# CS 111: Program Design I

## Lecture 10: Python Basics Concluded & File Intro

Robert H. Sloan & Richard Warner

University of Illinois at Chicago

Oct. 3, 2019

# FOR CONCLUDED (FOR NOW)

# Which this print #'s 1 through 3?

```
for x in range(1, 4):
   print(x)
   x = x + 1
```

A. Yes

~~B. No~~

~~C. Error~~

D. **Who cares which of A, B, or C because its style is so awful**

# Nagging counting

```
for j in range(42):
    print("Did YOU register on National Register to Vote
             Day?")

# It was about 1.5 weeks ago, and it's totally real
# You should have gotten email from UIC Chancellor
# about it
```

# This will print?

```
for x in range(0, 4, 3):
    print(x)
```

A. 3
B. 0, 3
C. 0, 1, 2, 3
D. 0, 1, 2, 3, 4
E. This will run forever

# More uses of for: Strings

- for i in range(len(st)):
  - makes i be each index of a character in a string
- if we just want the character and don't care about the position in the string, better to do
  - for char in st:
- However, sometimes we want to know index of character we are on

# vig again

- Needed to know where we were in key word to know how much to rotate
- One method:

```
key_index = 0
for char in plaintext:
    encrypt char: rotate by key[key_index]
    key_index = (key_index + 1) % len(key)
```

# Easier?

```
for i in range(len(plaintext)):
    encrypt plaintext[i] # rotate by key[i % len(key)]
```

# MORE ABOUT LISTS

# For web crawler

- Will need to have lists of things
  - e.g., pages we might still need to visit linked to from current webpage
- Just printing things out not our ultimate solution
- And string would be inconvenient way to store a whole collection of URLs

# Lists index/slice same as strings

```
          0      1              2
>>> ls = [1, 'Brennan', ['Joe', 'Donald'] ]

>>> len(ls)
3
>>> ls[1]
'Brennan'
>>>ls[3]
error!
>>> ls[2]
['Joe', 'Donald]
```

# Lists index/slice same as strings

```
          0      1             2
>>> ls = [1, 'Brennan', ['Joe', 'Donald'] ]
```

What is `ls[2][0]`?

A. 1

B. Syntax error

C. ['Joe', 'Donald']

D. 'Joe'

# Addition in lists (similar to strings)

```
>>> my_list = [17, 42, 52]
>>> my_list = my_list + 100
Barf
>>> new_list = my_list + [100]
>>> new_list
[17, 42, 52, 100]
>>> new_list = new_list + new_list
>>> new_list
[17, 42, 52, 100, 17, 42, 52, 100]
```

# Much of string stuff works for lists

- indexing
- slicing
- adding
- len
- in

- So you already know a lot about lists

# True, False or Error

- ls = [0, 'Brandeis', ['Joe', 'Donald']]
- 'Joe' in ls

A. True

B. False

C. Error

# List method append

- You can build up a list by adding as for strings, but can also use list method append:

```
>>> ls = [1, 3, 33]
>>> ls.append(17)
>>> ls
[1, 3, 33, 17]
```

# Pattern to build up list with append

```
ls = []
while still working on it:    # or perhaps for
    do some work to get item
    ls.append(item)
rest of function
```

# Pythonistas and very very very long lines

- Best practice is to keep individual line below 80 characters in length
  - Continuation *inside open* ()s is automatically same line. So:

```
def a_fun(zero, first, second, third,
              fourth, fifth, sixth):
    output = (first + second + third
          + fourth + fifth + sixth)
```

# Less stylish but if all else fails

- At end of line of Python code backslash character \ indicates continuation

```
VeryLong.left_hand_side \
    = even_longer.right_hand_side()
```

# HIERARCHICAL DECOMPOSITION

# Hierarchical Decomposition

- Solving problems by breaking them into subproblems
- One of a handful of popular methods for problem solving by programmers
- A good fit for Python

# Problem

- How do we count the number of each character of the alphabet in a string?

- (To break the Caesar cipher! E, T, and A, are the most common letters in English, in that order.)

# Problem

- How do we count the number of each character of the alphabet in a string?
  - I.e., how many A's, B's, C's, …, Y's and Z's?

- (To break Caesar cipher! E, T, and A, are most common letters in English, in that order.)

# Counting letters

- for each letter in the alphabet, count that letter in the string
- Subproblem identified
- So how do I count a letter's occurrence in a given string s?
    - Start my count at 0
    - go through string, check if I have that letter, and if so add 1 to my count

# for the subfunction, going to Python

```
def count_letter(letter, s):
    '''Counts # of letter in string s'''
    count = 0
    for char in s:
        if char == letter:
            count += 1
    return count
```

# Then

- Would have something that called that function for each letter in alphabet on string s
- Notice how close to code!

```
for each letter in alphabet
    count = count_letter(letter, s)
    print(letter, ': ', count)

# Assumes alphabet defined
```

# Bigger problem

- How do we build a web crawler?

# Crawl all pages reachable from start

- List of pages to visit, initially start
- Take a page from the list
- Get its text      # need to learn how to do this
- Get all the links in that page
- Add them to the list of pages to visit
- Keep going until list of pages to visit is empty

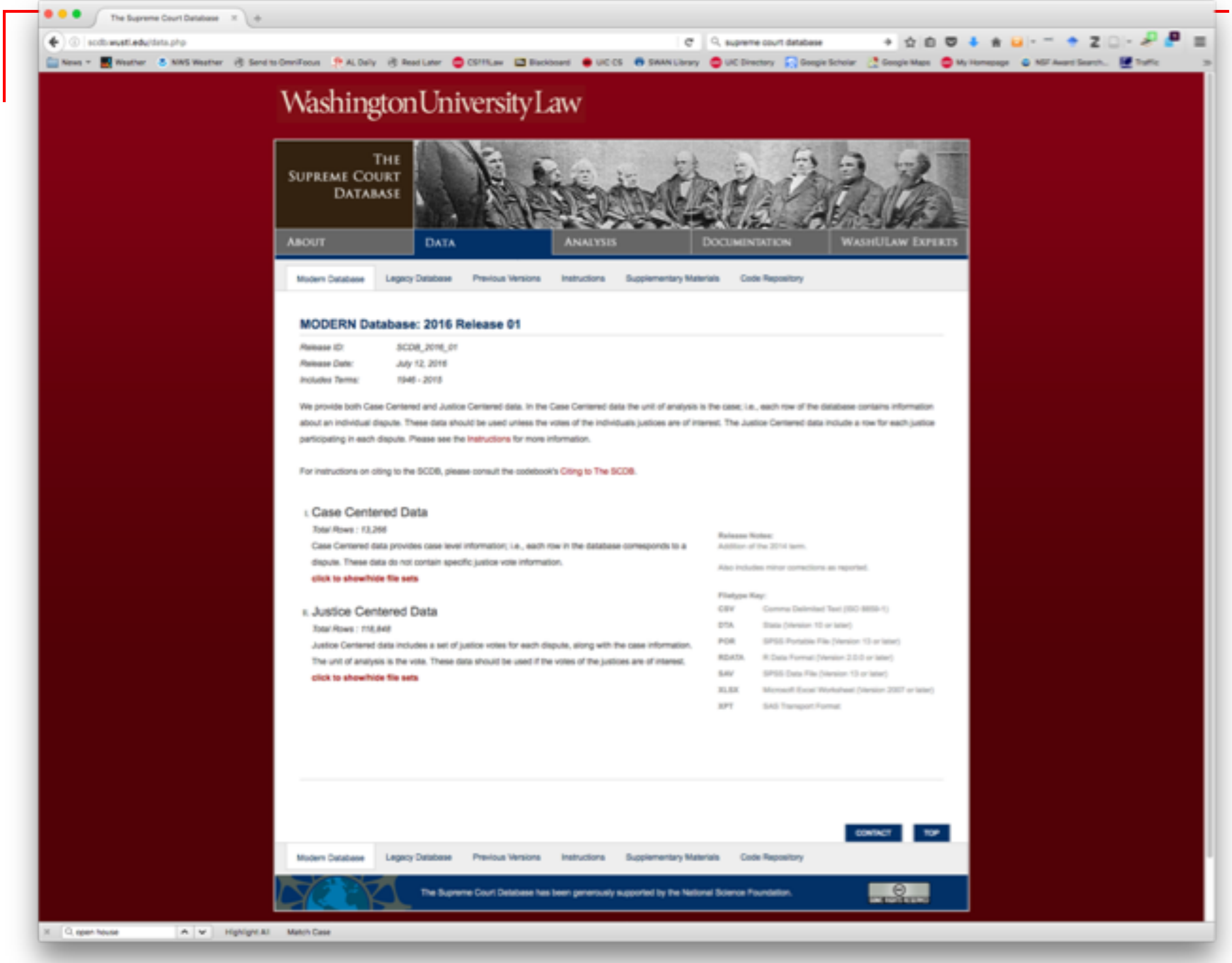
- Hmm. Not exactly right

# Crawl all pages reachable from start

- List of pages to visit, initially start
- while that list is not empty:
  - Take a page from the list
  - Get its text # need to learn how to do this
  - remove that page from to-visit list, add it to already-visited lsit
  - Get all the links in that page
  - for each link
    - if not already in visited list
    - add it
- Hmm. Much closer

# FILES

```
"caseId","docketId","caseIssuesId","voteId","dateDecision","decisionType","usCite","sctCite","ledCite","lexisCite","term","naturalCourt","chief","docket","caseName","dateArgument","dateRea
rg","petitioner","petitionerState","respondent","respondentState","jurisdiction","adminAction","adminActionState","threeJudgeFdc","caseOrigin","caseOriginState","caseSource","caseSourceS
tate","lcDisagreement","certReason","lcDisposition","lcDispositionDirection","declarationUncon","caseDisposition","caseDispositionUnusual","partyWinning","precedentAlteration","voteUncle
ar","issue","issueArea","decisionDirection","decisionDirectionDissent","authorityDecision1","authorityDecision2","lawType","lawSupp","lawMinor","majOpinWriter","majOpinAssigner","splitVot
e","majVotes","minVotes","justice","justiceName","vote","opinion","direction","majority","firstAgreement","secondAgreement"
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-01",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,86,"HHBurton",2,1,1,1,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-02",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,84,"RHJackson",1,1,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-03",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,81,"WODouglas",1,1,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-04",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,80,"FFrankfurter",4,2,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-05",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,79,"SFReed",1,1,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-06",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,78,"HLBlack",1,2,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-07",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,85,"WBRutledge",1,1,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-08",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,82,"FMurphy",1,1,2,2,,
"1946-001","1946-001-01","1946-001-01-01","1946-001-01-01-01-09",11/18/1946,1,"329 U.S. 1","67 S. Ct. 6","91 L. Ed. 3","1946 U.S. LEXIS
1724",1946,1301,"Vinson","24","HALLIBURTON OIL WELL CEMENTING CO. v. WALKER et al., DOING BUSINESS AS DEPTHOGRAPH
CO.",1/9/1946,10/23/1946,198,,172,,6,,,0,51,6,29,,0,11,2,1,1,3,0,1,1,0,80180,8,2,0,4,,6,600,"35 U.S.C. § 33",78,78,1,8,1,87,"FMVinson",1,1,2,2,,
"1946-002","1946-002-01","1946-002-01-01","1946-002-01-01-01-01",11/18/1946,1,"329 U.S. 14","67 S. Ct. 13","91 L. Ed. 12","1946 U.S. LEXIS
1725",1946,1301,"Vinson","12","CLEVELAND v. UNITED STATES",10/10/1945,10/17/1946,100,,27,,1,,,0,123,52,30,,0,4,2,1,1,2,0,0,0,0,10500,1,1,0,4,,6,600,"18 U.S.C. §
398",81,87,1,6,3,86,"HHBurton",1,1,1,2,,
"1946-002","1946-002-01","1946-002-01-01","1946-002-01-01-01-02",11/18/1946,1,"329 U.S. 14","67 S. Ct. 13","91 L. Ed. 12","1946 U.S. LEXIS
1725",1946,1301,"Vinson","12","CLEVELAND v. UNITED STATES",10/10/1945,10/17/1946,100,,27,,1,,,0,123,52,30,,0,4,2,1,1,2,0,0,0,0,10500,1,1,0,4,,6,600,"18 U.S.C. §
398",81,87,1,6,3,84,"RHJackson",2,3,2,1,,
"1946-002","1946-002-01","1946-002-01-01","1946-002-01-01-01-03",11/18/1946,1,"329 U.S. 14","67 S. Ct. 13","91 L. Ed. 12","1946 U.S. LEXIS
1725",1946,1301,"Vinson","12","CLEVELAND v. UNITED STATES",10/10/1945,10/17/1946,100,,27,,1,,,0,123,52,30,,0,4,2,1,1,2,0,0,0,0,10500,1,1,0,4,,6,600,"18 U.S.C. §
398",81,87,1,6,3,81,"WODouglas",1,2,1,2,,
"1946-002","1946-002-01","1946-002-01-01","1946-002-01-01-01-04",11/18/1946,1,"329 U.S. 14","67 S. Ct. 13","91 L. Ed. 12","1946 U.S. LEXIS
1725",1946,1301,"Vinson","12","CLEVELAND v. UNITED STATES",10/10/1945,10/17/1946,100,,27,,1,,,0,123,52,30,,0,4,2,1,1,2,0,0,0,0,10500,1,1,0,4,,6,600,"18 U.S.C. §
398",81,87,1,6,3,80,"FFrankfurter",1,1,1,2,,
"1946-002","1946-002-01","1946-002-01-01","1946-002-01-01-01-05",11/18/1946,1,"329 U.S. 14","67 S. Ct. 13","91 L. Ed. 12","1946 U.S. LEXIS
1725",1946,1301,"Vinson","12","CLEVELAND v. UNITED STATES",10/10/1945,10/17/1946,100,,27,,1,,,0,123,52,30,,0,4,2,1,1,2,0,0,0,0,10500,1,1,0,4,,6,600,"18 U.S.C. §
398",81,87,1,6,3,79,"SFReed",1,1,1,2,,
```



## MODERN Database: 2016 Release 01

| | |
|---|---|
| *Release ID:* | *SCDB_2016_01* |
| *Release Date:* | *July 12, 2016* |
| *Includes Terms:* | *1946 - 2015* |

We provide both Case Centered and Justice Centered data. In the Case Centered data the unit of analysis is the case; i.e., each row of the database contains information about an individual dispute. These data should be used unless the votes of the individuals justices are of interest. The Justice Centered data include a row for each justice participating in each dispute. Please see the Instructions for more information.

For instructions on citing to the SCDB, please consult the codebook's Citing to The SCDB.

i. Case Centered Data

*Total Rows : 13,266*

Case Centered data provides case level information; i.e., each row in the database corresponds to a dispute. These data do not contain specific justice vote information.

**click to show/hide file sets**

ii. Justice Centered Data

*Total Rows : 118,848*

Justice Centered data includes a set of justice votes for each dispute, along with the case information. The unit of analysis is the vote. These data should be used if the votes of the justices are of interest.

**click to show/hide file sets**

Release Notes:
Addition of the 2014 term.

Also includes minor corrections as reported.

Filetype Key:

| | |
|---|---|
| CSV | Comma Delimited Text (ISO 8859-1) |
| DTA | Stata (Version 10 or later) |
| POR | SPSS Portable File (Version 13 or later) |
| RDATA | R Data Format (Version 2.0.0 or later) |
| SAV | SPSS Data File (Version 13 or later) |
| XLSX | Microsoft Excel Worksheet (Version 2007 or later) |
| XPT | SAS Transport Format |

The Supreme Court Database has been generously supported by the National Science Foundation.

# Files

- Text
  - Collections of lines
    - Line = sequence of characters terminated by End Of Line (aka EOL, newline)
      - Can get End of Line explicitly with \n
- Binary: program (or zipped zip file)

# Open()

Start working with a file by using **open()** function.

Returns **file** object, most commonly used with two arguments.

Syntax:

***file_object = open(filename, mode)***

Mode:

- **'r'** file will only be read
- **'w'** only writing (an existing file with the same name will be *erased*)
- **'a'** for appending; any data written to the file is automatically added to the end.
- **'r+'** opens the file for both reading and writing.

# File functions (after creating file with open)

file.write(*String*)

file.read(*num_characters*)

file.readline()
file.readlines()

file.close()