SI 506: Programming I Fall 2019

Lecture 14

Anthony Whyte <arwhyte@umich.edu>
Lecturer III, School of Information
715 N. University Ave, Ann Arbor, MI 48109
Roumanis Square, 2nd floor ("the loft")





preliminaries





Exercise Paths

lectures/lecture_I4/
lecture_I4_solution.py
zen_input.txt





Midterm problem manual grading GSIs not finished yet



> 400 points earned





2nd half





Weeks 1-7

Python topics

- values, types, and variables
- statements and expressions
- operators (select list)
 - arithmetic (+, -, *, /, //)
 - assignment (=, +=)
 - comparison (==, !=, >, <, >=, <=)
 - logical (and, or, not)
 - membership (in, not in)
- built-in functions (input(), int(),
 float(), len(), open(), range(),
 sorted(), str(), type())

- strings, string methods
- lists, list methods
- indexing and slicing
- loops (for, while)
 - use of counters in loops $(i = 0 \dots i +=1)$
 - control statements (continue, break)
- conditional statements (if, else)
 - truth value testing (test object for truth value in if or while statements)
- functions
 - parameters, optional parameters
 - return statements
- reading from and writing to files
 - using the with statement





Weeks 8-15

weeks I-7 topics +

- data types
 - dictionaries
 - tuples
- modules
 - CSV
 - json (encode/decode)
 - pathlib
 - requests
- functions
 - lambdas (anonymous functions)
- lists
 - list comprehensions
- classes

- local dev environment
 - Python install
 - source code editor/IDE
 - command line
- debugging
- file types
 - *.csv
 - *.json
- data structures
 - structured data
 - semi-structured data
- RESTful APIs
 - HTTP request/response
 - JSON

final individual project assignment





VS Code / GitBash

Windows 10 users

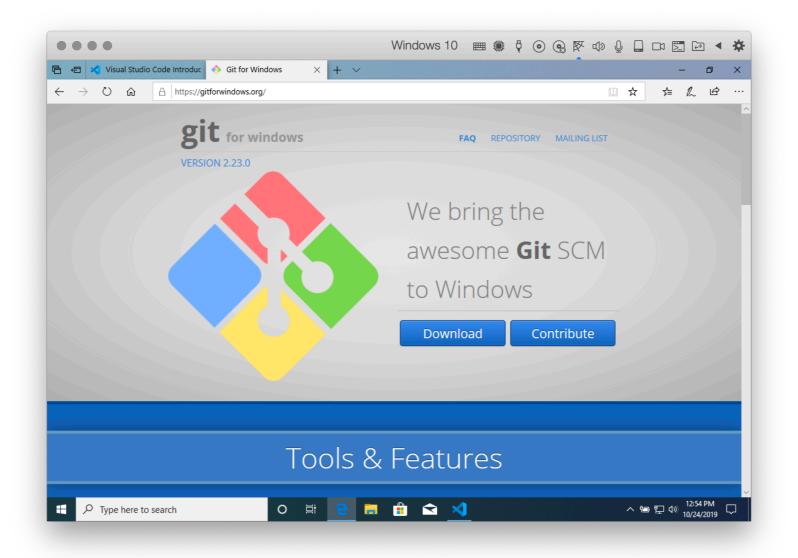




VS Code / GitBash: new guide

install Git tool set; configure VS Code to use Bash shell

http://bit.ly/2BF3t0n







commands





Mac vs Windows Paths

Windows backwards slash (must escape)

```
Mac (/Users/arwhyte/lecture_14/):
source_path = './source/zen_input.txt'
source_path = /Users/arwhyte/lecture_14/source/zen_input.txt

Windows:
source_path = '.\\source\\zen_input.txt'
source_path = 'C:\\Users\\arwhyte\\lecture_14\\source\\zen_input.txt'
```





Commands view directory contents

Mac/*nix	Win cmd	Description
ls	dir	List files and subdirectories in current directory.
ls -l	dir /A	List files and subdirectories in current directory in long format (includes permissions, size, modification date)
ls -a	dir /A:H	Include hidden files in listing (-a flag, /A:H switch).





Commands change directories

Mac/*nix	Win cmd	Description
cd <filepath></filepath>	cd <filepath></filepath>	Change directory
cd	cd	Change to parent directory (one level up)
cd//	cd//	Change directories (two levels up). Add/ for each level up
cd ~	cd %userprofile%	Change to user's home directory
cd /	cd\	Change to root directory





Commands

create empty file; delete file; view file contents

Mac/*nix	Win cmd	Description
touch <file></file>	copy NUL <file></file>	Mac/*nix: touch modifies a file's timestamp. But if the file does not exist it creates it. Windows: copy NUL <file> will generate an empty file.</file>
rm <file></file>	del <file></file>	Delete a file.
cat <file></file>	type <file></file>	View file contents.
head <file></file>		View first 10 lines of file contents. Windows: no exact equivalent. DOS command more +n <file> will output all lines after the first n lines.</file>
tail <file></file>		View last 10 lines of file contents.
tail -f <file></file>		View file contents as it grows, starting with last 10 lines.





Commands create / remove directories

Mac/*nix	Win cmd	Description
mkdir <dir></dir>	md <dir>, mkdir <dir></dir></dir>	Create a new directory.
rm dir <dir></dir>	rd <dir>, rmdir <dir></dir></dir>	Delete an empty directory
rm -rf <dir></dir>	rd /S /Q <dir>, rmdir /S / Q <dir></dir></dir>	Force delete directory and all contents recursively. /S = remove all subdirectories; /Q = quiet mode.





Commands copy / move files and directories

Mac/*nix	Win cmd	Description
<pre>cp <source/> <destination></destination></pre>	copy <source/> <destination></destination>	Copy a file to a new location.
cp -r <source/> <destination></destination>	xcopy /E <source/> <destination></destination>	Copy a directory and contents recursively.
mv <source/> <destination></destination>	move <source/> <destination></destination>	Move a file or folders.





Commands clear the screen

Mac/*nix	Win cmd	Description
clear	cls	Clears the screen.





finis





directors cut





pathlib module





pathlib module: Path class

cross platform solution: use it to return path objects

```
import pathlib
```

```
path = pathlib.Path('lecture_13_pathlib.py')
print(f"data type = {type(path)}\n")
```

Mac:

```
data type = <class 'pathlib.PosixPath'>
```

Windows:

```
data type = <class 'pathlib.WindowsPath'>
```





check if path exists, is a directory, or is a file

```
import pathlib
path = pathlib.Path('lecture_13_pathlib.py')
```

```
path exists = path.exists()
is file = path.is_file()
is dir = path.is_dir() # False
if path.exists():
```

print(f"path exists.\n")





```
home directory:.home()
import pathlib
path = pathlib.Path('lecture_13_pathlib.py')
home_directory = path.home()
```

print(f"Home directory = {home_directory}\n")

Mac:

Home directory = /Users/arwhyte

Windows:

Home directory = C:\users\arwhyte





```
current working directory:.cwd()
import pathlib
path = pathlib.Path('lecture_13_pathlib.py')
cwd = path.cwd()
print(f"cwd = {cwd}\n")
Mac:
cwd = /Users/arwhyte/lectures/lecture 13
Windows:
cwd = C:\users\arwhyte\lectures\lecture 13
```



path components

```
import pathlib
path = pathlib.Path('lecture_13_pathlib.py')
print(
    f"path.anchor = {path.anchor}",
    f"path.name = {path.name}",
    f"path.stem = {path.stem}",
    f"path.suffix = {path.suffix}",
    f"path.parent = {path.parent}",
    sep='\n'
```





```
path components: name
```

```
import pathlib
path = pathlib.Path('lecture_13_pathlib.py')
file_name = path.name
print(f"File name = {file name}\n")
Mac:
cwd = lecture 13 pathlib.py
Windows:
```

```
cwd = lecture_13_pathlib.py
```





```
absolute file path: .absolute()
```

```
import pathlib
```

```
path = pathlib.Path('lecture_13_pathlib.py')
```

```
file_path_abs = path.absolute()
```

```
print( f"Absolute file path = {file_path_abs}\n")
```

Mac:

```
CWd = /Users/arwhyte/lectures/lecture_13/lecture_13_pathlib.py
```

Windows:

```
CWd = C:\users\arwhyte\lectures\lecture_13\lecture_13_pathlib.py
```





When your code misbehaves debug flowchart

Attribute Error

You are calling a method on the wrong type of object

SyntaxError

You've forgotten the quotes around a string

You have forgotten to put a colon at the end of a def/if/for line

You have different number of open and close brackets in a statement

TypeError

You're trying to use an operator on the wrong type of objects

An object which you expect to have a value is actually None

You've used non-integer numbers in a list slice

You've called a method/ function with the wrong number or type of arguments

Indentation Error

You've used a mixture of tabs and spaces You haven't indented all

lines in a block equally

My code isn't working:-(

Start here...

Do you get an

error when you

run the code?

Does the code

use loops or if

statements?

Two numbers which should

be equal are not

You are comparing a number

with a string representation

of a number (e.g. if 3 == "3")

A complex condition is not

giving the expected result

The order of precedence in the

condition is ambiguous - add

some parentheses

What type of error do you get?

NameError

You've misspelt a variable, function or method name

> You've forgotten to import a module

> You've forgotten to define a variable

Your code uses a variable outside the scope where it's defined

Your code calls a function before it's defined

You're trying to print a single word and have forgotten the quotes

IOError

You're trying to open a file that doesn't exist

KeyError

You're trying to look up a key that doesn't exist in a dict

http://pythonforbiologists.com

A variable that should contain a value does not

You are storing the return value of a function which You are printing an object changes the variable itself (e.g. sort)

A number which should be a fraction is coming out as zero in Python 2

You are dividing integers rather than floats. Convert the numbers to floats or from __future__ import division

I'm trying to print a value but getting a weirdlooking string

(e.g. a FileObject) when you want the result of calling a method on the object

A regular expression is not matching when I expect it to

You have forgotten to use raw strings or escape backslash characters

I am reading a file but getting no input

You have already read the contents of the file earlier in the code, so the cursor is at the end.

neithei

loops

A list which should have a value for every iteration only has a single value

You have defined the list inside the loop: move it outside

A loop which uses the range function misses out the last value

The range function is exclusive at the finish: increase it by one.

I am trying to loop over a collection of strings, but am getting individual characters

You are iterating over a string by mistake

I am trying to write multiple lines to a file but only getting a single one You have opened the file inside the loop: move it

also check.





Slide deck revisions

errata: corrections and other changes

Slide no(s). Fix ver. Description
v1p1



