

Lab 2

CS50 for MBAs

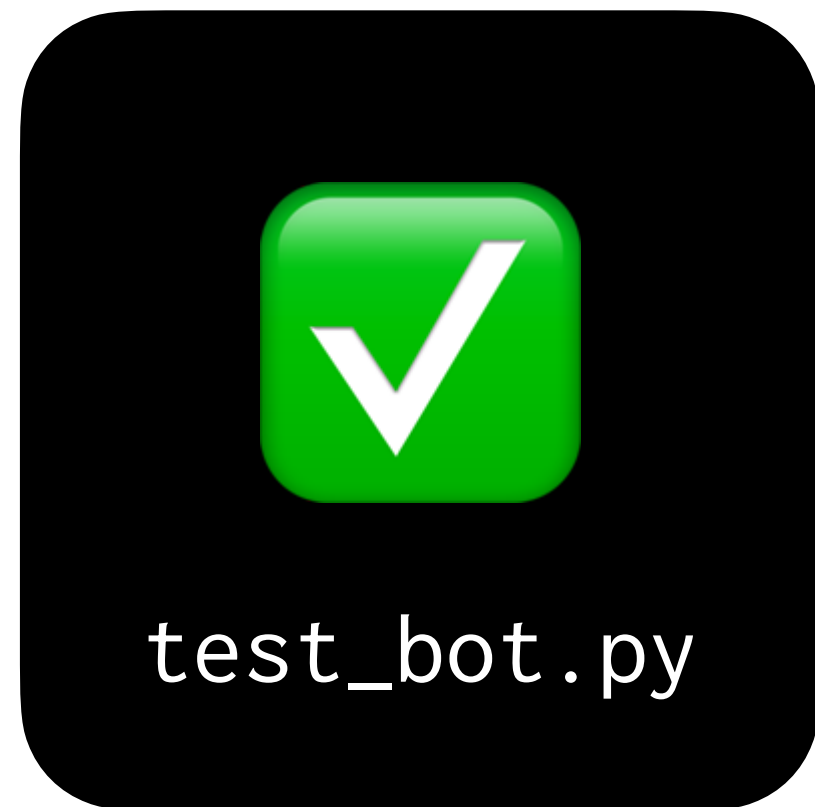
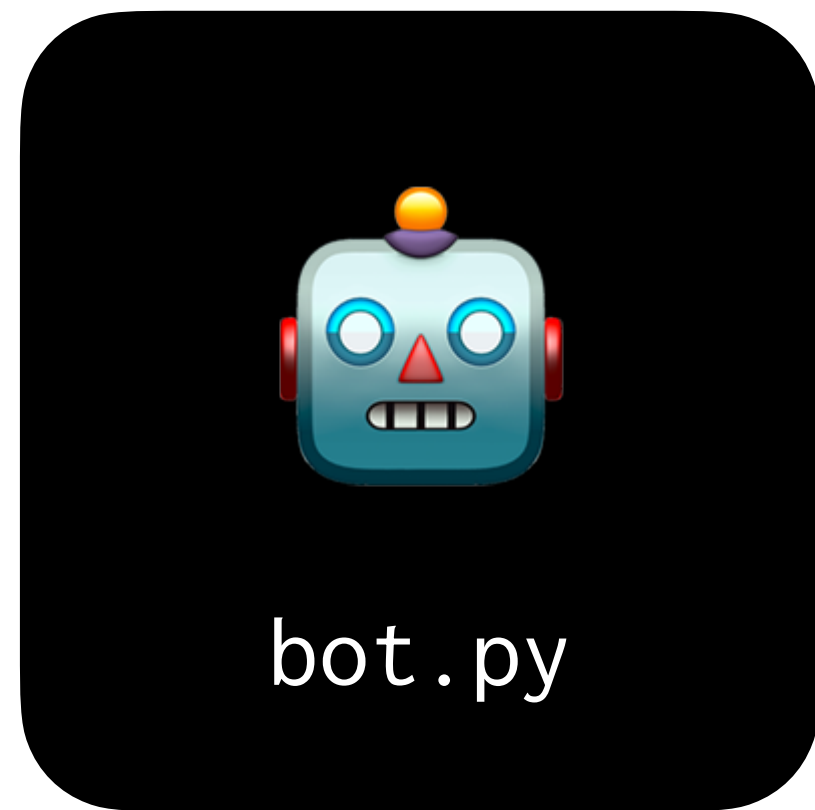
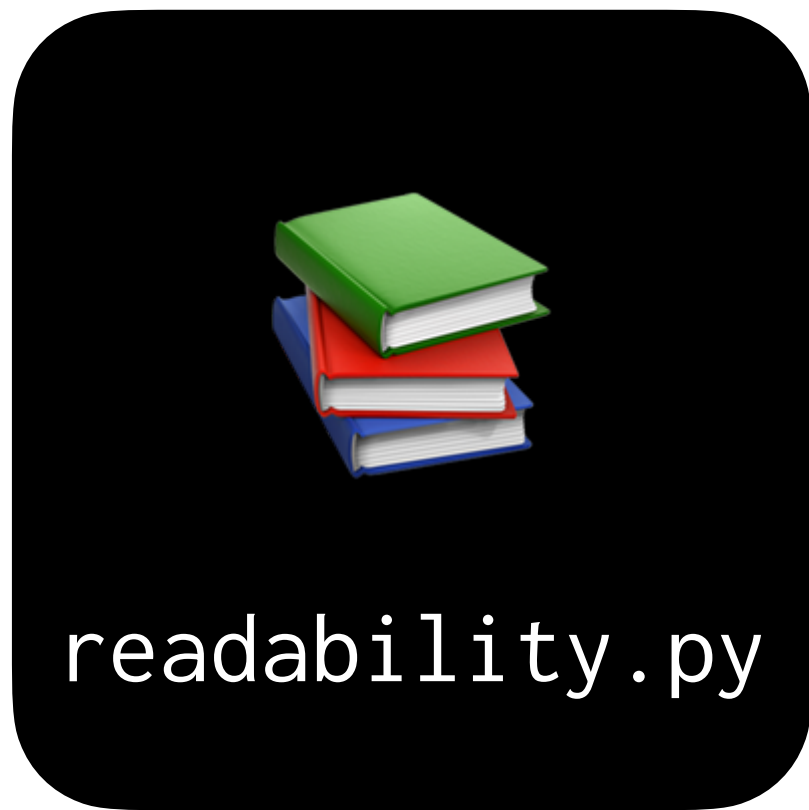
carterzenke.me/lab

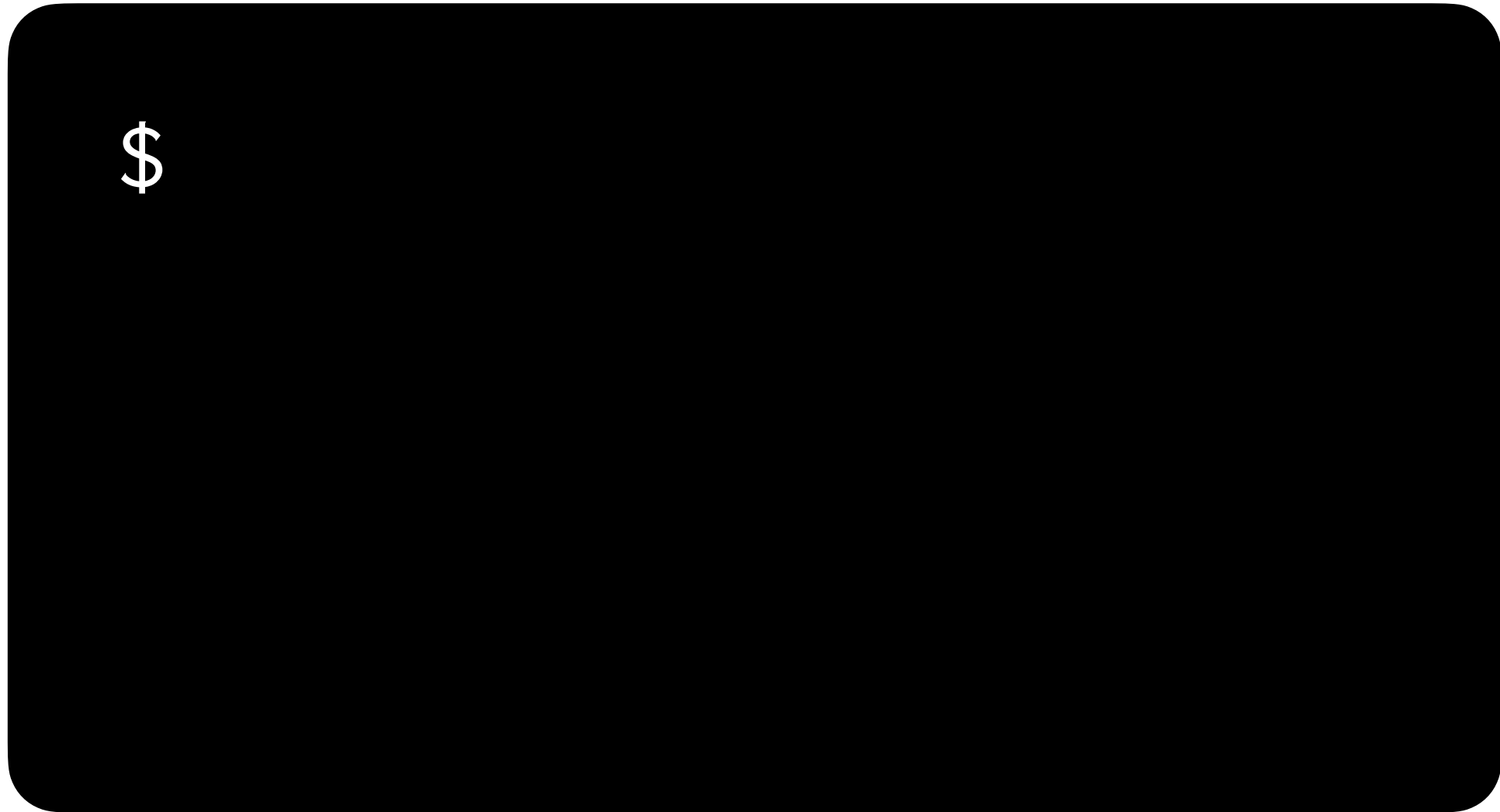


readability.py



bot.py





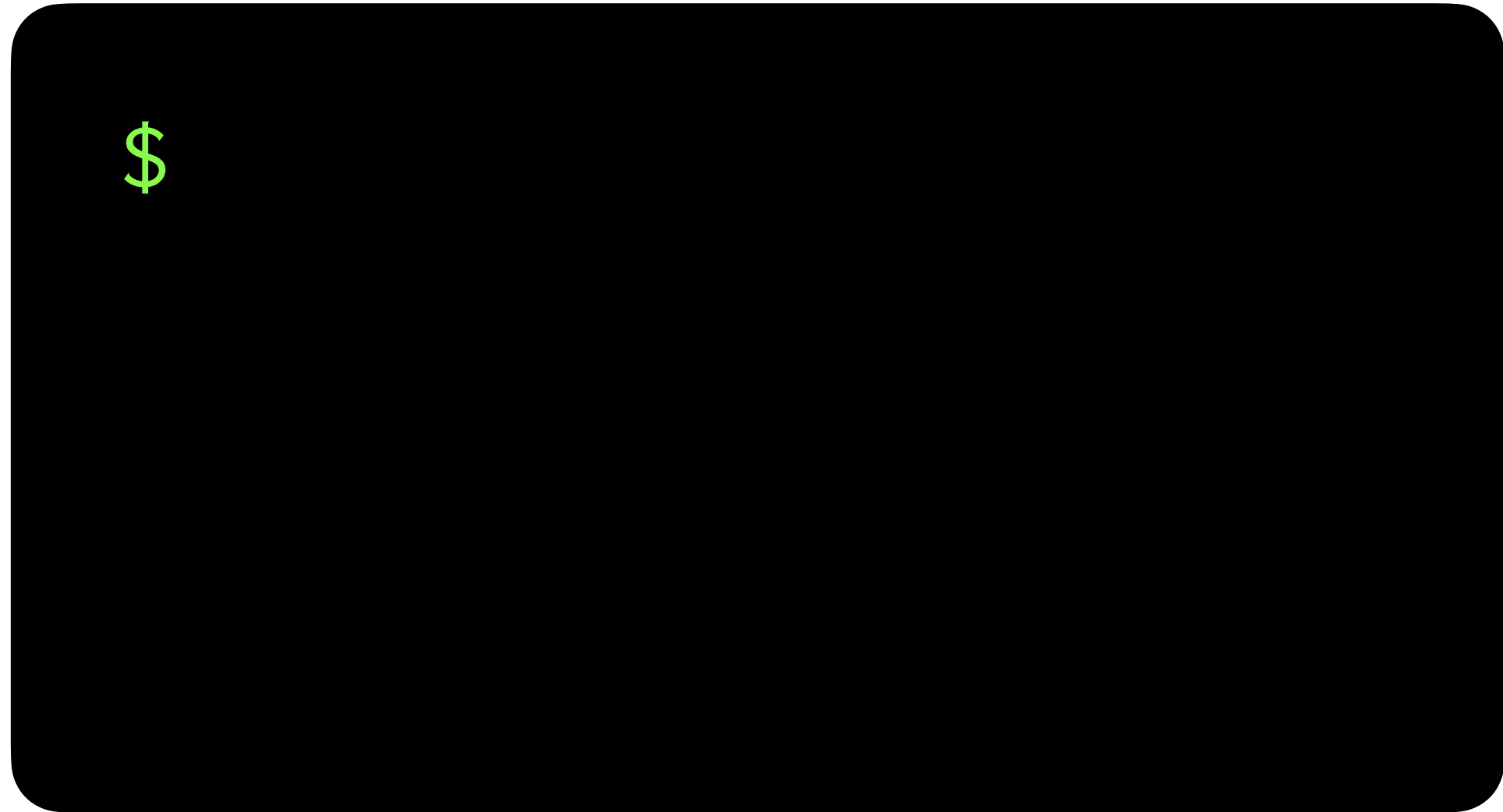
Home Directory



foo.py



bar.py



Home Directory



foo.py



bar.py

```
$ ls
```



Home Directory



foo.py



bar.py

```
$ ls  
foo.py bar.py
```



Home Directory



foo.py



bar.py


```
$ mkdir lab2
```



Home Directory

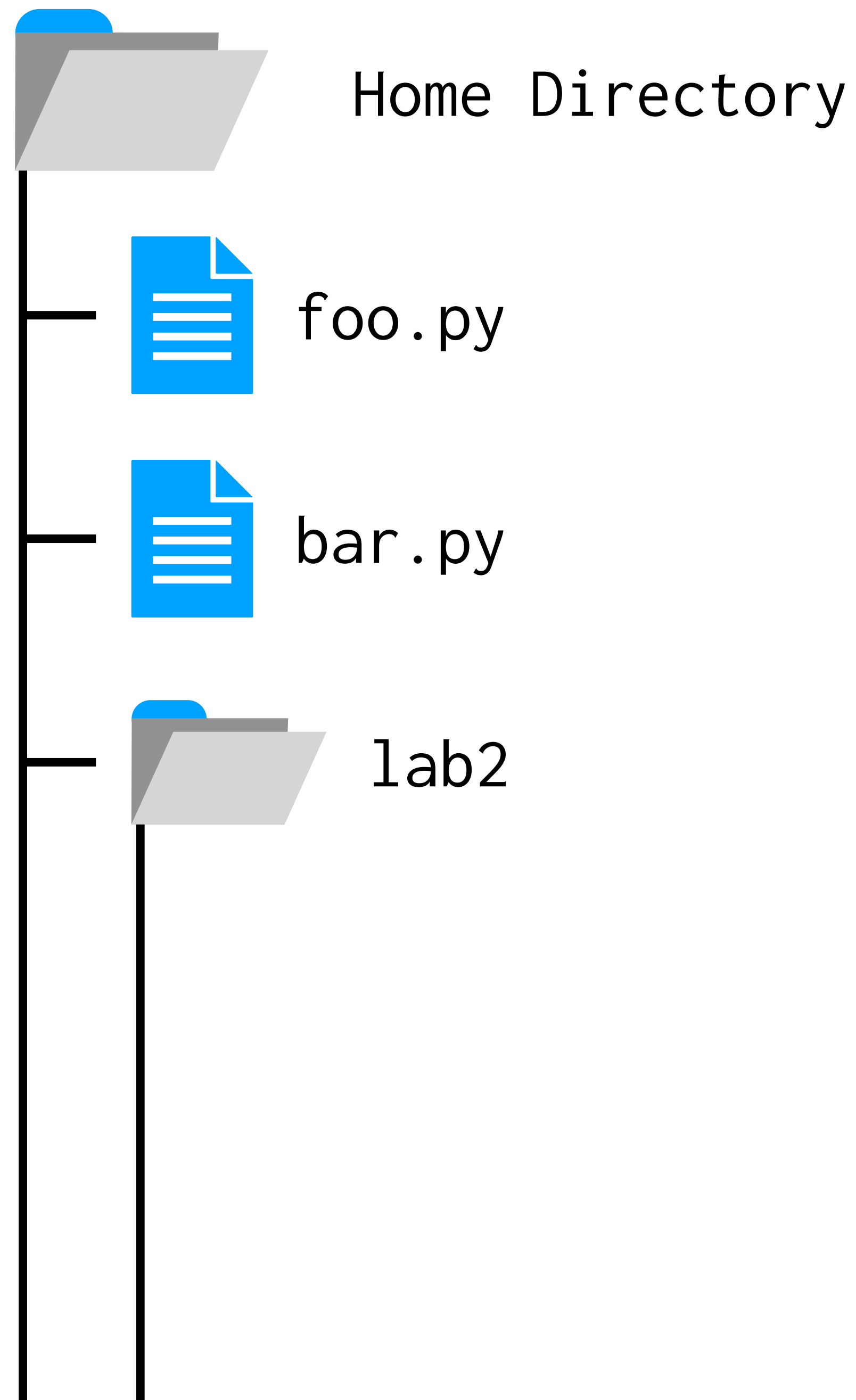


foo.py

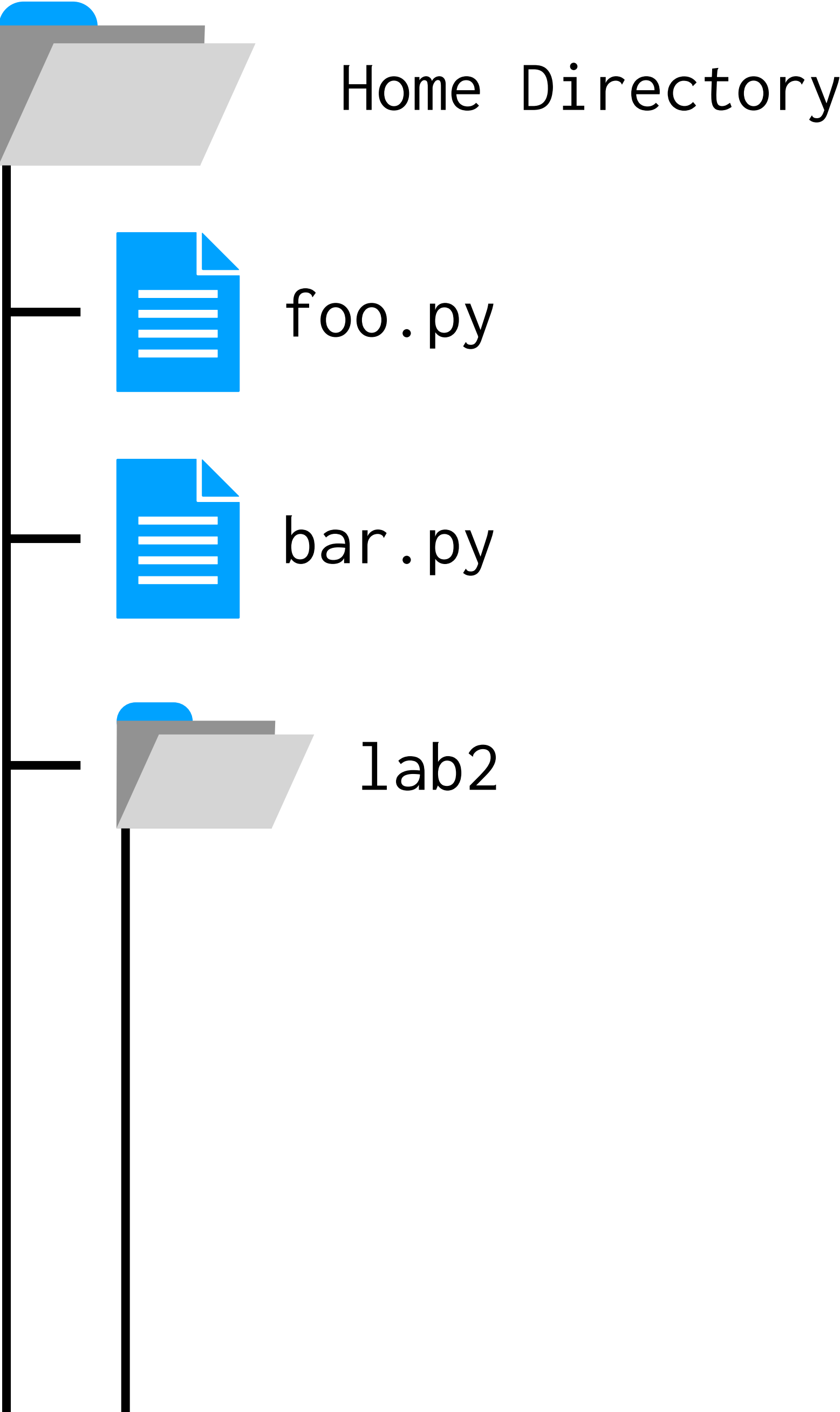


bar.py

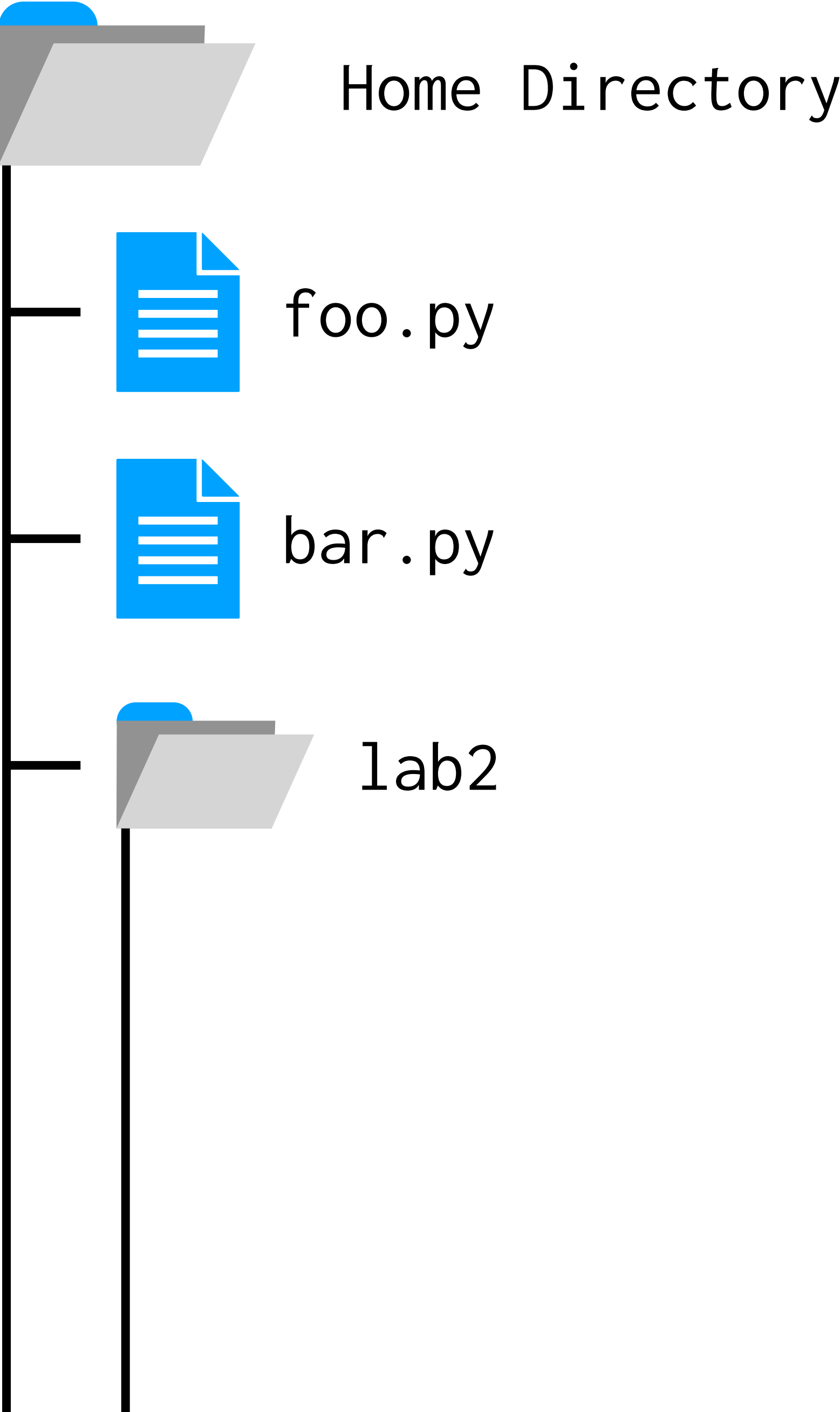
```
$ mkdir lab2
```



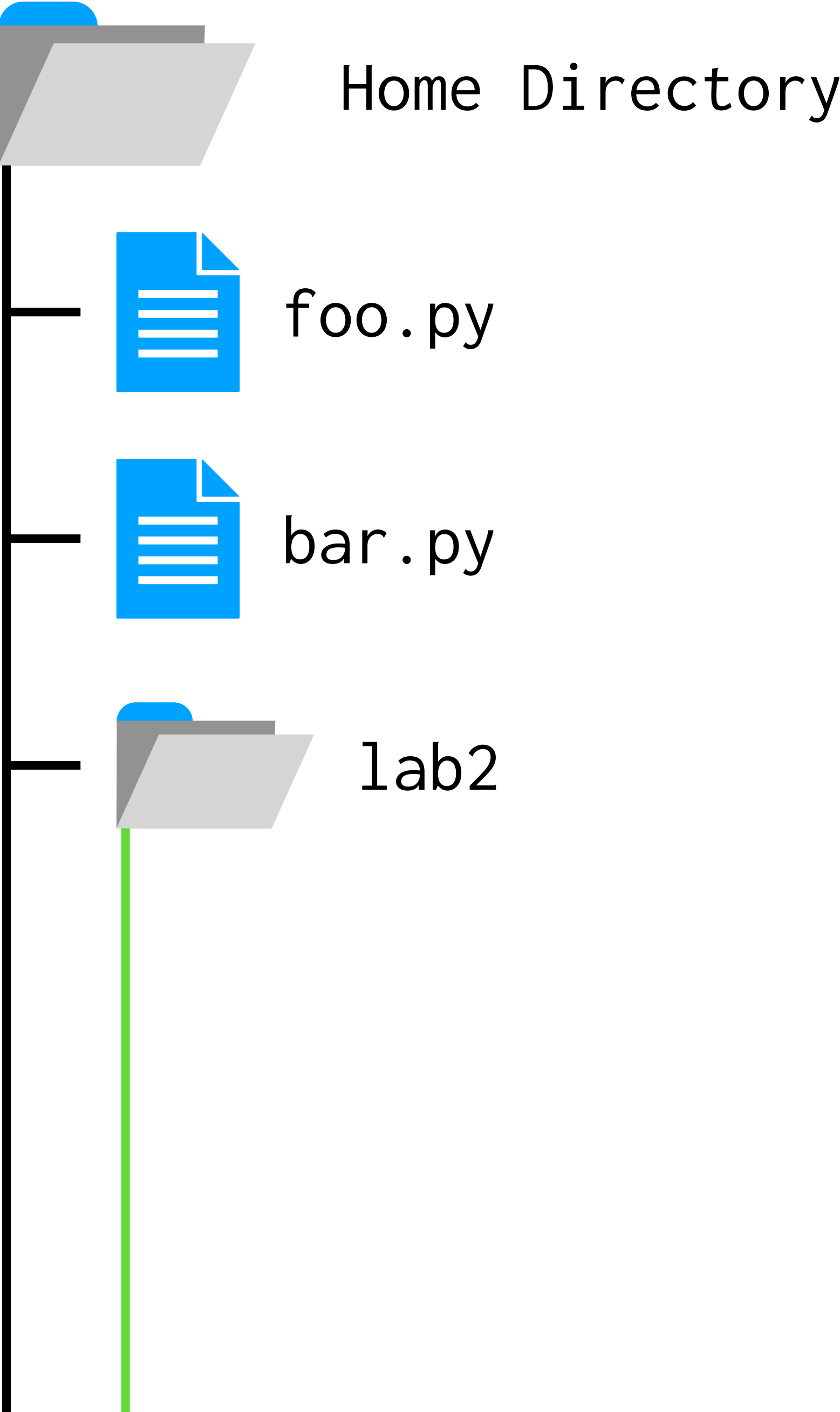
```
$ cd lab2
```



```
lab2/ $
```



```
lab2/ $
```



Common terminal commands

- `ls` List files in current "directory" (folder)
- `mkdir NAME` Make a new directory called NAME
- `cd NAME` Change directory to one called NAME
- `cd ..` Change directory to the folder above
- `code NAME` Open a file named NAME

Common terminal commands

- `ls` Lis files in current "directory" (folder)
- `mkdir NAME` Make a new directory called NAME
- `cd NAME` Change directory to one called NAME
- `cd ..` Change directory to the directory above
- `code NAME` Open a file named NAME



Readability

One fish, two fish, red fish, blue fish.

The literature on disciplinary identity emphasizes that identification with a discipline is best conceived not as a static trait, but as an evolving process. The process of identification occurs not solely within the student's mind, but in the interaction between the mind and the environment.

```
for c in text:  
    print(c)
```

"In the great
green room"

```
for c in text:  
    print(c)
```



"In the great
_
green room"

```
for c in text:  
    print(c)
```



"In the great

green room"

```
for c in text:  
    print(c)
```



"In the great
green room"

```
for c in text:  
    print(c)
```



"In the great
green room"

```
for c in text:  
    print(c)
```

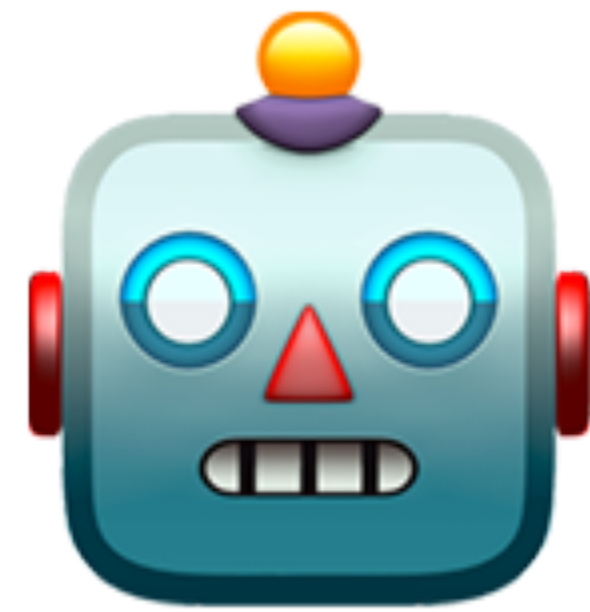


"In the great
green room"


```
for c in text:  
    print(c)
```



"In the great
green room"



Bot

hello, goodbye

- Define a **detect_goodbye** function that takes a variable, **text**, as input.
- Return **True** if "goodbye" is in **text**. Return **False** otherwise.
- Make your bot respond with "Goodbye!" in return!

Flags

`re.IGNORECASE`

ignore case while matching

HELLO, GOODBYE

- Define a **detect_goodbye** function that takes a variable, **text**, as input.
- Return **True** if "goodbye" is in **text**. Return **False** otherwise.
- Match "goodbye" **case-insensitively**.

Regular Expressions

```
re.search(r"hello", text)
```

text

"hello"

```
re.search(r"hello", text)
```

text

hello


```
re.search(r"hello", text)
```

text

"hellllloo"

Building Regular Expressions

<code>.</code>	match any character except a newline (<code>\n</code>)
<code>*</code>	match 0 or more repetitions
<code>+</code>	match 1 or more repetitions
<code>?</code>	match 0 or 1 repetition
<code>{m}</code>	match m repetitions
<code>{m,n}</code>	match m-n repetitions

```
re.search(r"hell+o", text)
```

text

"hellllo"

```
re.search(r"hell+o", text)
```

text

"hellllo"

```
re.search(r"hell+o", text)
```

text

hellllo

```
re.search(r"hell+o", text)
```

text

hellllo

```
re.search(r"hell+o", text)
```

text

hellllo

```
re.search(r"hell+o", text)
```

text

hellllo


```
re.search(r"hell+o", text)
```

text

hellllo

```
re.search(r"hell+o", text)
```

text

"hellllo"

Gooodgoodbye!

- Update your **detect_goodbye** function.
- Return **True** for each of the following:
 - "goodbye"
 - "gooodgoodbye"
 - "goodbyyyye"
 - "gooodbyyyye"

Building Regular Expressions

`[]` match any character in brackets

`[^]` match any character NOT in brackets

```
re.search(r"h[eu]llo", text)
```

text

"hullo"

```
re.search(r"h[eu]llo", text)
```

text

"hullo"

```
re.search(r"h[eu]llo", text)
```

text

hullo

```
re.search(r"h[eu]llo", text)
```

text

hullo

Building Regular Expressions

`()` a group of characters

`a|b` match a OR b

```
re.search(r"hello|i)", text)
```

text

"hello"

```
re.search(r"h(ello|i)", text)
```

text

"hello"

```
re.search(r"h(ello|i)", text)
```

text

hello

```
re.search(r"h(ello|i)", text)
```

text

"hi"

```
re.search(r"h(ello|i)", text)
```

text

hi

```
re.search(r"h(e|llo|i)", text)
```

text

hi

Goodday!

- Update your **detect_goodbye** function to accept other forms of saying goodbye!

Unit Tests

Capturing Input

```
match = re.search(r"I'm (.+)", text)
```

text

"I'm Carter"

```
match = re.search(r"I'm (.+)", text)
```

text

"I'm Carter"

```
match = re.search(r"I'm (.+)", text)
if matches:
    return matches.group(1)
```

text

"I'm Carter"

```
match = re.search(r"I'm (.+)", text)
if matches:
    return matches.group(1)
```

text

"I'm Carter"

matches.group(1)

"Carter"

Submission

- **Submit code files to Gradescope** by Thursday, February 9, 3:10 PM.
- Graded based on completion, but please double check to be sure your files are named correctly:
 - bot.py **not** bot (1).py
- Submit the lab feedback form