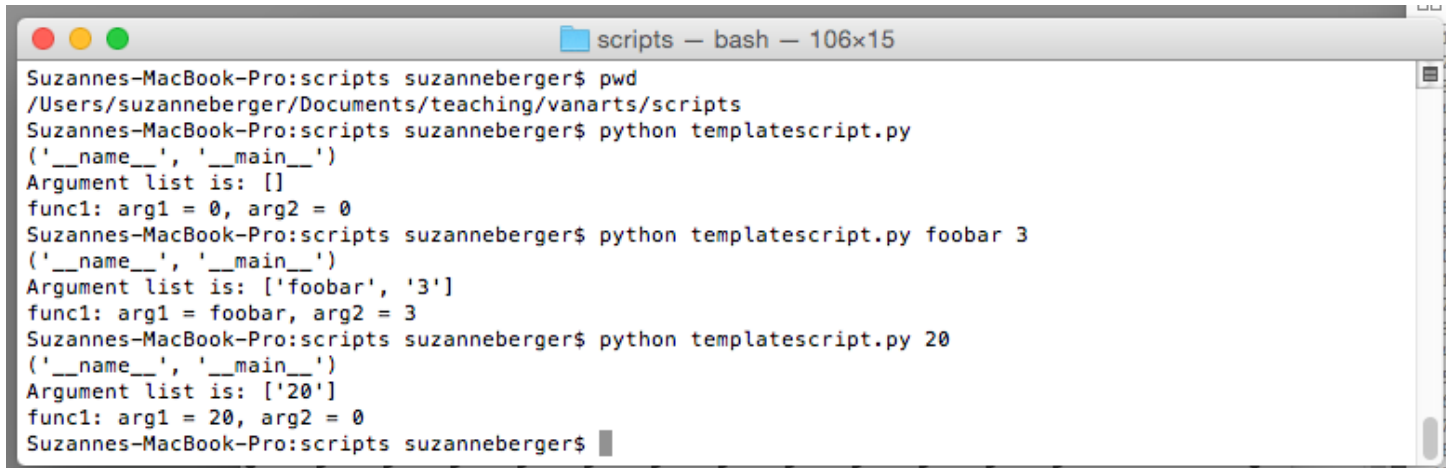# Running or Importing Python Scripts
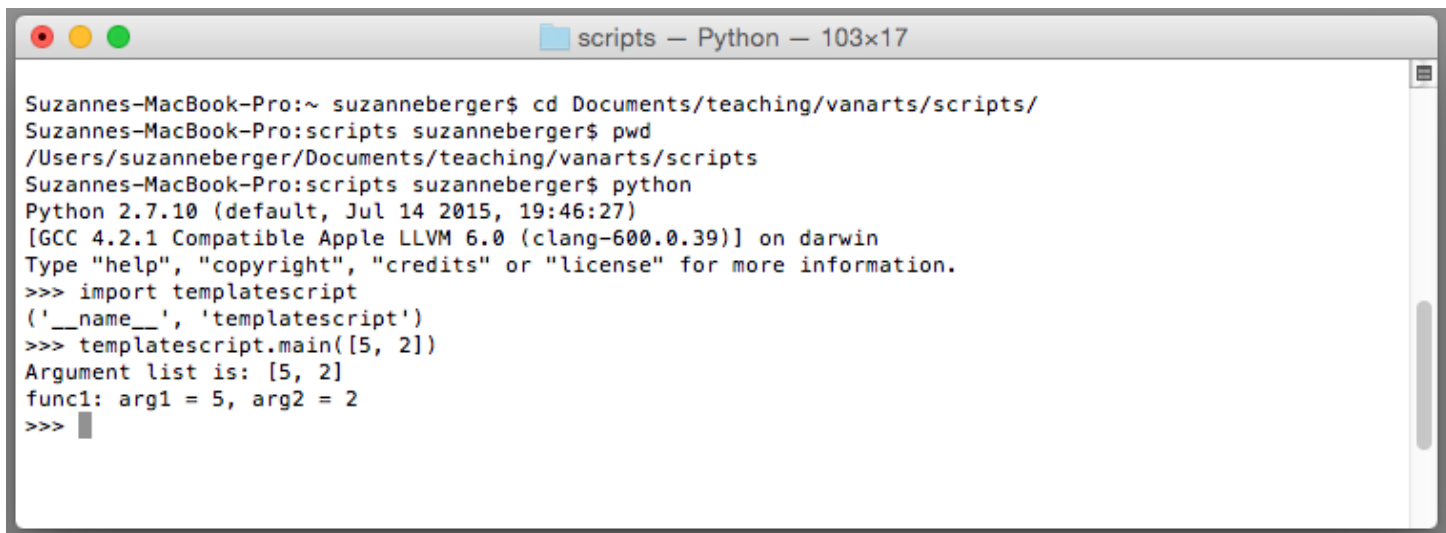## By: Suzanne Berger, Date: September 16, 2018

Note that first three examples were produced on MacBook Pro.

Running script directly from command line when script is contained in same directory as current working directory.

```
Suzannes-MacBook-Pro:scripts suzanneberger$ pwd
/Users/suzanneberger/Documents/teaching/vanarts/scripts
Suzannes-MacBook-Pro:scripts suzanneberger$ python templatescript.py
('__name__', '__main__')
Argument list is: []
func1: arg1 = 0, arg2 = 0
Suzannes-MacBook-Pro:scripts suzanneberger$ python templatescript.py foobar 3
('__name__', '__main__')
Argument list is: ['foobar', '3']
func1: arg1 = foobar, arg2 = 3
Suzannes-MacBook-Pro:scripts suzanneberger$ python templatescript.py 20
('__name__', '__main__')
Argument list is: ['20']
func1: arg1 = 20, arg2 = 0
Suzannes-MacBook-Pro:scripts suzanneberger$
```

Importing script in python interpreter when script is in same directory as current working directory.
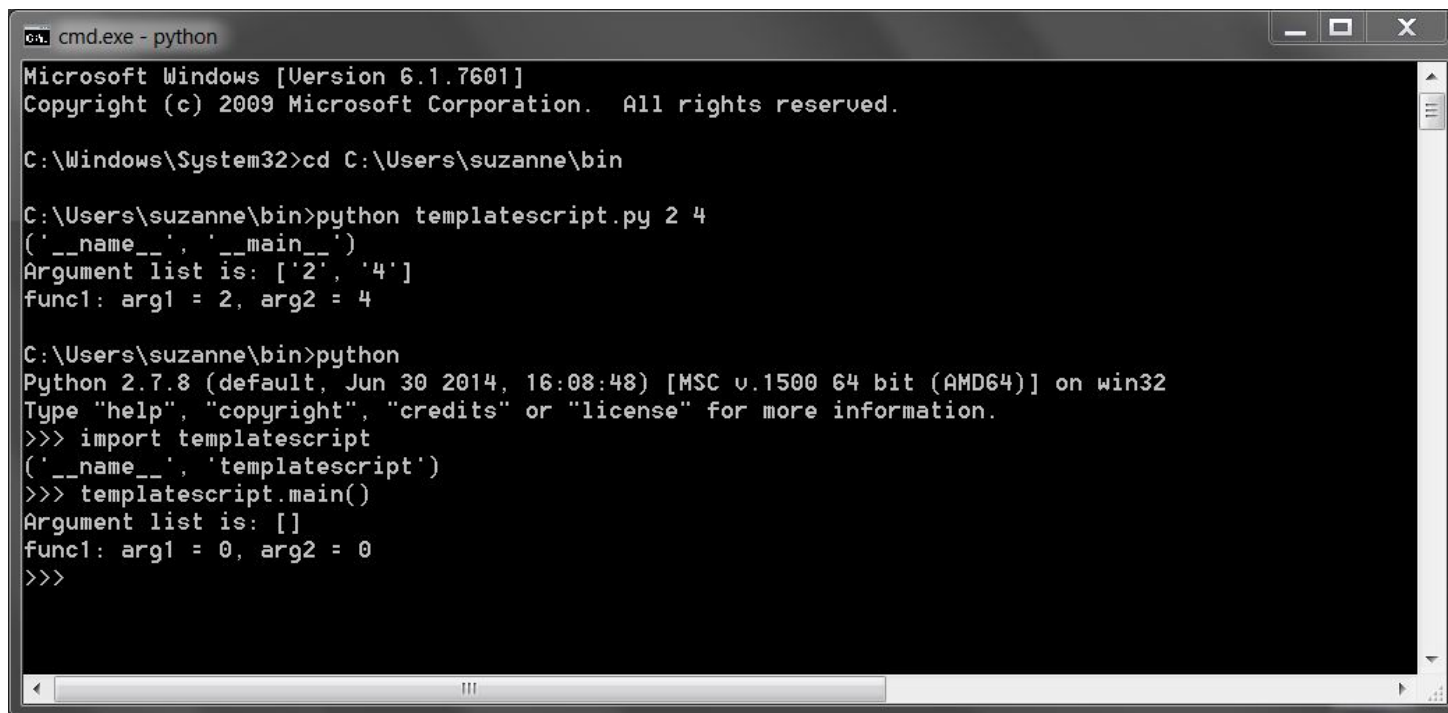
```
Suzannes-MacBook-Pro:~ suzanneberger$ cd Documents/teaching/vanarts/scripts/
Suzannes-MacBook-Pro:scripts suzanneberger$ pwd
/Users/suzanneberger/Documents/teaching/vanarts/scripts
Suzannes-MacBook-Pro:scripts suzanneberger$ python
Python 2.7.10 (default, Jul 14 2015, 19:46:27)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import templatescript
('__name__', 'templatescript')
>>> templatescript.main([5, 2])
Argument list is: [5, 2]
func1: arg1 = 5, arg2 = 2
>>>
```
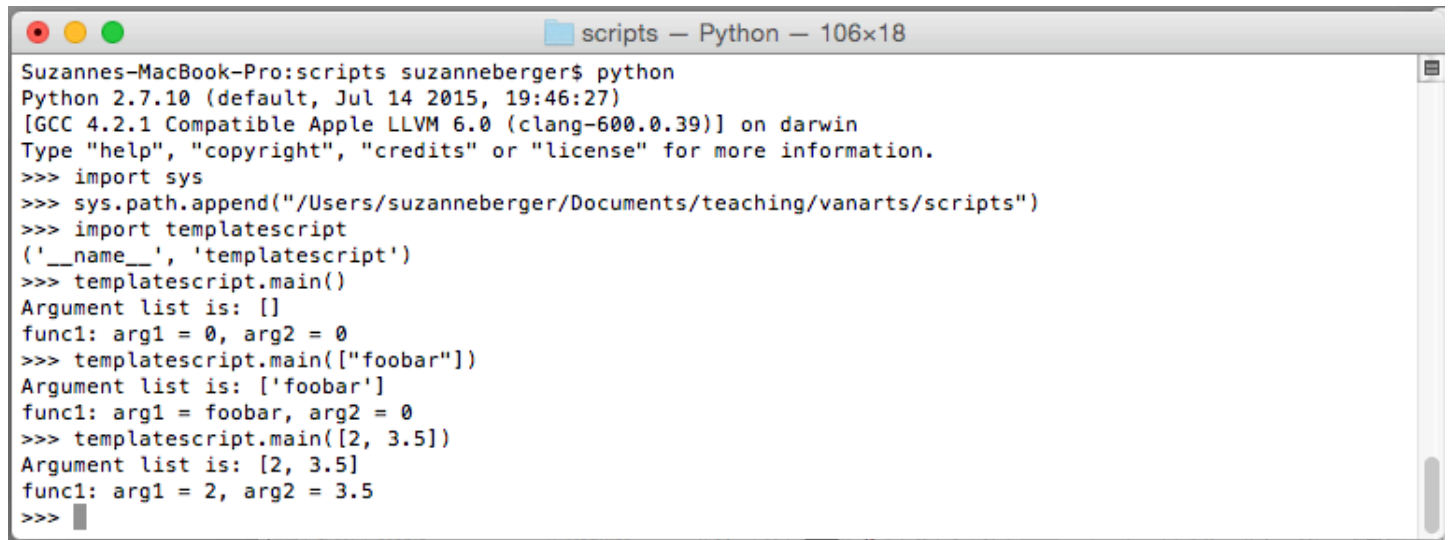
Importing script in python interpreter when full script path needs to be added to *sys.path* .

```
Suzannes-MacBook-Pro:scripts suzanneberger$ python
Python 2.7.10 (default, Jul 14 2015, 19:46:27)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path.append("/Users/suzanneberger/Documents/teaching/vanarts/scripts")
>>> import templatescript
('__name__', 'templatescript')
>>> templatescript.main()
Argument list is: []
func1: arg1 = 0, arg2 = 0
>>> templatescript.main(["foobar"])
Argument list is: ['foobar']
func1: arg1 = foobar, arg2 = 0
>>> templatescript.main([2, 3.5])
Argument list is: [2, 3.5]
func1: arg1 = 2, arg2 = 3.5
>>>
```

This example was produced on very old Windows 7 laptop ... but it worked ;). I saved script in '*bin*' folder under my home directory and executed python from that folder using the '*cd*' command to change directories.

```
cmd.exe - python

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\System32>cd C:\Users\suzanne\bin

C:\Users\suzanne\bin>python templatescript.py 2 4
('__name__', '__main__')
Argument list is: ['2', '4']
func1: arg1 = 2, arg2 = 4

C:\Users\suzanne\bin>python
Python 2.7.8 (default, Jun 30 2014, 16:08:48) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import templatescript
('__name__', 'templatescript')
>>> templatescript.main()
Argument list is: []
func1: arg1 = 0, arg2 = 0
>>>
```