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")

UMSI

preliminaries

UMSI

Exercise

Paths

lectures/lecture_14/

lecture_14_solution.py

zen_input.txt

UMSI

Midterm problem manual grading

GSIs not finished yet

~50%

> 400 points earned

UMSI

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` … `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

UMSI

Weeks 8-15

weeks 1-7 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

UMSI

VS Code / GitBash

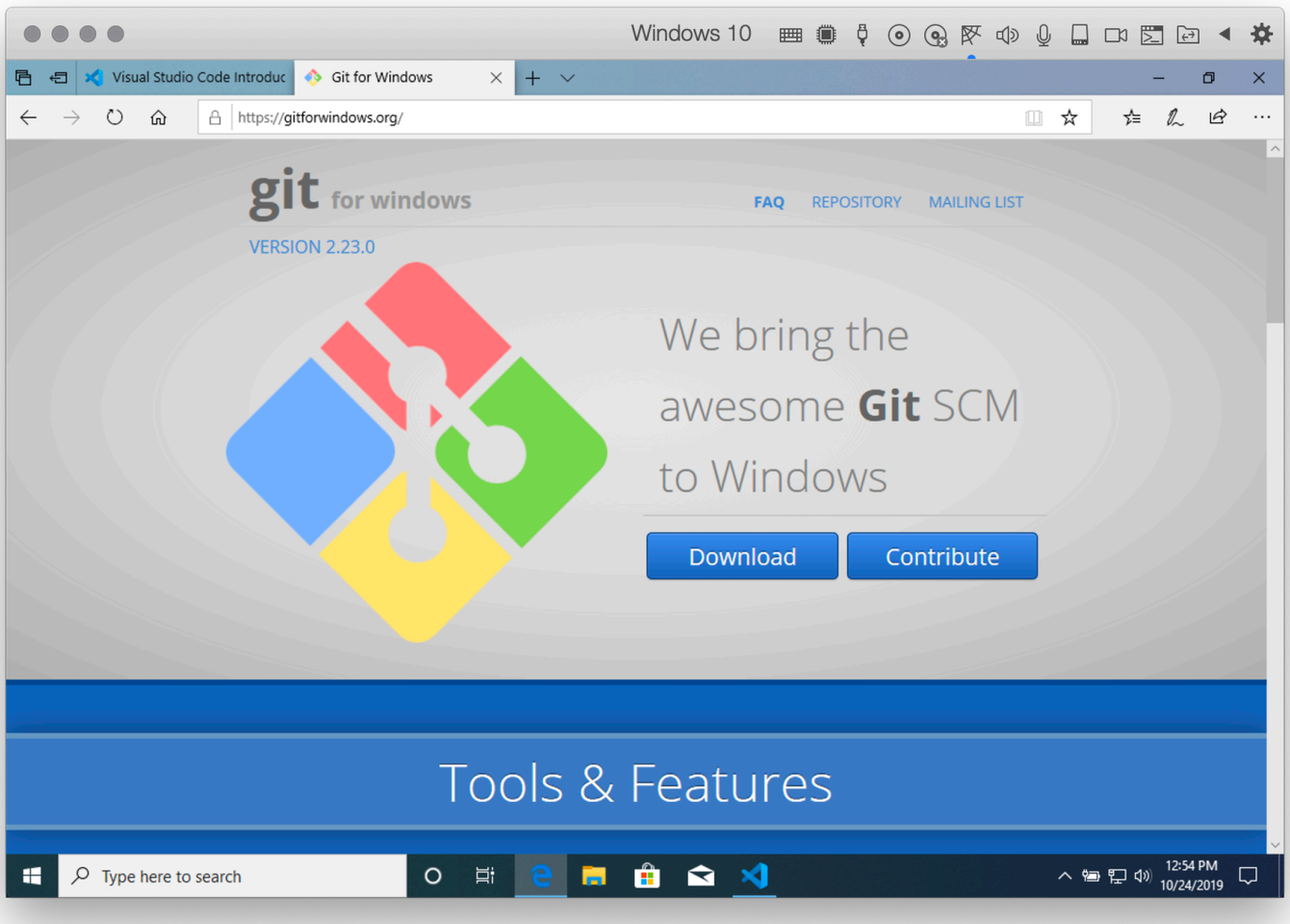
Windows 10 users

UMSI

VS Code / GitBash: new guide

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

http://bit.ly/2BF3t0n



UMSI

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'
```


UMSI

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>	cd <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>	copy NUL <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.
rm <file>	del <file>	Delete a file.
cat <file>	type <file>	View file contents.
head <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.
tail <file>		View last 10 lines of file contents.
tail -f <file>		View file contents as it grows, starting with last 10 lines.

Commands

create / remove directories

Mac/*nix	Win cmd	Description
mkdir <dir>	md <dir>, mkdir <dir>	Create a new directory.
rm dir <dir>	rd <dir>, rmdir <dir>	Delete an *empty* directory
rm -rf <dir>	rd /S /Q <dir>, rmdir /S /Q <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
cp <source> <destination>	copy <source> <destination>	Copy a file to a new location.
cp -r <source> <destination>	xcopy /E <source> <destination>	Copy a directory and contents *recursively*.
mv <source> <destination>	move <source> <destination>	Move a file or folders.

Commands

clear the screen

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

finis

directors cut

UMSI

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'>
```

pathlib module: methods

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")
```

pathlib module: methods

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
```

pathlib module: methods

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
```


UMSI

pathlib module: methods

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'
)
```


UMSI

pathlib module: methods

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
```


UMSI

pathlib module: methods

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:

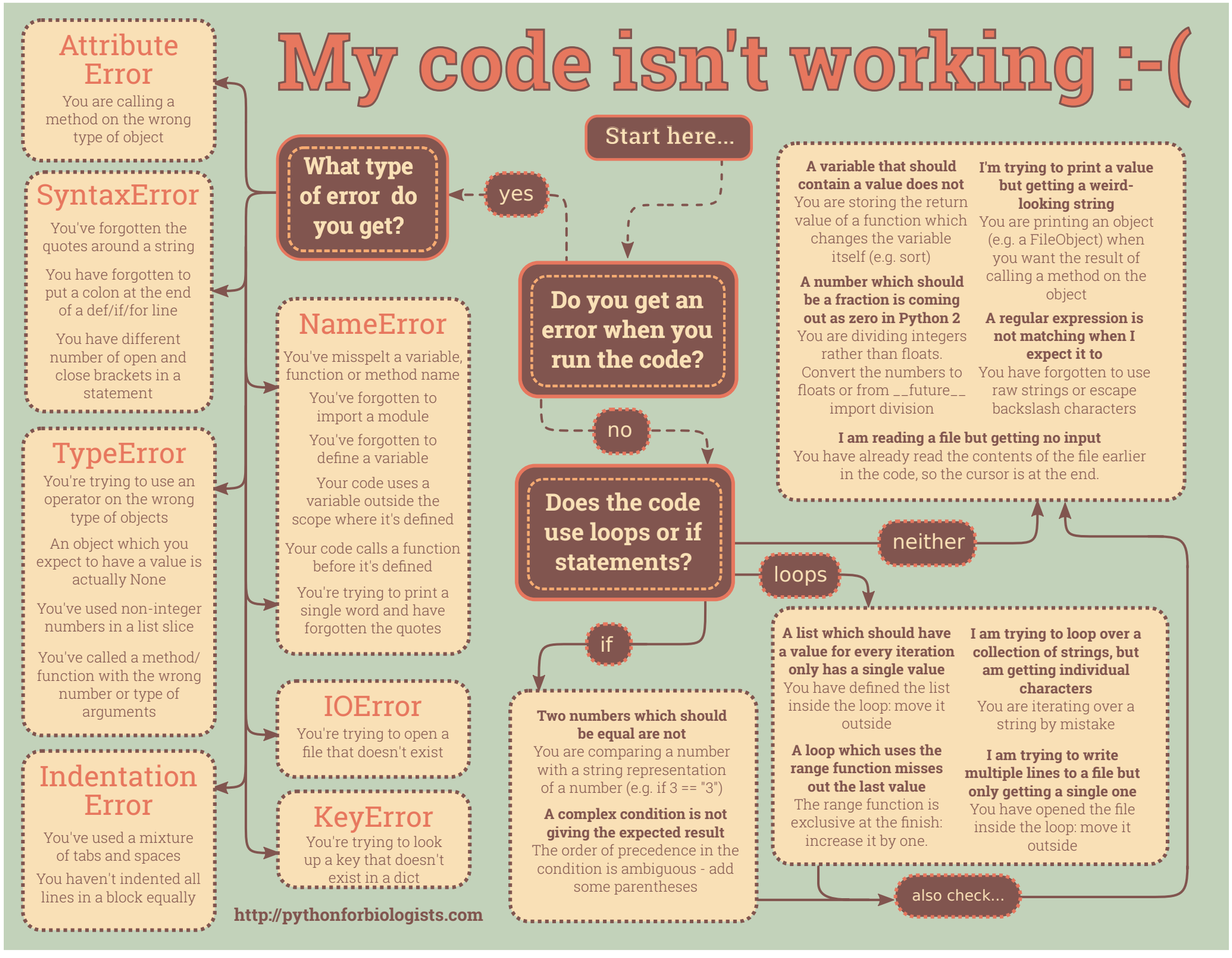
```
cwd = /Users/arwhyte/lectures/lecture_13/lecture_13_pathlib.py
```

Windows:

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

When your code misbehaves

debug flowchart



UMSI

Slide deck revisions

errata: corrections and other changes

Slide no(s).	Fix ver.	Description
	v1p1	