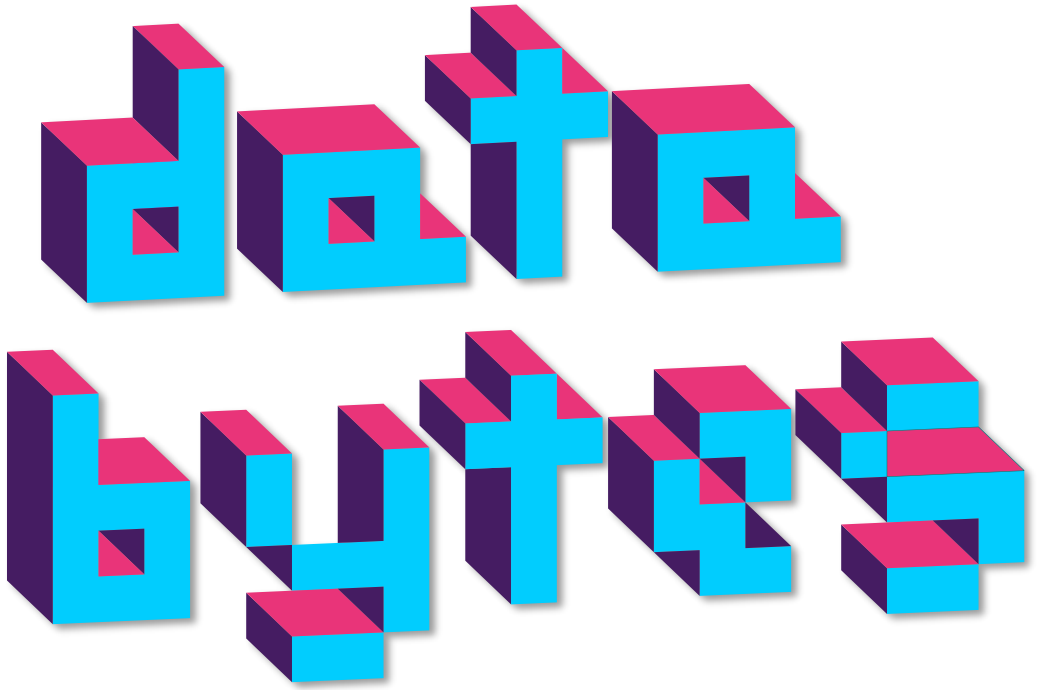
Debugging your Python Code

Peter Lawson

JHU Data Services



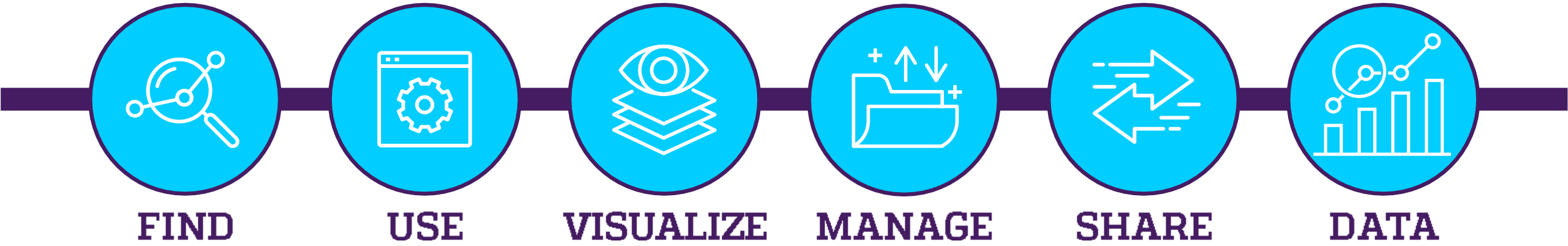
github.com/jhu-data-services

dataservices.library.jhu.edu

dataservices@jhu.edu

JHU Data Services

WE HELP FACULTY, RESEARCHERS AND STUDENTS:



What is a “bug”?

9/9

0800 Arctan started

1000 " stopped - arctan ✓ { 1.2700 · 9.037 847 025
9.037 846 795 correct

13" sec (032) MP - MC ~~1.9826442000~~ ~~2.130476415 (-3)~~ 4.615925059(-2)

(033) PRO 2 2.130476415

correct 2.130676415

Relays 6-2 in 033 failed special speed test

In Relay " 11.00 test.

Relay 2145
Relay 3370

Relays changed

1700 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545

Relay #70 Panel F
(moth) in relay.

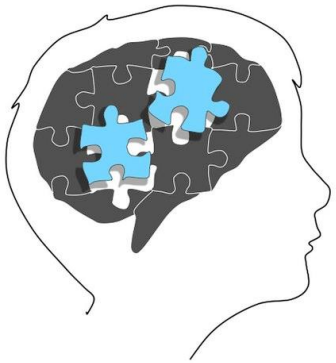
First actual case of bug being found.

~~1655~~ 1630 Arctangent started.

Main Types of Bugs

Syntax Error

Runtime Error



Logic Error

Syntax Error

Syntax Error

Arises from code that doesn’t follow the rules of the language

Syntax Error

Example

```
print "Hello world!"
```

Runtime Error

Runtime Error

Errors that arise when the code is run.

Runtime Error

Example

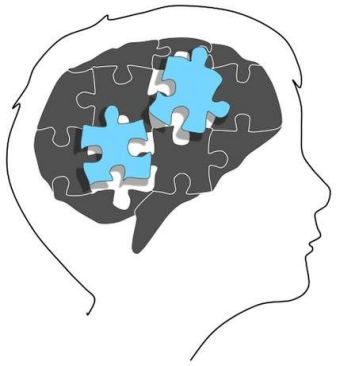
```python
def division(numerator, denominator):
    quotient = numerator/denominator
    return(f'{numerator} divided by {denominator} is {quotient:.2f}')

division(1, 0)
```

$$\text{Quotient} = \frac{Numerator}{Denominator}$$

```
Traceback (most recent call last):
  File "c:\Users\plawson6\Documents\debugging_python.py", line 5, in <module>
    division(1, 0)
  File "c:\Users\plawson6\Documents\debugging_python.py", line 2, in division
    quotient = numerator/denominator
ZeroDivisionError: division by zero
```

Logic Error



Logic Error

The code does not run as the programmer expected due to a mistake in how they wrote the code.

Logic Error

Example

```
def equation(value1, value2):
    return(value1 + value2 ** 2)
```

“I want to sum two values, then square the result”

```
print(equation(1, 2))
```

Expected **9**, got **5**

```
print(equation(3, 3))
```

Expected **36**, got **12**

Logic Error

Example

```
def equation(value1, value2):
    return(value1 + value2 ** 2)
```

Forgot about PEMDAS order of operations!

```
print(equation(1, 2))
```

Expected **9**, got **5**

```
print(equation(3, 3))
```

Expected **36**, got **12**

Logic Error

Example

```
print(equation(1, 2))
```

Expected **9**, got **5**

With Bug

```
def equation(value1, value2):
    return(value1 + value2 ** 2)
```

Step 1: `value2 ** 2`

Step 2: `value1 +` Step 1 result

Result: 5

Fixed Logic Error

```
def equation(value1, value2):
    return((value1 + value2) ** 2)
```

Step 1: `value1 + value2`

Step 2: Step 1 result `** 2`

Result: 9

Debugging:

The process of
identifying and fixing bugs

Reading a Traceback Error

```
def func1(a, b):
    return a / b

def func2(x):
    a = x
    b = x - 1
    return func1(a, b)

func2(1)
```

Reading a Traceback Error

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Input In [9], in <cell line: 1>()
----> 1 func2(1)

Input In [8], in func2(x)
      5 a = x
      6 b = x - 1
----> 7 return func1(a, b)

Input In [8], in func1(a, b)
      1 def func1(a, b):
----> 2     return a / b

ZeroDivisionError: division by zero
```

Reading a Traceback Error

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
Input In [9], in <cell line: 1>()
----> 1 func2(1)

Input In [8], in func2(x)
      5 a = x
      6 b = x - 1
----> 7 return func1(a, b)

Input In [8], in func1(a, b)
      1 def func1(a, b):
----> 2     return a / b

ZeroDivisionError: division by zero
```

Error Message

Reading a Traceback Error

```
ZeroDivisionError
Input In [9], in <cell line: 1>()
----> 1 func2(1)

Input In [8], in func2(x)
      5 a = x
      6 b = x - 1
----> 7 return func1(a, b)

Input In [8], in func1(a, b)
      1 def func1(a, b):
----> 2     return a / b

ZeroDivisionError: division by zero
```

Traceback (most recent call last)

Call Stack

Python Debugger

pdb - Python Debugger

A built in debugger for Python

ipdb - Python Debugger

A debugger for Jupyter Notebook

Fall 2022 Data Bytes Schedule

All sessions are on Mondays from 12 to 1 pm.

Finding Maps and Map Data
Sept 12th

Choosing a Python IDE
Sept 19th

Creating Infographics in Business Analyst
Sept 26th

Debugging your Python Code
Oct 3rd

Speeding up your Python Code
Oct 10th

Introduction to Leaflet
Oct 17th

Advanced StoryMaps Tips and Tricks
Oct 24th

Introduction to APIs in R
Oct 31st 

GIS and Maps Programming

More info at:
bit.ly/data-bytes

Thank you for attending!

Please complete our survey at:
bit.ly/data-bytes-survey

Questions?