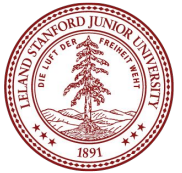
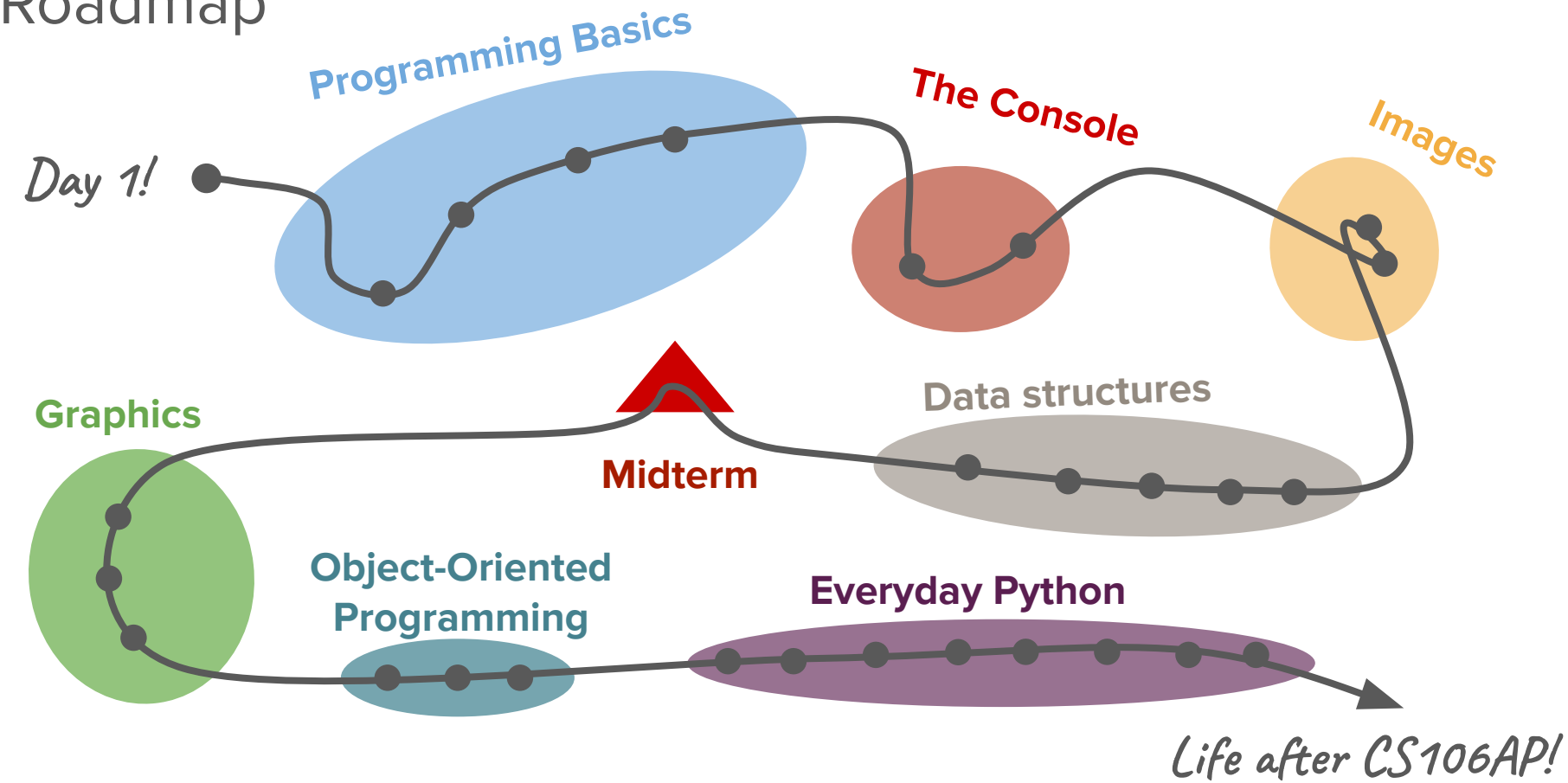
Files and the Command Line

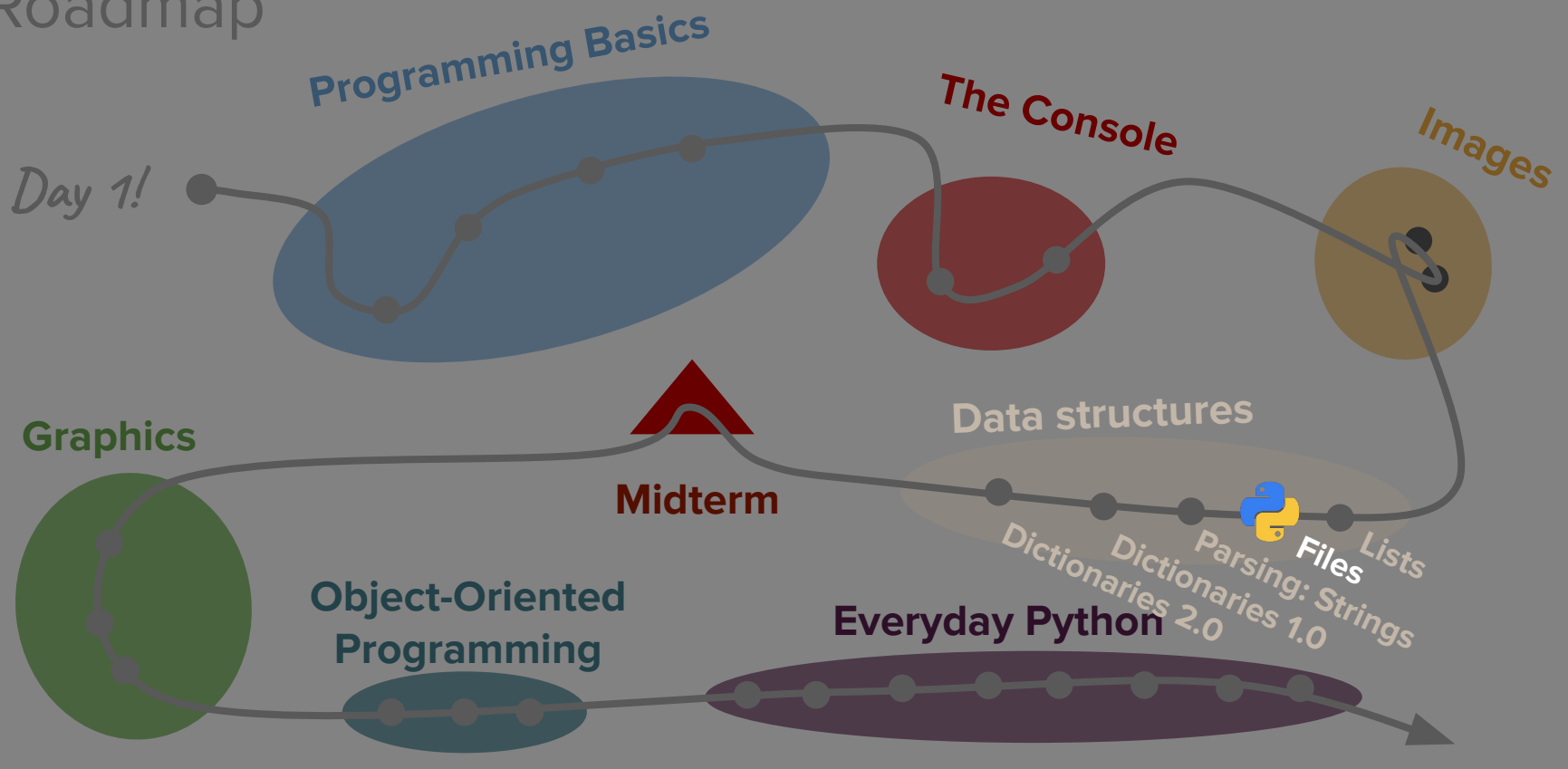
CS106AP Lecture 11



Roadmap


Programming Basics
Day 1!
The Console
Images
Graphics
Midterm
Data structures
Object-Oriented Programming
Everyday Python
Life after CS106AP!

Roadmap


Programming Basics
Day 1!
The Console
Images
Graphics
Midterm
Data structures
Files
Lists
Parsing: Strings
Dictionaries 1.0
Dictionaries 2.0
Object-Oriented Programming
Everyday Python
Life after CS106AP!

Today's questions

How do computers store and organize data?

How can I give instructions to my computer outside of a program?

Today's topics

1. Review
2. File Reading
3. The Command Line

   Arguments
4. What's next?

Review

What is a list?

```
[1, 2, 3, 4, 5]

['a', 'b', 'b', 'd']

[True]

[1, 'a', 2, 'b', True]

[]
```

Definition

List

A data type for storing values in a linear collection.

How to inspect a list

```
>>> letters = ['a', 'b', 'c', 'd']

>>> letters[0]

'a'

>>> letters[1:]

['b', 'c', 'd']
```

Other general functions

```
>>> letters = ['a', 'b', 'c', 'd']

>>> len(letters)

4

>>> print(letters)

['a', 'b', 'c', 'd']
```

How can I change what's in a list?

```
>>> lst = [1, 2, 3, 4, 5]

>>> lst.append(6)

>>> lst

[1, 2, 3, 4, 5, 6]

>>> lst += [7, 8]

[1, 2, 3, 4, 5, 6, 7, 8]
```

How can I remove something from a list?

```
>>> lst = [1, 2, 3, 4, 5]
>>> last_elem = lst.pop()
>>> last_elem
5
```

pop() removes the last element in a list and returns it. You can also pass an index into pop().

How can I check if something's in a list?

```
>>> fruits = ['apple', 'banana', 'mango', 'kiwi']

>>> 'mango' in fruits

True

>>> 'broccoli' in fruits

False

>>> 'broccoli' not in fruits
```

How can I loop over a list?

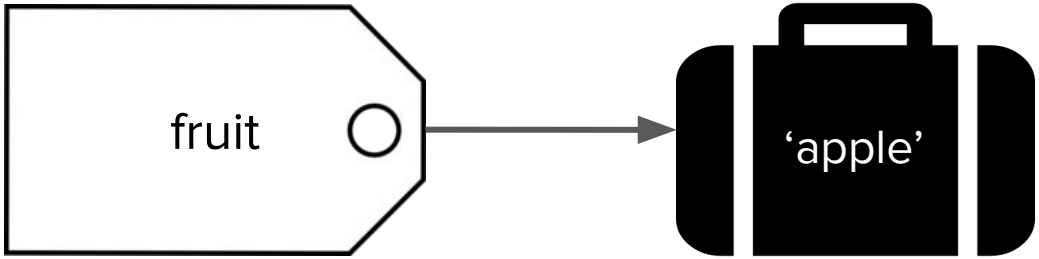
```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
```

like a foreach loop over a string!

How can I loop over a list?

```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
```

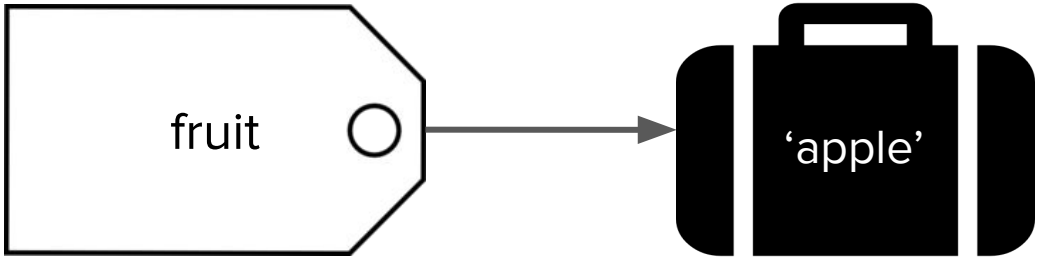
like a foreach loop over a string!



How can I loop over a list?

```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
apple
```

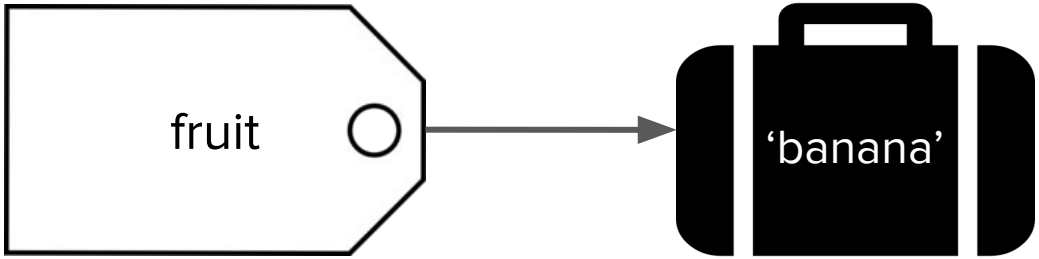
like a foreach loop over a string!



How can I loop over a list?

```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
apple
```

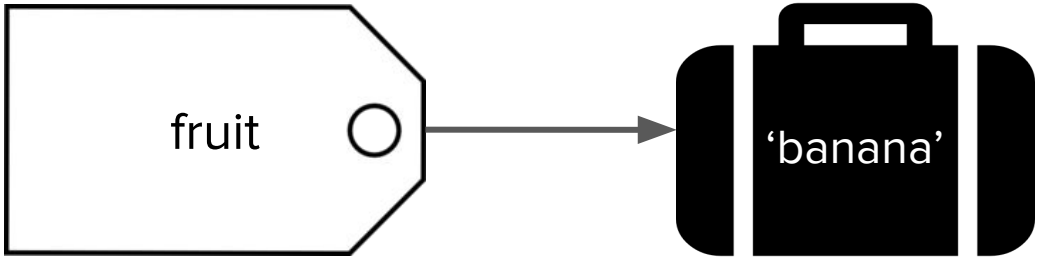
like a foreach loop over a string!



How can I loop over a list?

```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
apple
banana
```

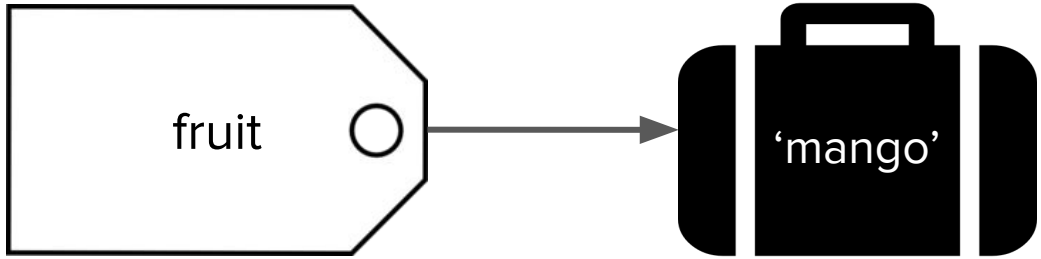
like a foreach loop over a string!



How can I loop over a list?

```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
apple
banana
```

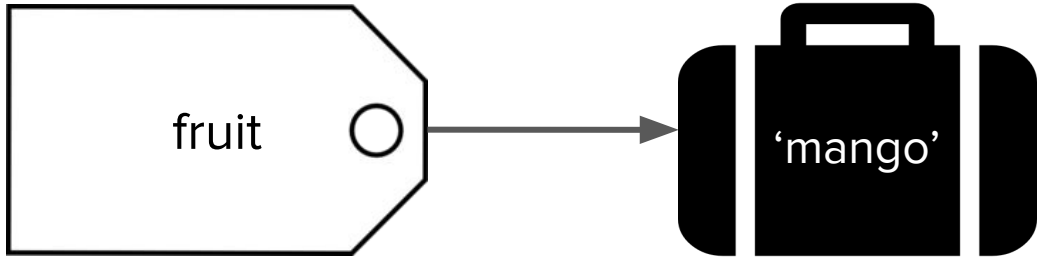
like a foreach loop over a string!



How can I loop over a list?

```
>>> fruits = ['apple', 'banana', 'mango']
>>> for fruit in fruits:
...     print(fruit)
apple
banana
mango
```

like a foreach loop over a string!



Making a list from a string

```
>>> s = 'I am comprised of words'
>>> words = s.split()
>>> words
['I', 'am', 'comprised', 'of', 'words']
>>> s
'I am comprised of words'
```

you can also pass in a delimiter, which says where to split the string

Advanced For Range Loops and Slicing

```
>>> for i in range(4, 0, -1):
```

```
>>> lst = [1, 2, 3, 4]
>>> lst[::-1]
[4, 3, 2, 1]
```

(start_index, end_index, step)

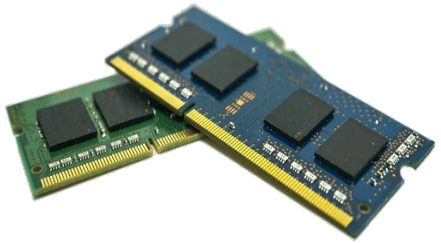
How do computers store and organize data?

Files!

But first… a note on storage

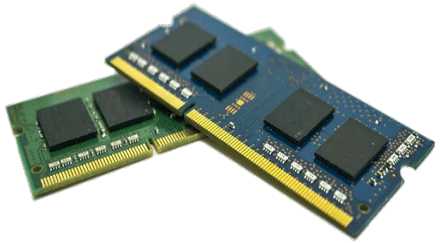
But first... a note on storage

When we're running a program, variables and information are stored on RAM (Random Access Memory)



But first... a note on storage

When we're running a program, variables and information are stored on RAM (Random Access Memory)



When we're not running a program and we want to save information, we store it on our hard drive (also called disk)

File Reading

Virtually all programs that you've used **read files** from disk at some point:

File Reading

Virtually all programs that you've used **read files** from disk at some point:

- Word processing (documents)

File Reading

Virtually all programs that you've used **read files** from disk at some point:

- Word processing (documents)
- Web browser (cookies)

File Reading

Virtually all programs that you've used **read files** from disk at some point:

- Word processing (documents)
- Web browser (cookies)
- Games (saved progress)

File Reading

Virtually all programs that you've used **read files** from disk at some point:

- Word processing (documents)
- Web browser (cookies)
- Games (saved progress)
- PyCharm (Python files)

File Reading

Virtually all programs that you've used **read files** from disk at some point:

- Word processing (documents)
- Web browser (cookies)
- Games (saved progress)
- PyCharm (Python files)
- Music player (songs)

File Reading

A file is a series of **bits** (ones and zeros).

- in plain text, bits represent characters
- in JPEGs, bits encode information about the structure of an image
- in MP3 files, bits encode frequency information

File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```

File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```


```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```

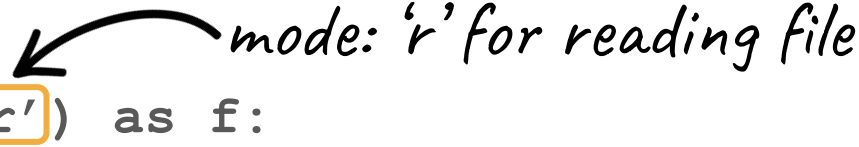
File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```


```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```

File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

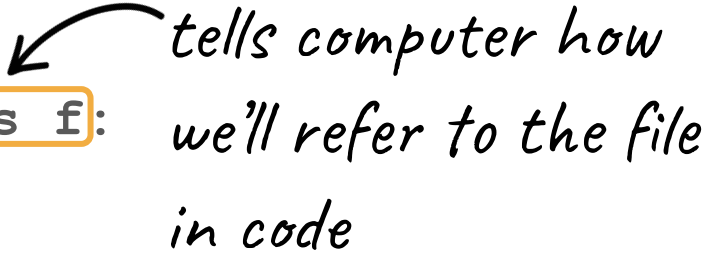



```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```

File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```



File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

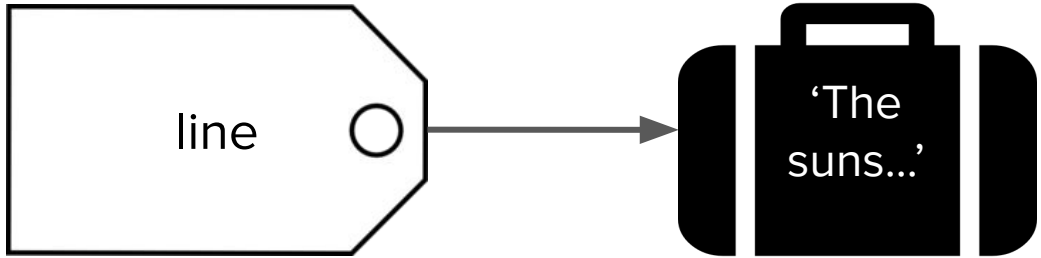
```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```

we can loop through files, just like strings or lists!

File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

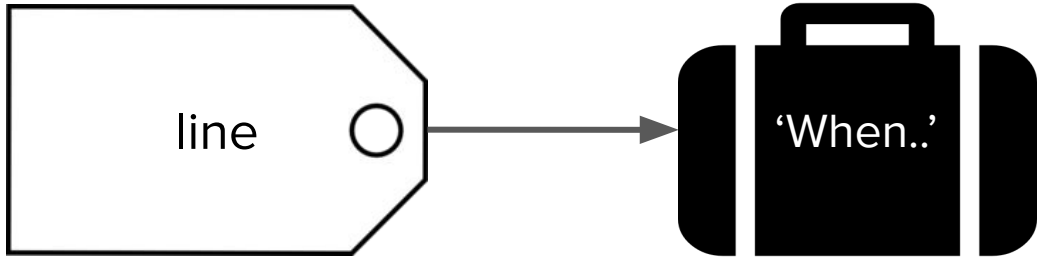
```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```



File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

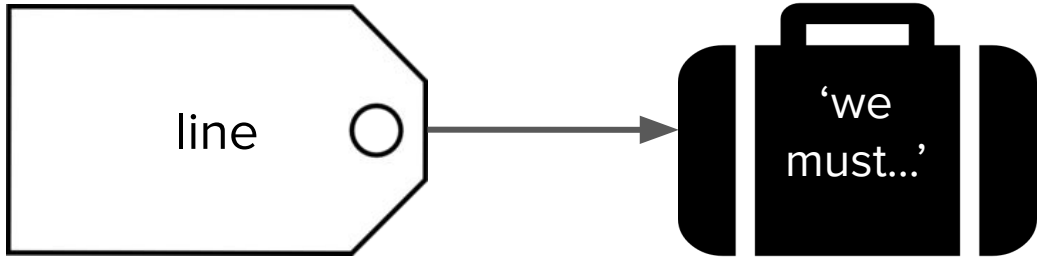
```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```



File Reading – catullus.txt

```
0 The suns are able to fall and rise:
1 When that brief light has fallen for us,
2 we must sleep a never ending night.
```

```
with open('catullus.txt', 'r') as f:
    for line in f:
        print(line)
```



Think/Pair/Share:

Write a function that returns the longest word contained in a file.

(hint: decompose! find the longest word in a single line)

How can I give instructions to my computer outside of a program?

Command Line

Definition

Command Line/Terminal
Text interface for giving instructions to the computer. These instructions are relayed to the computer's operating system.

Command Line

Definition

Command Line/Terminal

Text interface for giving instructions to the computer. These instructions are relayed to the computer's operating system.

Definition

Python Console/Interpreter

An interactive program that allows us to write Python code and run it line-by-line.

Command Line

Definition

Command Line/Terminal

Text interface for giving instructions to the computer. These instructions are relayed to the computer's operating system.

specific to Python

Definition

Python Console/Interpreter

An interactive program that allows us to write Python code and run it line-by-line.

Command Line

Definition

not specific to Python

Command Line/Terminal

Text interface for giving instructions to the computer. These instructions are relayed to the computer's operating system.

Definition

Python Console/Interpreter

An interactive program that allows us to write Python code and run it line-by-line.

Command Line

you can run the Python interpreter from the command line!

Definition

Command Line/Terminal

Text interface for giving instructions to the computer. These instructions are relayed to the computer's operating system.

Definition

Python Console/Interpreter

An interactive program that allows us to write Python code and run it line-by-line.

Command Line

In the command line, you can:

- run Python scripts
- navigate to different directories
- copy, move, and delete files
- and more!

Command Line

In the command line, you can:

- **run Python scripts**
- navigate to different directories
- copy, move, and delete files
- and more!

Command Line Usage

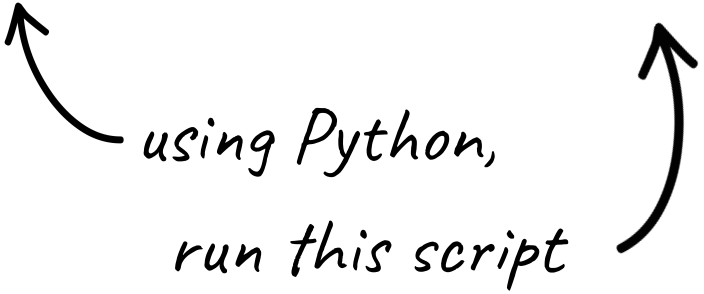
```
python3 script_name.py
```

Command Line Usage

```
python3 script_name.py
```

Command Line Usage

```
python3 script_name.py
```



Think/Pair/Share:

Write a program that prints text with certain characters removed.

Inputs:

filename, chars to remove (string)

Command Line Usage

```
python3 DeleteCharacters.py
```

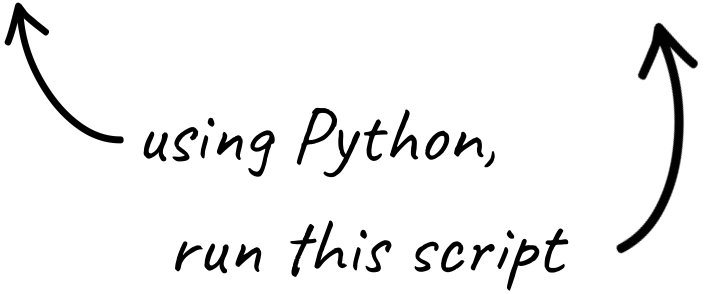
How can we easily change what file/characters we're running our script on?

Command Line Usage

```
python3 script_name.py
```

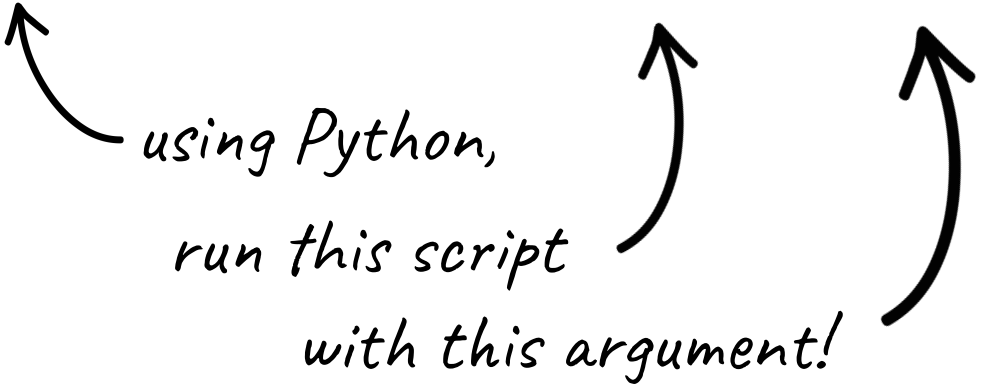
Command Line Usage

```
python3 script_name.py
```



Command Line Usage

```
python3 script_name.py filename.txt
```



Arguments

```
python3 script_name.py filename.txt
```

this is called an argument.

Arguments

```
python3 script_name.py filename.txt
```

it's an additional piece of information we're passing to our program.

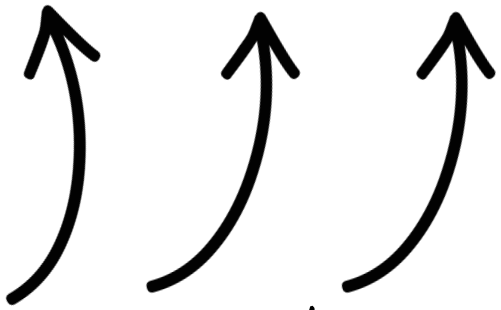
Arguments

```
python3 script_name.py filename.txt
```

we can use it in our code!

Arguments

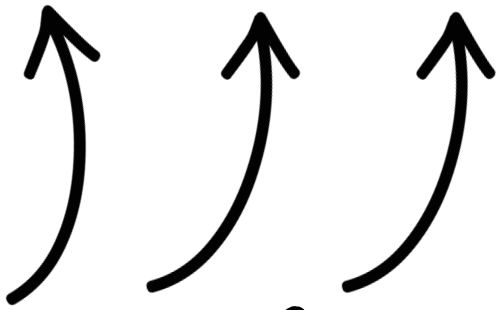
```
python3 DeleteCharacters.py -chars aei poem.txt
```



you can have multiple arguments!

Arguments

```
python3 DeleteCharacters.py -chars aei poem.txt
```



how do we use them in our code?

[DEMO]

Arguments

```
def main():
    args = sys.argv[1:]
    if len(args) == 1:
        filename = args[0]
```

Arguments

```
def main():
    args = sys.argv[1:]
    if len(args) == 1:
        filename = args[0]
```

we use this syntax to get a list of string arguments (that we wrote on the command line)

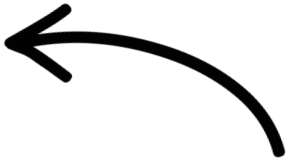
Arguments

```
def main():
    args = sys.argv[1:]
    if len(args) == 1:
        filename = args[0]
```

Common pattern: checking for length of args

Arguments

```
def main():
    args = sys.argv[1:]
    if len(args) == 1:
        filename = args[0]
```



Now we use that info!

Think/Pair/Share:

Write a program that counts the number of words in a file.

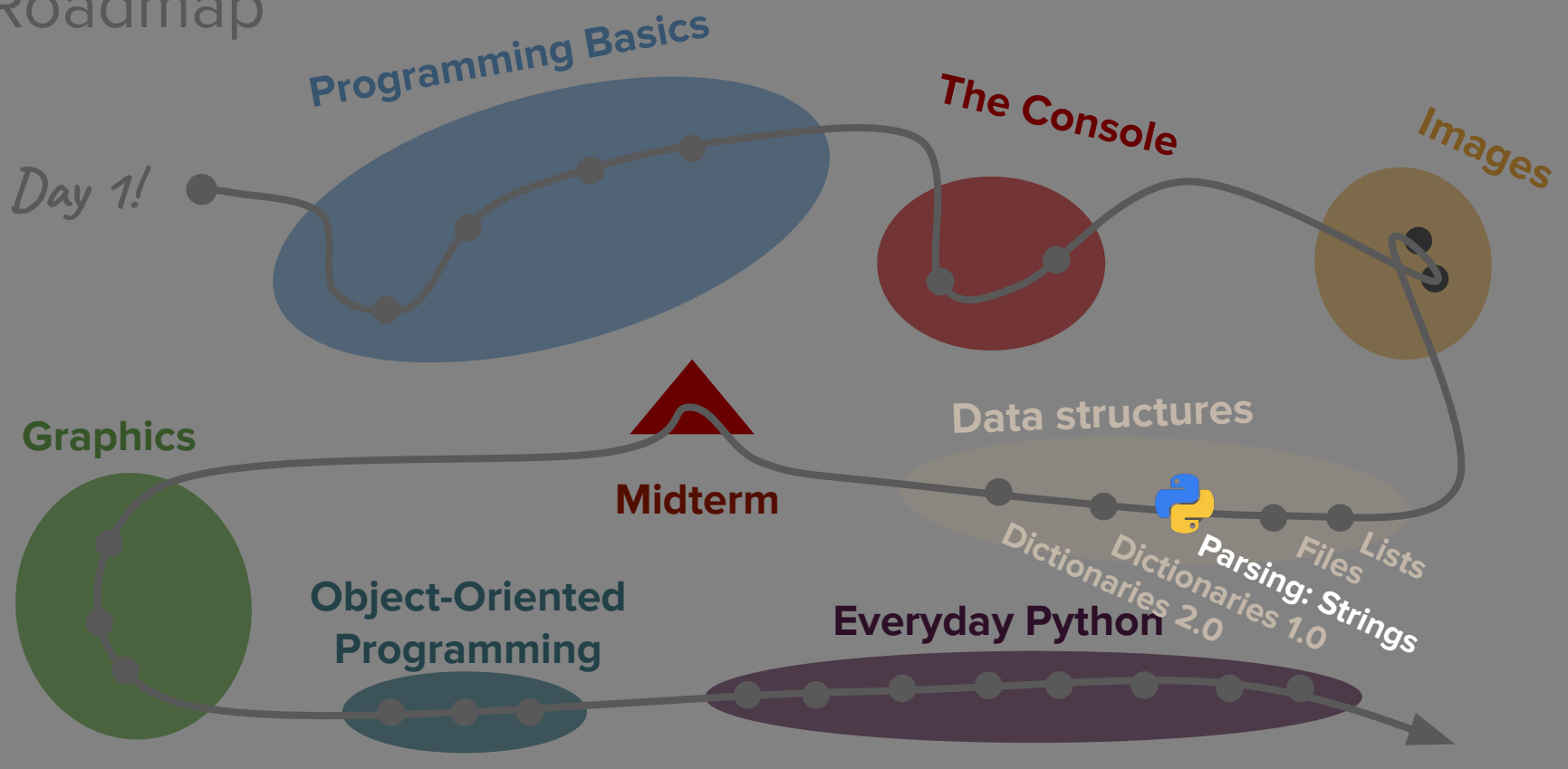
Input: filename

In Sum

- **Files are a way of storing information on your computer long-term**
  - You can read that information in and use it in your programs!
- **The command line is a way of giving instructions to your computer**
  - We can use it to run programs!
- **We can pass information into our programs using the command line**
  - This comes in the form of *arguments*, which are processed as a list of strings

What’s next?

Roadmap


Programming Basics
Day 1!
The Console
Images
Graphics
Midterm
Data structures
Lists
Files
Parsing: Strings
Dictionaries 1.0
Dictionaries 2.0
Object-Oriented Programming
Everyday Python
Life after CS106AP!

What’s next?

- **Parsing**
  - How do we separate junk from valuable information?
  - How do we organize massive amounts of data?