

Programmeertalen: Quick guide on debugging

How to Troubleshoot and Fix Your Code Without Pulling Your Hair Out

Jurre Brandsen

Universiteit van Amsterdam

March 6, 2020

Troubleshooting; Why is it important?

"My code doesn't work and I'm hopelessly stuck. Please help me."

How to do it?

- You could use a debugger to troubleshoot your code.
- You could use unit-tests to see what code works (Recall the Inleiding Programmeren assignment where you already have done this once ;-)) and whether code still works.
- But for now, you could use these tips.

#1 Sir PrintALot

```
while stack:  
    candidate = stack[-1]  
    game.board[candidate.x][candidate.y] = candidate.value
```

- Let's say I am not sure if I popped the correct candidate, How do I make sure?

#1 Sir PrintALot

- On every single line of code, you should have a sense of what all of the variables values' are. If you're not sure, print them out!

```
while stack:  
    candidate = stack[-1]  
    print("current candidate is %s" % candidate)  
    game.board[candidate.x][candidate.y] = candidate.value  
    print("Coordinate (%s, %s) has value %s" % (candidate.x, candidate.y,
```

```
>1 Candidate(positions=[], chain=[(8, 8, 7), (8, 8, 5)])  
>2 Coordinate (0, 2) has value 5
```

- Sometimes a simple "got here" is sufficient if your interested if you made a mistake in your control flow like if-statements or for-loops

#2 Figure out what you want from your future code

- Write down what you want done on a piece of paper or as comments in your file.
 - ▶ These can be one-liners about the functionality you need.
 - ▶ You could then search the internet for examples of each bit of functionality.
- Execute the examples that the course provides you! (Todolist, Kittysshop, etc.)

#3 Start with code that already works

- You should be running any script changes – it's really not possible to test and run your code *too* often.
- Plus, every time you run your code, you're getting feedback on your work. Is it getting closer to what you want, or is it suddenly failing?

#4 Read the error message

Not knowing how to fix an error will not only waste a lot of time, but even worse it can demotivate you to the point where you feel dump and give up learning how to program.

The Python code below is supposed to create a dictionary and print that dictionary out.

```
data = ("Name": "John", "Surname": "Smith")  
print(data)
```

However, when executed, the code produces the following output:

```
File "classmates.py", line 1  
data = ("Name": "John", "Surname": "Smith")  
      ^  
SyntaxError: invalid syntax
```


#4 Read the error message

```
File "classmates.py", line 1
data = ("Name": "John", "Surname": "Smith")
      ^
SyntaxError: invalid syntax
```

Now here are the steps on how to understand and solve that error.

- ❶ Read the error from the beginning
 - ▶ The error occurred in *classmates.py*
- ❷ Look at the error type
 - ▶ In this case the error type is a `SyntaxError`. That means you have written something that doesn't follow the Python syntax rules.
- ❸ Look at the details of the error
 - ▶ On the right of `SyntaxError` you have the detailed information about the error. In this case this information is "invalid syntax" and you also have an arrow character pointing upward. That error is pointing to the colon character. The arrow is trying to say that the colon doesn't belong there
- ❹ Time to use your logic
 - ▶ You are creating a tuple. That's fine. But then after you write the first item ("Name" in this case) you were supposed to write a comma to separate that item from the next item, but you used a colon instead, so the interpreter is saying that a colon is not syntactically correct to use with round brackets.

#4 Read the error message

Now we can solve this error by choosing either one of two solutions:

```
>>> data = ("Name", "John", "Surname", "Smith")
>>> print(data)
('Name', 'John', 'Surname', 'Smith')
>>> data = {"Name": "John", "Surname": "Smith"}
>>> print(data)
{'Name': 'John', 'Surname': 'Smith'}
```

- Sometimes though errors are much more complex than this, so in that case copying and pasting the last line of the error (SyntaxError: invalid syntax in this case) on Google will usually show up forums posts with answers on fixing that particular error.

#5 Consult the internet

- Sometimes though errors are much more complex than this, so in that case copying and pasting the last line of the error (SyntaxError: invalid syntax in this case) on Google will usually show up forums posts with answers on fixing that particular error.

Sunday, January 26, 2020














- × 15:14:27  python - Applying the Sobel filter using scipy - Stack Overflow
- × 15:08:51  python - How to convolve images using scipy - Stack Overflow
- × 15:05:18  python - How to do convolution matrix operation in numpy? - Stack Overflow
- × 15:02:45  python - Display image as grayscale using matplotlib - Stack Overflow
- × 15:02:12  python - what does the flatten means in ndimage.imread()? - Stack Overflow
- × 14:58:25  python - Laplacian sharpening - grey image as result - Stack Overflow
- × 14:58:10  python - Buildin a sparse 2D laplacian matrix using SciPy modules - Stack Overflow
- × 14:53:23  numpy - Discrete Laplacian (del2 equivalent) in Python - Stack Overflow
- × 14:46:56  python - How do you change the size of figures drawn with matplotlib? - Stack Overflow
- × 14:42:26  python - scipy.misc module has no attribute imread? - Stack Overflow
- × 14:39:52  python - What is the difference between scipy.ndimage.imread and matplotlib.pyplot.imread? - Stack Overflow
- × 14:33:03  signal processing - How do I perform a convolution in python with a variable-width Gaussian? - Stack Overflow
- × 14:30:54  python - discrete proximity to gaussian distribution using numpy - Stack Overflow

Figure: Example of Stack Overflow questions in a programming session

#6 Comment-out code

- You could take advantage of the comment feature that resides in every programming language by temporarily “commenting out” code that you don’t want to lose track of, but that you just don’t want running right now.
- If your script is long, you can comment out parts of the code that are unrelated to the specific changes you’re working on. This might make it run faster and make it easier to search the remainder of the code for the mistake.

#7 How to ask for help

Okay okay, so you've really tried everything and it seems like nothing is working. Now you feel like you're really ready to ask someone else for help.

Before asking anyone about a bug in your code, it's important that you make sure you have all of the following components of an excellent question:

- 1 Explain what you're trying to do
- 2 Show the code that's giving the error
- 3 Show the entire stack trace including the error message
- 4 Explain 2-3 thing that you've tried already and why they didn't work

Sometimes, in the simple process of going through these items in your mind and writing them down, a new solution becomes obvious. This is called the "**Rubber Duck**" method.

```
      , ~ ~ .  
      ( 7 ) _ _ ,  
(\ _ _ _ ) == ' - '  
 \ .   ) )  
  \ ` - ' /  
~ ~ ~ ~ ~ ~ ~ ~ ~ ~
```