

Exercise 25: Even More Practice

We're going to do some more practice involving functions and variables to make sure you know them well. This exercise should be straightforward for you to type in, break down, and understand.

However, this exercise is a little different. You won't be running it. Instead you will import it into Python and run the functions yourself.

```
1  def break_words(stuff):
2      """This function will break up words for us."""
3      words = stuff.split(' ')
4      return words
5
6  def sort_words(words):
7      """Sorts the words."""
8      return sorted(words)
9
10 def print_first_word(words):
11     """Prints the first word after popping it off."""
12     word = words.pop(0)
13     print word
14
15 def print_last_word(words):
16     """Prints the last word after popping it off."""
17     word = words.pop(-1)
18     print word
19
20 def sort_sentence(sentence):
21     """Takes in a full sentence and returns the sorted
22     words."""
23     words = break_words(sentence)
24     return sort_words(words)
25
26 def print_first_and_last(sentence):
27     """Prints the first and last words of the
28     sentence."""
29     words = break_words(sentence)
30     print_first_word(words)
31     print_last_word(words)
32
33 def print_first_and_last_sorted(sentence):
34     """Sorts the words then prints the first and last
```

```
34 one."""
35     words = sort_sentence(sentence)
        print_first_word(words)
        print_last_word(words)
```

First, run this like normal with `python ex25.py` to find any errors you have made. Once you have found all of the errors you can and fixed them, you will then want to follow the `wyss` section to complete the exercise.

What You Should See

In this exercise we're going to interact with your `.py` file inside the `python` interpreter you used periodically to do calculations. You run that from the shell like this:

```
$ python
Python 2.7.1 (r271:86832, Jun 16 2011, 16:59:05)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)] on darwin
Type "help", "copyright", "credits" or "license" for
more information.
>>>
```

Yours will look a little different from mine, but once you see the `>>>` prompt you can then type Python code in and it will run immediately.

Here's what it looks like when I do it:

```
Python 2.7.1 (r271:86832, Jun 16 2011, 16:59:05)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)] on darwin
Type "help", "copyright", "credits" or "license" for
more information.
>>> import ex25
>>> sentence = "All good things come to those who
wait."
>>> words = ex25.break_words(sentence)
>>> words
['All', 'good', 'things', 'come', 'to', 'those', 'who',
'wait.']
>>> sorted_words = ex25.sort_words(words)
>>> sorted_words
['All', 'come', 'good', 'things', 'those', 'to',
'wait.', 'who']
```

```

>>> ex25.print_first_word(words)
All
>>> ex25.print_last_word(words)
wait.
>>> wrods
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'wrods' is not defined
>>> words
['good', 'things', 'come', 'to', 'those', 'who']
>>> ex25.print_first_word(sorted_words)
All
>>> ex25.print_last_word(sorted_words)
who
>>> sorted_words
['come', 'good', 'things', 'those', 'to', 'wait.']
>>> sorted_words = ex25.sort_sentence(sentence)
>>> sorted_words
['All', 'come', 'good', 'things', 'those', 'to',
'wait.', 'who']
>>> ex25.print_first_and_last(sentence)
All
wait.
>>> ex25.print_first_and_last_sorted(sentence)
All
who

```

Let's break this down line by line to make sure you know what's going on:

- Line 5: you import *your* `ex25.py` Python file, just like other imports you have done. Notice you do not need to put the `.py` at the end to import it. When you do this you make a `module` that has all your functions in it to use.
- Line 6: you made a `sentence` to work with.
- Line 7: you use the `ex25` module and call your first function `ex25.break_words`. The `.` (dot, period) symbol is how you tell Python, "Hey, inside `ex25` there's a function called `break_words` and I want to run it."
- Line 8: we just type `words`, and Python will print out what's in that variable (line 9). It looks weird but this is a `list` which you will learn about later.
- Lines 10-11: we do the same thing with `ex25.sort_words` to get a sorted sentence.
- Lines 13-16: we use `ex25.print_first_word` and `ex25.print_last_word` to get the first and last word printed out.

- Line 17: is interesting. I made a mistake and typed the `words` variable as `wrods` so Python gave me an error on Lines 18-20.
- Line 21-22: is where we print the modified words list. Notice that since we printed the first and last one, those words are now missing.

The remaining lines are for you to figure out and analyze in the Study Drills.

Study Drills

1. Take the remaining lines of the `wyss` output and figure out what they are doing. Make sure you understand how you are running your functions in the `ex25` module.
2. Try doing this: `help(ex25)` and also `help(ex25.break_words)`. Notice how you get help for your module, and how the help is those odd `"""` strings you put after each function in `ex25`? Those special strings are called `documentation comments` and we'll be seeing more of them.
3. Typing `ex25.` is annoying. A shortcut is to do your import like this: `from ex25 import *` which is like saying, "Import everything from `ex25`." Programmers like saying things backward. Start a new session and see how all your functions are right there.
4. Try breaking your file and see what it looks like in Python when you use it. You will have to quit Python with CTRL-D (CTRL-Z on windows) to be able to reload it.

Common Student Questions

I get a `None` printed out for some of the functions.

You probalby have a function that is missing the `return` at the end. Go backward through the file like I taught you and confirm that every line is right.

I get `-bash: import: command not found` when I type `import ex25`.

Pay attention to what I'm doing in the `wyss` section. I'm doing this in *Python* not in the Terminal. That means you first run Python.

I get `ImportError: No module named ex25.py` when I type `import ex25.py`.

Don't add the `.py` to the end. Python knows the file ends in `.py` so you just type `import ex25`.

I get `SyntaxError: invalid syntax` when I run this.

That means you have something like a missing `(` or `"` or similar syntax error on that line or above it. Any time you get that error, start at the line it mentions and check that it's right, then go backward checking each line above that.

How can the `words.pop(0)` be changing the `words` variable then?

That's a complicated question, but in this case `words` is a list, and because of that you can give it commands and it'll retain the results of those commands. This is similar to how files and many other things worked when you were working with `f.readline()`.

When should I `print` vs. `return` in a function?

You need to understand that `print` is only for printing to the screen and that you can actually do both `print` and `return` a value. When you understand this then you'll see that the question is kind of a pointless question. You use `print` when you want to print. You use `return` when you want to return.

Purchase The Videos For \$29.59

For just \$29.59 you can get access to all the videos for [Learn Python The Hard Way](#), **plus** a PDF of the book and no more popups all in this one location. For \$29.59 you get:

- All 52 videos, 1 per exercise, almost 2G of video.
- A PDF of the book.

- Email help from the author.
- [See a list of everything you get before you buy.](#)

When you buy the videos they will immediately show up **right here** without any hassles.



[Already Paid? Reactivate Your Purchase Right Now!](#)


Buying Is Easy

Buying is easy. Just fill out the form below and we'll get started.

Full Name

Email Address

☒    Pay With Credit Card (by Stripe™)

☐  Use your PayPal™ account.

Buy Learn Python The Hard Way, 3rd Edition



Copyright (C) 2010 Zed. A. Shaw