building a shell from scratch

Redbrick AGM

Wednesday 3rd April - CG05.

- Chair
- Vice-Chair
- Secretary
- Public Relations Officer
- Treasurer
- Graphical Design Officer
- Events Officer x2
- Ordinary Member x2
- System Administrator x3
- Webmaster
- Helpdesk x2

today's material

- Code → github.com/theycallmemac/sesh
- Slides → jamesmcdermott.ie/sesh

what is a shell?

- Interface (kernel metaphor)
- Interpreter (like python3 interpreter)
- Environment (it’s surroundings describe it and its functionality)
- Tool (and that means it’s replaceable)

why do we use shells?

- Simple (or at least as simple as interacting with your machines resources can be)
- Fast (Yeah GUI is nice, but it’s much faster to operate through a shell)

how do they work?

- Read (they interpret the user input)
- Tokenize (they split the input into arguments)
- Execute (they execute these arguments)

Let’s look at a simple example in Python

types of shells

ksh

csh

sh

fish

zsh

tcsh

bash

am I qualified?

- Yes
- … and No

Cheap plug: I wrote a shell called ezsh

Check it out at ezsh.jamesmcdermott.ie

today's workshop

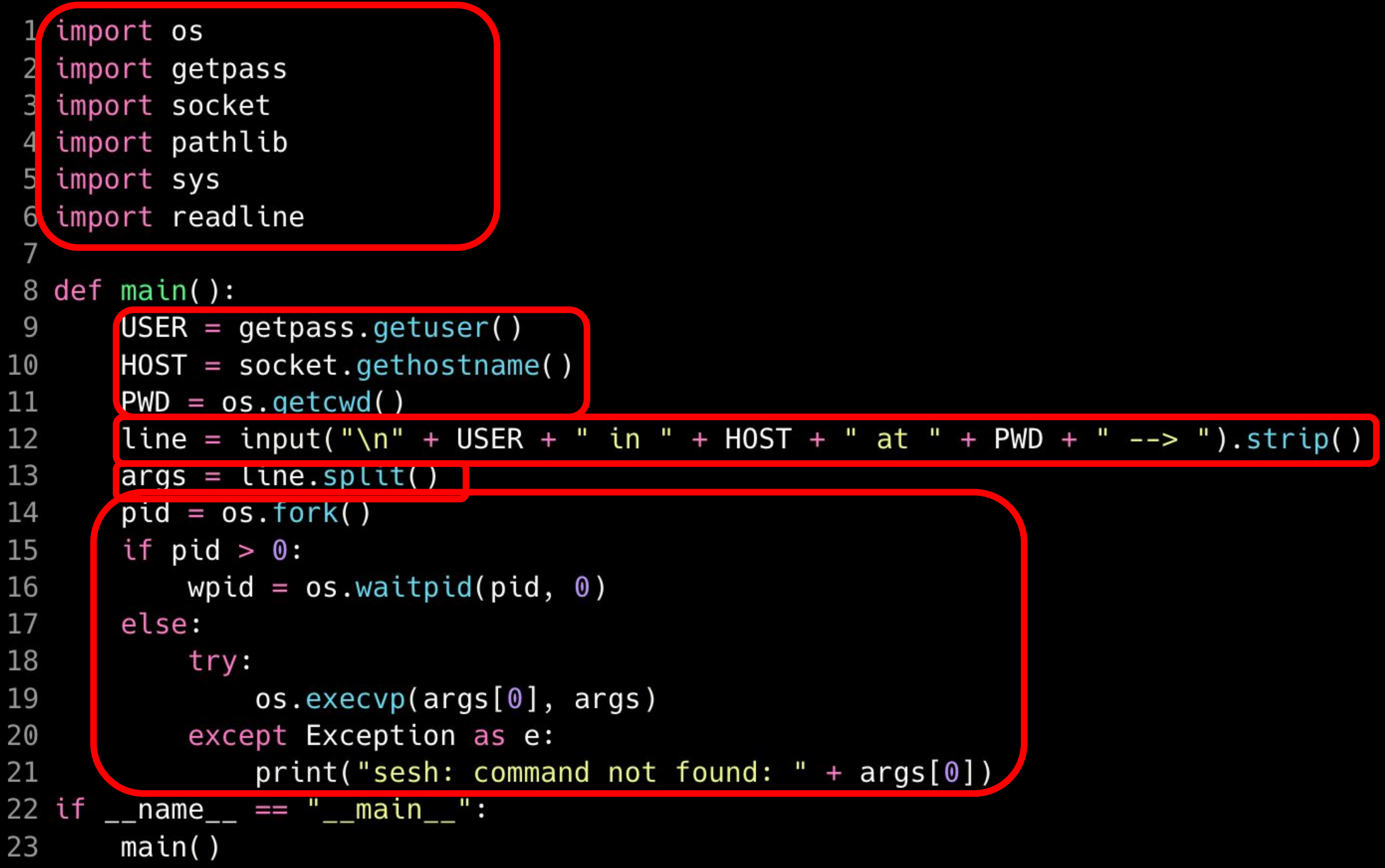
- 6 Sessions
    - 5 writing code
    - 1 asking questions
    - Reflection

simple
easy
shell

simple
easy
shell

session one

- Mocking a basic shell
    - Read
    - Write
    - Execute



```python
import os
import getpass
import socket
import pathlib
import sys
import readline

def main():
    USER = getpass.getuser()
    HOST = socket.gethostname()
    PWD = os.getcwd()
    line = input("\n" + USER + " in " + HOST + " at " + PWD + " --> ").strip()
    args = line.split()
    pid = os.fork()
    if pid > 0:
        wpid = os.waitpid(pid, 0)
    else:
        try:
            os.execvp(args[0], args)
        except Exception as e:
            print("sesh: command not found: " + args[0])
if __name__ == "__main__":
    main()
```

session two

- Looping

- Modularisation

```python
def main():
    while True:
        USER = getpass.getuser()
        HOST = socket.gethostname()
        PWD = os.getcwd()
        line = read_line(USER, HOST, PWD)
        args = tokenize(line)
        execute(args)
if __name__ == "__main__":
    main()
```

```python
def read_line(USER, HOST, PWD):
    line = input("\n" + USER + " in " + HOST + " at " + PWD + " --> ").strip()
    return line

def tokenize(line):
   args = line.split()
   return args

def execute(args):
    pid = os.fork()
    if pid > 0:
        wpid = os.waitpid(pid, 0)
    else:
        try:
            os.execvp(args[0], args)
        except Exception as e:
            print("sesh: command not found: " + args[0])
```

session three

- Making Built In Commands
  - cd
  - quit
- Launch and Execute Functions

```python
def launch(args):
    pid = os.fork()
    if pid > 0:
        wpid = os.waitpid(pid, 0)
    else:
        try:
            os.execvp(args[0], args)
        except Exception as e:
            print("sesh: command not found: " + args[0])

def execute(args):
    try:
        if len(args) == 0:
            pass
        elif "cd" == args[0]:
            cd("".join(args[1:]))
        elif "quit" == args[0]:
            quit()
        else:
            launch(args)
    except EOFError as e:
         print("")
```

```python
def cd(args):
    try:
        if len(args) == 0:
            home_dir = str(pathlib.Path.home())
            os.chdir(home_dir)
        else:
            os.chdir(args)
    except Exception as e:
        print("cd: no such file or directory: " + args)


def quit():
    sys.exit(0)
```

session four

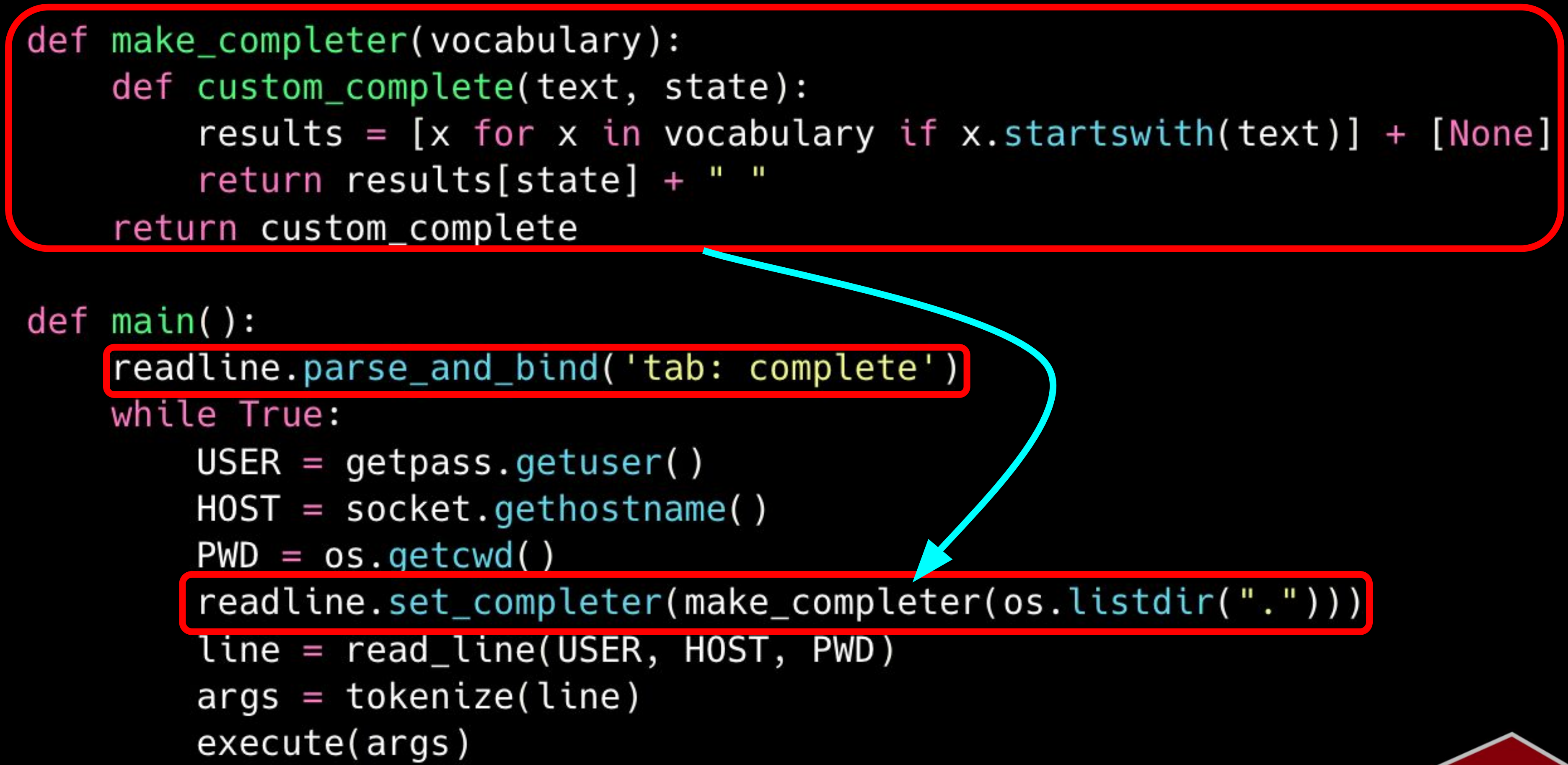
- Let’s make it colourful
- Autocompletion

```
def color_yellow(word):
    return "\033[1;91m" + word +  "\033[00m"

def color_pink(word):
    return "\033[1;95m" + word +  "\033[00m"

def color_green(word):
    return "\033[1;92m" + word +  "\033[00m"

def read_line(USER, HOST, PWD):
    line = input("\n" + color_yellow(USER) + " in " + color_pink(HOST) + " at " + color_green(PWD) + " --> ")
    return line.strip()
```



```python
def make_completer(vocabulary):
    def custom_complete(text, state):
        results = [x for x in vocabulary if x.startswith(text)] + [None]
        return results[state] + " "
    return custom_complete

def main():
    readline.parse_and_bind('tab: complete')
    while True:
        USER = getpass.getuser()
        HOST = socket.gethostname()
        PWD = os.getcwd()
        readline.set_completer(make_completer(os.listdir(".")))
        line = read_line(USER, HOST, PWD)
        args = tokenize(line)
        execute(args)
```

session five

- sesh interpreter

```
def execute(args):
    try:
        if len(args) == 0:
            pass
        elif "cd" == args[0]:
            cd("".join(args[1:]))
        elif "quit" == args[0]:
            quit()
        elif "sesh" == args[0]:
            sesh("".join(args[1:]))
        else:
            launch(args)
    except EOFError as e:
        print("")

def sesh(args):
    try:
        for line in open(args, "r"):
            execute(line.split())
    except Exception as e:
        print("sesh: cannot access " + args + ": No such file or directory")
```

session six

- chat and question time:
  - i/o redirection
  - pipes
  - logical operations
  - background processes
  - other built in commands
  - questions?

today's material

- Code → github.com/theycallmemac/sesh
- Slides → jamesmcdermott.ie/sesh