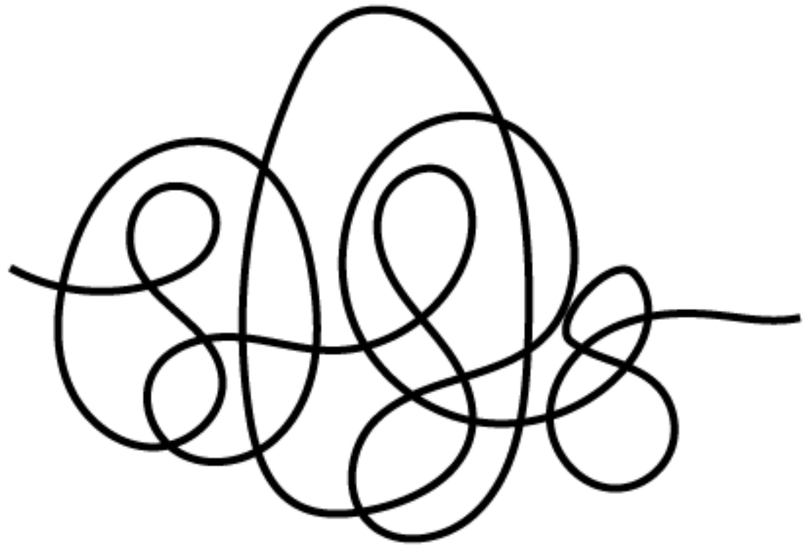
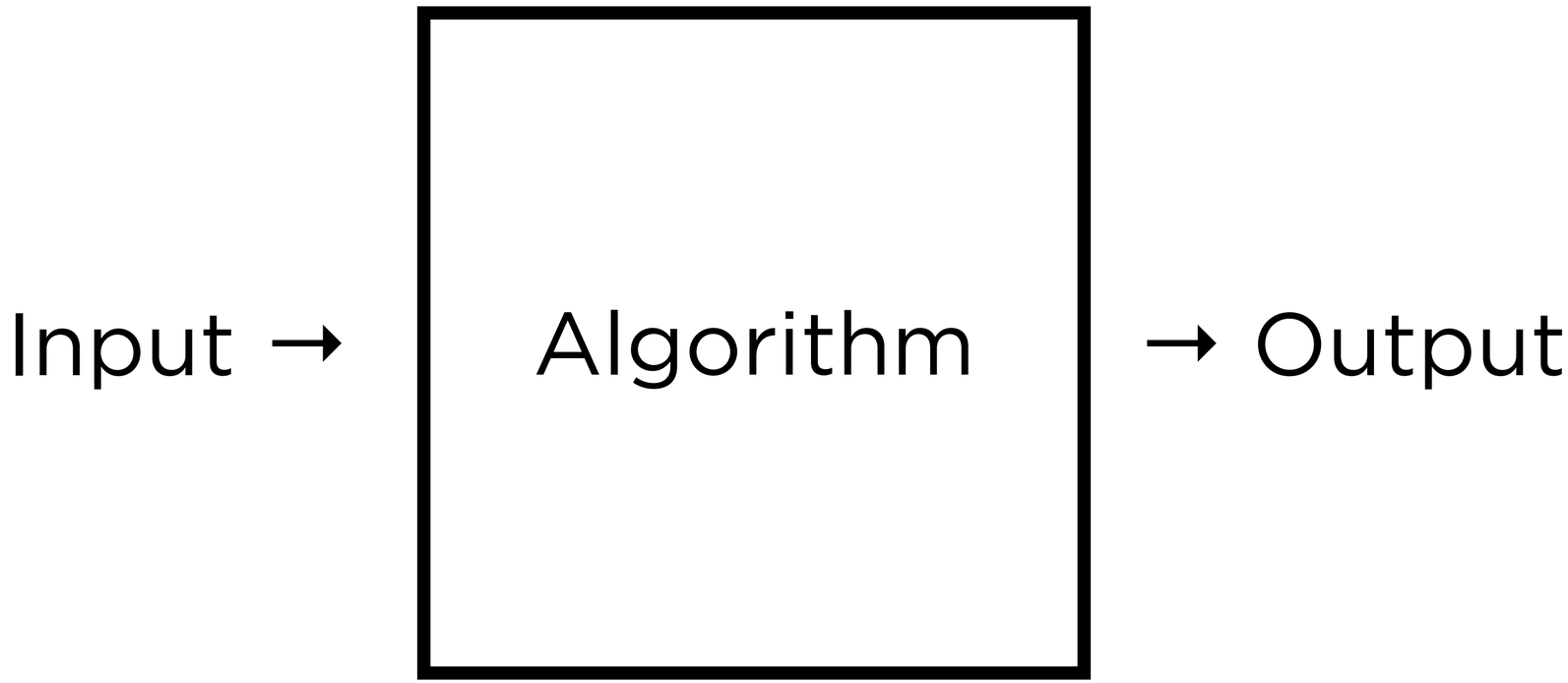


Lab 1

CS50 for MBAs

carterzenke.me/lab

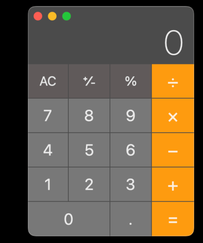

Input →
Algorithm
→ Output

- Work an example yourself
- Create an algorithm after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by testing it

- Work an example yourself
- Create an algorithm after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by testing it

Think, Pair, Share

- How would you explain a **function** to someone new to computer science?
- How would you describe a **variable** in a single sentence?
- We used our first **loops** and **conditionals** in `twttr.py`. How would you explain those to someone just joining the course?



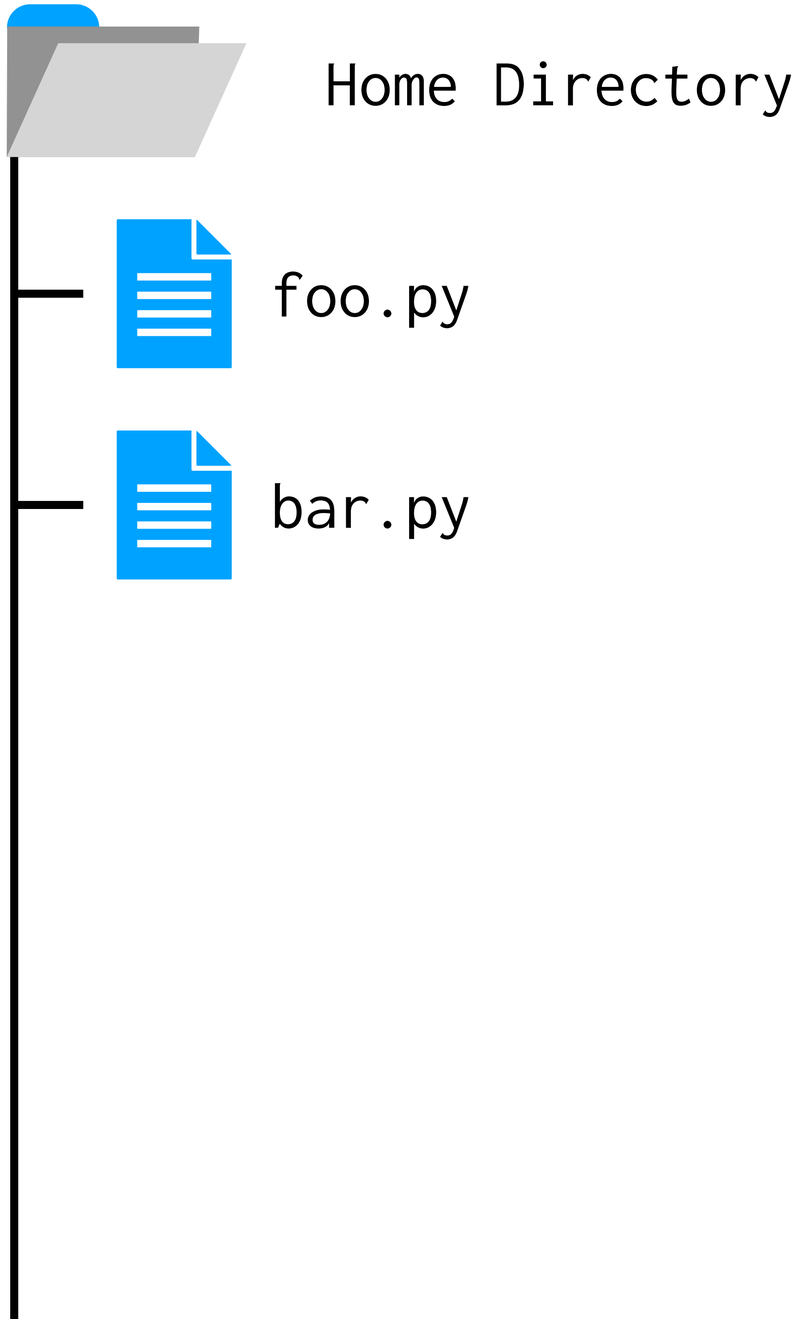
calculator.py

nutrition.py

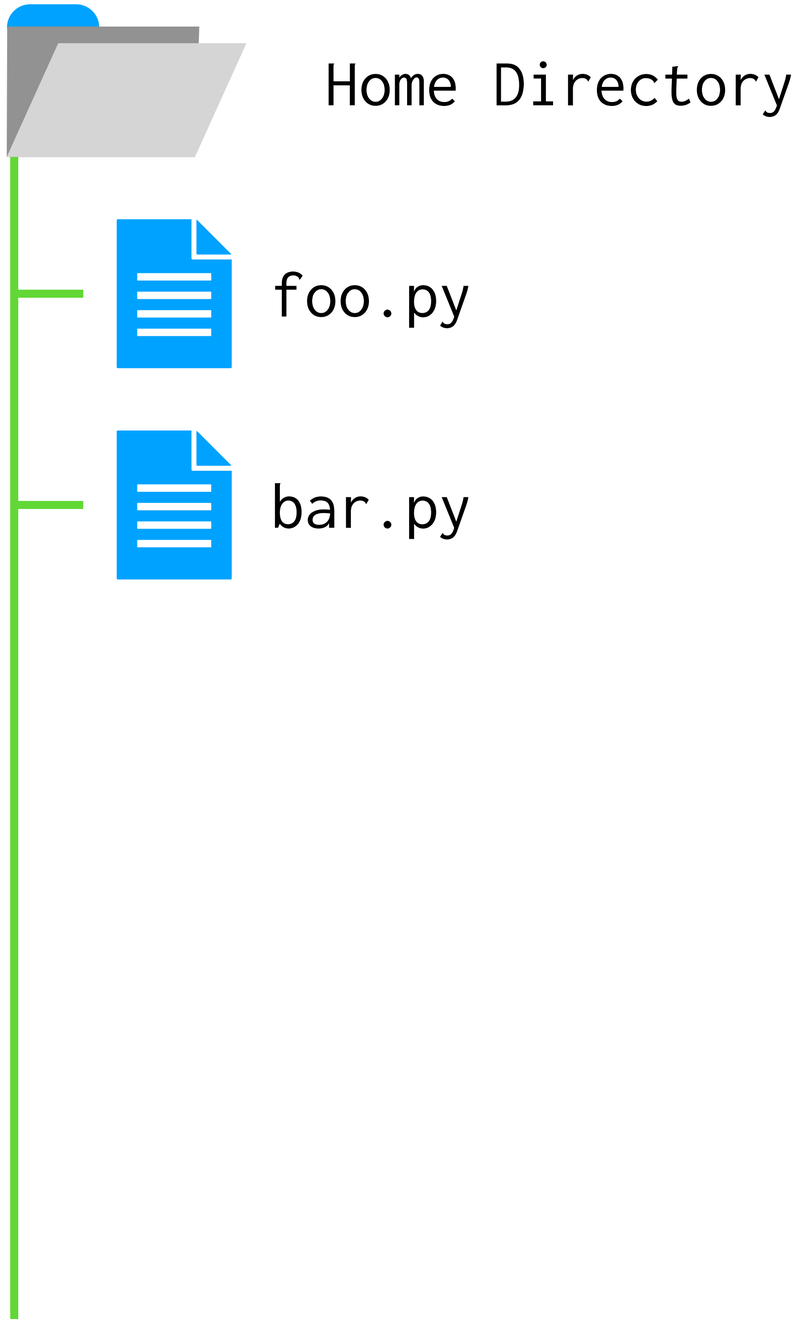
coke.py

cash.py

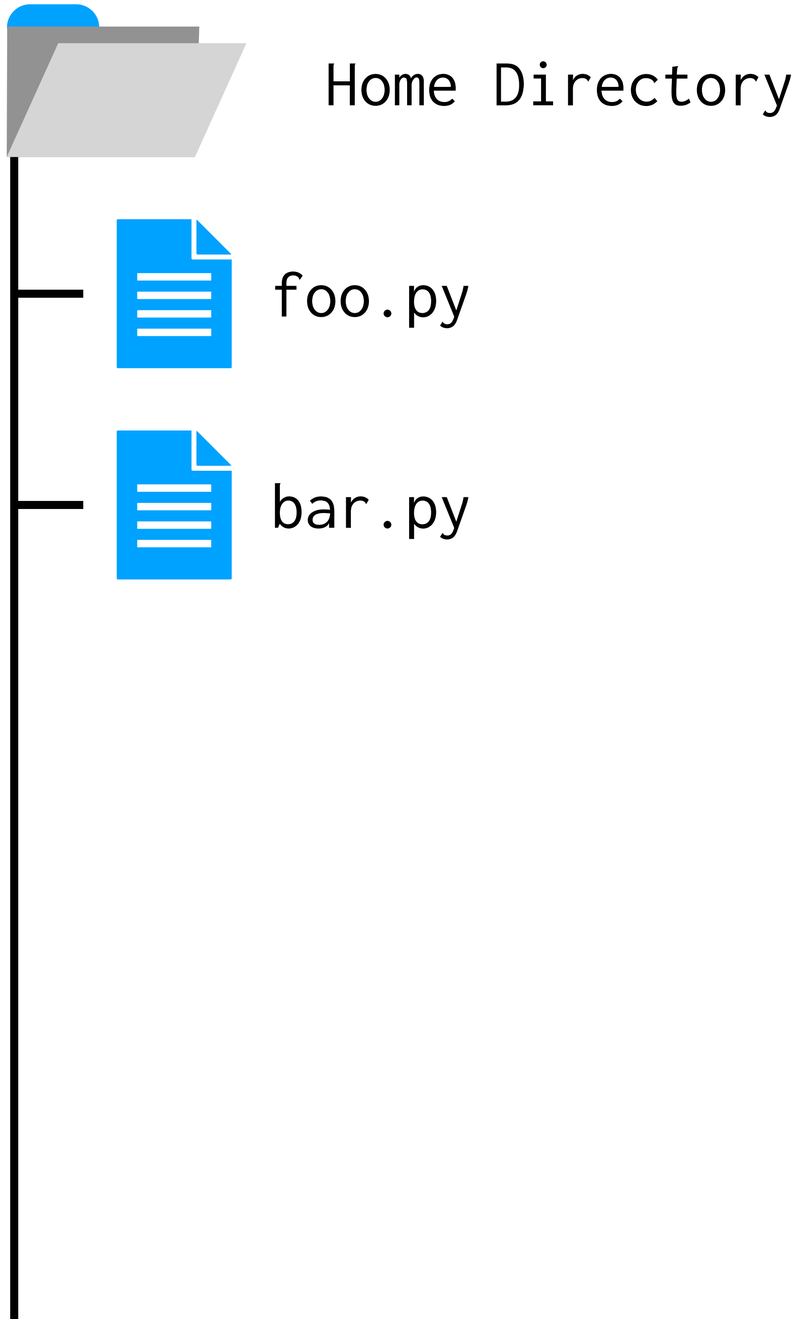

$


Home Directory
foo.py
bar.py

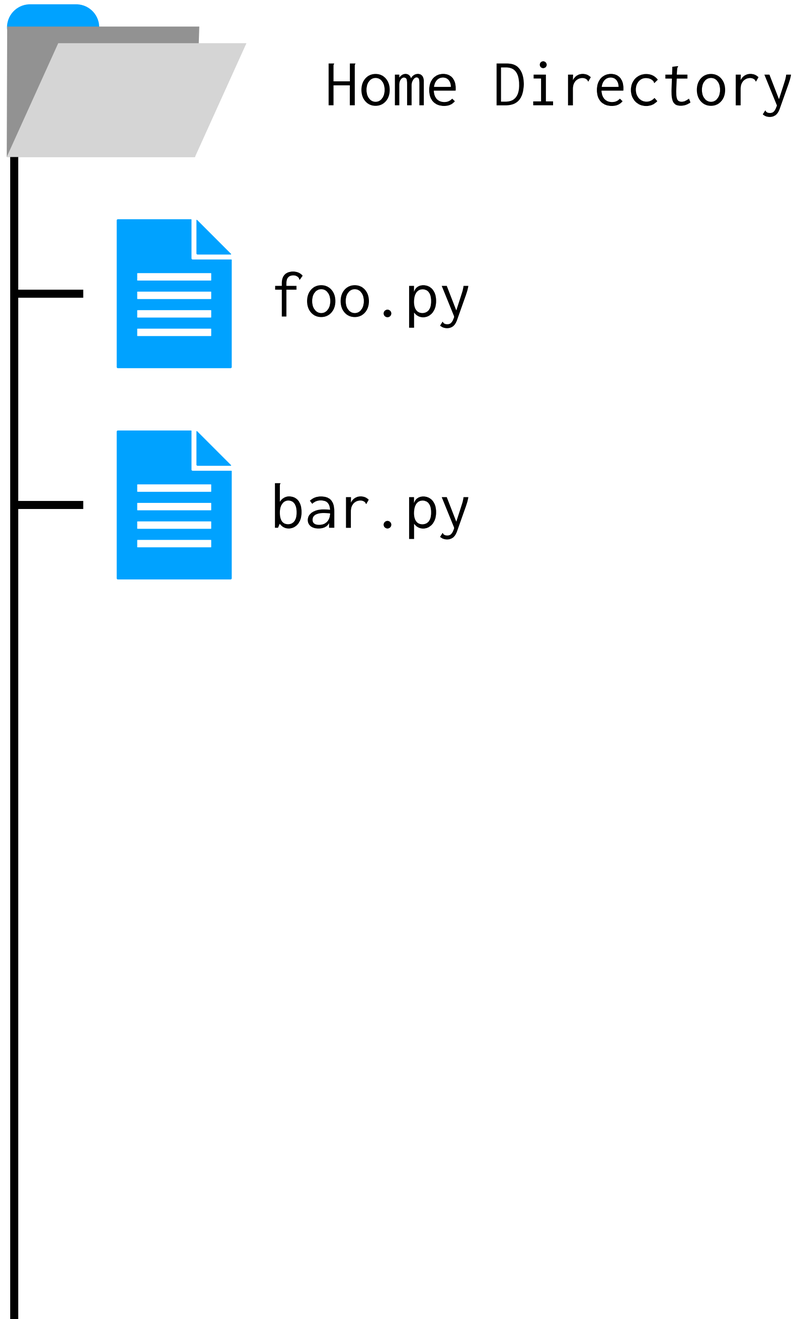

$


Home Directory
foo.py
bar.py

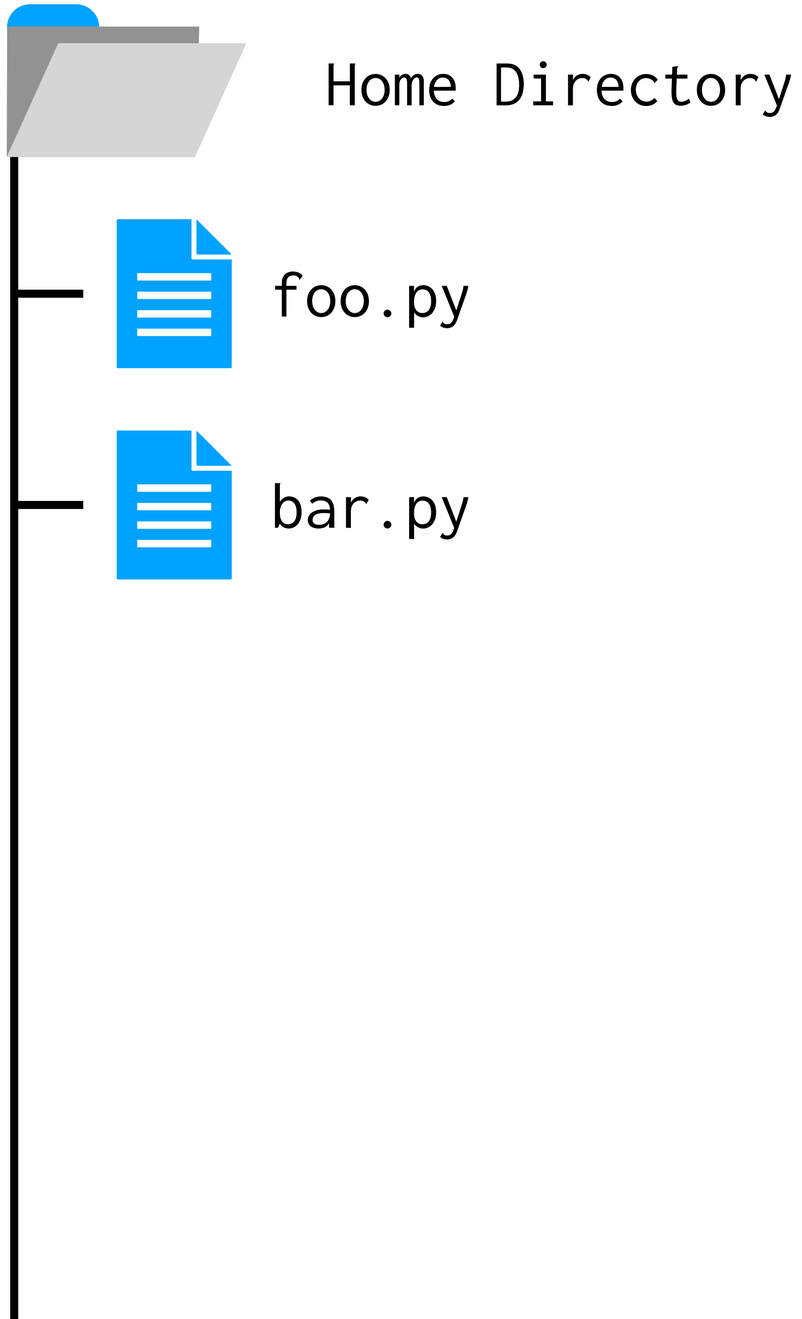
```
$ ls
```



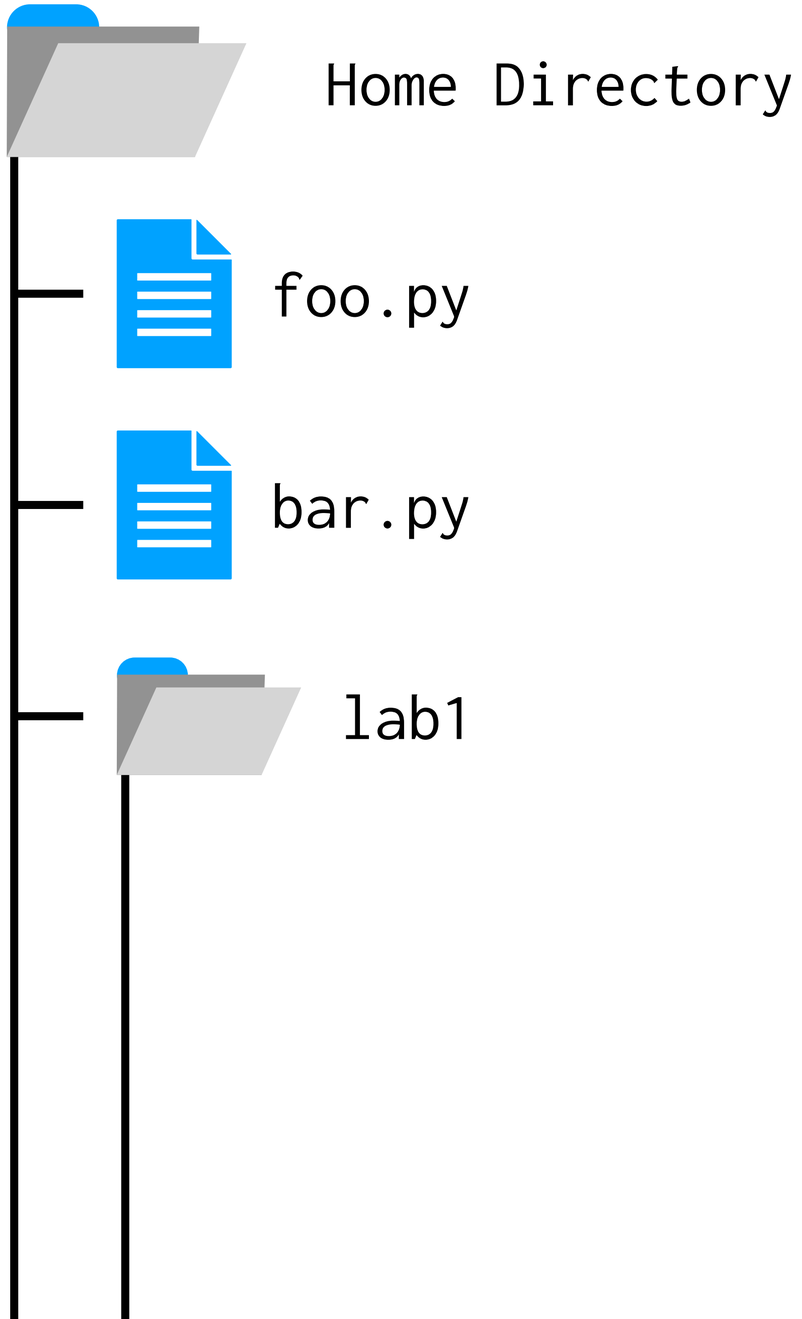
```
$ ls
foo.py  bar.py
```



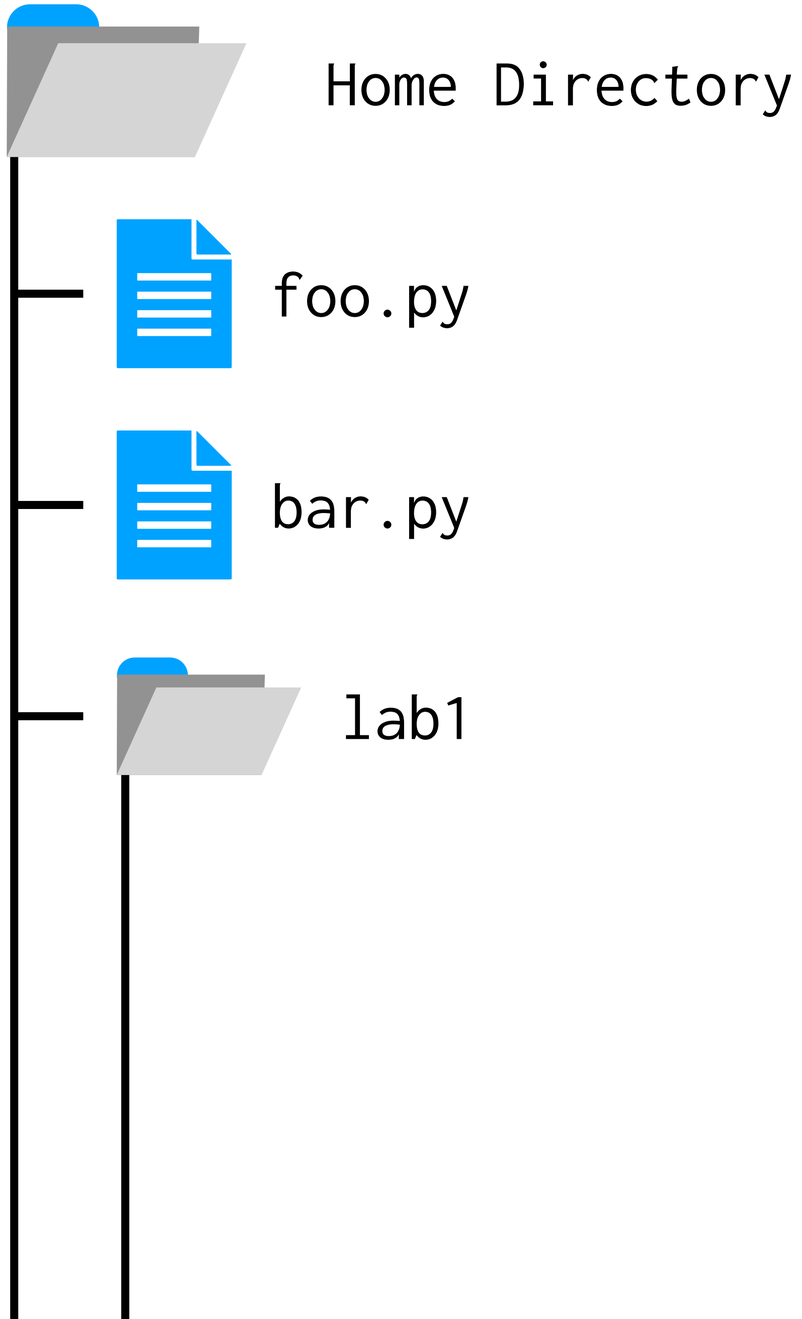
```
$  mkdir lab1
```



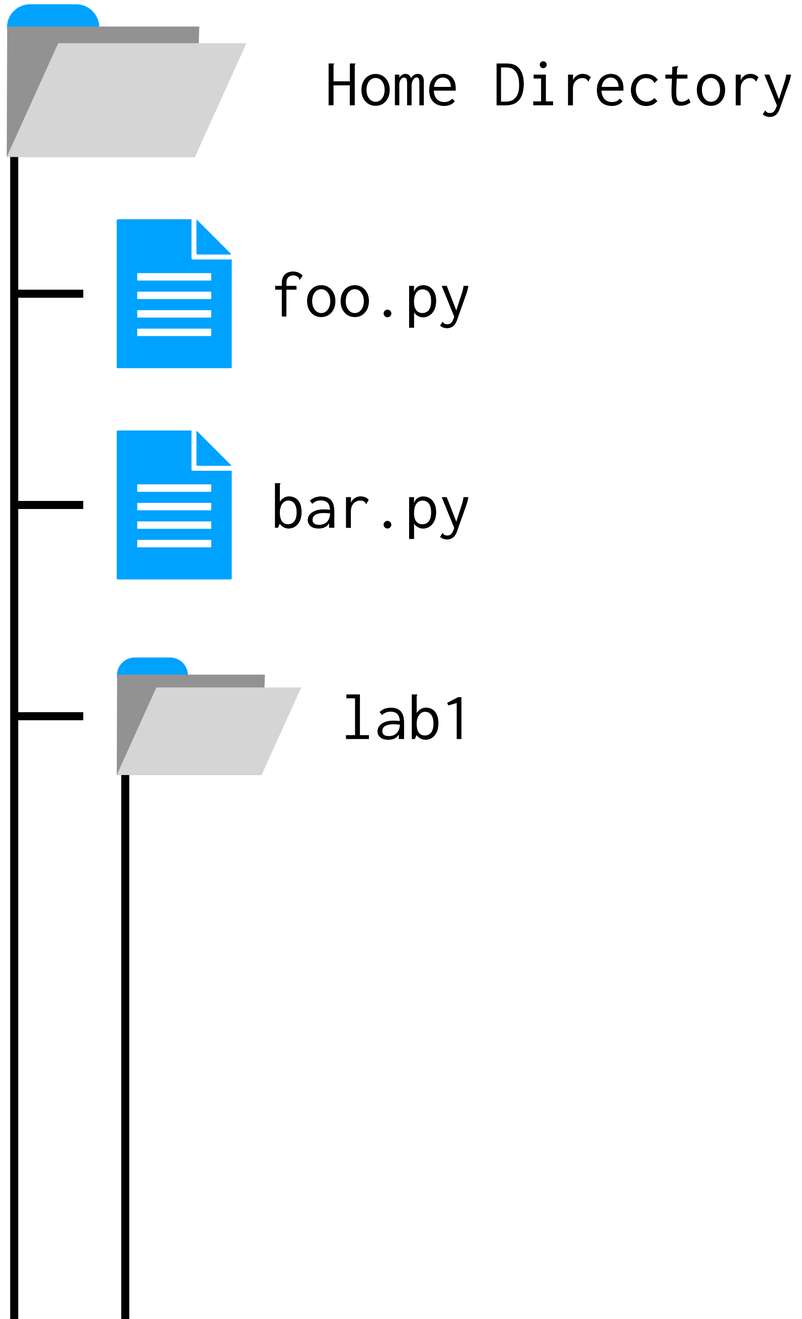
```
$  mkdir lab1
```



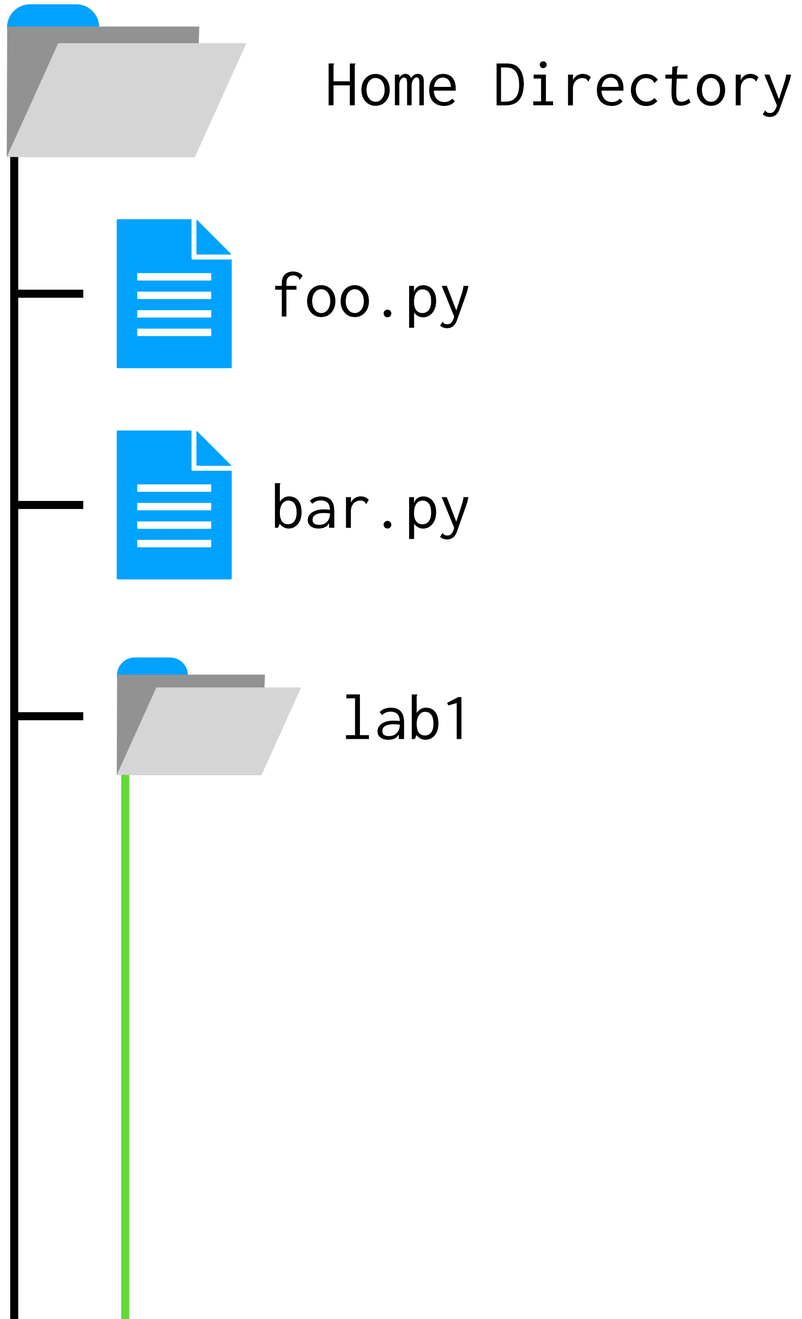
```
$  cd lab1
```



```
lab1/ $
```



```
lab1/ $
```

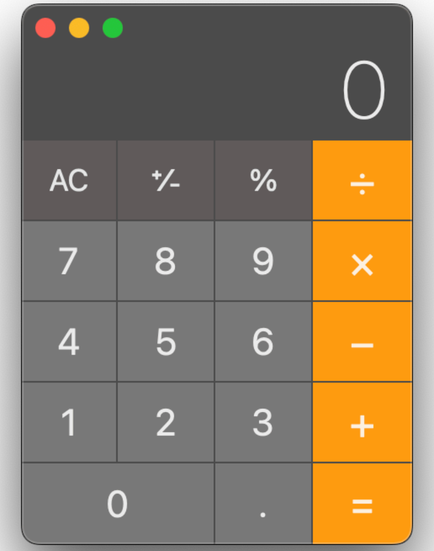


Common terminal commands

- `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` 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 directory above
- `code NAME` Open a file named NAME



Calculator

calculator.py

```
x = input("What's x? ")
y = input("What's y? ")

print(x + y)
```

```
What's x? 1
What's y? 2
12
```

calculator.py

```
x = int(input("What's x? "))
y = int(input("What's y? "))

print(x + y)
```

```
What's x? 1
What's y? 2
3
```

```
$ python calculator.py
What's x? cat

Traceback (most recent call last)
...
ValueError: invalid literal int() with base 10:'cat'
```

```
$ python calculator.py
What's x? cat

Traceback (most recent call last)
...
ValueError: invalid literal int() with base 10:'cat'
```

Exceptions

```
try:
    ...
except ...:
    ...
```

```
try:
    x = int(input("What's x? "))
except ...:
    ...
```

```
try:
    x = int(input("What's x? "))
except ValueError:
    ...
```

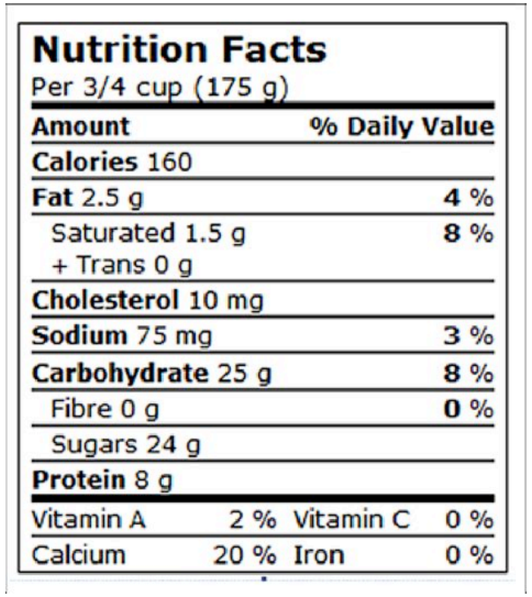
```
try:
    x = int(input("What's x? "))
except ValueError:
    x = 0
```

```
while True:
    try:
        x = int(input("What's x? "))
    except ValueError:
        continue
```

```
while True:
    try:
        x = int(input("What's x? "))
        break
    except ValueError:
        continue
```

Loop Controls

- `continue` Move to the next cycle (iteration) of loop
- `break` Exit the loop entirely



Nutrition Facts

Per 3/4 cup (175 g)

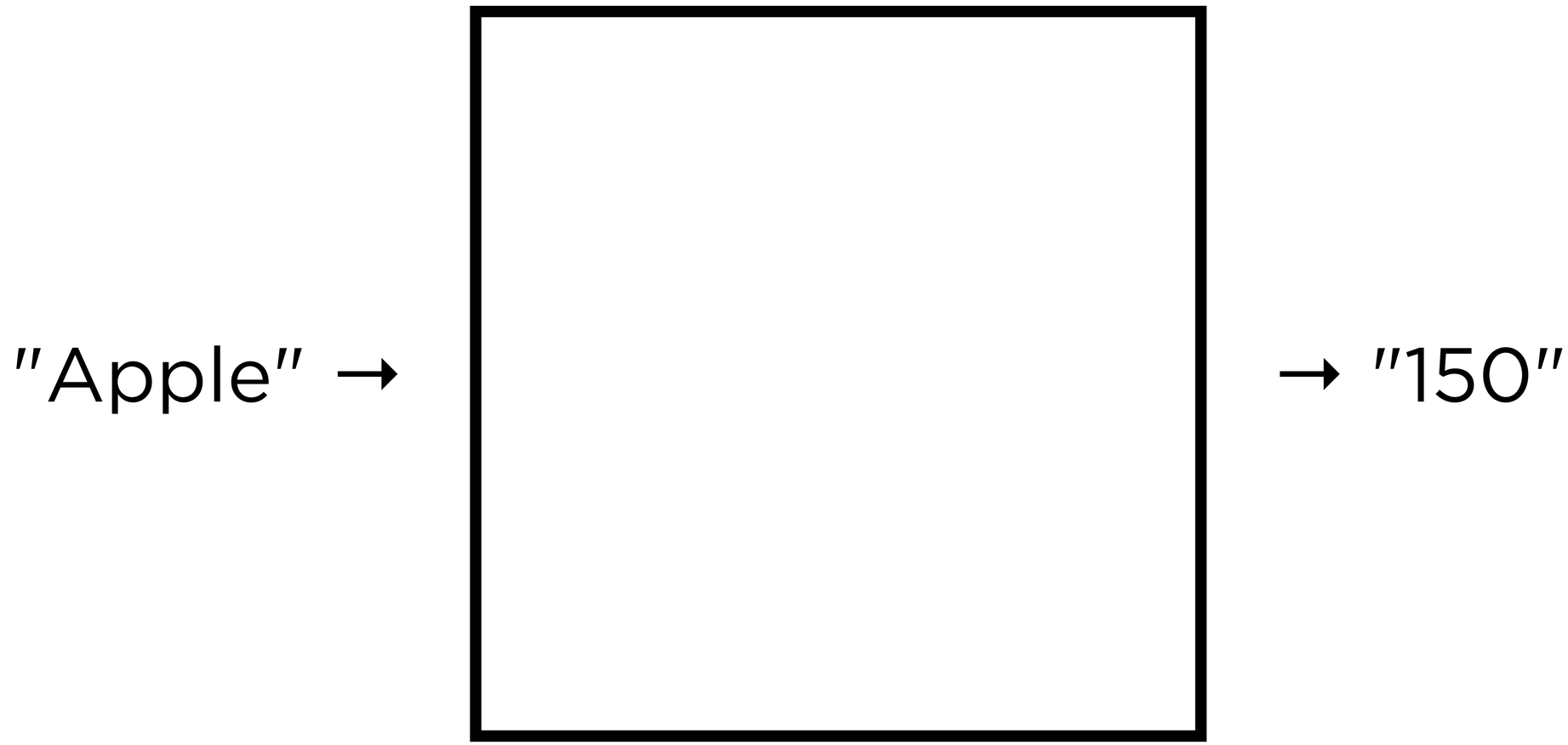
Amount	% Daily Value
Calories 160	
Fat 2.5 g	**4** %
Saturated 1.5 g + Trans 0 g	**8** %
Cholesterol 10 mg	
Sodium 75 mg	**3** %
Carbohydrate 25 g	**8** %
Fibre 0 g	**0** %
Sugars 24 g	
Protein 8 g	

Vitamin A	2 %	Vitamin C	0 %
Calcium	20 %	Iron	0 %

Nutrition

```
$ python nutrition.py
Strawberries
Calories: 50
```

```
$ python nutrition.py
Apple
Calories: 130
```


"Apple" →
→ "150"

Dictionaries

```
fruits = {
    "apple": 130,
    "strawberries": 50
}
```

Key	Value
apple	130
strawberries	50

```
fruits = {
    "apple": 130,
    "strawberries": 50
}

fruits["strawberries"]
```

Key	Value
apple	130
strawberries	50

```
fruits = {
    "apple": 130,
    "strawberries": 50
}

fruits["apple"]
```

Key	Value
apple	130
strawberries	50

Exceptions

```
fruits = {
    "apple": 130,
    "strawberries": 50
}

fruits["chocolate"]
```

Key	Value
apple	130
strawberries	50

```
fruits = {
    "apple": 130,
    "strawberries": 50
}

fruits["chocolate"]
```

Key	Value
apple	130
strawberries	50

KeyError

```
fruits = {
    "apple": 130,
    "strawberries": 50
}

try:
    fruits["chocolate"]
except KeyError:
    print("Not here!")
```

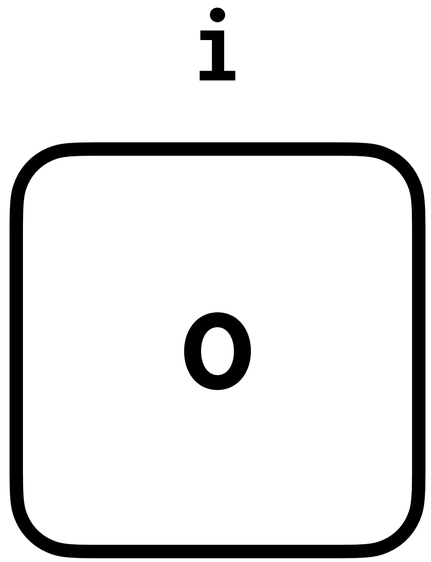
Key	Value
apple	130
strawberries	50

Coke Machine

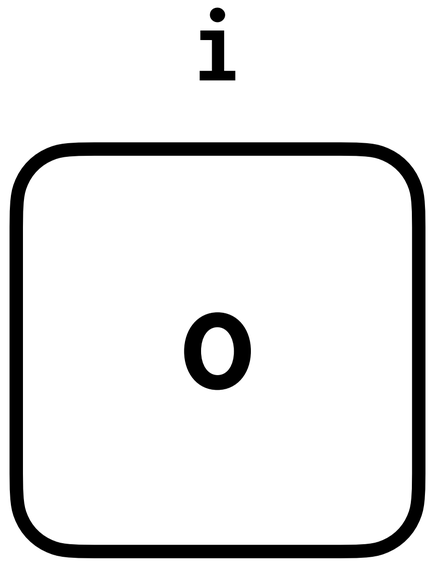
While Loops

```
i = 0
while i < 3:
    i += 1
```

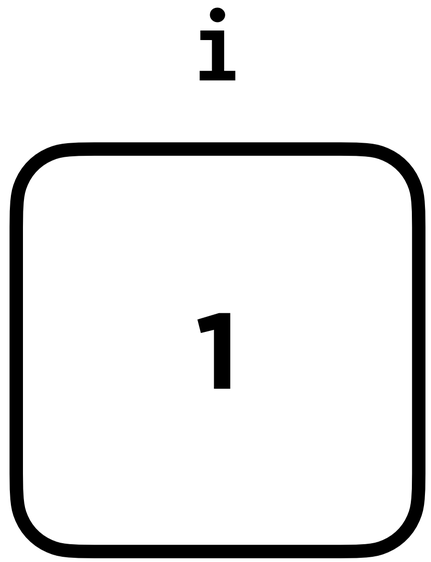
```
i = 0
while i < 3:
    i += 1
```



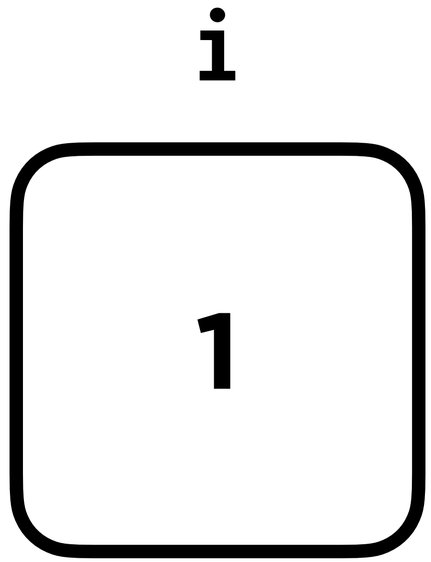
```
i = 0
while i < 3:
    i += 1
```



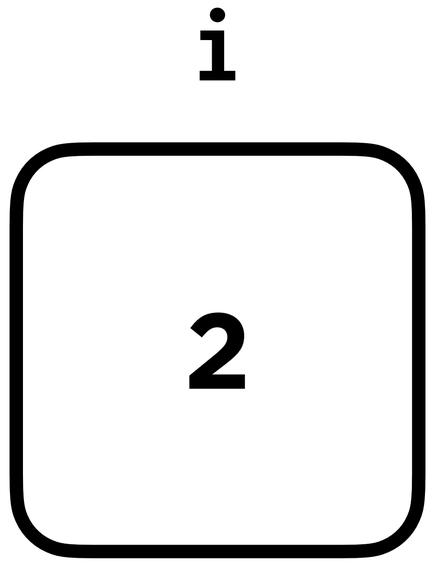
```
i = 0
while i < 3:
    i += 1
```



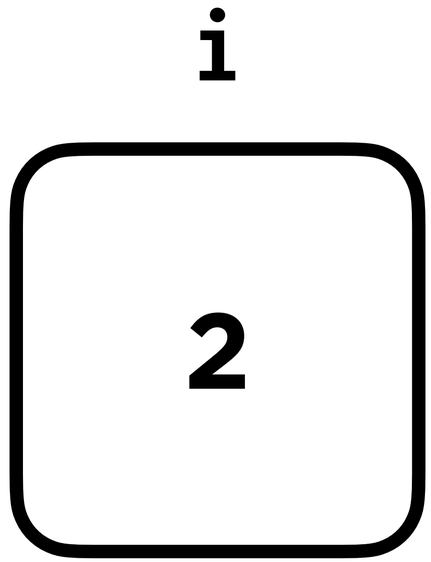
```
i = 0
while i < 3:
    i += 1
```



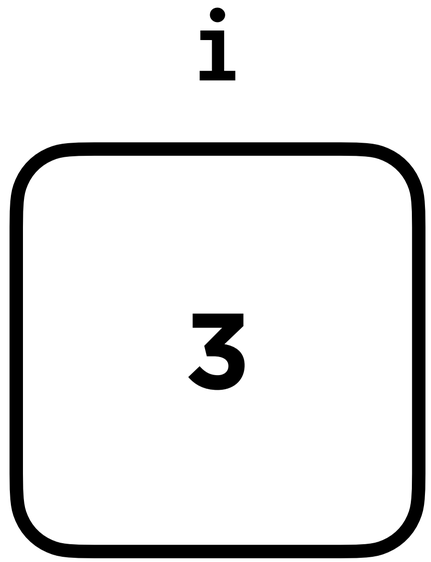
```
i = 0
while i < 3:
    i += 1
```



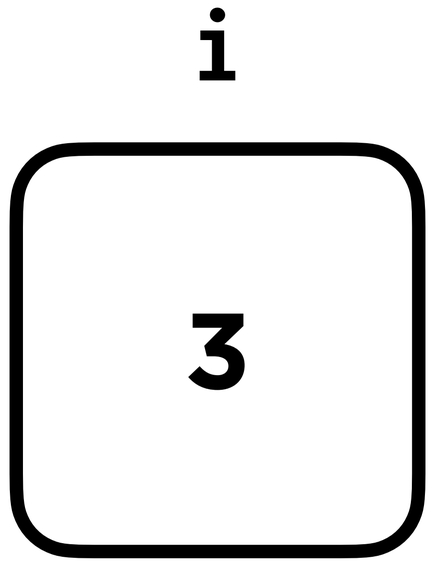
```
i = 0
while i < 3:
    i += 1
```



```
i = 0
while i < 3:
    i += 1
```



```
i = 0
while i < 3:
    i += 1
```



Pseudocode

```
# While amount owed is > 0
    # Accept coin from user
    # Check if valid coin
        # Subtract coin from amount owed
```

Cash


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR

25¢


LIBERTY
IN GOD WE TRUST
2005
P

10¢


GOD WE TRUST
LIBERTY
2004
S

5¢


IN GOD WE TRUST
LIBERTY
2005
D

1¢

$1.00


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
S
QUARTER DOLLAR

25¢


LIBERTY
IN GOD WE TRUST
2005
P

10¢


LIBERTY
IN GOD WE TRUST
2004
S

5¢


IN GOD WE TRUST
LIBERTY
2005
D

1¢

$0.95


UNITED STATES OF AMERICA
IN GOD WE TRUST
LIBERTY
QUARTER DOLLAR
LIBERTY
IN GOD WE TRUST
2005
GOD WE TRUST
LIBERTY 2004

25¢ 10¢ 5¢ 1¢

Submission

- **Submit code files to Gradescope** by Friday, February 3, 3:10 PM.
- Graded based on completion, but please double check to be sure your files are named correctly:
  - `coke.py` **not** `coke (1).py`