FoCS:

Markov Chains, Dictionaries, and Files

Niema Moshiri
UC San Diego SPIS 2019

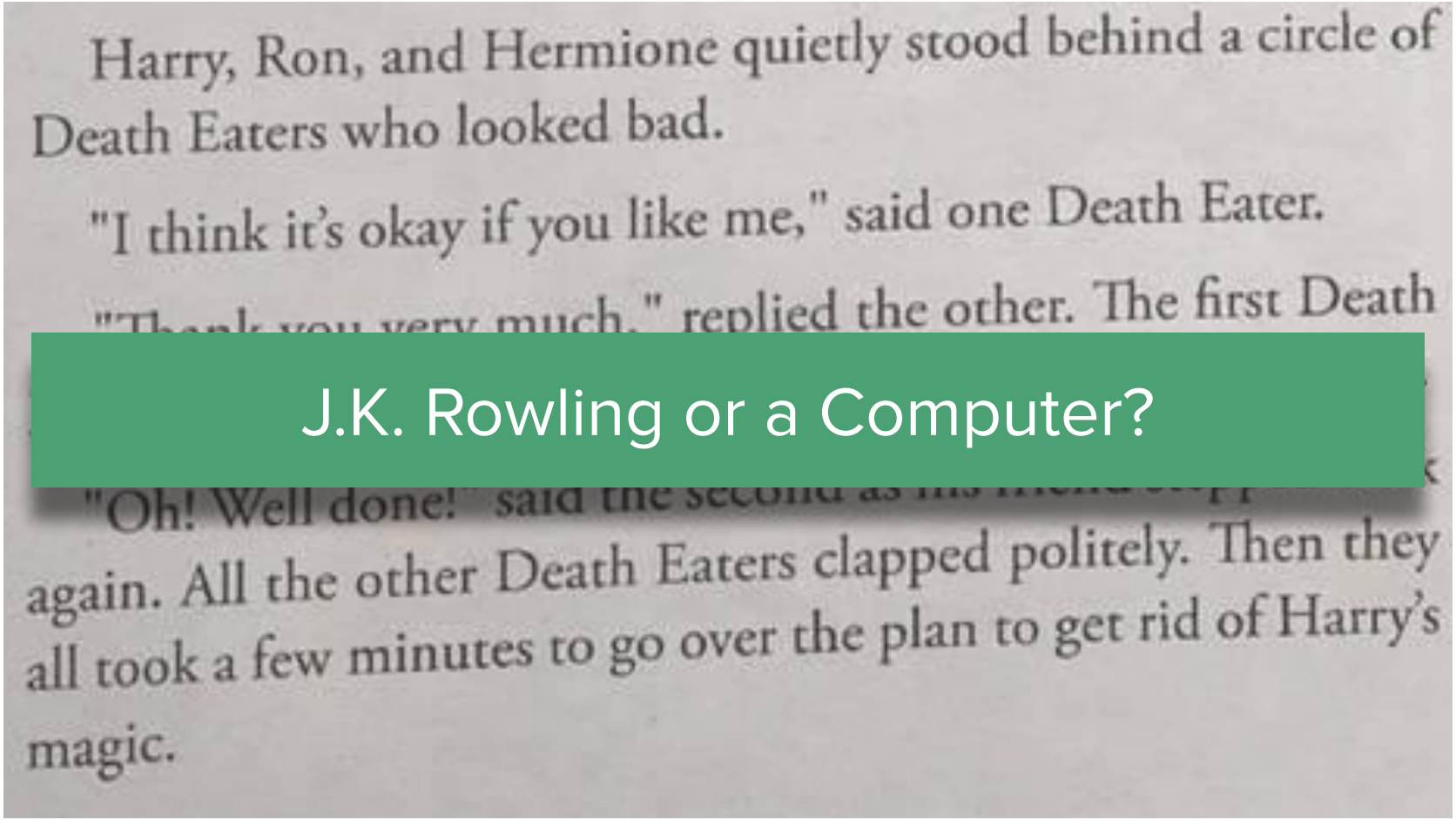
Programs that can **write**

Harry, Ron, and Hermione quietly stood behind a circle of Death Eaters who looked bad.

"I think it's okay if you like me," said one Death Eater.

"Thank you very much," replied the other. The first Death Eater confidently leaned forward to plant a kiss on his cheek.

"Oh! Well done!" said the second as his friend stepped back again. All the other Death Eaters clapped politely. Then they all took a few minutes to go over the plan to get rid of Harry's magic.


Harry, Ron, and Hermione quietly stood behind a circle of Death Eaters who looked bad.
"I think it's okay if you like me," said one Death Eater.
"Thank you very much," replied the other. The first Death
"Oh! Well done!" said the second as his friend stopped
again. All the other Death Eaters clapped politely. Then they all took a few minutes to go over the plan to get rid of Harry's magic.
J.K. Rowling or a Computer?

Creating a Program to Write

- What would be a reasonable **first word** to start a sentence?

Creating a Program to Write

- What would be a reasonable **first word** to start a sentence?
- What would be a reasonable **next word** to follow the first?

Creating a Program to Write

- What would be a reasonable **first word** to start a sentence?
- What would be a reasonable **next word** to follow the first?
- What would be a reasonable **next word** to follow the second?

Creating a Program to Write

- What would be a reasonable **first word** to start a sentence?
- What would be a reasonable **next word** to follow the first?
- What would be a reasonable **next word** to follow the second?
- ...

Creating a Program to Write

- What would be a reasonable **first word** to start a sentence?
- What would be a reasonable **next word** to follow the first?
- What would be a reasonable **next word** to follow the second?
- ...
- What would be a reasonable way to **end a sentence**?

Creating a Program to Write

- What would be a reasonable **first word** to start a sentence?
- What would be a reasonable **next word** to follow the first?
- Can we learn these from reality?
- …
- What would be a reasonable way to **end a sentence**?

Who Wrote What?

It's a dangerous business, _____, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one that rises against them and strikes back!

It’s a gift to exist, and with existence comes suffering. There’s no escaping that. I don’t want it to have happened. I want it to not have happened, but if you are grateful for your life, then you have to be grateful for all of it.

No matter where you are, whether it's a quarter mile away or halfway across the world, you'll always be with me.

Bilbo Baggins (J.R.R. Tolkien)

It's a dangerous business, **Frodo**, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one that rises against them and strikes back!

It’s a gift to exist, and with existence comes suffering. There’s no escaping that. I don’t want it to have happened. I want it to not have happened, but if you are grateful for your life, then you have to be grateful for all of it.

No matter where you are, whether it's a quarter mile away or halfway across the world, you'll always be with me.

Who Wrote What?

Albus Dumbledore (J.K. Rowling)

It's a dangerous business, **Frodo**, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one that rises against them and strikes back!

It’s a gift to exist, and with existence comes suffering. There’s no escaping that. I don’t want it to have happened. I want it to not have happened, but if you are grateful for your life, then you have to be grateful for all of it.

No matter where you are, whether it's a quarter mile away or halfway across the world, you'll always be with me.

Who Wrote What?

It's a dangerous business, **Frodo**, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one that rises against them and strikes back!

It’s a gift to exist, and with existence comes suffering. There’s no escaping that. I don’t want it to have happened. I want it to not have happened, but if you are grateful for your life, then you have to be grateful for all of it.

No matter where you are, whether it's a quarter mile away or halfway across the world, you'll always be with me.

Stephen Colbert

Who Wrote What?

It's a dangerous business, **Frodo**, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

Have you any idea how much tyrants fear the people they oppress? All of them realize that, one day, amongst their many victims, there is sure to be one that rises against them and strikes back!

It’s a gift to exist, and with existence comes suffering. There’s no escaping that. I don’t want it to have happened. I want it to not have happened, but if you are grateful for your life, then you have to be grateful for all of it.

No matter where you are, whether it's a quarter mile away or halfway across the world, you'll always be with me.

Dom Toretto (Fast & Furious)

Mathematical Models

- A **mathematical model** is a description of a system using math

Mathematical Models

- A **mathematical model** is a description of a system using math
- A **probabilistic model** is a mathematical model with randomness

Mathematical Models

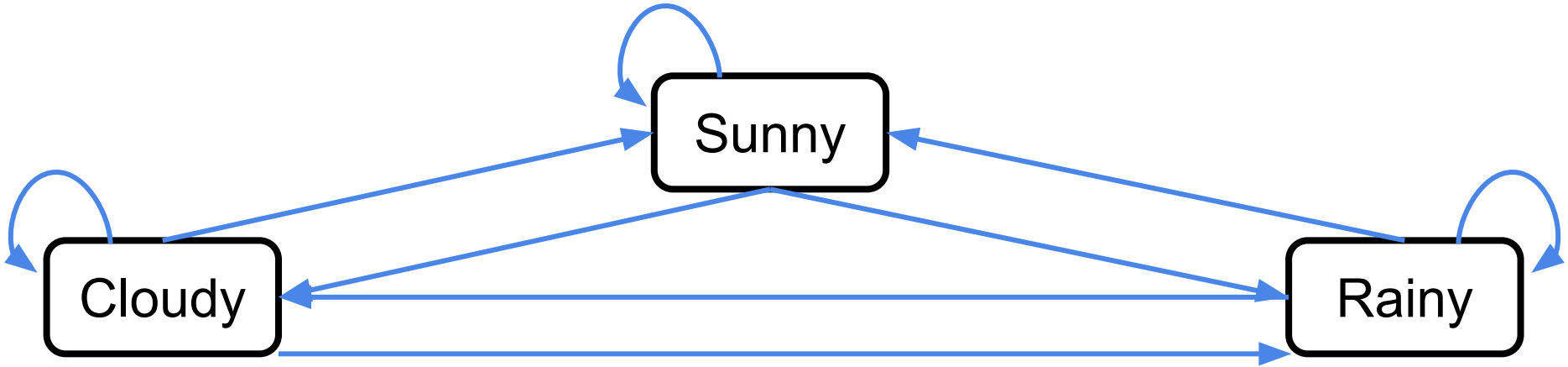
- A **mathematical model** is a description of a system using math
- A **probabilistic model** is a mathematical model with randomness
- We can **train** a probabilistic model to try to capture reality

Mathematical Models

- A **mathematical model** is a description of a system using math
- A **probabilistic model** is a mathematical model with randomness
- We can **train** a probabilistic model to try to capture reality
    - Can we design a probabilistic model to learn writing style?

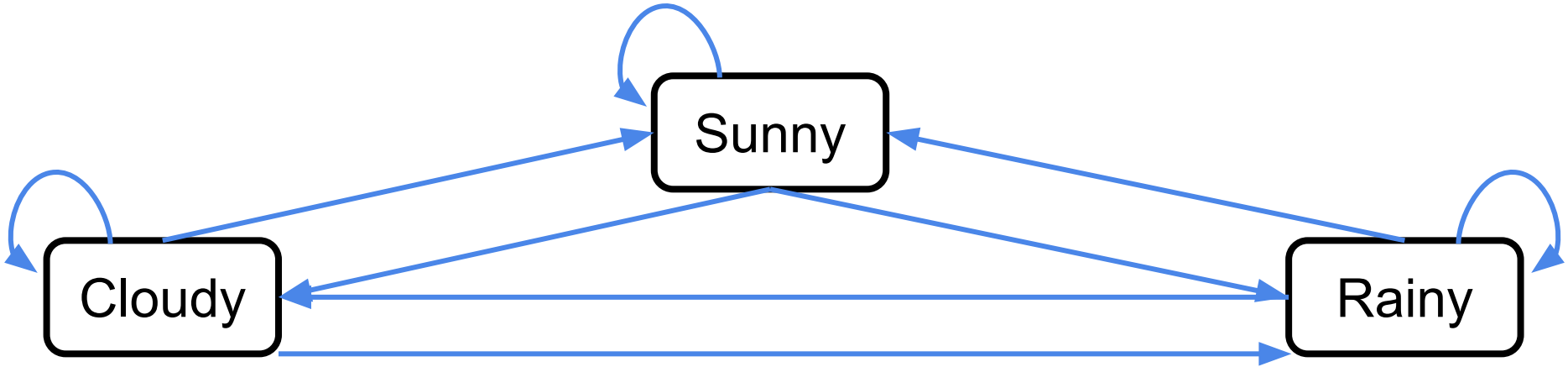
Markov Chains

- **Markov Chains:** A probabilistic model defined by states & transitions



Markov Chains

- **Markov Chains:** A probabilistic model defined by states & transitions
    - In a **first-order** Markov chain, your current state only depends on the state you were in **one step** ago (“memoryless property”)



Markov Chains

- **Markov Chains:** A probabilistic model defined by states & transitions
  - In a **first-order** Markov chain, your current state only depends on the state you were in **one step** ago (“memoryless property”)
    - You can also have higher-order (e.g. second, third, etc.) Markov chains, but they will not be discussed in this lecture

Markov Chains to Model Writing

- We can create a Markov chain in which states are words

Markov Chains to Model Writing

- We can create a Markov chain in which states are words
    - What words have we seen?

Markov Chains to Model Writing

- We can create a Markov chain in which states are words
  - What words have we seen?
  - How often do we transition from word *u* to word *v*?

Markov Chains to Model Writing

- We can create a Markov chain in which states are words
  - What words have we seen?
  - How often do we transition from word *u* to word *v*?
  - In what state (word) do we start?

Markov Chains to Model Writing

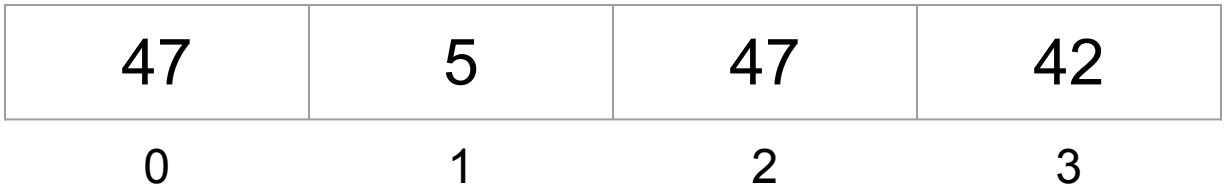
- We can create a Markov chain in which states are words
  - What words have we seen?
  - How often do we transition from word *u* to word *v*?
  - In what state (word) do we start?
  - How do we end a sentence?

Python Dictionaries

Lists vs. Dictionaries

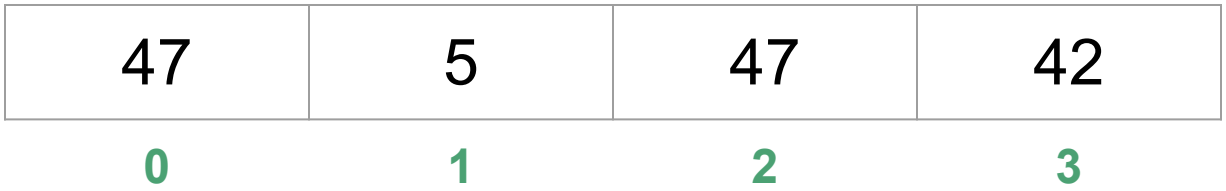
Lists vs. Dictionaries

- A **list** is a *sequential* container



Lists vs. Dictionaries

- A **list** is a *sequential* container
    - Elements are looked up by their **location** (or **index**) in the list



Lists vs. Dictionaries

- A **list** is a *sequential* container
    - Elements are looked up by their **location** (or **index**) in the list
- A **dictionary** is an *unordered* container of (*key*, *value*) pairs

Electric	Fire	Water	Grass
Pikachu	Charmander	Squirtle	Bulbasaur

Lists vs. Dictionaries

- A **list** is a *sequential* container
  - Elements are looked up by their **location** (or **index**) in the list
- A **dictionary** is an *unordered* container of (*key*, *value*) pairs
  - Elements (values) are looked up by their corresponding **keys**

Electric	Fire	Water	Grass
Pikachu	**Charmander**	**Squirtle**	**Bulbasaur**

Lists vs. Dictionaries

- A **list** is a *sequential* container

Design a dictionary!

- A **dictionary** is an *unordered* container of (*key*, *value*) pairs
  - Elements (values) are looked up by their corresponding **keys**

Electric	Fire	Water	Grass
Pikachu	**Charmander**	**Squirtle**	**Bulbasaur**

Practice Problem: Counting Occurrences

- Given a list of words *words*, build a dictionary *counts* as follows:

Practice Problem: Counting Occurrences

- Given a list of words *words*, build a dictionary *counts* as follows:
    - The dictionary’s keys are words

Practice Problem: Counting Occurrences

- Given a list of words *words*, build a dictionary *counts* as follows:
    - The dictionary's keys are words
    - For any key *u*, *counts*[*u*] is the number of times *u* appears in *words*

Practice Problem: Counting Occurrences

- Given a list of words *words*, build a dictionary *counts* as follows:
  - The dictionary's keys are words
  - For any key *u*, *counts*[*u*] is the number of times *u* appears in *words*

```
counts = dict() # empty dictionary
for word in words:
    if _______________:
        _______________
    else:
        _______________
```

```
# Add the key 'hello'
# and set its value to 0
counts['hello'] = 0

# Get the value associated
# with 'hello'
counts['hello']

# Check if 'hello' is a key
'hello' in counts
```

Practice Problem: Counting Occurrences

- Given a list of words *words*, build a dictionary *counts* as follows:
  - The dictionary's keys are words
  - For any key *u*, *counts*[*u*] is the number of times *u* appears in *words*

```
counts = dict() # empty dictionary
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
# Add the key 'hello'
# and set its value to 0
counts['hello'] = 0

# Get the value associated
# with 'hello'
counts['hello']

# Check if 'hello' is a key
'hello' in counts
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    'spam': 1
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 1
}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    'spam': 1
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 1
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 2
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 2
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 2
}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    'spam': 2,
    'poptarts': 1
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 2,
    ‘poptarts’: 1
}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    'spam': 2,
    'poptarts': 1
}
```

Practice Problem: Counting Occurrences

```
words = [‘spam’, ‘spam’, ‘poptarts’, ‘spam’]
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    ‘spam’: 2,
    ‘poptarts’: 1
}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

```
counts = dict()
for word in words:
    if word not in counts:
        counts[word] = 1
    else:
        counts[word] += 1
```

```
counts = {
    'spam': 2,
    'poptarts': 1
}
```

Practice Problem: Counting Occurrences

```
words = ['spam', 'spam', 'poptarts', 'spam']
```

Files!!

But where will we get so many words??

```
cou
for
        counts[word] = 1
    else:
        counts[word] += 1
```

```
}
```

Reading Files in Python

Reading Files in Python

- In Python, you can use the `open()` function to open a file

Reading Files in Python

- In Python, you can use the `open()` function to open a file
  - `f = open('the_hobbit.txt') # f is a file object`

Reading Files in Python

- In Python, you can use the `open()` function to open a file
  - `f = open('the_hobbit.txt') # f is a file object`
- You can use the `read()` function of a file object to read the contents

Reading Files in Python

- In Python, you can use the `open()` function to open a file
  - `f = open('the_hobbit.txt') # f is a file object`
- You can use the `read()` function of a file object to read the contents
  - `text = f.read() # text is a string`

Reading Files in Python

- In Python, you can use the `open()` function to open a file
  - `f = open('the_hobbit.txt') # f is a file object`
- You can use the `read()` function of a file object to read the contents
  - `text = f.read() # text is a string`
- You can use the split() function of a string to split the words apart

Reading Files in Python

- In Python, you can use the `open()` function to open a file
  - `f = open('the_hobbit.txt') # f is a file object`
- You can use the `read()` function of a file object to read the contents
  - `text = f.read() # text is a string`
- You can use the split() function of a string to split the words apart
  - `words = text.split() # words is a list of strings`

Example: Reading Files in Python

```
f = open('the_hobbit.txt')
text = f.read()
words = text.split()
```

```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

Example: Reading Files in Python

```
f = open('the_hobbit.txt')
text = f.read()
words = text.split()
```


```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

Example: Reading Files in Python

```
f = open('the_hobbit.txt')
text = f.read()
words = text.split()
```


```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

text

```
"In a hole in the ground..."
```

Example: Reading Files in Python

```
f = open('the_hobbit.txt')
text = f.read()
words = text.split()
```

f

```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

text

```
"In a hole in the ground..."
```

```
words = ['In', 'a', 'hole',
'in', 'the', 'ground',
'there', 'lived', 'a',
'hobbit.', 'Not', 'a',
'nasty,', 'dirty,', 'wet',
'hole,', ...]
```

Example: Reading Files in Python

How many words are there in total?

d…”

```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

```
words = [‘In’, ‘a’, ‘hole’,
‘in’, ‘the’, ‘ground’,
‘there’, ‘lived’, ‘a’,
‘hobbit.’, ‘Not’, ‘a’,
‘nasty,’, ‘dirty,’, ‘wet’,
‘hole,’, ...]
```

Example: Reading Files in Python

```
len(words)
```

How many words are there in total?

d…”

```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

```
words = [‘In’, ‘a’, ‘hole’,
‘in’, ‘the’, ‘ground’,
‘there’, ‘lived’, ‘a’,
‘hobbit.’, ‘Not’, ‘a’,
‘nasty,’, ‘dirty,’, ‘wet’,
‘hole,’, ...]
```

Example: Reading Files in Python

How many unique words are there?

```
In a hole in the ground there lived
a hobbit. Not a nasty, dirty, wet
hole, filled with the ends of worms
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

```
words = [‘In’, ‘a’, ‘hole’,
‘in’, ‘the’, ‘ground’,
‘there’, ‘lived’, ‘a’,
‘hobbit.’, ‘Not’, ‘a’,
‘nasty,’, ‘dirty,’, ‘wet’,
‘hole,’, ...]
```

Example: Reading Files in Python

How many unique words are there?

How many instances of each unique word are there?

```
In
a h
hol
and an oozy smell, nor yet a dry,
bare, sandy hole with nothing in it
to sit down on or to eat: it was a
hobbit-hole, and that means comfort.
```

the_hobbit.txt

```
d…”

le’,

‘hobbit.’, ‘Not’, ‘a’,
‘nasty,’, ‘dirty,’, ‘wet’,
‘hole,’, ...]
```

Implementing a Markov Model using a Python Dictionary

Implementing a Markov Model using a Dictionary

- We will use a Python dictionary to implement a Markov model

Implementing a Markov Model using a Dictionary

- We will use a Python dictionary to implement a Markov model
    - What are the keys?

Implementing a Markov Model using a Dictionary

- We will use a Python dictionary to implement a Markov model
  - What are the keys?
  - What are the values?

Implementing a Markov Model using a Dictionary

- We will use a Python dictionary to implement a Markov model
  - What are the keys?
  - What are the values?
  - How will we depict the start of a sentence?

Implementing a Markov Model using a Dictionary

- We will use a Python dictionary to implement a Markov model
  - What are the keys?
  - What are the values?
  - How will we depict the start of a sentence?
    - `None` can be used as “start of sentence” state