

Drop comments...

```
def binary_search(list_of_numbers: list, number: int) -> int:

    low = 0
    high = len(list_of_numbers) - 1

    while (low <= high):
        mid = (low + high) // 2
        if number == list_of_numbers[mid]:
            return mid
        elif number < list_of_numbers[mid]:
            high = mid
        else:
            low = mid + 1

    return -1
```

43

```
BinarySearch [~/BinarySearch] - .../venv/binary_search.py
BinarySearch [~/BinarySearch] > venv > binary_search.py
binary_search.py binary_search_test.py
binary_search.py binary_search_test.py pyvenv.cfg
External Libraries Scratches and Consoles

def binary_search(list_of_numbers: list, number: int) -> int:
    # Set boundaries
    low, high = 0, len(list_of_numbers) - 1
    # Check subarray
    while (low <= high):
        mid = (low + high) // 2
        if number == list_of_numbers[mid]:
            return mid
        elif number < list_of_numbers[mid]:
            high = mid
        else:
            low = mid + 1
    # Not found - complain
    raise ValueError('Not in the list')

binary_search()

Run: binary_search_test
/Users/jparkers/BinarySearch/venv/bin/python /Users/jparkers/BinarySearch/venv/binary_search_test.py
.....
Ran 8 tests in 0.000s
OK
Process finished with exit code 0
```

44

Unit Tests

```
BinarySearch [-/BinarySearch] - .../venv/binary_search_test.py
BinarySearch ~/BinarySearch
binary_search.py binary_search_test.py
Project venv .pytest_cache bin include lib binary_search.py binary_search_test.py pyvenv.cfg External Libraries Scratches and Consoles
1:Project
1-Project
binary_search.py
binary_search_test.py
1 import unittest
2
3 from binary_search import binary_search
4
5
6 # Tests adapted from 'problem-specifications//canonical-data.json' @ v1.1.0
7
8 class BinarySearchTest(unittest.TestCase):
9     def test_finds_value_in_array_with_one_element(self):
10         self.assertEqual(binary_search([6], 6), 0)
11
12     def test_finds_value_in_middle_of_array(self):
13         self.assertEqual(binary_search([1, 3, 4, 6, 8, 9, 11], 6), 3)
14
15     def test_finds_value_at_beginning_of_array(self):
16         self.assertEqual(binary_search([1, 3, 4, 6, 8, 9, 11], 1), 0)
17
18
BinarySearchTest
```

45

After running Unit Tests

```
BinarySearch [-/BinarySearch] - .../venv/binary_search_test.py
BinarySearch ~/BinarySearch
binary_search.py binary_search_test.py
Project venv .pytest_cache bin include lib binary_search.py binary_search_test.py pyvenv.cfg External Libraries
Run: binary_search_test x
Run: /Users/jparker/BinarySearch/venv/bin/python /Users/jparker/BinarySearch/venv/binary_search_test.py
Ran 8 tests in 0.000s
OK
Process finished with exit code 0
External Libraries
Run: binary_search_test x
Run: /Users/jparker/BinarySearch/venv/bin/python /Users/jparker/BinarySearch/venv/binary_search_test.py
Ran 8 tests in 0.000s
OK
Process finished with exit code 0
Terminal Python Console Run TODO Event Log
```

Passes all the unit tests

All the tests pass! Great!

That doesn't mean you have a working function
Maybe you haven't thought hard enough about tests

Dream up the tests **before** you write the code
You become protective of code you have written

Let's add a test for things that aren't there

47

Put failing test in main

```
BinarySearch [-/BinarySearch] - .../venv/binary_search.py
BinarySearch ~/BinarySearch
binary_search.py binary_search_test.py
Project venv .pytest_cache bin include lib binary_search.py binary_search_test.py pyvenv.cfg External Libraries Scratches and Consoles
1 def binary_search(list_of_numbers: list, number: int) -> int:
2
3     # Set boundaries
4     low, high = 0, len(list_of_numbers) - 1
5
6     # Check subarray
7     while (low <= high):
8         mid = (low + high) // 2
9         if number == list_of_numbers[mid]:
10             return mid
11         elif number < list_of_numbers[mid]:
12             high = mid
13         else:
14             low = mid + 1
15
16     # Not found - complain
17     raise ValueError('Not in the list')
18
19     # Add test case
20     binary_search([1, 3, 4, 6, 8, 9, 11], 7)
```

We will put this in the test cases

48

Run program, not tests

A screenshot of the PyCharm IDE interface. The project tree on the left shows a 'BinarySearch' project with files like 'binary_search.py', 'binary_search_test.py', and 'pyvenv.cfg'. The code editor on the right contains Python code for a binary search algorithm. Two red arrows point from the text 'Breakpoints' to the red circular markers at line 8 and line 18, which indicate where the program will stop during execution. A context menu is open over the code, with the 'Run' option highlighted. Below the menu, two options are listed: '1. binary_search_test' and '2. binary_search'. The second option, 'binary_search', is highlighted with a blue arrow.

Breakpoints

49

Select debug, not run

A screenshot of the PyCharm IDE interface, similar to the previous one but with a different selection in the Run menu. The 'Run' menu is open at the top, and the 'Debug' option is highlighted with a red arrow. The menu also includes 'Run', 'Edit Configurations...', 'Attach to Process...', 'Edit Configurations...', and 'Import Tests from File'. The code editor and project tree are visible in the background.

Variables are visible

A screenshot of the PyCharm IDE during a debug session. The code editor shows the same binary search code. The bottom of the screen features the debugger interface. In the 'Variables' panel, there are two entries: 'list_of_numbers = [list <class 'list': [1, 3, 4, 6, 8, 9, 11]]' and 'number = (int) 7'. A red arrow points from the text 'Variables are visible' to this panel.

Variables

Click to resume

Resume after breakpoint

A screenshot of the PyCharm IDE during a debug session, showing the code editor with a breakpoint at line 8. The bottom of the screen shows the debugger interface. A red arrow points to the 'Resume' button in the toolbar, which is highlighted with a red border. The text 'Click to resume' is overlaid next to the button.

We can remove/add breakpoints

A screenshot of a code editor showing a Python script named `binary_search.py`. The code implements a binary search algorithm. A red arrow points from the text "New breakpoint stops between lines 8 and 9" to a new breakpoint set on line 9. The code is as follows:

```
1 def binary_search(list_of_numbers; list, number; int) -> int: list_of_numbers[0]
2
3     # Set boundaries
4     low, high = 0, len(list_of_numbers) - 1 low: 4 high: 6
5
6     # Check subarray
7     while (low <= high):
8         mid = (low + high) // 2 mid: 3
9         if number == list_of_numbers[mid]:
10             return mid
11         elif number < list_of_numbers[mid]:
12             high = mid
13         else:
14             low = mid + 1
15
16     # Not found - complain
17     raise ValueError('Not in the list')
18
binary_search() -> while (low <= high)
```

New breakpoint stops between lines 8 and 9

53

Watch high and low

A screenshot of a debugger showing the same code as above. The variables pane on the right shows the current state of the variables: `high` is 6, `list_of_numbers` is [1, 3, 4, 6, 8, 9, 11], `low` is 4, `mid` is 5, and `number` is 7. The code is identical to the previous screenshot.

54

High and low are stuck

A screenshot of a debugger showing the same code. The variables pane now shows `high` is 4, `list_of_numbers` is [1, 3, 4, 6, 8, 9, 11], `low` is 4, `mid` is 4, and `number` is 7. The code is identical to the previous screenshots, but the values of `high`, `low`, and `mid` are all stuck at 4.

55

What is the fix?

The debugger can help you find the problem
You have to figure out what to change
When you find a new bug
Add new tests that spot bug

56