CMPU 366 · Computational Linguistics

Processing Text

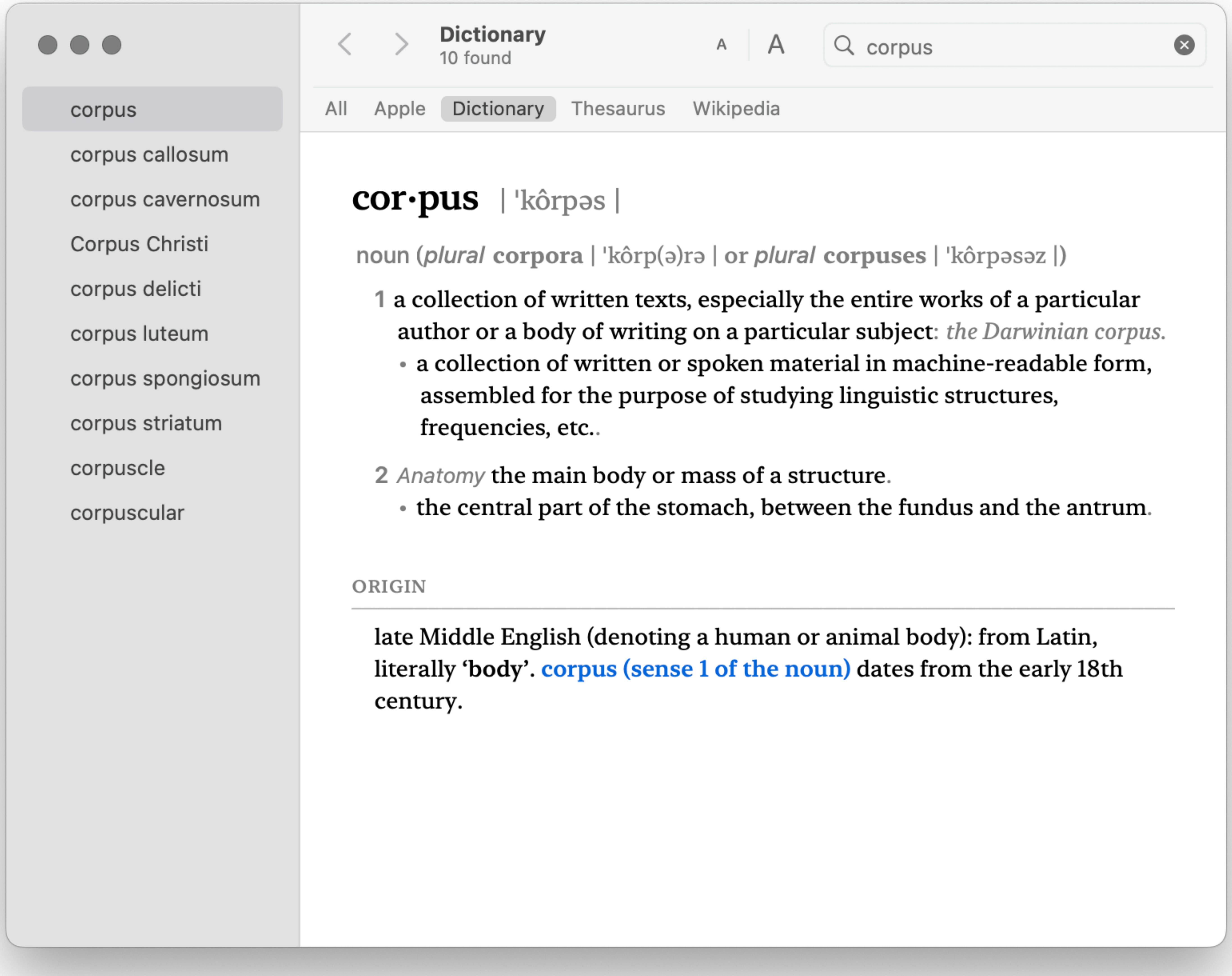
8 September 2025

VASSAR COLLEGE
A·D·1861

Where are we?

Our textbook is called *Speech and Language Processing*, but this class will be focused on text.

Why might we do that?



cor·pus | ˈkôrpəs |

noun (*plural* **corpora** | ˈkôrp(ə)rə | or *plural* **corpuses** | ˈkôrpəsəz |)

1 a collection of written texts, especially the entire works of a particular author or a body of writing on a particular subject: *the Darwinian corpus.*
- a collection of written or spoken material in machine-readable form, assembled for the purpose of studying linguistic structures, frequencies, etc..

2 *Anatomy* the main body or mass of a structure.
- the central part of the stomach, between the fundus and the antrum.

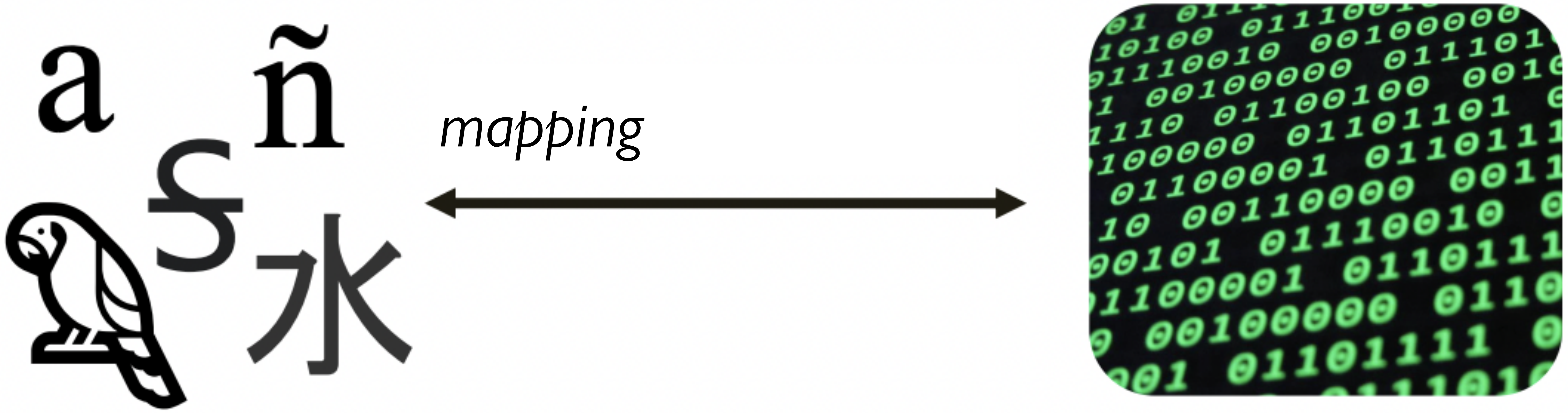
ORIGIN

late Middle English (denoting a human or animal body): from Latin, literally **‘body’**. corpus (sense 1 of the noun) dates from the early 18th century.

What *isn't* a text corpus?

Text on computers

Encodings



ASCII

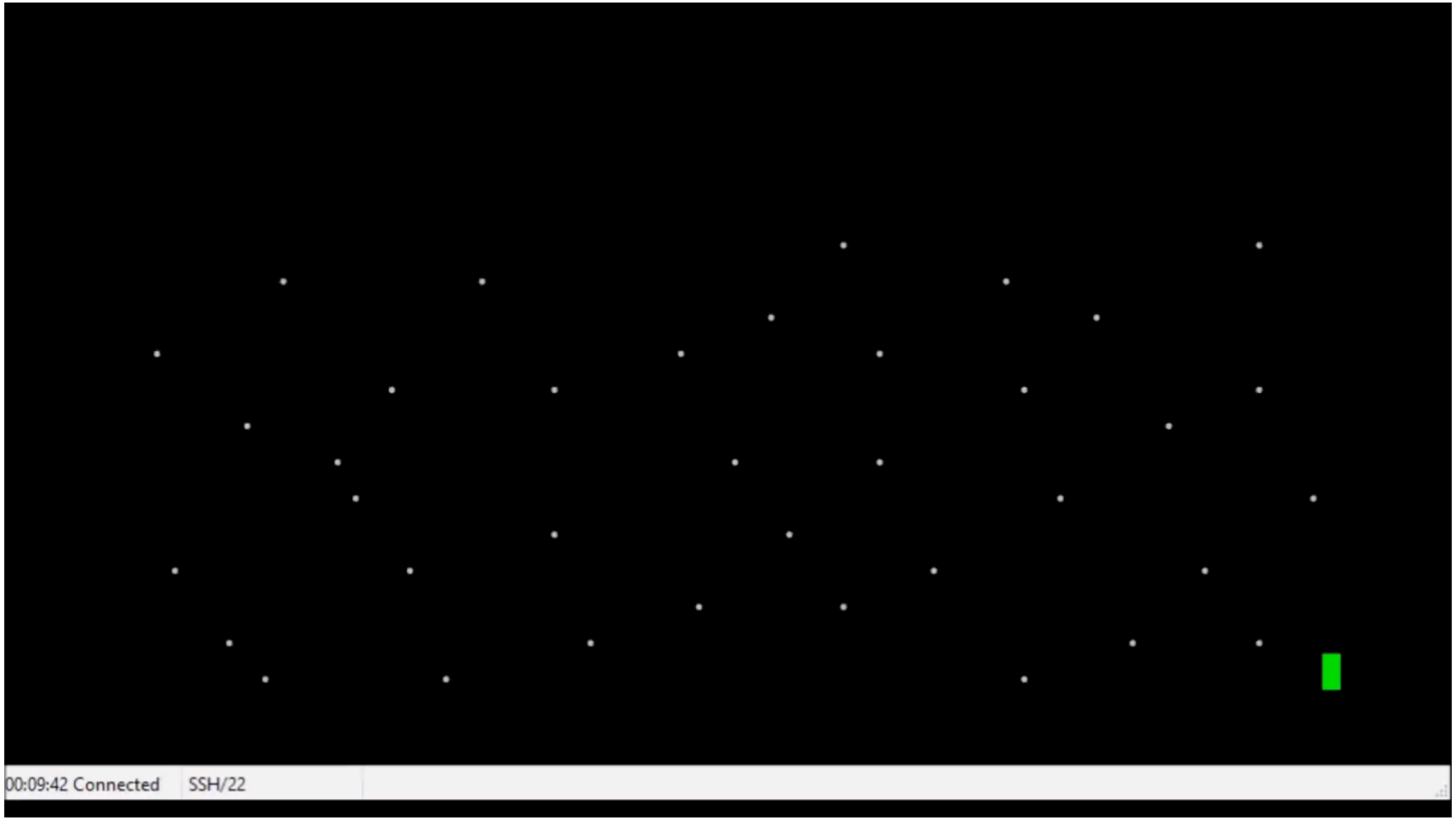
ASCII

```
           _  .   .   .
         .' '; '-' '-'|-.
        (     '------.'  )
         ;            \ /
          :     '   ' |/
          '._._       \    .;
         .-'   ;--.    '--' /
        /      \-'---.___.'
       |     / 7 \(>o<) /\
       |     | \ |  .  \   \
       |=====|   |  .   \ |-)
        |-'-'   ./_.-._.\|"
        '-.----'        |
          |       |     |
          |     | |   | |
          |_____|_|___|_|
snd       (-------',----'.
           '_'_____'_____'
```

Babar

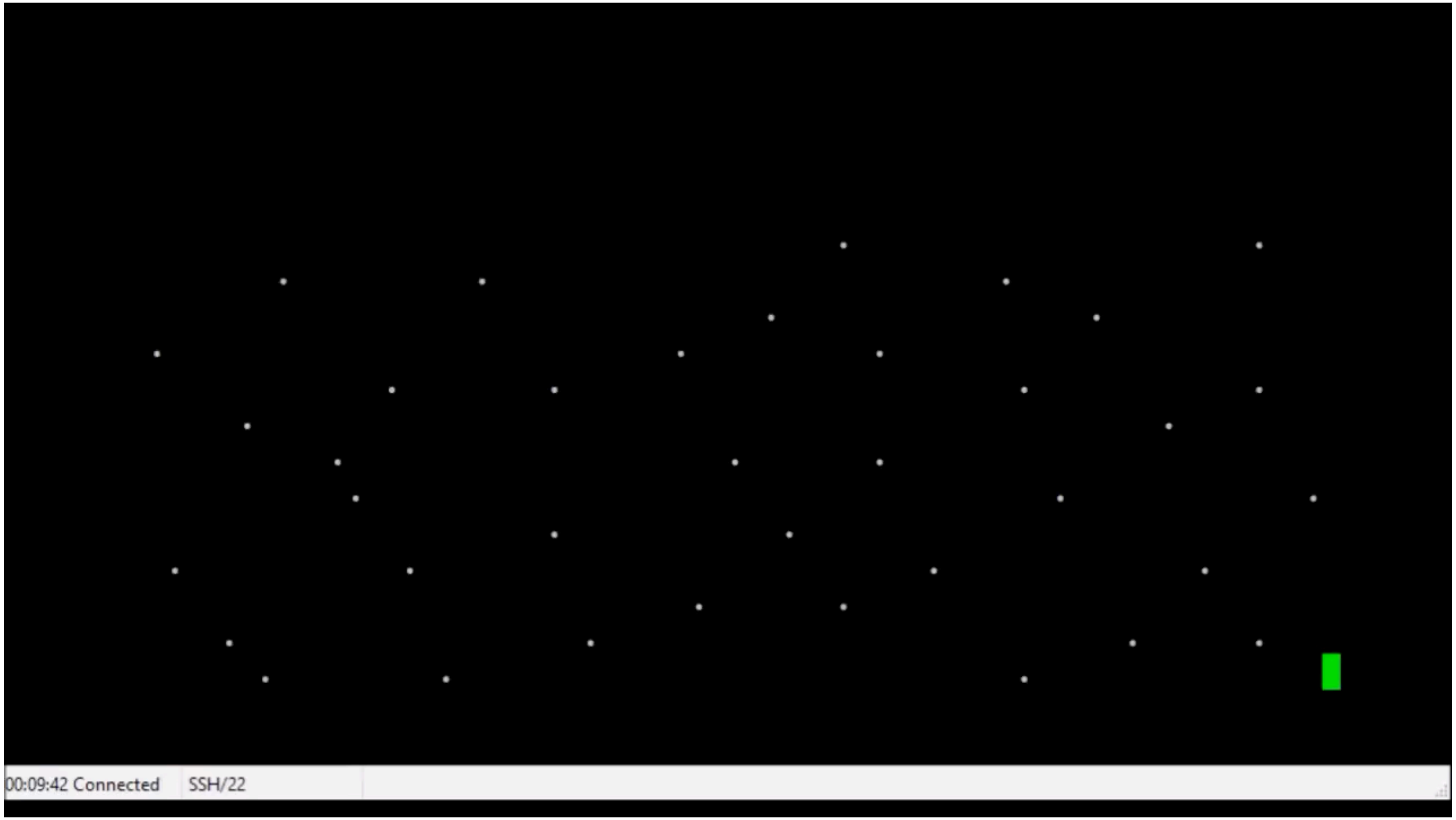
By Shanaka Dias

ASCII



```
telnet towel.blinkenlights.nl
```

ASCII



00:09:42 Connected SSH/22

```
telnet towel.blinkenlights.nl
```

ASCII

American Standard Code for Information Interchange

`01000001`	`01010011`	`01000011`	`01001001`	`01001001`	*Bits*
65	83	67	73	73	*Code point*
`"A"`	`"S"`	`"C"`	`"I"`	`"I"`	*Character*

```
Dec  Char                           Dec  Char     Dec  Char     Dec  Char
---------                           ---------     ---------     ----------
  0  NUL (null)                      32  SPACE     64  @         96  `
  1  SOH (start of heading)          33  !         65  A         97  a
  2  STX (start of text)             34  "         66  B         98  b
  3  ETX (end of text)               35  #         67  C         99  c
  4  EOT (end of transmission)       36  $         68  D        100  d
  5  ENQ (enquiry)                   37  %         69  E        101  e
  6  ACK (acknowledge)               38  &         70  F        102  f
  7  BEL (bell)                      39  '         71  G        103  g
  8  BS  (backspace)                 40  (         72  H        104  h
  9  TAB (horizontal tab)            41  )         73  I        105  i
 10  LF  (NL line feed, new line)    42  *         74  J        106  j
 11  VT  (vertical tab)              43  +         75  K        107  k
 12  FF  (NP form feed, new page)    44  ,         76  L        108  l
 13  CR  (carriage return)           45  -         77  M        109  m
 14  SO  (shift out)                 46  .         78  N        110  n
 15  SI  (shift in)                  47  /         79  O        111  o
 16  DLE (data link escape)          48  0         80  P        112  p
 17  DC1 (device control 1)          49  1         81  Q        113  q
 18  DC2 (device control 2)          50  2         82  R        114  r
 19  DC3 (device control 3)          51  3         83  S        115  s
 20  DC4 (device control 4)          52  4         84  T        116  t
 21  NAK (negative acknowledge)      53  5         85  U        117  u
 22  SYN (synchronous idle)          54  6         86  V        118  v
 23  ETB (end of trans. block)       55  7         87  W        119  w
 24  CAN (cancel)                    56  8         88  X        120  x
 25  EM  (end of medium)             57  9         89  Y        121  y
 26  SUB (substitute)                58  :         90  Z        122  z
 27  ESC (escape)                    59  ;         91  [        123  {
 28  FS  (file separator)            60  <         92  \        124  |
 29  GS  (group separator)           61  =         93  ]        125  }
 30  RS  (record separator)          62  >         94  ^        126  ~
 31  US  (unit separator)            63  ?         95  _        127  DEL
```

*These are **all** of the characters in ASCII.*

1963 ASCII, a 7-bit standard (with an unused 8th bit)

1987–2000 ISO-8859, a series of 256-character 8-bit standards

Examples: Latin-1, Latin/Cyrillic, Latin/Greek, Latin/Hebrew,

But CJKV languages need bigger character sets, e.g., Big5 (two bytes), GB 18030 and ISO 2022 (more complex).

How do you tell which one a file is using?

Today we mostly use *Unicode*. Unicode isn't an encoding; it's a listing of code points.

It can be encoded in different ways:

- UTF-32 is a 32-bit fixed-length encoding
- UTF-8 and UTF-16 are variable-length encodings

Character	*Code point*	*UTF-8 bits*
A	U+0041	`01000001`
ਐ	U+0A10	`11100000 10101000 10010000`

Unicode® 16.0.0

2024 September 10 (Announcement)

This page summarizes the important changes for the Unicode Standard, Version 16.0.0. This version supersedes all previous versions of the Unicode Standard.

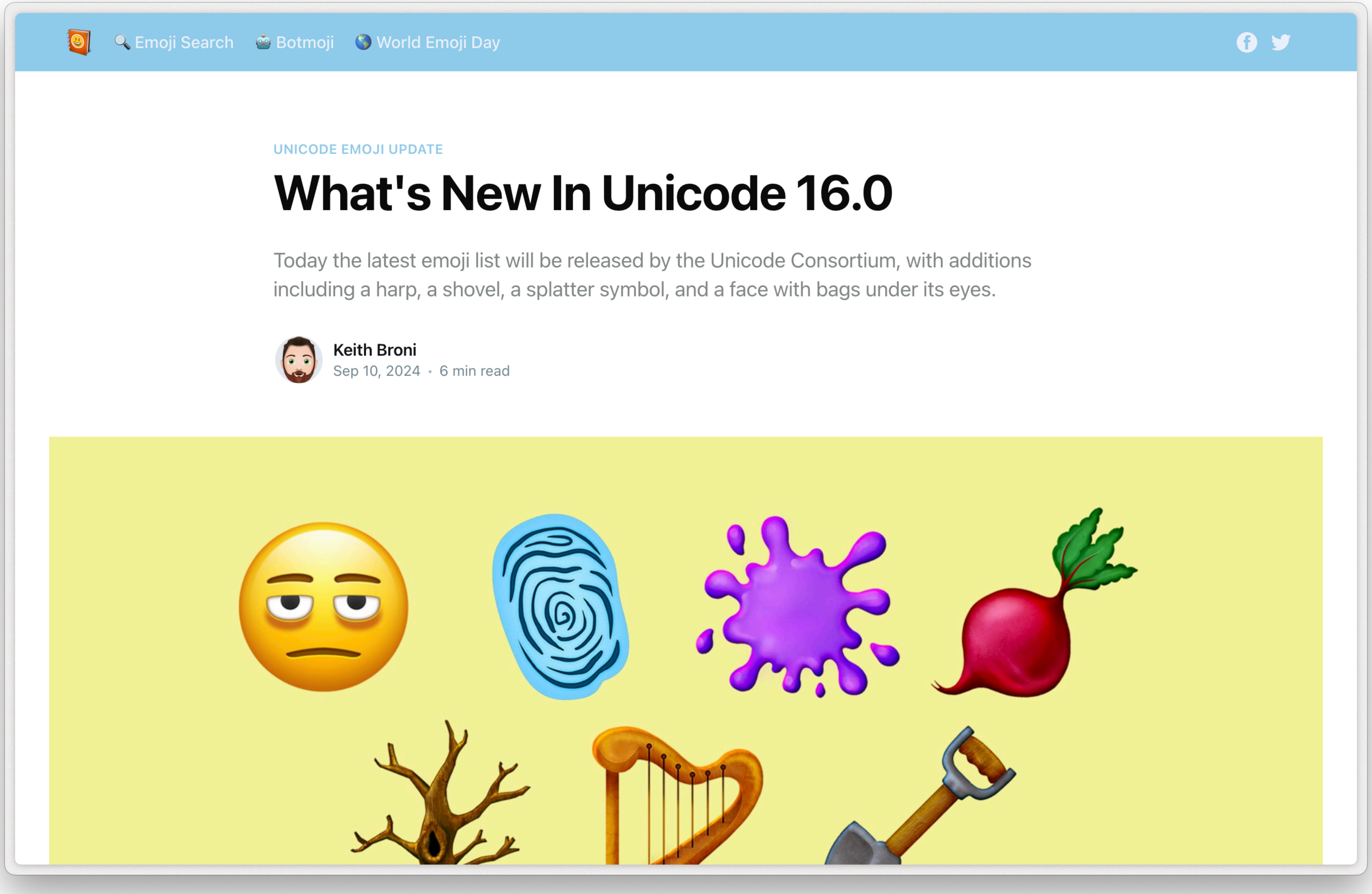
A. Summary

Unicode 16.0 adds 5185 characters, for a total of 154,998 characters. The new additions include seven new scripts:

- Garay is a modern-use script from West Africa.
- Gurung Khema, Kirat Rai, Ol Onal and Sunuwar are four modern-use scripts from Northeast India and Nepal.
- Todhri is an historic script used for Albanian.
- Tulu-Tigalari is an historic script from Southwest India.

Other character additions include seven new emoji characters plus 3,995 additional Egyptian Hieroglyphs and over 700 symbols from legacy computing environments.

In addition to new characters, new “Moji Jōhō Kiban” (文字情報基盤) Japanese source references have been added for over 36,000 CJK unified ideographs. These are



UNICODE EMOJI UPDATE

What's New In Unicode 16.0

Today the latest emoji list will be released by the Unicode Consortium, with additions including a harp, a shovel, a splatter symbol, and a face with bags under its eyes.

Keith Broni
Sep 10, 2024 · 6 min read

㋿

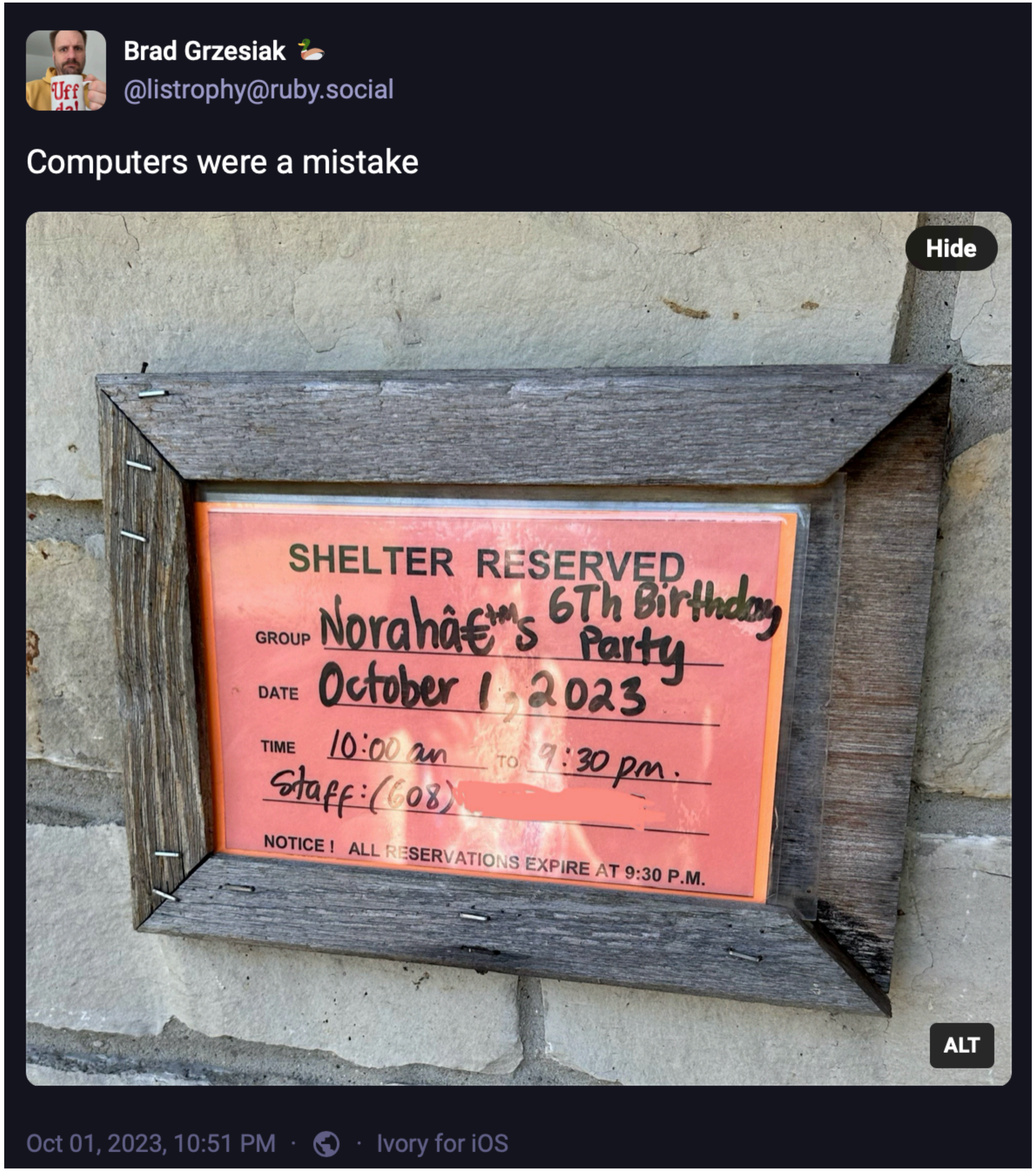
Introduced by Unicode 12.1 in 2019

Computers have gotten better at typography, but they’ve made things more difficult for NLP.

Computers have gotten better at typography, but they’ve made things more diﬃcult for NLP.

Things that can cause trouble:

- Contextual characters (“quotation marks” rather than "quotation marks")
- Newlines (LF, CR, or CRLF)
- Ligatures (scoﬀ)
- Combining characters (jwz)



Brad Grzesiak 🦆

@listrophy@ruby.social

Computers were a mistake

SHELTER RESERVED

GROUP Norahâ€™s 6Th Birthday Party

DATE October 1, 2023

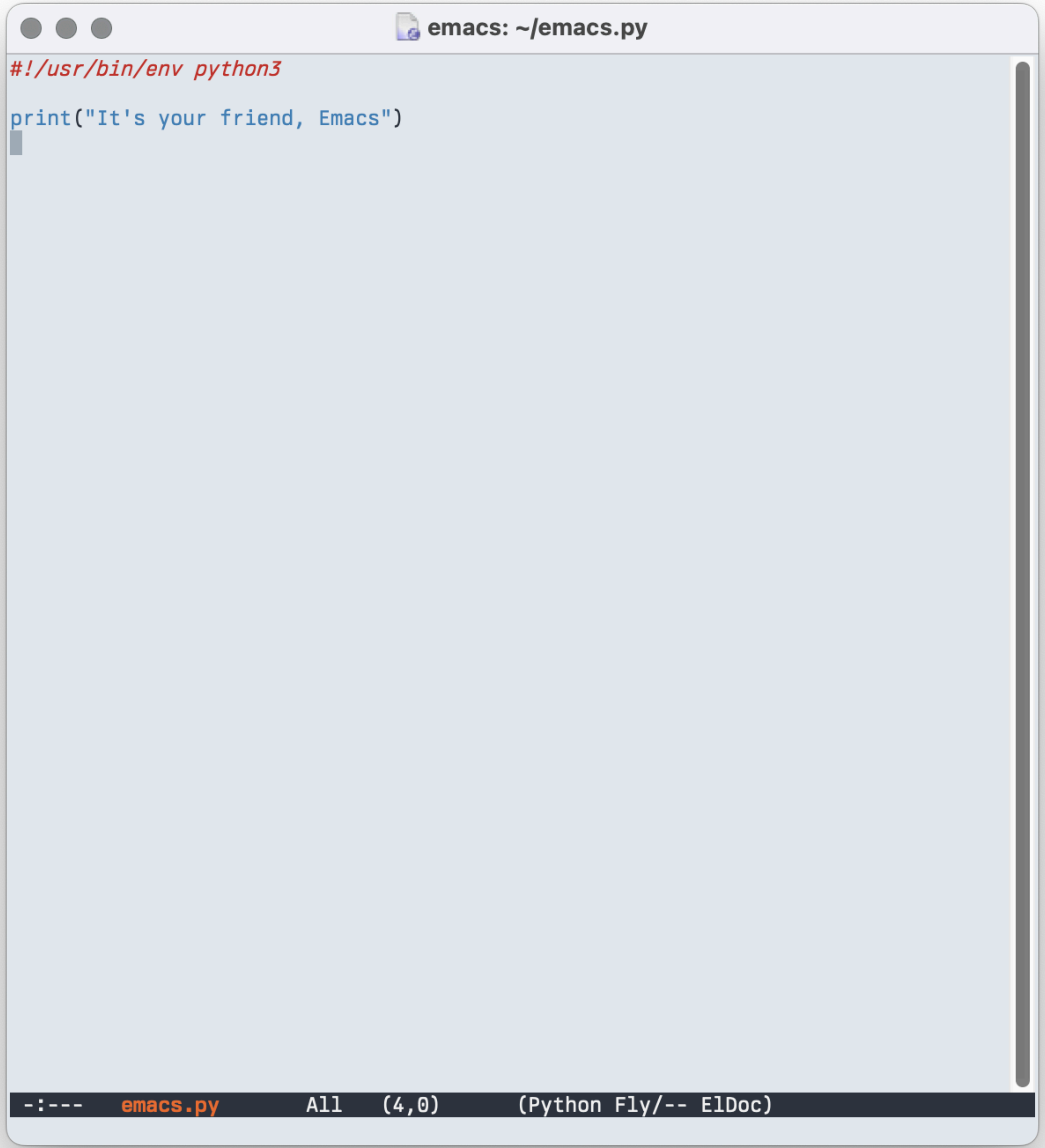
TIME 10:00 an TO 9:30 pm.

Staff: (608) [redacted]

NOTICE ! ALL RESERVATIONS EXPIRE AT 9:30 P.M.

Oct 01, 2023, 10:51 PM · 🌐 · Ivory for iOS

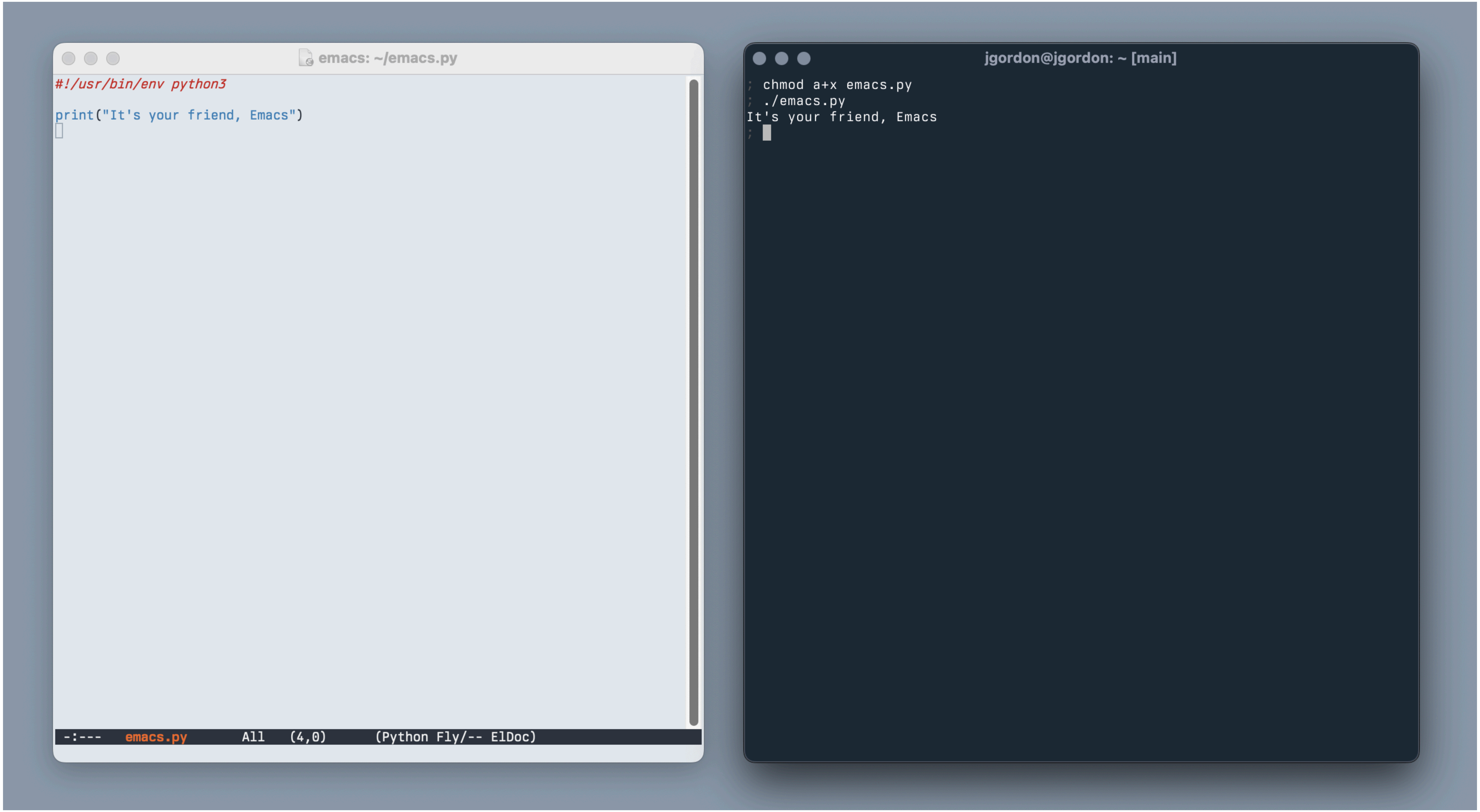
Introduction to Python development



emacs: ~/emacs.py

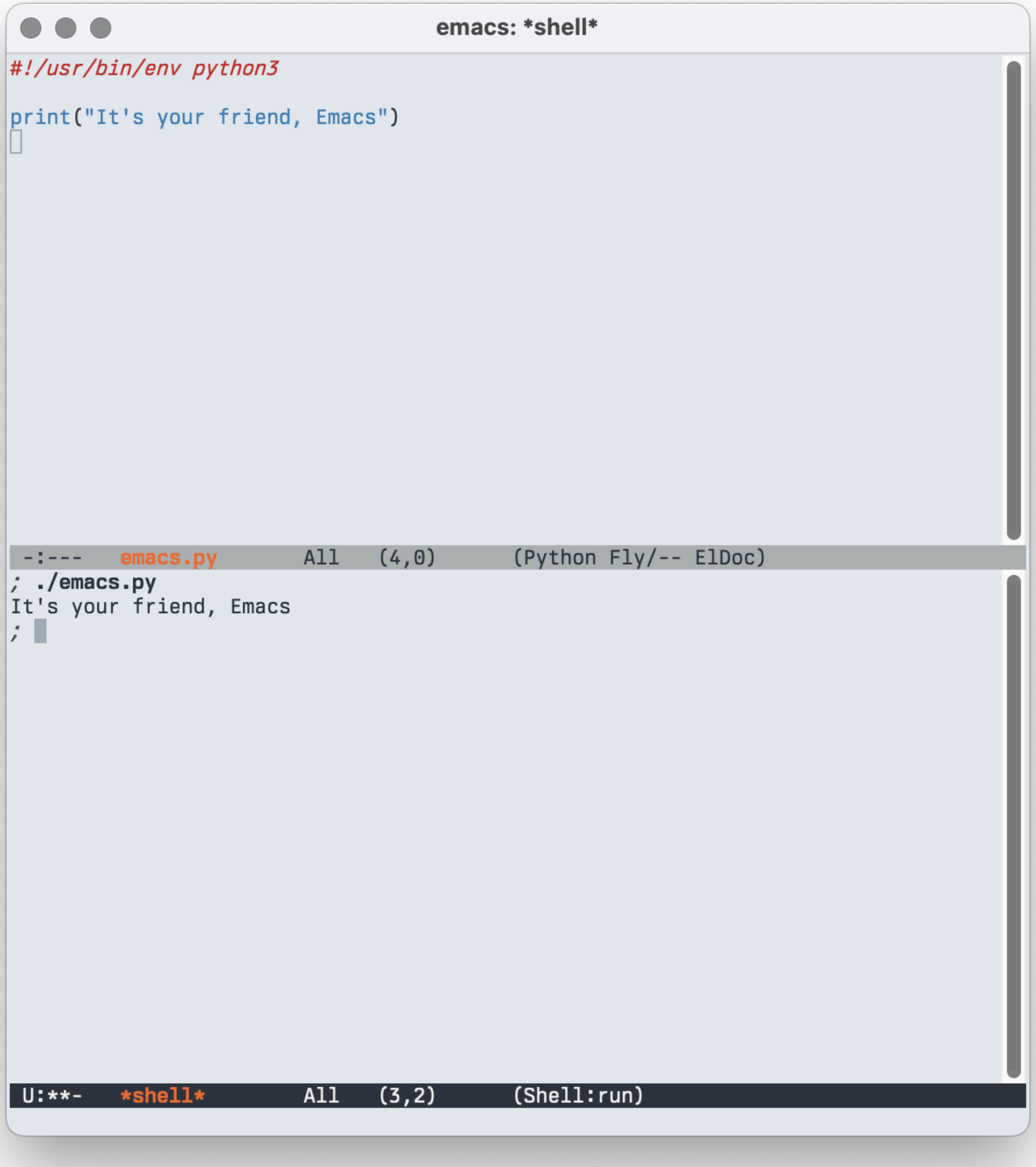
```
#!/usr/bin/env python3

print("It's your friend, Emacs")

```

```
-:---  emacs.py      All   (4,0)      (Python Fly/-- ElDoc)
```


emacs: ~/emacs.py
-:--- emacs.py All (4,0) (Python Fly/-- ElDoc)
jgordon@jgordon: ~ [main]

```
#!/usr/bin/env python3

print("It's your friend, Emacs")
```

```
; chmod a+x emacs.py
; ./emacs.py
It's your friend, Emacs
;
```



```
#!/usr/bin/env python3

print("It's your friend, Emacs")

 -:---    emacs.py      All   (4,0)      (Python Fly/-- ElDoc)
; ./emacs.py
It's your friend, Emacs
;
 U:**-    *shell*       All   (3,2)      (Shell:run)
```

`Ctrl-x 2`

to split vertically

`M-x shell`

M means the "meta" key, which is what you and I call "Escape".

Other options:

`vim`

`gedit`

BBEdit

VSCode*

…

Other options:

`vim`

`gedit`

BBEdit

VSCode*

…

** I know this is the one you want to choose, but it makes things more complicated by using its own Python runtime rather than what's installed on the system. If you choose to use it, you're on your own.*

Jupyter notebooks (like Google Colab) are great, and they're widely used in NLP and data science.

Feel free to use them for experimentation or when you're working on your final project, but – unless otherwise noted – the assignments you turn in should be normal, well-written Python files (`.py`), not notebooks (`.ipynb`).

Demo: Files and encodings in Python

As an example of a file to read, we will use a relatively small, unannotated document from Project Gutenberg.

> Project Gutenberg is an online collection of texts whose copyright has expired. It contains texts in many languages.

Download the First Project Gutenberg Collection of Edgar Allan Poe:

> `gutenberg.org/cache/epub/1062/pg1062.txt`

```python
with open("pg1062.txt") as f:
    for line in f:
        print(line)
```

Because we didn’t specify the encoding, Python will use its default.

What’s the default? 🤷‍♀️

It depends on the system you’re running on!

```python
with open("pg1062.txt", encoding="utf-8-sig") as f:
    for line in f:
        print(line)
```

EDGAR POE

LE SCARABÉE D'OR

Traduction de Charles Baudelaire

ILLUSTRATIONS EN COULEURS ET EN NOIR
PAR
GEORGES ROCHEGROSSE

PARIS
LIBRAIRIE DES AMATEURS
A. FERROUD. — F. FERROUD, Successeur
127, Boulevard Saint-Germain, 127
1926

Download

`gutenberg.org/ebooks/67094.txt.utf-8`

Compare the output

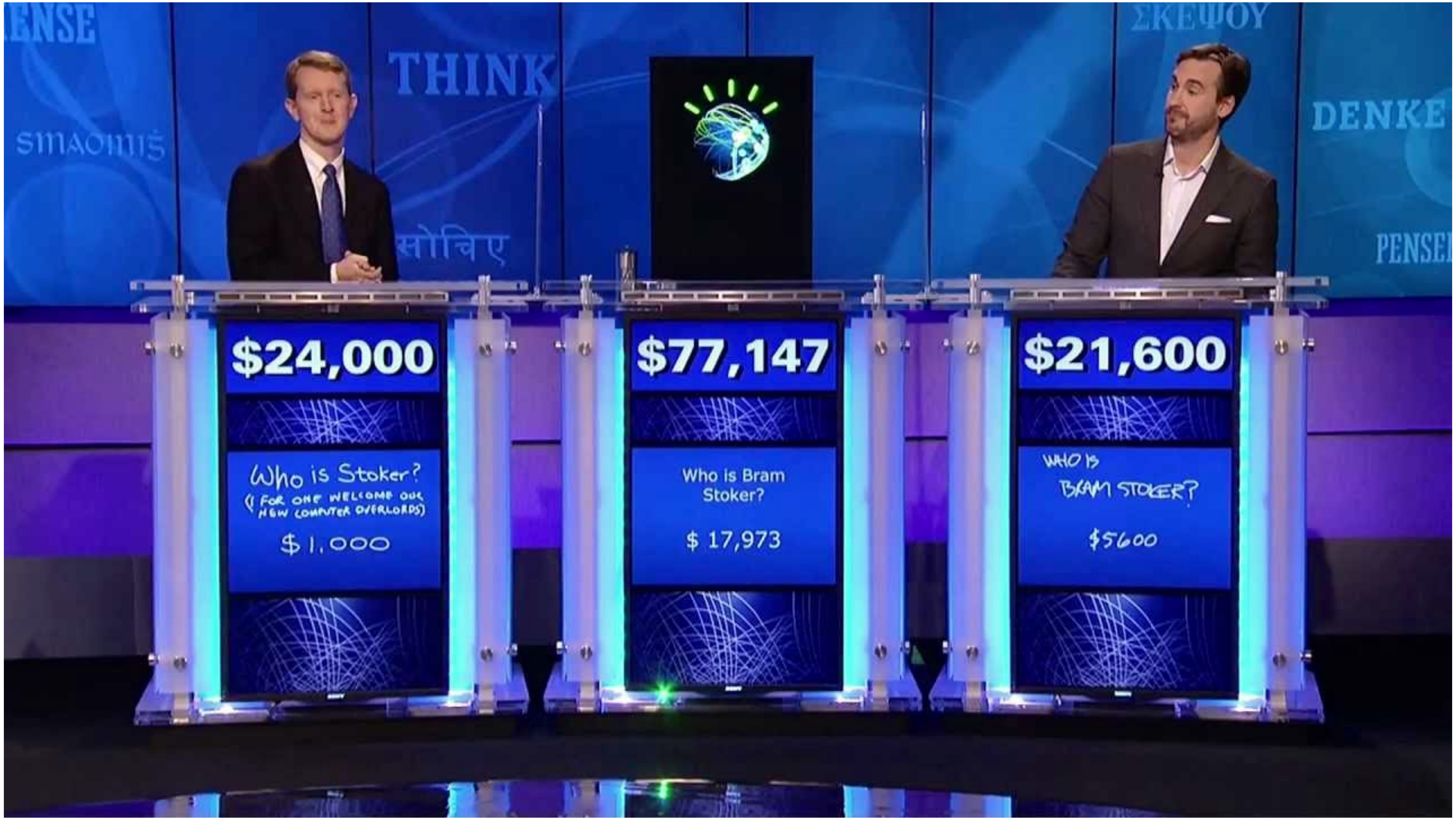
```python
with open("pg67094.txt", encoding="utf-8-sig") as f:
    for line in f:
        print(line)
```

```python
with open("pg67094.txt", encoding="latin1") as f:
    for line in f:
        print(line)
```

If we want to *write* a file, change the `"r"` mode to `"w"`:

```
with open("myoutfile.txt", "w") as f:
    print("First line of output", file=f)
    print("Second line of output", file=f)
    print("Here's a number:", 5, file=f)
```

Search



February 2011

LITERARY CHARACTER APB, $400:

His victims include Charity Burbage, Mad Eye Moody & Severus Snape; he'd be easier to catch if you'd just name him!

BEATLES PEOPLE, $200:

“And anytime you feel the pain, hey” this guy
“refrain, don’t carry the world upon your shoulders”

OLYMPIC ODDITIES, $800:

In the 2004 opening ceremonies, a sole member of this team opened the Parade of Nations; the rest of his team closed it.

Some of these questions are really hard for people because you need to know so much trivia, and our meat minds are bad at that.

But computers can store a lot; Watson had 21.6 TB of storage, back in 2011.

LINUS’S LAW: “Given enough eyeballs, all bugs are shallow.”

“Given enough text, all questions are easy.”

Regular expressions in Python

RE functions to know:

```
re.search

re.match

re.finditer

re.findall

re.compile

re.sub
```

```python
import re

s = "Hello there"

m = re.search(r"\b(t?here)\b", s)

print(m.group(1))
```

Says this is a regular expression. Otherwise, `\b` *won't be interpreted correctly as the class of word-boundary markers.*

```
import re

s = "Hello there"

m = re.search(r"\b(t?here)\b", s)

print(m.group(1))
```

```python
import re

s = "Hello there"

m = re.match(r"\b(t?here)\b", s)

print(m.group(1))
```

```
import re

s = "Hello there"

m = re.match(r"\b(t?here)\b", s)

print(m.group(1))
```

Error! re.match only matches from the beginning of the string. It’s equivalent to starting the RE with ^.

```python
import re

s = "Hello there, hello here, hello everywhere"

for m in re.finditer(r"\b(t?here)\b", s):
    print(m.group(1))
```

```python
import re

s = "Hello there, hello here, hello everywhere"

for match in re.findall(r"\b(t?here)\b", s):
    print(match)
```

```python
import re

s = "Hello there"

prog = re.compile(r"(Hello|howdy)")

m = prog.match("Hello there")
print(m.group(1))

m = prog.match("Howdy partner")
print(m.group(1))
```

```python
import re

s = "Hello there"

prog = re.compile(r"(Hello|howdy)")

m = prog.match("Hello there")
print(m.group(1))

m = prog.match("Howdy partner")
print(m.group(1))
```

Compiling lets us efficiently re-use a regex.

```python
import re

s = "Hello there"

t = re.sub("(Hello|Hi) there", r"\1", s)

print(t)
```

Practice: Information Extraction

Output will be triples (entity1, relation, entity2), e.g.,

```
("Vassar College", "located in", "Poughkeepsie, NY")

("Grace Hopper", "born in", "1906")
```

“[Regular expressions] are particularly useful for searching in texts, when we have a *pattern* to search for and a *corpus* of texts to search through. A regular expression search function will search through the corpus, returning all texts that match the pattern.”

Jurafsky & Martin, § 2.1

Consider learning when people were born.

Consider learning when people were born.

What do you search for?

Let’s try some of these out!

As our corpus, we’ll look at an old snapshot of English Wikipedia:

```
/data/366/wikipedia
```

`born in [0-9]{4}`

Don't want to match places or other descriptions, e.g.,

born in New York

born in poverty

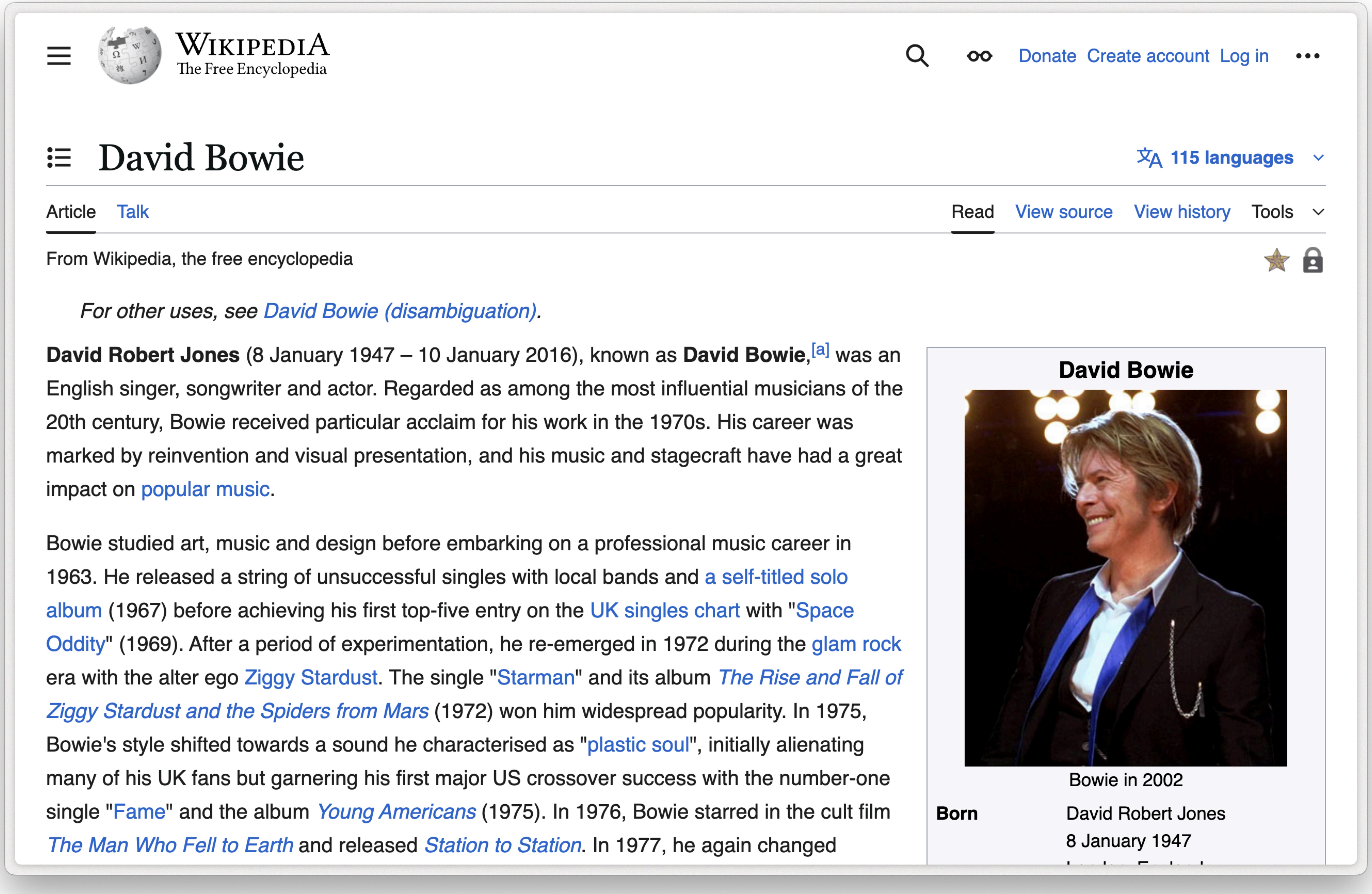
```python
#!/usr/bin/env python3

import fileinput
import re

prog = re.compile(r"((?:(?:[A-Z][a-z]+) )+)\(born .*([0-9]{4})\)")

for line in fileinput.input():
    m = prog.search(line)

    name = m.group(1).strip()
    year = m.group(2)
    print(f"(\"{name}\", \"born in\", \"{year}\")")
```

```
born in ([0-9]{4}|[0-9]+ (AD|BC))
born in ([0-9]{4}|[0-9]+ (AD|BC|CE|BCE))
```

born on the 8th of May, 1885

Another pattern:

```
born on .+ [0-9]{4}
```



David Bowie

From Wikipedia, the free encyclopedia

For other uses, see David Bowie (disambiguation).

David Robert Jones (8 January 1947 – 10 January 2016), known as **David Bowie**,[a] was an English singer, songwriter and actor. Regarded as among the most influential musicians of the 20th century, Bowie received particular acclaim for his work in the 1970s. His career was marked by reinvention and visual presentation, and his music and stagecraft have had a great impact on popular music.

Bowie studied art, music and design before embarking on a professional music career in 1963. He released a string of unsuccessful singles with local bands and a self-titled solo album (1967) before achieving his first top-five entry on the UK singles chart with "Space Oddity" (1969). After a period of experimentation, he re-emerged in 1972 during the glam rock era with the alter ego Ziggy Stardust. The single "Starman" and its album *The Rise and Fall of Ziggy Stardust and the Spiders from Mars* (1972) won him widespread popularity. In 1975, Bowie's style shifted towards a sound he characterised as "plastic soul", initially alienating many of his UK fans but garnering his first major US crossover success with the number-one single "Fame" and the album *Young Americans* (1975). In 1976, Bowie starred in the cult film *The Man Who Fell to Earth* and released *Station to Station*. In 1977, he again changed

David Bowie

Bowie in 2002

Born	David Robert Jones 8 January 1947

We could keep going!

These kind of searches let us learn lots of information that’s stated in text.

Which companies bought which other companies.

What state is a town in?

Which musicians made which albums?

As we work on this information retrieval program, we've been trying to fix two kinds of errors:

1 *False positives*: Matching strings that we shouldn't have matched (e.g., *born in humble circumstances*)

2 *False negatives*: Not matching things that we should have matched (e.g., *born on the first of January, 1901*)

Error types

	Actually says when someone was born	*Doesn’t actually says when someone was born*
Program thinks it says when someone was born	*True positive* ✓	*False positive* ✗
Program thinks it doesn’t say when someone was born	*False negative* ✗	*True negative* ✓

In NLP, we’re always dealing with these kinds of errors.

Reducing the error rate for an application often involves two antagonistic efforts:

- Increasing accuracy or precision (minimizing false positives)
- Increasing coverage or recall (minimizing false negatives)

Acknowledgments

This class incorporates material from:

- Carolyn Anderson, Wellesley College
- Katrin Erk, University of Texas at Austin
- Katie Keith, Williams College
- Xanda Schofield, Harvey Mudd College
- Jurafsky & Martin, *Speech and Language Processing*, 3rd ed. draft