Debugging

Why `print()` isn't good enough

Matthew Dupree

Definitions

Debugging

Verb:

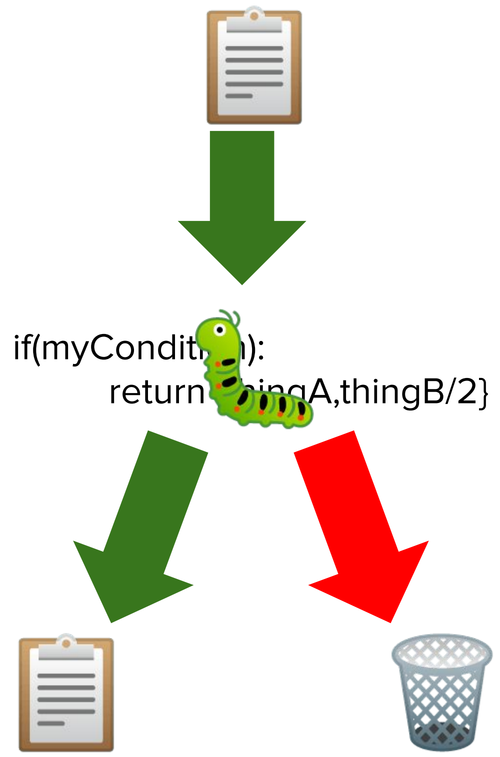
the process of **identifying** and removing [**bugs**] from computer hardware or software

~Oxford English Dictionary

Bug (Software)

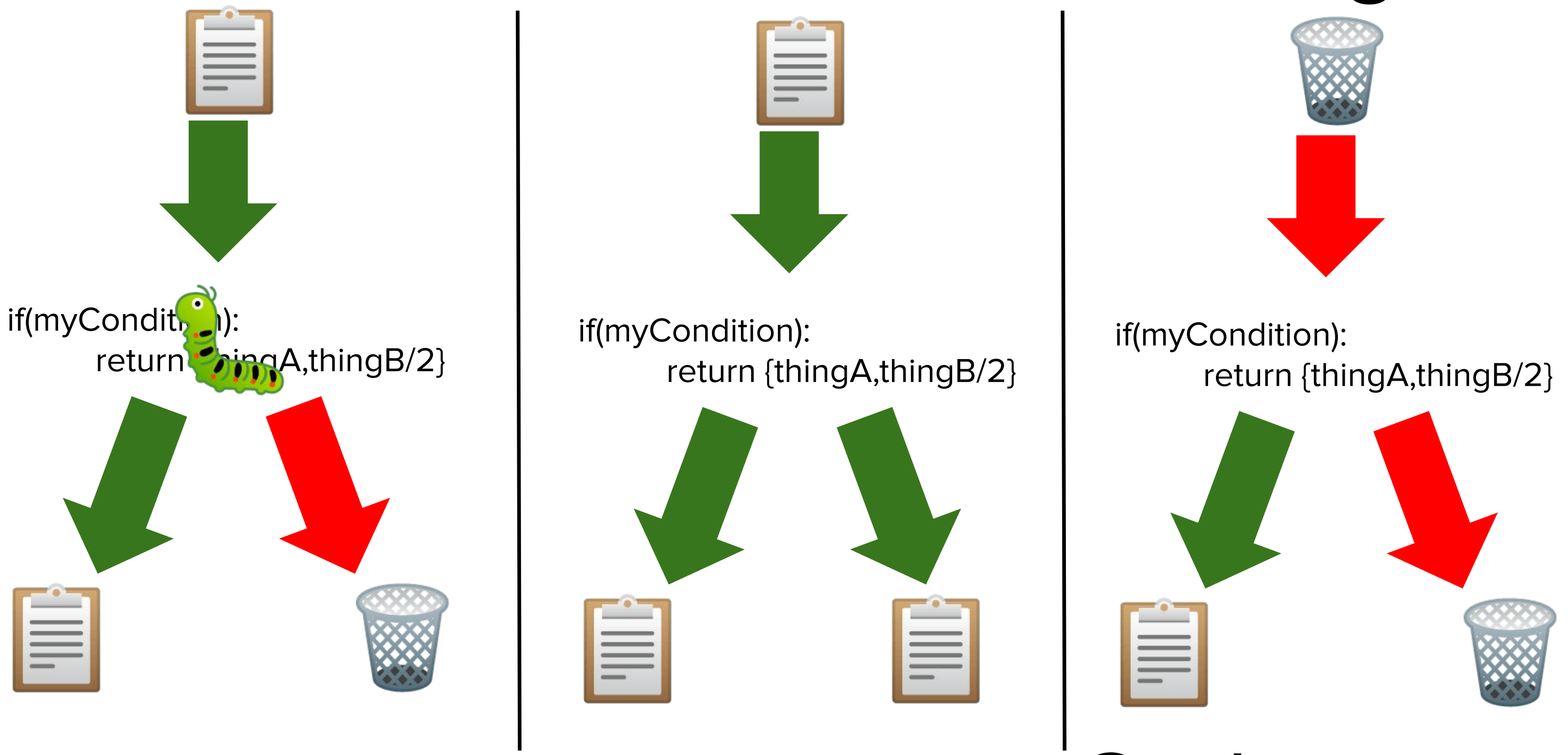
an error, flaw, failure or fault in a computer program or system that causes it to **produce an incorrect or unexpected result**, or to
behave in unintended ways.

~Wikipedia page: Software Bug



```
if(myCondit[illegible]n):
          return[illegible]ingA,thingB/2}
```

Garbage in,

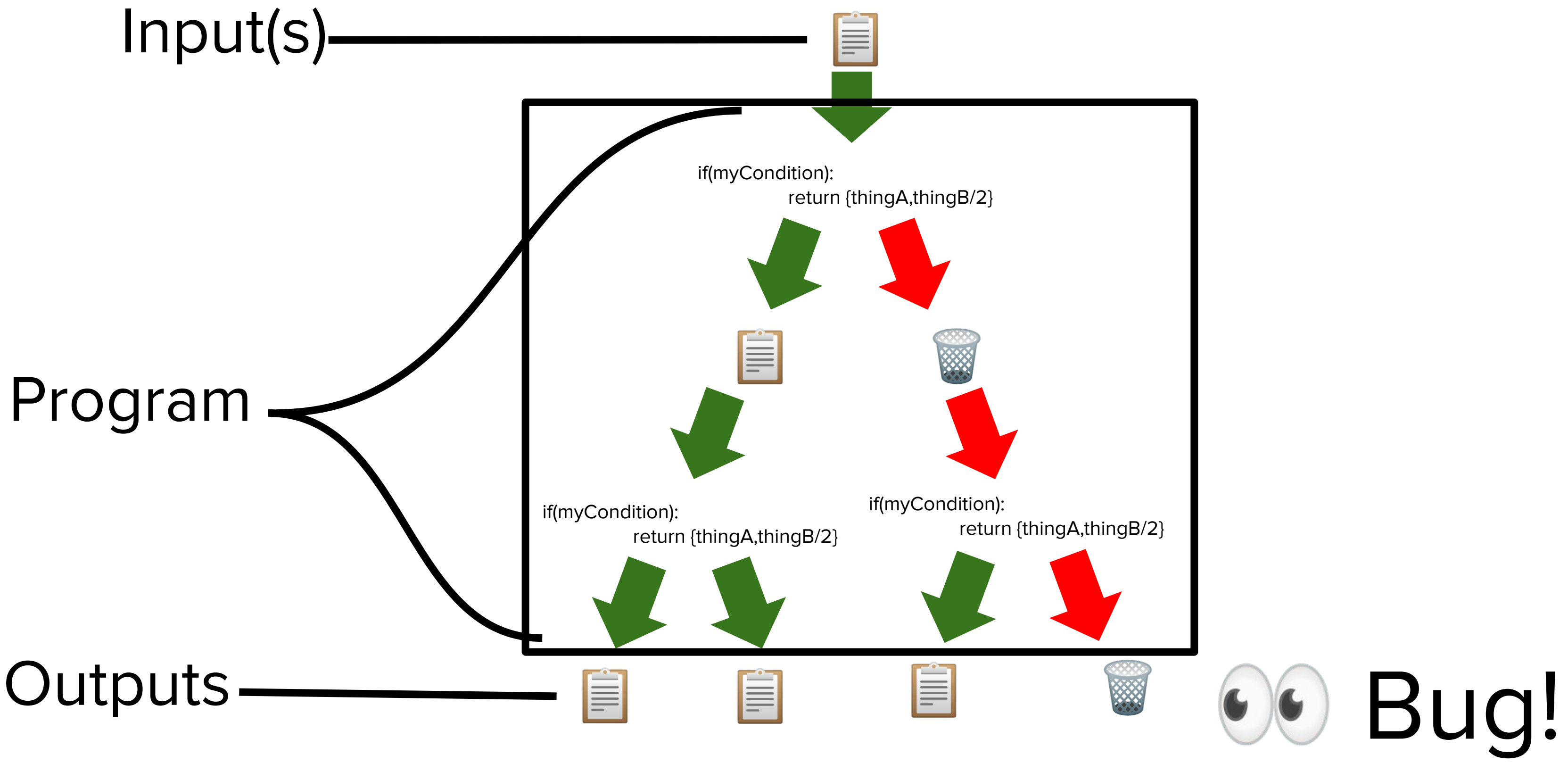


```
if(myCondit   ):
        return    ingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

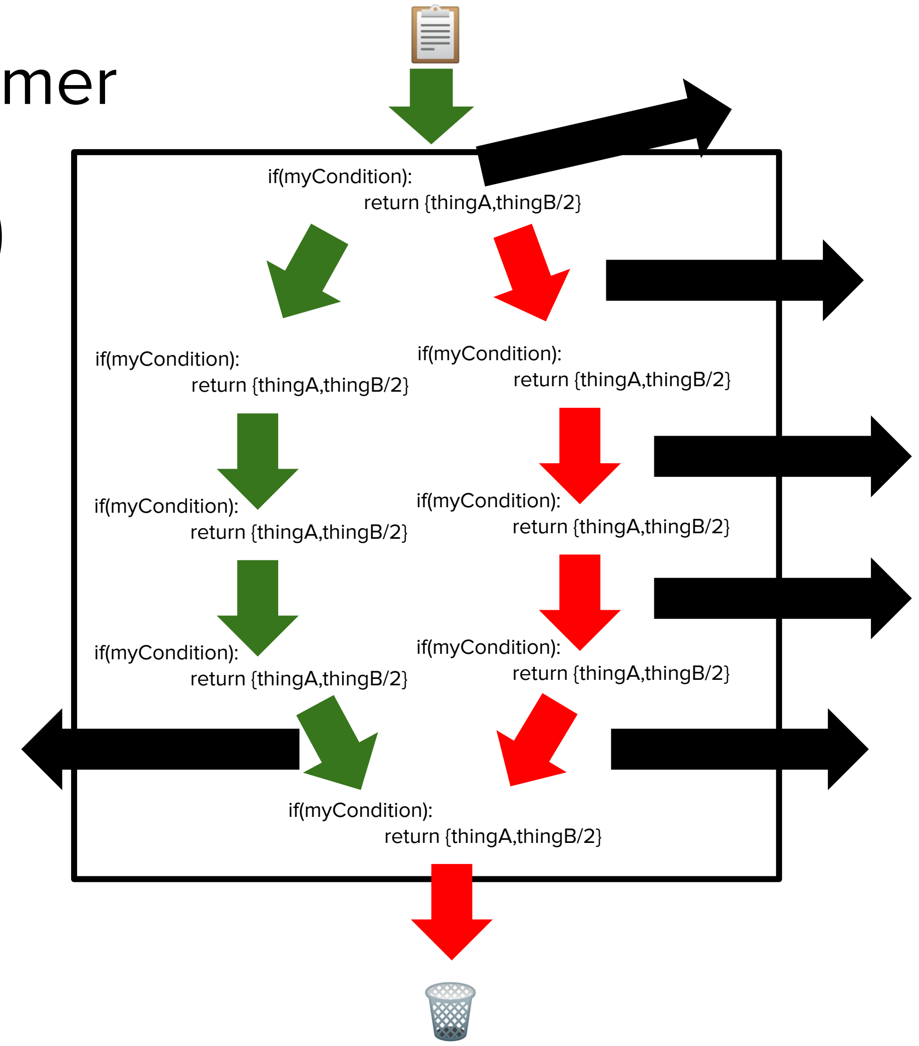
```
if(myCondition):
        return {thingA,thingB/2}
```

Garbage out

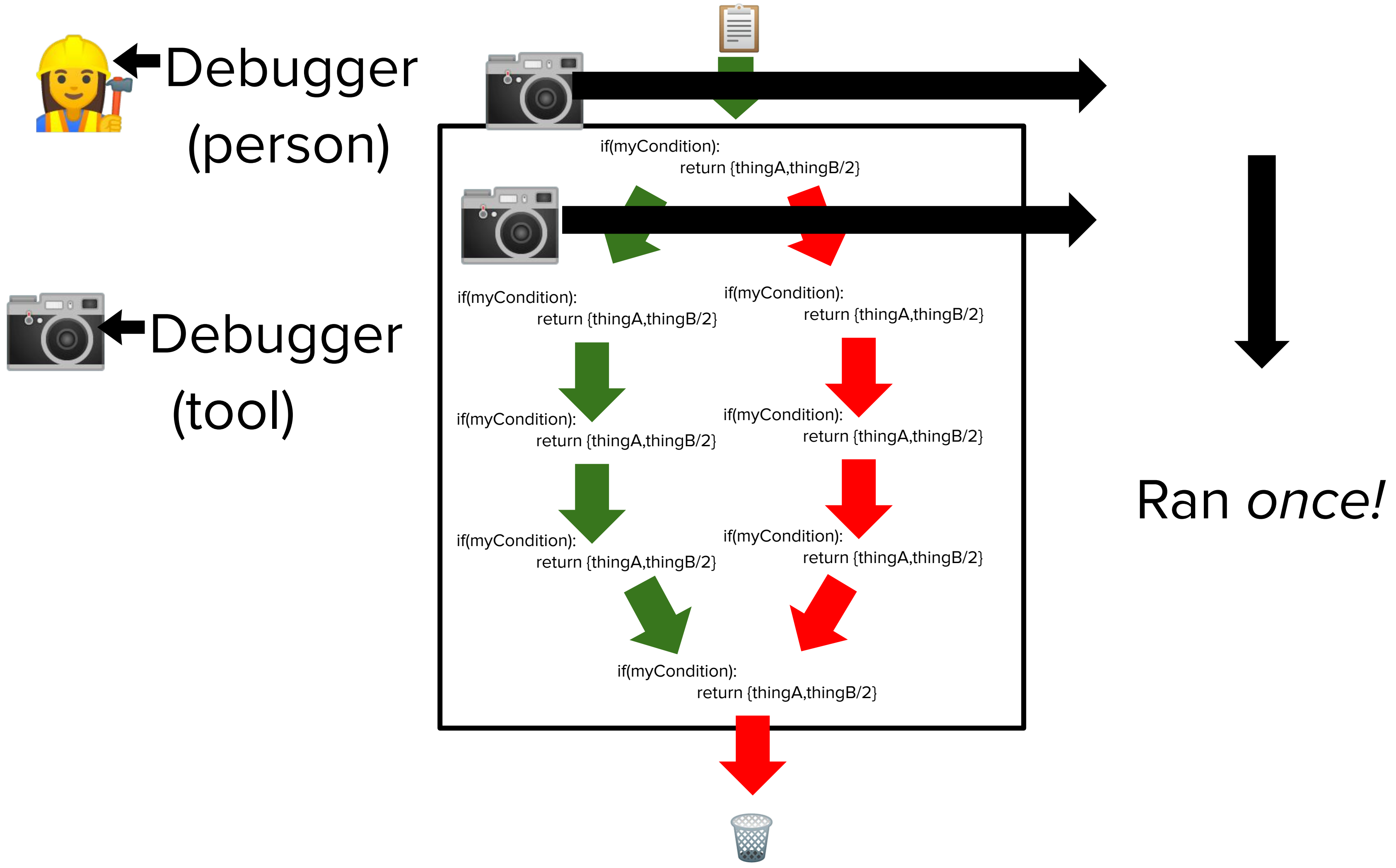

Input(s)
if(myCondition):
return {thingA,thingB/2}
Program
if(myCondition):
return {thingA,thingB/2}
if(myCondition):
return {thingA,thingB/2}
Outputs
Bug!

←Programmer ♀

←print()



Had to restart the program 5 times.



```
if(myCondition):
        return {thingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

```
if(myCondition):
        return {thingA,thingB/2}
```

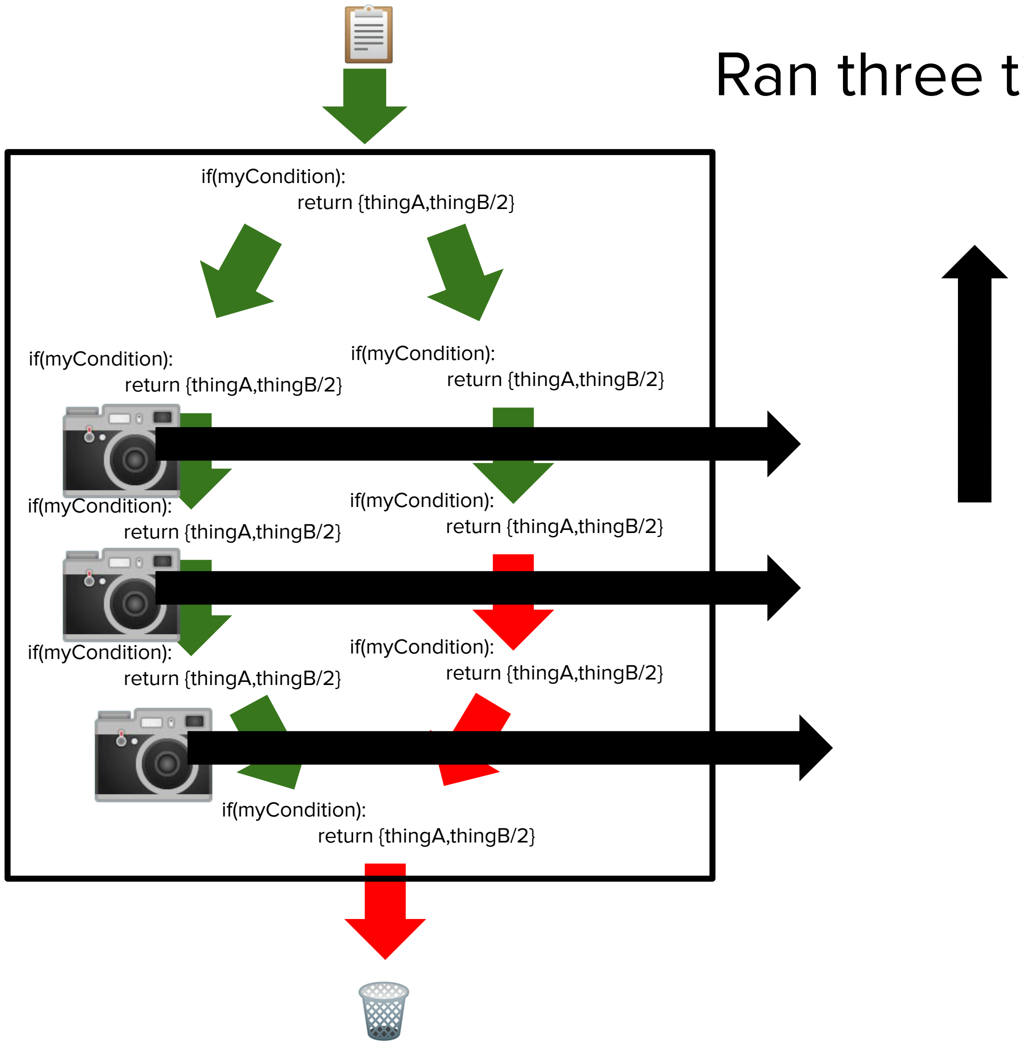
```
if(myCondition):
        return {thingA,thingB/2}
```

♀ ← Debugger (person)

📷 ← Debugger (tool)

🛑 ← Breakpoint

Ran three times!



```
if(myCondition):
          return {thingA,thingB/2}

if(myCondition):
          return {thingA,thingB/2}
if(myCondition):
          return {thingA,thingB/2}

if(myCondition):
          return {thingA,thingB/2}
if(myCondition):
          return {thingA,thingB/2}

if(myCondition):
          return {thingA,thingB/2}
if(myCondition):
          return {thingA,thingB/2}

if(myCondition):
          return {thingA,thingB/2}
```

Pros/Cons

`Print()`

+ `Print()` is in every program!
- More reproduction of the bug
- Have to select which items to print

Debugger

- Requires another program
+ With the right program, may only need to reproduce once!
+ Provides all program state at selected points in time*

*some debuggers can time travel

In many cases, a debugger takes less effort than `print()`

Why don’t more people use debuggers, then? 😕

- Debuggers are ~~new*~~ tools

- Learning new tools takes effort

*the first debugger I could find was the **Atari 2600 BASIC programming cartridge** from **1980**. Not new in the slightest!

Who’s ready to learn?

Python *Fibonacci* in VSCode & PDB

Learning extended, time permitting:

Python *Bubble Sort* in VSCode and PDB

Important Operations: Basics

- `python3 -m pdb myAwesomePython.py` 📷
    - Python DeBugger
    - Opens the Python DeBugger running `myAwesomePython.py`
- r 🏃‍♀️
    - run
    - Start the program at the beginning and go until an error or breakpoint

Important Operations: Navigation in time

- s - 👢
  - step (into)
  - Advance time by 1 line of code, entering function calls.
- n - ➡️
  - next (line)
  - Step by 1 line of code, fast-forwarding function calls.
  - (Also called “step over”)
- c - ▶️
  - continue
  - Resume the program from the current line of code
  - Stops again at the next breakpoint or error

These don’t work when stopped by errors. They <u>terminate</u> the program!

Important Operations: Breakpoints

- b 147 🛑
  - break 147
  - Sets a breakpoint on line 147 of the active Python file

We didn’t talk about how to clear or ignore breakpoints. Unfortunately, debuggers disagree on that. Look it up in your debugger of choice!

Important
Operations: Showing values

- p myVariableName 🖨️
  - print myVariableName
  - Displays the value of myVariableName at the current point in time
- display myVariableName 📺
  - Prints myVariableName every time execution stops (via breakpoint, step, or next)
  - How do we make that stop?
- undisplay

Fin

Wondering where to go to learn more?

Python:
https://docs.python.org/3/library/pdb.html

C/C++:
https://www.cs.swarthmore.edu/~newhall/unixhelp/howto_gdb.php

MATLAB:
https://www.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html