OReilly - 2024-02February-22

February 22, 2024

1 Agenda: Day 3 (Dictionaries and files)

1. Q&A
2. Dictionaries
   - What are dictionaries?
   - Defining / retrieving
   - Modifying dictionaries
   - Accumulating using dictionaries
   - Accumulating the unknown
   - Looping over dicts
   - How do dictionaries work?
3. Files
   - What does it mean to work with files?
   - Reading from files
   - Turning files into data structures
   - Writing to files, as well (using `with`)

[1]:
```
# you can run both mylist += [80, 90, 100] and mylist += (50, 60, 70) and it␣
 ↪works for
# both in the same way. But you cannot run mylist.append['abc'] or
# mylist.append[40], why?
```

[2]:
```
mylist = [10, 20, 30]

# when we use += on a list, += looks to its right
# and runs a "for" loop on what it sees.  Each element we
# get in that loop is appended to mylist
mylist += [80, 90, 100]
```

[3]:
```
# the above is basically the same as saying

mylist = [10, 20, 30]

for one_item in [80, 90, 100]:
    mylist.append(one_item)

mylist
```

[3]: [10, 20, 30, 80, 90, 100]

[4]:
```
# let's try this with a tuple!

mylist = [10, 20, 30]

# += doesn't care what we have on the right side! It will run
# a for loop on whatever we give it.
mylist += (80, 90, 100)

mylist
```

[4]: [10, 20, 30, 80, 90, 100]

[5]:
```
# why can't we run mylist.append['abc'] or mylist.append[40]?

# append is a method
# we need to use () to invoke it
# whatever is in the () is appended

mylist.append('abc')
mylist
```

[5]: [10, 20, 30, 80, 90, 100, 'abc']

[6]:
```
# if you were to say

mylist.append['abc']

# the above means (to Python): Find mylist, and grab its "append" method. Then,␣
 ↪on the method,
# retrieve whatever is at index 'abc'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[6], line 3
      1 # if you were to say
----> 3 mylist.append['abc']
      5 # the above means (to Python): Find mylist, and grab its "append" metho.
 ↪ Then, on the method,
      6 # retrieve whatever is at index 'abc'

TypeError: 'builtin_function_or_method' object is not subscriptable
```

[7]:
```
mylist.append(40)
mylist
```

[7]:
```
[10, 20, 30, 80, 90, 100, 'abc', 40]
```

[ ]:
```
## From last session
# In google colab i ran the program to get a number from user as input and␣
 ↪display it
# 1 * 10 ** 3
# 2 * 10 ** 2
# 3 * 10 ** 1

# i didn't have to do index -1. when i ran

user_input = input(' enter a number :  ').strip()
for index,one_char in enumerate(user_input):
    print(f'{one_char} * 10 ** {len(user_input) - index}')

# output is
# 1 * 10 ** 3
# 2 * 10 ** 2
# 3 * 10 ** 1

# v/s if i do index - 1
# output is
# 1 * 10 ** 2
# 2 * 10 ** 1
# 3 * 10 ** 0
```

[ ]:
```
user_input = input(' enter a number :  ').strip()
for index,one_char in enumerate(user_input):
    print(f'{one_char} * 10 ** {len(user_input) - index}')
```

[2]:
```
user_input = '4321'   # this is 4*10**3  3*10**2  2*10**1   1*10**0

for index,one_char in enumerate(user_input):
    print(f'{one_char} * 10 ** {len(user_input) - index - 1}')
```

```
4 * 10 ** 3
3 * 10 ** 2
2 * 10 ** 1
1 * 10 ** 0
```

2 Dictionaries

Dictionaries (aka “dicts” in the Python world) are the most important data structure in Python. They aren’t unique to Python! Many other languages have a similar data structure, often called:

- hash tables
- hashes
- hash maps

- maps
- key-value stores
- name-value stores
- associative arrays

The basic idea behind a dictionary is as follows:

When we store things in a list, we determine the value, but Python normally determines the index. If we want, we can replace a value at a given index, but normally, we're just adding to the end of a list.

That's fine, except that the index numbers don't really have any meaning. What if I want to have a list of people in my company? Wouldn't it be better if I could store them not at arbitrary indexes (0, 1, 30, 256) but rather at indexes corresponding to their user IDs?

In other words: I'd like to determine not just the values, but also the indexes we use to store those values.

If I'm already wishing, I'd like to use not just integers, but also strings as indexes. Those could really come in handy.

What I've just described is (basically) a dictionary!

- We decide on both the keys (what we call the indexes in a dict) and the values
- The keys can be any immutable value (typically, integers and strings, but theoretically also floats and tuples)
- The values can be absolutely anything at all
- Each key must be unique in a dict, so each key can appear at most once

Dicts are incredibly efficient and fast, as well as convenient. They're a big win for everyone.

3 Defining dicts

- We define a dictionary using `{}` (curly braces)
- Each key-value pair is defined with a colon separating the key from the value
- The pairs are separated from one another using commas.
-

[3]:
```
d = {'a':10, 'b':20, 'c':30}

type(d)
```

[3]:
```
dict
```

[4]:
```
# how many pairs are in this dict?

len(d)
```

[4]:
```
3
```

[5]:
```
# what if I want to retrieve a value via a key?
d['a']   # put the key in square brackets
```

[5]:
```
10
```

[6]:
```
# what if I try to retrieve a key that doesn't exist?
d['x']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[6], line 2
      1 # what if I try to retrieve a key that doesn't exist?
----> 2 d['x']

KeyError: 'x'
```

[7]:
```
# how can I know if a key is in a dict?
# we can ask with "in"
# "in" *only* searches in the keys; it ignores the values

'a' in d
```

[7]:
```
True
```

[8]:
```
10 in d
```

[8]:
```
False
```

[9]:
```
# we can use variables

s = 'a'
d[s]   # this first evaluates s, getting back 'a', and then evaluates d['a']
```

[9]:
```
10
```

[10]:
```
# we saw that we can get a value via a key.
# can we do the opposite? The answer: no

# dicts are one way, from keys to values
# we could search through a dict, pair by pair, for a value and get its␣
 ↪corresonding key
# but if you're doing that, you're almost certainly doing something wrong

# remember: keys are unique, but values aren't (or don't have to be)
```

4 Exercise: Restaurant

1. Define a dict, `menu`, representing a menu in a restaurant. The keys will be the menu items (strings), and the values will be the prices of those items (integers).
2. Set a variable, `total`, to be 0.
3. Ask the user repeatedly to order something.
   - If they give us the empty string, stop asking and print the total.
   - If they give us the name of something on the menu (i.e., a key in our `menu` dict), then add the price to the total, and print the item, price, and new total.
   - If they give us the name of something *not* on the menu, then scold them lightly.
4. Print `total`.

Example:

```
Order: sandwich
sandwich costs 10, total is 10
Order: tea
tea costs 5, total is 15
Order: elephant
we are fresh out of elephant today!
Order: [ENTER]
Your total is 15
```

[ ]:

[ ]:

[11]:
```
menu = {'sandwich':10, 'tea':5, 'cookie':3, 'apple':2}
total = 0

while True:
    s = input('Order: ').strip()

    if s == '':   # empty string? break out of the loop
        break

    # is the user's input a key in our dict?
    if s in menu:
        price = menu[s]   # get the price for the user's order
        total += price    # increase total by this price
        print(f'{s} costs {price}; total is now {total}')

    else:
        print(f'Sorry, we are fresh out of {s} today.')

print(total)
```

```
Order:  sandwich
sandwich costs 10; total is now 10
```

```
Order:  tea

tea costs 5; total is now 15

Order:  elephant

Sorry, we are fresh out of elephant today.

Order:

15
```

5 When would I use a dict this way?

If you have key-value associations that might make sense throughout your program, you might well do this:

- Month names and month numbers
- Month numbers and month names
- User IDs and usernames
- Usernames and further data
- IP address and computer names

[12]:
```python
menu = {'sandwich':10.50, 'tea':5.75, 'cookie':3.25, 'apple':2.99}
total = 0

while True:
    s = input('Order: ').strip()

    if s == '':   # empty string? break out of the loop
        break

    # is the user's input a key in our dict?
    if s in menu:
        price = menu[s]   # get the price for the user's order
        total += price    # increase total by this price
        print(f'{s} costs {price}; total is now {total}')

    else:
        print(f'Sorry, we are fresh out of {s} today.')

print(total)
```

```
Order:  apple

apple costs 2.99; total is now 2.99

Order:  sandwich

sandwich costs 10.5; total is now 13.49

Order:
```

```
13.49
```

6 Mutable dictionaries

We've seen that some data structures in Python are immutable (e.g., `int`, `float`, `str`, and `tuple`). But we know that others are mutable (e.g., `list`). Which is true for dictionaries?

Answer: Dicts are mutable. They can be changed:

- We can modify existing key-value pairs
- We can add new key-value pairs
- We can remove existing key-value pairs

Every key needs to have a value, and every value needs to have a key. So there's no such thing as removing a key but keeping the value around, or of removing a value and keeping the key around.

[13]:
```
d = {'a':10, 'b':20, 'c':30}

# let's update the value associated with 'b'
d['b'] = 25   # assigning to an existing key updates the value for that key
d
```

[13]:
```
{'a': 10, 'b': 25, 'c': 30}
```

[14]:
```
# I can even use += with an existing dict value
d['b'] += 1  # this will add 1 to the existing value

d
```

[14]:
```
{'a': 10, 'b': 26, 'c': 30}
```

[15]:
```
# what happens if I try to update a value for a key that doesn't exist?

d['x'] += 1
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[15], line 3
      1 # what happens if I try to update a value for a key that doesn't exist?
----> 3 d['x'] += 1

KeyError: 'x'
```

[16]:
```
# what if I want to add a new key-value pair?
# we know, from lists, that we can use "append"
# BUT NOT IN DICTS!

# In dictionaries, to add a new key-value pair, all we do is assign
```

```
d['z'] = 1234

# it looks just like updating! But it's really adding a new key-value pair
```

[17]:
```
d
```

[17]:
```
{'a': 10, 'b': 26, 'c': 30, 'z': 1234}
```

[18]:
```
d['a', 'b'] += 1    # no, this doesn't work
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[18], line 1
----> 1 d['a', 'b'] += 1    # no, this doesn't work

KeyError: ('a', 'b')
```

[19]:
```
# Removing key-value pairs
# we can use the "pop" method to remove a key-value pair -- specify the key,␣
 ↪and the value is returned + removed

d.pop('z')
```

[19]:
```
1234
```

[20]:
```
d
```

[20]:
```
{'a': 10, 'b': 26, 'c': 30}
```

[21]:
```
# can we add multiple key-value pairs to a dict?
# yes -- we can do that by defining a second dict, and using the | (union)␣
 ↪method

d
```

[21]:
```
{'a': 10, 'b': 26, 'c': 30}
```

[22]:
```
new_stuff = {'c':77, 'd':88, 'e':99}

d | new_stuff    # this returns a new dict -- doens't affect either d or␣
 ↪new_stuff
```

[22]:
```
{'a': 10, 'b': 26, 'c': 77, 'd': 88, 'e': 99}
```

7 Next up

- Accumulating in dicts (known keys)

- Accumulating in dicts (unknown keys)

We've seen that we can define a dict and then use it as a read-only database inside of a program. But we can also use it as a read-write data structure.

One of the most common ways to use dicts in this format is to define it at the start of the program with some keys and some initial values, such as 0. Then, as the program proceeds, you add to the values.

In other words: The keys don't change (no new ones, no removals) but the values do.

[23]:
```
counts = {'a':0, 'e':0, 'i':0, 'o':0, 'u':0}

s = 'This is a bunch of letters for my course'

for one_character in s:
    if one_character in counts:      # if the current character is a key in our␣
 ↪dict
        counts[one_character] += 1   # add 1 to the value!

print(counts)
```

```
{'a': 1, 'e': 3, 'i': 2, 'o': 3, 'u': 2}
```

8 Exercise: Vowels, digits, and others (dict edition)

1. Define a dict in which the keys are `vowels`, `digits`, and `others`, and all values are 0.
2. Ask the user, repeatedly, to enter a string.
   - If they enter an empty string, stop asking
3. Go through each character in the string they gave you
   - If the character is a vowel, add 1 to `vowels`
   - If the character is a digit, add 1 to `digits`
   - Otherwise, add 1 to `others`
4. In the end, print the dict with the counts.

[24]:
```
counts = {'vowels':0,
          'digits':0,
          'others':0}

while True:
    s = input('Enter text: ').strip()

    if s == '':    # if we got the empty string, exit the loop
        break

    for one_character in s:
        if one_character in 'aeiou':
            counts['vowels'] += 1
        elif one_character.isdigit():
```

```
            counts['digits'] += 1
        else:
            counts['others'] += 1

print(counts)
```

```
Enter text:  hello!! 123
Enter text:  what about now?!? 456
Enter text:
```

```
{'vowels': 7, 'digits': 6, 'others': 19}
```

[ ]:
```
# AB

dicto={'vowels':0,'digits':0,'others':0}
while True:
    s = input("Enter a string? ").strip().lower()
    if s == '':
        break
    for one_character in s:
        if one_character in 'aeoiuy':
            dicto['vowels'] += 1
        elif one_character.isdigit():
            dicto['digits'] += 1
        else:
            dicto['others'] += 1
print(dicto)
```

[25]:
```
# MK
# can we define a dict

d = {'vowels':'aeiou'}

d['vowels']
```

[25]:
```
'aeiou'
```

[26]:
```
vowels = 'aeiou'
```

9 Loops and dicts

We know that we can iterate over a number of different data structures:

- string, we get one character at a time
- list, we get one element at a time
- tuple, we get one element at a time

This raises an obivous question: Can we iterate over dicts? If so, what do we get?

[27]:
```
d = {'a':10, 'b':20, 'c':30}

for one_item in d:
    print(one_item)
```

```
a
b
c
```

[28]:
```
# iterating over a dict gives us the dictionary's keys
# we can use this to iterate over the dict

for one_key in d:
    print(f'{one_key}: {d[one_key]}')
```

```
a: 10
b: 20
c: 30
```

[29]:
```
# some people discover that there is a "keys" method for dicts
# what am I going to get from this?

# exactly the same result as iterating over d
# **EXCEPT** that it's much slower

for one_key in d.keys():
    print(f'{one_key}: {d[one_key]}')
```

```
a: 10
b: 20
c: 30
```

[30]:
```
# there is also a dict.values method
# that returns all of the values in a special data structure that's sort of␣
 ↪(but not really) a list

d.values()
```

[30]:
```
dict_values([10, 20, 30])
```

[31]:
```
# if you want to search or iterate over the values, you can use this
```

[33]:
```
# there is a better way to iterate over a dict
# the dict.items method returns one key-value pair for each iteration

for t in d.items():  # get a (key, value) tuple with each iteration
    key, value = t   # unpacking to get the key and value
    print(f'{key}: {value}')
```

```
a: 10
b: 20
c: 30
```

[34]:
```
# we can use unpacking directly in the for loop!

for key, value  in d.items():  # get a (key, value) tuple with each iteration
    print(f'{key}: {value}')
```

```
a: 10
b: 20
c: 30
```

10 Paradigm 3 for dict usage

- In paradigm 1, we define a dict and treat it as a read-only database, never changing it
- In paradigm 2, we define a dict and modify the values, but not the keys (no new ones, no removal)
- In paradigm 3, we start with an *empty* dict, adding new keys as needed and updating values as needed

Paradigm 3 is perfect for when you don't know what keys or values you'll get, but you know what to do with them when you get them.

Imagine that you'll get user IDs and names for users on your system. You don't know in advance all of the users' names and IDs. But you can still define a dict and expect that when new users show up, you'll add a new pair to the dict.

[35]:
```
# counting characters
# what if I want to count *all* of the characters in a string?
# do I really want to define a new dict with all characters in its keys?
# instead, I'll start with an empty dict, and add new keys (characters) as we␣
 ↪encounter them.

counts = {}

s = 'This is another amazing sentence that I can use in my Python course'

for one_character in s:
    if one_character in counts:
        counts[one_character] += 1   # add 1 to the count if we've seen it␣
 ↪before
    else:
        counts[one_character] = 1    # otherwise, start the count with 1

for key, value in counts.items():
    print(f'{key}: {value}')
```

```
T: 1
```

```
h: 4
i: 4
s: 5
 : 12
a: 5
n: 7
o: 3
t: 5
e: 6
r: 2
m: 2
z: 1
g: 1
c: 3
I: 1
u: 2
y: 2
P: 1
```

11 Exercise: Rainfall

1. Define an empty dict, `rainfall`. Each key-value pair that we add will be a string (a key, the name of a city) and an integer (a value, an amount of rain, in mm, that fell there).
2. Ask the user to repeatedly enter the name of a city.
   - If we get an empty string, exit from the loop.
3. If we got a city name, ask the user how many rain fell there?
4. Check:
   - If the city already exists in the dict as a key, add the rainfall to the existing value
   - If the city does *not* exist in the dict, then add a new key-value pair, the city and the rainfall
5. Iterate over our dict, printing the cities and amounts.

Example:

```
City: a
Rain: 5
City: b
Rain: 4
City: a
Rain: 3
City: [ENTER]

a: 8
b: 4
```

[40]:
```
rainfall = {}

while True:
```

```
    city_name = input('City: ').strip()

    if city_name == '':
        break

    mm_rain = input('Rain: ').strip()
    mm_rain = int(mm_rain)

    if city_name in rainfall:
        rainfall[city_name] += mm_rain   # this is good if we've seen the city before
    else:
        rainfall[city_name] = mm_rain    # this is good if the city is new

print(rainfall)
```

```
City:  a
Rain:  5
City:  b
Rain:  4
City:  a
Rain:  3
City:
```

```
{'a': 8, 'b': 4}
```

[41]:
```
for key, value in rainfall.items():
    print(f'{key}: {value}')
```

```
a: 8
b: 4
```

[ ]:
```
a
b

5
4
3
```

[42]:
```
# when we iterate over the "items" method in a dict,
# we're getting a new (key, value) pair with each iteration
# we can capture each pair with two variables, as I've done here:

for key, value in rainfall.items():
    print(f'{key}: {value}')
```

```
a: 8
b: 4
```

[43]:
```
# if we iterate over the dict (i.e., just the dict itself),
# then we get the keys, and not the values. So trying to use
# two variables will probably fail, and if it works, it'll be bad.

for key, value in rainfall:
    print(f'{key}: {value}')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[43], line 5
      1 # if we iterate over the dict (i.e., just the dict itself),
      2 # then we get the keys, and not the values. So trying to use
      3 # two variables will probably fail, and if it works, it'll be bad.
----> 5 for key, value in rainfall:
      6     print(f'{key}: {value}')

ValueError: not enough values to unpack (expected 2, got 1)
```

[44]:
```
# if I declare a variable, but I don't give it a value, what value does it have?

# you can't declare a variable in Python without giving it a value
# the way we define a variable for the first time is by assigning to it

# there is a None value in Python, and that's sorta kinda like undefined/null␣
 ↪in other languages
```

12 Next up

1. How dicts work
2. Beginning of working with files

You'll want to download some files onto your system for us to practice with. They are available from here:

https://files.lerner.co.il/exercise-files.zip

13 How do dicts work?

There are two important things to know about dicts:

1. We can use any immutable Python value as a key. We have great flexibility in defining our dict.
2. If we search for a value in a list, the amount of time it takes to search depends on the length of the list – the longer the list, the longer it takes. By contrast, the size of the dictionary has very little impact on how long it takes to find a key in a dict.

How is that possible? How can we have both fast searching and any key we want?

The answer is: A hash function. This is a special function that takes the key as an input, and returns an integer as an output. The integer we get back tells Python where to store the key-value pair.

This means that if I try to store `d['a'] = 10` in my dict, Python runs `hash('a')`, which returns a number, which tells it where to store the key-value pair. Then if I run `'a' in d`, Python runs `hash('a')`, knows where to go, and finds out if our key-value pair is there.

[45]:
```
d = {'a':10, 'b':20, 'c':30}

d['a'] += 5  # add 5 to whatever is already in d['a']
d
```

[45]:
```
{'a': 15, 'b': 20, 'c': 30}
```

[46]:
```
d['x'] += 5  # can I do this?  No, because 'x' isn't (yet) a key in d
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[46], line 1
----> 1 d['x'] += 5  # can I do this?  No, because 'x' isn't (yet) a key in d

KeyError: 'x'
```

In general, if I say

```
x += 1
```

that's the same as saying

```
x = x + 1
```

And if I say

```
d['x'] += 1
```

that's the same as

```
d['x'] = d['x'] + 1   # as you can see, d['x'] needs to already exist!
```

14 Files

We use files every single day with our computers. But what are they?

Files are a way for us to take data structures in the memory of our computer and store them when the program/computer isn't around, or on. In the same way, I can take the file to another computer and use it there, instead of on the orginal computer where I created it.

There are lots of types of files. The easiest to deal with contain only text. (Not Word, Excel, PowerPoint, or PDF.)

Let's say a program wants to work with a file on the disk. Can it?

Yes, but indirectly. A program has to go through the operating system, which acts as a traffic cop.

Disks are (even at their fastest) thousands of times slower than our computers. If every time we want to read from a file, we actually do – or every time we want to write to a file, we actually do – the computer would get very slow.

The OS makes sure that these operations happen in a way that doesn't interfere with the program's running.

To work with a file, we ask the OS to give us an agent, or a handhold, that we can use to work with the file. Everything we want to do, we do via that agent. That ensures the OS can continue to monitor things.

14.0.1 In practice

To work with a file, we use the `open` function. By default, it assumes that we want to read from a file. We get back a "file object," or a "file-like object" (as they are now called). Via that file object, we can read from the file.

I asked you to download a zipfile with several text files we'll be using in this class. My suggestion is that you put them in the same directory as you're running Jupyter or (if you're not using it) in the same directory as the Python programs you're writing/running.

[48]:
```
# I'm going to start with a standard file that exists on Unix systems
# it's called /etc/passwd

f = open('/etc/passwd')

type(f)  # what kind of data structure is this?  (text file object, with a␣
 ↪fancy name)
```

[48]:
```
_io.TextIOWrapper
```

[49]:
```
# what is your printed representation?
f
```

[49]:
```
<_io.TextIOWrapper name='/etc/passwd' mode='r' encoding='UTF-8'>
```

[50]:
```
# how can I read from the file?
# option 1: use the "read" method

# read returns a string, the entire contents of the file

s = f.read()
```

[51]:
```
# let's display the file!
print(s)
```

```
##
# User Database
#
```

```
# Note that this file is consulted directly only when the system is running
# in single-user mode.  At other times this information is provided by
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information about
# Open Directory.
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/uucico
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install Assistant:/var/empty:/usr/bin/false
_lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
_postfix:*:27:27:Postfix Mail Server:/var/spool/postfix:/usr/bin/false
_scsd:*:31:31:Service Configuration Service:/var/empty:/usr/bin/false
_ces:*:32:32:Certificate Enrollment Service:/var/empty:/usr/bin/false
_appstore:*:33:33:Mac App Store Service:/var/db/appstore:/usr/bin/false
_mcxalr:*:54:54:MCX AppLaunch:/var/empty:/usr/bin/false
_appleevents:*:55:55:AppleEvents Daemon:/var/empty:/usr/bin/false
_geod:*:56:56:Geo Services Daemon:/var/db/geod:/usr/bin/false
_devdocs:*:59:59:Developer Documentation:/var/empty:/usr/bin/false
_sandbox:*:60:60:Seatbelt:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65:mDNSResponder:/var/empty:/usr/bin/false
_ard:*:67:67:Apple Remote Desktop:/var/empty:/usr/bin/false
_www:*:70:70:World Wide Web Server:/Library/WebServer:/usr/bin/false
_eppc:*:71:71:Apple Events User:/var/empty:/usr/bin/false
_cvs:*:72:72:CVS Server:/var/empty:/usr/bin/false
_svn:*:73:73:SVN Server:/var/empty:/usr/bin/false
_mysql:*:74:74:MySQL Server:/var/empty:/usr/bin/false
_sshd:*:75:75:sshd Privilege separation:/var/empty:/usr/bin/false
_qtss:*:76:76:QuickTime Streaming Server:/var/empty:/usr/bin/false
_cyrus:*:77:6:Cyrus Administrator:/var/imap:/usr/bin/false
_mailman:*:78:78:Mailman List Server:/var/empty:/usr/bin/false
_appserver:*:79:79:Application Server:/var/empty:/usr/bin/false
_clamav:*:82:82:ClamAV Daemon:/var/virusmails:/usr/bin/false
_amavisd:*:83:83:AMaViS Daemon:/var/virusmails:/usr/bin/false
_jabber:*:84:84:Jabber XMPP Server:/var/empty:/usr/bin/false
_appowner:*:87:87:Application Owner:/var/empty:/usr/bin/false
_windowserver:*:88:88:WindowServer:/var/empty:/usr/bin/false
_spotlight:*:89:89:Spotlight:/var/empty:/usr/bin/false
_tokend:*:91:91:Token Daemon:/var/empty:/usr/bin/false
_securityagent:*:92:92:SecurityAgent:/var/db/securityagent:/usr/bin/false
_calendar:*:93:93:Calendar:/var/empty:/usr/bin/false
_teamsserver:*:94:94:TeamsServer:/var/teamsserver:/usr/bin/false
_update_sharing:*:95:-2:Update Sharing:/var/empty:/usr/bin/false
_installer:*:96:-2:Installer:/var/empty:/usr/bin/false
```

```
_atsserver:*:97:97:ATS Server:/var/empty:/usr/bin/false
_ftp:*:98:-2:FTP Daemon:/var/empty:/usr/bin/false
_unknown:*:99:99:Unknown User:/var/empty:/usr/bin/false
_softwareupdate:*:200:200:Software Update
Service:/var/db/softwareupdate:/usr/bin/false
_coreaudiod:*:202:202:Core Audio Daemon:/var/empty:/usr/bin/false
_screensaver:*:203:203:Screensaver:/var/empty:/usr/bin/false
_locationd:*:205:205:Location Daemon:/var/db/locationd:/usr/bin/false
_trustevaluationagent:*:208:208:Trust Evaluation Agent:/var/empty:/usr/bin/false
_timezone:*:210:210:AutoTimeZoneDaemon:/var/empty:/usr/bin/false
_lda:*:211:211:Local Delivery Agent:/var/empty:/usr/bin/false
_cvmsroot:*:212:212:CVMS Root:/var/empty:/usr/bin/false
_usbmuxd:*:213:213:iPhone OS Device Helper:/var/db/lockdown:/usr/bin/false
_dovecot:*:214:6:Dovecot Administrator:/var/empty:/usr/bin/false
_dpaudio:*:215:215:DP Audio:/var/empty:/usr/bin/false
_postgres:*:216:216:PostgreSQL Server:/var/empty:/usr/bin/false
_krbtgt:*:217:-2:Kerberos Ticket Granting Ticket:/var/empty:/usr/bin/false
_kadmin_admin:*:218:-2:Kerberos Admin Service:/var/empty:/usr/bin/false
_kadmin_changepw:*:219:-2:Kerberos Change Password
Service:/var/empty:/usr/bin/false
_devicemgr:*:220:220:Device Management Server:/var/empty:/usr/bin/false
_webauthserver:*:221:221:Web Auth Server:/var/empty:/usr/bin/false
_netbios:*:222:222:NetBIOS:/var/empty:/usr/bin/false
_warmd:*:224:224:Warm Daemon:/var/empty:/usr/bin/false
_dovenull:*:227:227:Dovecot Authentication:/var/empty:/usr/bin/false
_netstatistics:*:228:228:Network Statistics Daemon:/var/empty:/usr/bin/false
_avbdeviced:*:229:-2:Ethernet AVB Device Daemon:/var/empty:/usr/bin/false
_krb_krbtgt:*:230:-2:Open Directory Kerberos Ticket Granting
Ticket:/var/empty:/usr/bin/false
_krb_kadmin:*:231:-2:Open Directory Kerberos Admin
Service:/var/empty:/usr/bin/false
_krb_changepw:*:232:-2:Open Directory Kerberos Change Password
Service:/var/empty:/usr/bin/false
_krb_kerberos:*:233:-2:Open Directory Kerberos:/var/empty:/usr/bin/false
_krb_anonymous:*:234:-2:Open Directory Kerberos
Anonymous:/var/empty:/usr/bin/false
_assetcache:*:235:235:Asset Cache Service:/var/empty:/usr/bin/false
_coremediaiod:*:236:236:Core Media IO Daemon:/var/empty:/usr/bin/false
_launchservicesd:*:239:239:_launchservicesd:/var/empty:/usr/bin/false
_iconservices:*:240:240:IconServices:/var/empty:/usr/bin/false
_distnote:*:241:241:DistNote:/var/empty:/usr/bin/false
_nsurlsessiond:*:242:242:NSURLSession
Daemon:/var/db/nsurlsessiond:/usr/bin/false
_displaypolicyd:*:244:244:Display Policy Daemon:/var/empty:/usr/bin/false
_astris:*:245:245:Astris Services:/var/db/astris:/usr/bin/false
_krbfast:*:246:-2:Kerberos FAST Account:/var/empty:/usr/bin/false
_gamecontrollerd:*:247:247:Game Controller Daemon:/var/empty:/usr/bin/false
_mbsetupuser:*:248:248:Setup User:/var/setup:/bin/bash
```

```
_ondemand:*:249:249:On Demand Resource Daemon:/var/db/ondemand:/usr/bin/false
_xserverdocs:*:251:251:macOS Server Documents Service:/var/empty:/usr/bin/false
_wwwproxy:*:252:252:WWW Proxy:/var/empty:/usr/bin/false
_mobileasset:*:253:253:MobileAsset User:/var/ma:/usr/bin/false
_findmydevice:*:254:254:Find My Device
Daemon:/var/db/findmydevice:/usr/bin/false
_datadetectors:*:257:257:DataDetectors:/var/db/datadetectors:/usr/bin/false
_captiveagent:*:258:258:captiveagent:/var/empty:/usr/bin/false
_ctkd:*:259:259:ctkd Account:/var/empty:/usr/bin/false
_applepay:*:260:260:applepay Account:/var/db/applepay:/usr/bin/false
_hidd:*:261:261:HID Service User:/var/db/hidd:/usr/bin/false
_cmiodalassistants:*:262:262:CoreMedia IO Assistants
User:/var/db/cmiodalassistants:/usr/bin/false
_analyticsd:*:263:263:Analytics Daemon:/var/db/analyticsd:/usr/bin/false
_fpsd:*:265:265:FPS Daemon:/var/db/fpsd:/usr/bin/false
_timed:*:266:266:Time Sync Daemon:/var/db/timed:/usr/bin/false
_nearbyd:*:268:268:Proximity and Ranging Daemon:/var/db/nearbyd:/usr/bin/false
_reportmemoryexception:*:269:269:ReportMemoryException:/var/db/reportmemoryexcep
tion:/usr/bin/false
_driverkit:*:270:270:DriverKit:/var/empty:/usr/bin/false
_diskimagesiod:*:271:271:DiskImages IO
Daemon:/var/db/diskimagesiod:/usr/bin/false
_logd:*:272:272:Log Daemon:/var/db/diagnostics:/usr/bin/false
_appinstalld:*:273:273:App Install Daemon:/var/db/appinstalld:/usr/bin/false
_installcoordinationd:*:274:274:Install Coordination
Daemon:/var/db/installcoordinationd:/usr/bin/false
_demod:*:275:275:Demo Daemon:/var/empty:/usr/bin/false
_rmd:*:277:277:Remote Management Daemon:/var/db/rmd:/usr/bin/false
_accessoryupdater:*:278:278:Accessory Update
Daemon:/var/db/accessoryupdater:/usr/bin/false
_knowledgegraphd:*:279:279:Knowledge Graph
Daemon:/var/db/knowledgegraphd:/usr/bin/false
_coreml:*:280:280:CoreML Services:/var/db/coreml:/usr/bin/false
_sntpd:*:281:281:SNTP Server Daemon:/var/empty:/usr/bin/false
_trustd:*:282:282:trustd:/var/empty:/usr/bin/false
_mmaintenanced:*:283:283:mmaintenanced:/var/db/mmaintenanced:/usr/bin/false
_darwindaemon:*:284:284:Darwin Daemon:/var/db/darwindaemon:/usr/bin/false
_notification_proxy:*:285:285:Notification Proxy:/var/empty:/usr/bin/false
_avphidbridge:*:288:288:Apple Virtual Platform HID
Bridge:/var/empty:/usr/bin/false
_biome:*:289:289:Biome:/var/db/biome:/usr/bin/false
_backgroundassets:*:291:291:Background Assets Service:/var/empty:/usr/bin/false
_mobilegestalthelper:*:293:293:MobileGestaltHelper:/var/empty:/usr/bin/false
_audiomxd:*:294:294:Audio and MediaExperience
Daemon:/var/db/audiomxd:/usr/bin/false
_terminusd:*:295:295:Terminus:/var/empty:/usr/bin/false
_neuralengine:*:296:296:AppleNeuralEngine:/var/db/neuralengine:/usr/bin/false
_oahd:*:441:441:OAH Daemon:/var/empty:/usr/bin/false
```

15 Don't use `read` in this way!

It's great that `read` returns a string with the file's contents.

But there are two problems:

1. It's a bit unweildy to work with a file in one big string.
2. What if the file we want to read is 10 TB in size? If you were to use `read` to grab such a file, it would kill off your Python program, and maybe give your entire computer a bad experience.

(If it's a short file, then that's probably fine.)

You can invoke `read` with an integer argument, telling it how many (max) characters to read. But then you have to deal with finding line endings, word endings, etc.

[53]:
```
# option 2: a for loop

# if you iterate over a file object in a "for" loop, you get each line of the␣
 ↪file,
# one at a time. The loop returns a string with each iteration, each string␣
 ↪ending with \n

f = open('/etc/passwd')       # this is an "absolute path," starting with /.
for one_line in f:            # get each line of the file, one at a time
    print(one_line.strip())   # print the line
```

```
##
# User Database
#
# Note that this file is consulted directly only when the system is running
# in single-user mode.  At other times this information is provided by
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information about
# Open Directory.
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/uucico
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install Assistant:/var/empty:/usr/bin/false
_lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
_postfix:*:27:27:Postfix Mail Server:/var/spool/postfix:/usr/bin/false
_scsd:*:31:31:Service Configuration Service:/var/empty:/usr/bin/false
_ces:*:32:32:Certificate Enrollment Service:/var/empty:/usr/bin/false
_appstore:*:33:33:Mac App Store Service:/var/db/appstore:/usr/bin/false
```

```
_mcxalr:*:54:54:MCX AppLaunch:/var/empty:/usr/bin/false
_appleevents:*:55:55:AppleEvents Daemon:/var/empty:/usr/bin/false
_geod:*:56:56:Geo Services Daemon:/var/db/geod:/usr/bin/false
_devdocs:*:59:59:Developer Documentation:/var/empty:/usr/bin/false
_sandbox:*:60:60:Seatbelt:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65:mDNSResponder:/var/empty:/usr/bin/false
_ard:*:67:67:Apple Remote Desktop:/var/empty:/usr/bin/false
_www:*:70:70:World Wide Web Server:/Library/WebServer:/usr/bin/false
_eppc:*:71:71:Apple Events User:/var/empty:/usr/bin/false
_cvs:*:72:72:CVS Server:/var/empty:/usr/bin/false
_svn:*:73:73:SVN Server:/var/empty:/usr/bin/false
_mysql:*:74:74:MySQL Server:/var/empty:/usr/bin/false
_sshd:*:75:75:sshd Privilege separation:/var/empty:/usr/bin/false
_qtss:*:76:76:QuickTime Streaming Server:/var/empty:/usr/bin/false
_cyrus:*:77:6:Cyrus Administrator:/var/imap:/usr/bin/false
_mailman:*:78:78:Mailman List Server:/var/empty:/usr/bin/false
_appserver:*:79:79:Application Server:/var/empty:/usr/bin/false
_clamav:*:82:82:ClamAV Daemon:/var/virusmails:/usr/bin/false
_amavisd:*:83:83:AMaViS Daemon:/var/virusmails:/usr/bin/false
_jabber:*:84:84:Jabber XMPP Server:/var/empty:/usr/bin/false
_appowner:*:87:87:Application Owner:/var/empty:/usr/bin/false
_windowserver:*:88:88:WindowServer:/var/empty:/usr/bin/false
_spotlight:*:89:89:Spotlight:/var/empty:/usr/bin/false
_tokend:*:91:91:Token Daemon:/var/empty:/usr/bin/false
_securityagent:*:92:92:SecurityAgent:/var/db/securityagent:/usr/bin/false
_calendar:*:93:93:Calendar:/var/empty:/usr/bin/false
_teamsserver:*:94:94:TeamsServer:/var/teamsserver:/usr/bin/false
_update_sharing:*:95:-2:Update Sharing:/var/empty:/usr/bin/false
_installer:*:96:-2:Installer:/var/empty:/usr/bin/false
_atsserver:*:97:97:ATS Server:/var/empty:/usr/bin/false
_ftp:*:98:-2:FTP Daemon:/var/empty:/usr/bin/false
_unknown:*:99:99:Unknown User:/var/empty:/usr/bin/false
_softwareupdate:*:200:200:Software Update
Service:/var/db/softwareupdate:/usr/bin/false
_coreaudiod:*:202:202:Core Audio Daemon:/var/empty:/usr/bin/false
_screensaver:*:203:203:Screensaver:/var/empty:/usr/bin/false
_locationd:*:205:205:Location Daemon:/var/db/locationd:/usr/bin/false
_trustevaluationagent:*:208:208:Trust Evaluation Agent:/var/empty:/usr/bin/false
_timezone:*:210:210:AutoTimeZoneDaemon:/var/empty:/usr/bin/false
_lda:*:211:211:Local Delivery Agent:/var/empty:/usr/bin/false
_cvmsroot:*:212:212:CVMS Root:/var/empty:/usr/bin/false
_usbmuxd:*:213:213:iPhone OS Device Helper:/var/db/lockdown:/usr/bin/false
_dovecot:*:214:6:Dovecot Administrator:/var/empty:/usr/bin/false
_dpaudio:*:215:215:DP Audio:/var/empty:/usr/bin/false
_postgres:*:216:216:PostgreSQL Server:/var/empty:/usr/bin/false
_krbtgt:*:217:-2:Kerberos Ticket Granting Ticket:/var/empty:/usr/bin/false
_kadmin_admin:*:218:-2:Kerberos Admin Service:/var/empty:/usr/bin/false
_kadmin_changepw:*:219:-2:Kerberos Change Password
```

```
Service:/var/empty:/usr/bin/false
_devicemgr:*:220:220:Device Management Server:/var/empty:/usr/bin/false
_webauthserver:*:221:221:Web Auth Server:/var/empty:/usr/bin/false
_netbios:*:222:222:NetBIOS:/var/empty:/usr/bin/false
_warmd:*:224:224:Warm Daemon:/var/empty:/usr/bin/false
_dovenull:*:227:227:Dovecot Authentication:/var/empty:/usr/bin/false
_netstatistics:*:228:228:Network Statistics Daemon:/var/empty:/usr/bin/false
_avbdeviced:*:229:-2:Ethernet AVB Device Daemon:/var/empty:/usr/bin/false
_krb_krbtgt:*:230:-2:Open Directory Kerberos Ticket Granting
Ticket:/var/empty:/usr/bin/false
_krb_kadmin:*:231:-2:Open Directory Kerberos Admin
Service:/var/empty:/usr/bin/false
_krb_changepw:*:232:-2:Open Directory Kerberos Change Password
Service:/var/empty:/usr/bin/false
_krb_kerberos:*:233:-2:Open Directory Kerberos:/var/empty:/usr/bin/false
_krb_anonymous:*:234:-2:Open Directory Kerberos
Anonymous:/var/empty:/usr/bin/false
_assetcache:*:235:235:Asset Cache Service:/var/empty:/usr/bin/false
_coremediaiod:*:236:236:Core Media IO Daemon:/var/empty:/usr/bin/false
_launchservicesd:*:239:239:_launchservicesd:/var/empty:/usr/bin/false
_iconservices:*:240:240:IconServices:/var/empty:/usr/bin/false
_distnote:*:241:241:DistNote:/var/empty:/usr/bin/false
_nsurlsessiond:*:242:242:NSURLSession
Daemon:/var/db/nsurlsessiond:/usr/bin/false
_displaypolicyd:*:244:244:Display Policy Daemon:/var/empty:/usr/bin/false
_astris:*:245:245:Astris Services:/var/db/astris:/usr/bin/false
_krbfast:*:246:-2:Kerberos FAST Account:/var/empty:/usr/bin/false
_gamecontrollerd:*:247:247:Game Controller Daemon:/var/empty:/usr/bin/false
_mbsetupuser:*:248:248:Setup User:/var/setup:/bin/bash
_ondemand:*:249:249:On Demand Resource Daemon:/var/db/ondemand:/usr/bin/false
_xserverdocs:*:251:251:macOS Server Documents Service:/var/empty:/usr/bin/false
_wwwproxy:*:252:252:WWW Proxy:/var/empty:/usr/bin/false
_mobileasset:*:253:253:MobileAsset User:/var/ma:/usr/bin/false
_findmydevice:*:254:254:Find My Device
Daemon:/var/db/findmydevice:/usr/bin/false
_datadetectors:*:257:257:DataDetectors:/var/db/datadetectors:/usr/bin/false
_captiveagent:*:258:258:captiveagent:/var/empty:/usr/bin/false
_ctkd:*:259:259:ctkd Account:/var/empty:/usr/bin/false
_applepay:*:260:260:applepay Account:/var/db/applepay:/usr/bin/false
_hidd:*:261:261:HID Service User:/var/db/hidd:/usr/bin/false
_cmiodalassistants:*:262:262:CoreMedia IO Assistants
User:/var/db/cmiodalassistants:/usr/bin/false
_analyticsd:*:263:263:Analytics Daemon:/var/db/analyticsd:/usr/bin/false
_fpsd:*:265:265:FPS Daemon:/var/db/fpsd:/usr/bin/false
_timed:*:266:266:Time Sync Daemon:/var/db/timed:/usr/bin/false
_nearbyd:*:268:268:Proximity and Ranging Daemon:/var/db/nearbyd:/usr/bin/false
_reportmemoryexception:*:269:269:ReportMemoryException:/var/db/reportmemoryexcep
tion:/usr/bin/false
```

```
_driverkit:*:270:270:DriverKit:/var/empty:/usr/bin/false
_diskimagesiod:*:271:271:DiskImages IO
Daemon:/var/db/diskimagesiod:/usr/bin/false
_logd:*:272:272:Log Daemon:/var/db/diagnostics:/usr/bin/false
_appinstalld:*:273:273:App Install Daemon:/var/db/appinstalld:/usr/bin/false
_installcoordinationd:*:274:274:Install Coordination
Daemon:/var/db/installcoordinationd:/usr/bin/false
_demod:*:275:275:Demo Daemon:/var/empty:/usr/bin/false
_rmd:*:277:277:Remote Management Daemon:/var/db/rmd:/usr/bin/false
_accessoryupdater:*:278:278:Accessory Update
Daemon:/var/db/accessoryupdater:/usr/bin/false
_knowledgegraphd:*:279:279:Knowledge Graph
Daemon:/var/db/knowledgegraphd:/usr/bin/false
_coreml:*:280:280:CoreML Services:/var/db/coreml:/usr/bin/false
_sntpd:*:281:281:SNTP Server Daemon:/var/empty:/usr/bin/false
_trustd:*:282:282:trustd:/var/empty:/usr/bin/false
_mmaintenanced:*:283:283:mmaintenanced:/var/db/mmaintenanced:/usr/bin/false
_darwindaemon:*:284:284:Darwin Daemon:/var/db/darwindaemon:/usr/bin/false
_notification_proxy:*:285:285:Notification Proxy:/var/empty:/usr/bin/false
_avphidbridge:*:288:288:Apple Virtual Platform HID
Bridge:/var/empty:/usr/bin/false
_biome:*:289:289:Biome:/var/db/biome:/usr/bin/false
_backgroundassets:*:291:291:Background Assets Service:/var/empty:/usr/bin/false
_mobilegestalthelper:*:293:293:MobileGestaltHelper:/var/empty:/usr/bin/false
_audiomxd:*:294:294:Audio and MediaExperience
Daemon:/var/db/audiomxd:/usr/bin/false
_terminusd:*:295:295:Terminus:/var/empty:/usr/bin/false
_neuralengine:*:296:296:AppleNeuralEngine:/var/db/neuralengine:/usr/bin/false
_oahd:*:441:441:OAH Daemon:/var/empty:/usr/bin/false
```

[54]:
```
for one_line in open('/etc/passwd'):      # get each line of the file, one at a␣
 ↪time
    print(one_line.strip())               # print the line
```

```
##
# User Database
#
# Note that this file is consulted directly only when the system is running
# in single-user mode.  At other times this information is provided by
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information about
# Open Directory.
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/uucico
```

```
_taskgated:*:13:13:Task Gate Daemon:/var/empty:/usr/bin/false
_networkd:*:24:24:Network Services:/var/networkd:/usr/bin/false
_installassistant:*:25:25:Install Assistant:/var/empty:/usr/bin/false
_lp:*:26:26:Printing Services:/var/spool/cups:/usr/bin/false
_postfix:*:27:27:Postfix Mail Server:/var/spool/postfix:/usr/bin/false
_scsd:*:31:31:Service Configuration Service:/var/empty:/usr/bin/false
_ces:*:32:32:Certificate Enrollment Service:/var/empty:/usr/bin/false
_appstore:*:33:33:Mac App Store Service:/var/db/appstore:/usr/bin/false
_mcxalr:*:54:54:MCX AppLaunch:/var/empty:/usr/bin/false
_appleevents:*:55:55:AppleEvents Daemon:/var/empty:/usr/bin/false
_geod:*:56:56:Geo Services Daemon:/var/db/geod:/usr/bin/false
_devdocs:*:59:59:Developer Documentation:/var/empty:/usr/bin/false
_sandbox:*:60:60:Seatbelt:/var/empty:/usr/bin/false
_mdnsresponder:*:65:65:mDNSResponder:/var/empty:/usr/bin/false
_ard:*:67:67:Apple Remote Desktop:/var/empty:/usr/bin/false
_www:*:70:70:World Wide Web Server:/Library/WebServer:/usr/bin/false
_eppc:*:71:71:Apple Events User:/var/empty:/usr/bin/false
_cvs:*:72:72:CVS Server:/var/empty:/usr/bin/false
_svn:*:73:73:SVN Server:/var/empty:/usr/bin/false
_mysql:*:74:74:MySQL Server:/var/empty:/usr/bin/false
_sshd:*:75:75:sshd Privilege separation:/var/empty:/usr/bin/false
_qtss:*:76:76:QuickTime Streaming Server:/var/empty:/usr/bin/false
_cyrus:*:77:6:Cyrus Administrator:/var/imap:/usr/bin/false
_mailman:*:78:78:Mailman List Server:/var/empty:/usr/bin/false
_appserver:*:79:79:Application Server:/var/empty:/usr/bin/false
_clamav:*:82:82:ClamAV Daemon:/var/virusmails:/usr/bin/false
_amavisd:*:83:83:AMaViS Daemon:/var/virusmails:/usr/bin/false
_jabber:*:84:84:Jabber XMPP Server:/var/empty:/usr/bin/false
_appowner:*:87:87:Application Owner:/var/empty:/usr/bin/false
_windowserver:*:88:88:WindowServer:/var/empty:/usr/bin/false
_spotlight:*:89:89:Spotlight:/var/empty:/usr/bin/false
_tokend:*:91:91:Token Daemon:/var/empty:/usr/bin/false
_securityagent:*:92:92:SecurityAgent:/var/db/securityagent:/usr/bin/false
_calendar:*:93:93:Calendar:/var/empty:/usr/bin/false
_teamsserver:*:94:94:TeamsServer:/var/teamsserver:/usr/bin/false
_update_sharing:*:95:-2:Update Sharing:/var/empty:/usr/bin/false
_installer:*:96:-2:Installer:/var/empty:/usr/bin/false
_atsserver:*:97:97:ATS Server:/var/empty:/usr/bin/false
_ftp:*:98:-2:FTP Daemon:/var/empty:/usr/bin/false
_unknown:*:99:99:Unknown User:/var/empty:/usr/bin/false
_softwareupdate:*:200:200:Software Update
Service:/var/db/softwareupdate:/usr/bin/false
_coreaudiod:*:202:202:Core Audio Daemon:/var/empty:/usr/bin/false
_screensaver:*:203:203:Screensaver:/var/empty:/usr/bin/false
_locationd:*:205:205:Location Daemon:/var/db/locationd:/usr/bin/false
_trustevaluationagent:*:208:208:Trust Evaluation Agent:/var/empty:/usr/bin/false
_timezone:*:210:210:AutoTimeZoneDaemon:/var/empty:/usr/bin/false
_lda:*:211:211:Local Delivery Agent:/var/empty:/usr/bin/false
```

```
_cvmsroot:*:212:212:CVMS Root:/var/empty:/usr/bin/false
_usbmuxd:*:213:213:iPhone OS Device Helper:/var/db/lockdown:/usr/bin/false
_dovecot:*:214:6:Dovecot Administrator:/var/empty:/usr/bin/false
_dpaudio:*:215:215:DP Audio:/var/empty:/usr/bin/false
_postgres:*:216:216:PostgreSQL Server:/var/empty:/usr/bin/false
_krbtgt:*:217:-2:Kerberos Ticket Granting Ticket:/var/empty:/usr/bin/false
_kadmin_admin:*:218:-2:Kerberos Admin Service:/var/empty:/usr/bin/false
_kadmin_changepw:*:219:-2:Kerberos Change Password
Service:/var/empty:/usr/bin/false
_devicemgr:*:220:220:Device Management Server:/var/empty:/usr/bin/false
_webauthserver:*:221:221:Web Auth Server:/var/empty:/usr/bin/false
_netbios:*:222:222:NetBIOS:/var/empty:/usr/bin/false
_warmd:*:224:224:Warm Daemon:/var/empty:/usr/bin/false
_dovenull:*:227:227:Dovecot Authentication:/var/empty:/usr/bin/false
_netstatistics:*:228:228:Network Statistics Daemon:/var/empty:/usr/bin/false
_avbdeviced:*:229:-2:Ethernet AVB Device Daemon:/var/empty:/usr/bin/false
_krb_krbtgt:*:230:-2:Open Directory Kerberos Ticket Granting
Ticket:/var/empty:/usr/bin/false
_krb_kadmin:*:231:-2:Open Directory Kerberos Admin
Service:/var/empty:/usr/bin/false
_krb_changepw:*:232:-2:Open Directory Kerberos Change Password
Service:/var/empty:/usr/bin/false
_krb_kerberos:*:233:-2:Open Directory Kerberos:/var/empty:/usr/bin/false
_krb_anonymous:*:234:-2:Open Directory Kerberos
Anonymous:/var/empty:/usr/bin/false
_assetcache:*:235:235:Asset Cache Service:/var/empty:/usr/bin/false
_coremediaiod:*:236:236:Core Media IO Daemon:/var/empty:/usr/bin/false
_launchservicesd:*:239:239:_launchservicesd:/var/empty:/usr/bin/false
_iconservices:*:240:240:IconServices:/var/empty:/usr/bin/false
_distnote:*:241:241:DistNote:/var/empty:/usr/bin/false
_nsurlsessiond:*:242:242:NSURLSession
Daemon:/var/db/nsurlsessiond:/usr/bin/false
_displaypolicyd:*:244:244:Display Policy Daemon:/var/empty:/usr/bin/false
_astris:*:245:245:Astris Services:/var/db/astris:/usr/bin/false
_krbfast:*:246:-2:Kerberos FAST Account:/var/empty:/usr/bin/false
_gamecontrollerd:*:247:247:Game Controller Daemon:/var/empty:/usr/bin/false
_mbsetupuser:*:248:248:Setup User:/var/setup:/bin/bash
_ondemand:*:249:249:On Demand Resource Daemon:/var/db/ondemand:/usr/bin/false
_xserverdocs:*:251:251:macOS Server Documents Service:/var/empty:/usr/bin/false
_wwwproxy:*:252:252:WWW Proxy:/var/empty:/usr/bin/false
_mobileasset:*:253:253:MobileAsset User:/var/ma:/usr/bin/false
_findmydevice:*:254:254:Find My Device
Daemon:/var/db/findmydevice:/usr/bin/false
_datadetectors:*:257:257:DataDetectors:/var/db/datadetectors:/usr/bin/false
_captiveagent:*:258:258:captiveagent:/var/empty:/usr/bin/false
_ctkd:*:259:259:ctkd Account:/var/empty:/usr/bin/false
_applepay:*:260:260:applepay Account:/var/db/applepay:/usr/bin/false
_hidd:*:261:261:HID Service User:/var/db/hidd:/usr/bin/false
```

```
_cmiodalassistants:*:262:262:CoreMedia IO Assistants
User:/var/db/cmiodalassistants:/usr/bin/false
_analyticsd:*:263:263:Analytics Daemon:/var/db/analyticsd:/usr/bin/false
_fpsd:*:265:265:FPS Daemon:/var/db/fpsd:/usr/bin/false
_timed:*:266:266:Time Sync Daemon:/var/db/timed:/usr/bin/false
_nearbyd:*:268:268:Proximity and Ranging Daemon:/var/db/nearbyd:/usr/bin/false
_reportmemoryexception:*:269:269:ReportMemoryException:/var/db/reportmemoryexcep
tion:/usr/bin/false
_driverkit:*:270:270:DriverKit:/var/empty:/usr/bin/false
_diskimagesiod:*:271:271:DiskImages IO
Daemon:/var/db/diskimagesiod:/usr/bin/false
_logd:*:272:272:Log Daemon:/var/db/diagnostics:/usr/bin/false
_appinstalld:*:273:273:App Install Daemon:/var/db/appinstalld:/usr/bin/false
_installcoordinationd:*:274:274:Install Coordination
Daemon:/var/db/installcoordinationd:/usr/bin/false
_demod:*:275:275:Demo Daemon:/var/empty:/usr/bin/false
_rmd:*:277:277:Remote Management Daemon:/var/db/rmd:/usr/bin/false
_accessoryupdater:*:278:278:Accessory Update
Daemon:/var/db/accessoryupdater:/usr/bin/false
_knowledgegraphd:*:279:279:Knowledge Graph
Daemon:/var/db/knowledgegraphd:/usr/bin/false
_coreml:*:280:280:CoreML Services:/var/db/coreml:/usr/bin/false
_sntpd:*:281:281:SNTP Server Daemon:/var/empty:/usr/bin/false
_trustd:*:282:282:trustd:/var/empty:/usr/bin/false
_mmaintenanced:*:283:283:mmaintenanced:/var/db/mmaintenanced:/usr/bin/false
_darwindaemon:*:284:284:Darwin Daemon:/var/db/darwindaemon:/usr/bin/false
_notification_proxy:*:285:285:Notification Proxy:/var/empty:/usr/bin/false
_avphidbridge:*:288:288:Apple Virtual Platform HID
Bridge:/var/empty:/usr/bin/false
_biome:*:289:289:Biome:/var/db/biome:/usr/bin/false
_backgroundassets:*:291:291:Background Assets Service:/var/empty:/usr/bin/false
_mobilegestalthelper:*:293:293:MobileGestaltHelper:/var/empty:/usr/bin/false
_audiomxd:*:294:294:Audio and MediaExperience
Daemon:/var/db/audiomxd:/usr/bin/false
_terminusd:*:295:295:Terminus:/var/empty:/usr/bin/false
_neuralengine:*:296:296:AppleNeuralEngine:/var/db/neuralengine:/usr/bin/false
_oahd:*:441:441:OAH Daemon:/var/empty:/usr/bin/false
```

[57]:
```
# I want to print all of the usernames from my system
# we need to ignore the lines that start with #

for one_line in open('/etc/passwd'):
    if one_line[0] != '#':
        print(one_line.split(':')[0])
```

```
nobody
root
daemon
```

```
_uucp
_taskgated
_networkd
_installassistant
_lp
_postfix
_scsd
_ces
_appstore
_mcxalr
_appleevents
_geod
_devdocs
_sandbox
_mdnsresponder
_ard
_www
_eppc
_cvs
_svn
_mysql
_sshd
_qtss
_cyrus
_mailman
_appserver
_clamav
_amavisd
_jabber
_appowner
_windowserver
_spotlight
_tokend
_securityagent
_calendar
_teamsserver
_update_sharing
_installer
_atsserver
_ftp
_unknown
_softwareupdate
_coreaudiod
_screensaver
_locationd
_trustevaluationagent
_timezone
_lda
```

```
_cvmsroot
_usbmuxd
_dovecot
_dpaudio
_postgres
_krbtgt
_kadmin_admin
_kadmin_changepw
_devicemgr
_webauthserver
_netbios
_warmd
_dovenull
_netstatistics
_avbdeviced
_krb_krbtgt
_krb_kadmin
_krb_changepw
_krb_kerberos
_krb_anonymous
_assetcache
_coremediaiod
_launchservicesd
_iconservices
_distnote
_nsurlsessiond
_displaypolicyd
_astris
_krbfast
_gamecontrollerd
_mbsetupuser
_ondemand
_xserverdocs
_wwwproxy
_mobileasset
_findmydevice
_datadetectors
_captiveagent
_ctkd
_applepay
_hidd
_cmiodalassistants
_analyticsd
_fpsd
_timed
_nearbyd
_reportmemoryexception
_driverkit
```

```
_diskimagesiod
_logd
_appinstalld
_installcoordinationd
_demod
_rmd
_accessoryupdater
_knowledgegraphd
_coreml
_sntpd
_trustd
_mmaintenanced
_darwindaemon
_notification_proxy
_avphidbridge
_biome
_backgroundassets
_mobilegestalthelper
_audiomxd
_terminusd
_neuralengine
_oahd
```

[59]:
```
# let's print only lines 5-10 of the file
# you might remember enumerate, which gives us the index in a for loop

for index, one_line in enumerate(open('/etc/passwd')):
    if index < 5:
        continue
    if index > 10:
        break
    print(one_line.strip())
```

```
# Open Directory.
#
# See the opendirectoryd(8) man page for additional information about
# Open Directory.
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
```

16 Exercise: Count vowels

1. Define a dict whose keys are the values (a, e, i, o, u) and whose values are all 0.
2. Open `linux-etc-passwd.txt`, one of the files in the zipfile I asked you to download.
3. Go through that file, one line at a time (in a `for` loop)
4. Go through each line, one character at a time, in a `for` loop) – a "nested" loop
5. If the character is a vowel, then add 1 to the count for that vowel in the dict.
6. When we're done reading through the file, print the dict.

[60]:
```
d = {'a':0, 'e':0, 'i':0, 'o':0, 'u':0}

for one_line in open('linux-etc-passwd.txt'):    # when we iterate over a file,␣
 ↪we get one line at a time
    for one_character in one_line:               # when we iterate over a␣
 ↪string, we get one character at a time
        if one_character in d:                   # is the character a key in d?
            d[one_character] += 1                # if so, then add 1 to its value

print(d)
```

```
{'a': 137, 'e': 122, 'i': 148, 'o': 123, 'u': 50}
```

17 Next up

1. More reading from files
2. Writing to files

18 Filenames

When we invoke `open`, we pass a filename, which is a string. There are three types of filenames that we can use:

- An absolute path, meaning a filename that starts with `/` (on Unix) or `c:\` (on Windows). This is unambiguous, and points to the same file no matter where you are located on the current computer, or where you run the program. For example, I used `/etc/passwd` before.
- Just a filename, with no slashes or backslashes in it. In such a case, the file must be in the current directory – meaning, where the program is located. If you're using Jupyter, then it has to be in the same directory as Jupyter is running. Before, when I looked through `linux-etc-passwd.txt`, it was in the current directory. This is great and easy to work with, but you have to know where the program will be run from.
- A name that doesn't start with a `/`, but contains one. This tells the program to look in the current directory for a subdirectory (subfolder), and then look for files in there. These could look like `files/data.txt` or `../../myfile.txt`. Notice that you can go up in the hierarchy with `..`

In order for the second two options to succeed, you need to know (a) where the program is located and (b) where the files are located. That's not always obvious.

19 Exercise: Word count

Unix comes with a bunch of utilities that are quite useful. One is called `wc` ("word count"). If you run it on a file, you'll get a report that tells you:

- How many lines are in the file (including blank lines)
- How many characters are in the file (including spaces and newlines, and other invisible characters)
- How many words are in the file (separated by whitespace)

I want you to write a program that, given a filename, produces such a report.

I've written a file that you can use to experiment on, `wcfile.txt`.

[61]:
```
# in Jupyter, you can run Unix commands by typing ! at the start of a line
# The Unix "cat" command displays the contents of a file

!cat wcfile.txt
```

```
This is a test file.

It contains 28 words and 20 different words.

It also contains 165 characters.

It also contains 11 lines.

It is also self-referential.

Wow!
```

[64]:
```
# setup
counts = {'lines':0,
          'chars':0,
          'words':0}

filename = 'wcfile.txt'

# calculations
for one_line in open(filename):
    counts['lines'] += 1
    counts['chars'] += len(one_line)
    counts['words'] += len(one_line.split())

# report
for key, value in counts.items():
    print(f'{key}: {value}')
```

```
lines: 11
chars: 165
words: 28
```

[ ]:
```

```

20 Writing to files

If we want to write to a file, we can open it – but we have to pass a second argument, indicating that we want to write to it.

(Yes, you can theoretically open a file for both reading and writing. But I suggest you not do so.)

You can say:

```
open('myfile.txt', 'w')   # the second argument means "for writing"
```

Note that if you open a file for writing, one of two things will be true: 1. You get an error, indicating that you cannot write to the file 2. You successfully open the file for writing, and now it exists with 0 bytes in it

This means: If you open a file for writing, and the file already exits, you have obliterated its contents.

Once we've open a file for writing, we can then use the `write` method to write to it. It's similar to `print`, except that it doesn't automatically add a newline after writing.

[65]:
```
f = open('myfile.txt', 'w')
f.write('abcd\n')
f.write('efghij\n')
f.write('klmnopqr\n')
```

[65]:
```
9
```

[66]:
```
# let's take a look at the file I've written

!cat myfile.txt
```

The file is empty because it would be wasteful for the OS to write to a file every time we ask it to do so. Instead, it writes what we want to a part of memory called a "buffer." When the buffer fills, then the data is really written to the disk.

Since we didn't write a lot of data, the buffer wasn't full, and nothing was truly written.

When will the buffer be flushed?

- We can call the `flush` method, which flushes the buffer
- We can call the `close` method, which stops us from having access to the file, and also flushes it
- When the program/Python exits, the buffer is automatically flushed

What if I know that I want to open a file, write to it, and then immediately close it (to ensure it's flushed)?

[67]:
```
# I could just explicitly close the file when I'm done with it

f = open('myfile.txt', 'w')
f.write('*abcd\n')
f.write('*efghij\n')
f.write('*klmnopqr\n')
f.close()
```

[68]:
```
!cat myfile.txt
```

```
*abcd
*efghij
*klmnopqr
```

Python supplies a special keyword, `with`, that automates the closing (and flushing) of the file when we're done with our block.

We just need to open the file with `with`

[69]:
```
with open('myfile.txt', 'w') as f:    # this is the same as f = open(...), but␣
 ↪auto-closes it
    f.write('**abcd\n')
    f.write('**efghij\n')
    f.write('**klmnopqr\n')
    # when Python exits from this block, f.close() will be invoked
```

[70]:
```
!cat myfile.txt
```

```
**abcd
**efghij
**klmnopqr
```

[74]:
```
# storing data to a file

d = {'a':10, 'b':20, 'c':30}

with open('config.txt', 'w') as f:
    for key, value in d.items():
        f.write(f'{key}:{value}\n')
```

[75]:
```
!cat config.txt
```

```
a:10
b:20
c:30
```

21 Next week: Functions!