

Debugging



Debugging

- Under-appreciated skill
- Success model
- Techniques

Under-appreciated Skill

- More time debugging than writing
- Learned skill

Success

- Success makes you happy
 - your brain releases a feel good chemical
- Organize programming for lots of little successes
 - your brain doesn't care all that much if you have successfully written hello world, or a fast fourier transform you still get that little release that makes you happy
 - When you are happy you want to go on and solve the next little problem
- Start small, get something small working, and then add to it

Success Model

- Start Small
- Get it Working
- Add a Small bit
- Keep it Working

Techniques

•Detective work

- look for clues
- evaluate clues
- generate hypothesis
- change code
- repeat

Techniques

- Comment out lines with errors
 - Evaluate clue
- Replace Input Values or Expressions with a constant value
 - Evaluate clue

Techniques

- Print statements are your friends
 - Use them to help you uncover what is really happening in your code
- Print variables
- Print variable type
- Print to say your program got to here

Techniques

- Work backward from the error
 - Many times an error message is caused by something that has happened before it in the program
 - Always remember that python evaluates a program top to bottom.

Techniques

•Hand Tracing Code

- Create a column for each variable
- Create a column for output
- Go through code line by line tracking variable changes and outputs

Hand Tracing Code

```
1  a = 5
2  b = 12
3
4  c = b + a
5
6  print ("c is: " + str(c) )
7
8  b = b + a
9
10 c = b * a
11
12 print ("c is: " + str(c) )
```

a	b	c	output

Techniques

- Editing and Running the same file

- not sure? put something obvious at beginning and test

Techniques

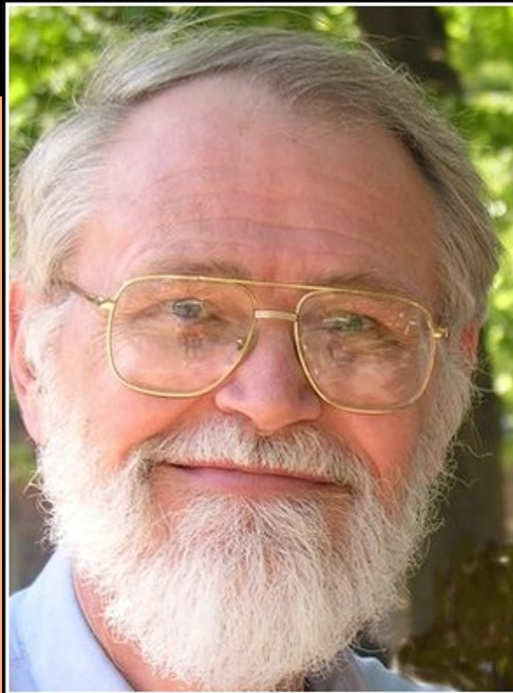
- Go for a walk

Getting Help

- Don't be offended
- Helper will
 - challenge your assumptions
 - validate and check your assumptions
- Goal (in this class)
 - NOT to make program work
 - help you learn how to make program work

Debugging is like
being the detective in a
crime movie where you're
also the murderer.

- Filipe Fortes -

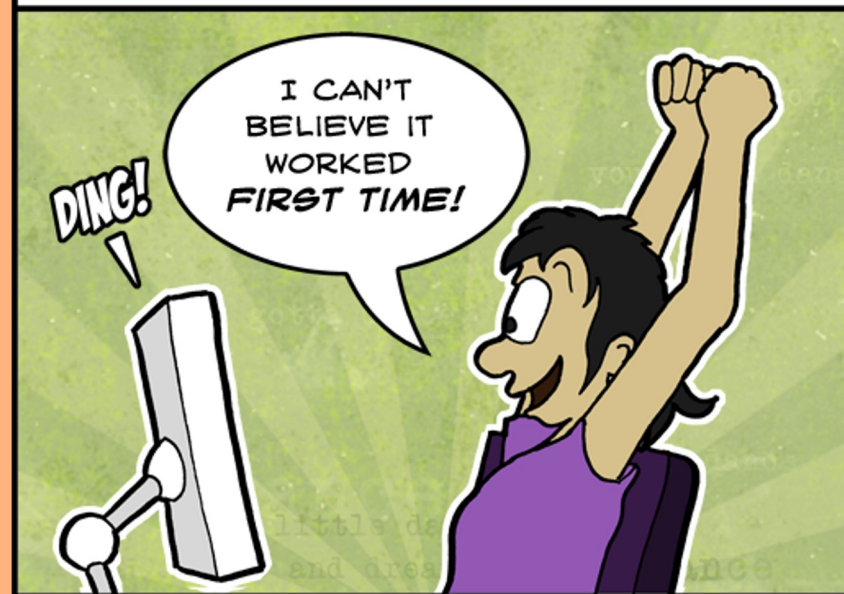


Debugging is twice as hard as writing
the code in the first place.
Therefore, if you write the code as
cleverly as possible, you are, by
definition, not smart enough to
debug it.

— *Brian Kernighan* —

AZ QUOTES

NOVICE PROGRAMMER



EXPERIENCED PROGRAMMER



Review

- Start small; keep it working
- Print statements
- Comment out lines
- Replace values/expressions with known constant
- Hand trace code
- Challenge assumptions
- Go for a Bike Ride