On your way in…(on the side table)

Hand-in:

1. Homework 2
   - Two Piles: SU Boxes <1700 and >= 1700

Pick-Up:

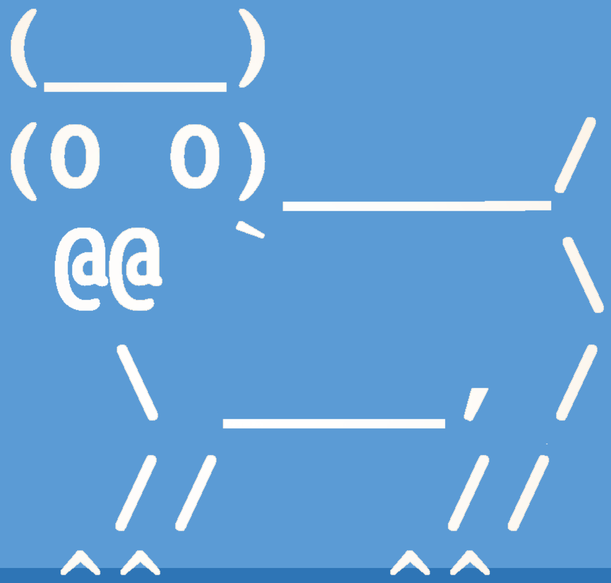
1. POGIL 18: More Lists and Strings

Welcome to CS 134!

Introduction to Computer Science

Iris Howley

-Strings-



Spring 2020

```
'/lab02/08ikh1        Grade = B+'
'/lab02/28gmh3        Grade = A'
```

In plain English, what is an *algorithm* for grabbing the user names and the letter grades from all strings that take the above form?

(Think the sandwich instructions from first day of class)

Take a couple minutes to discuss with a partner.

```
(___)
(O O)___
 @@   `
   \  ___
   //
 ^^
```

TODAY'S LESSON

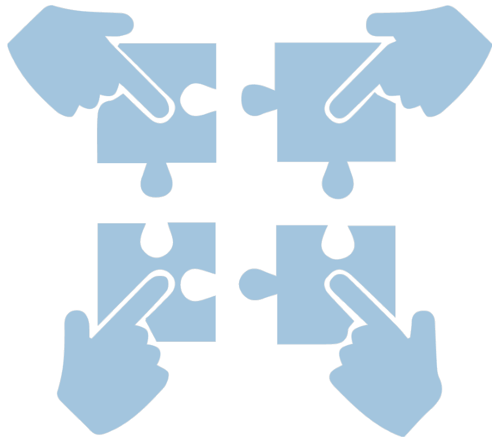
`Strings`

(A sequence of characters. Text data is everywhere!)

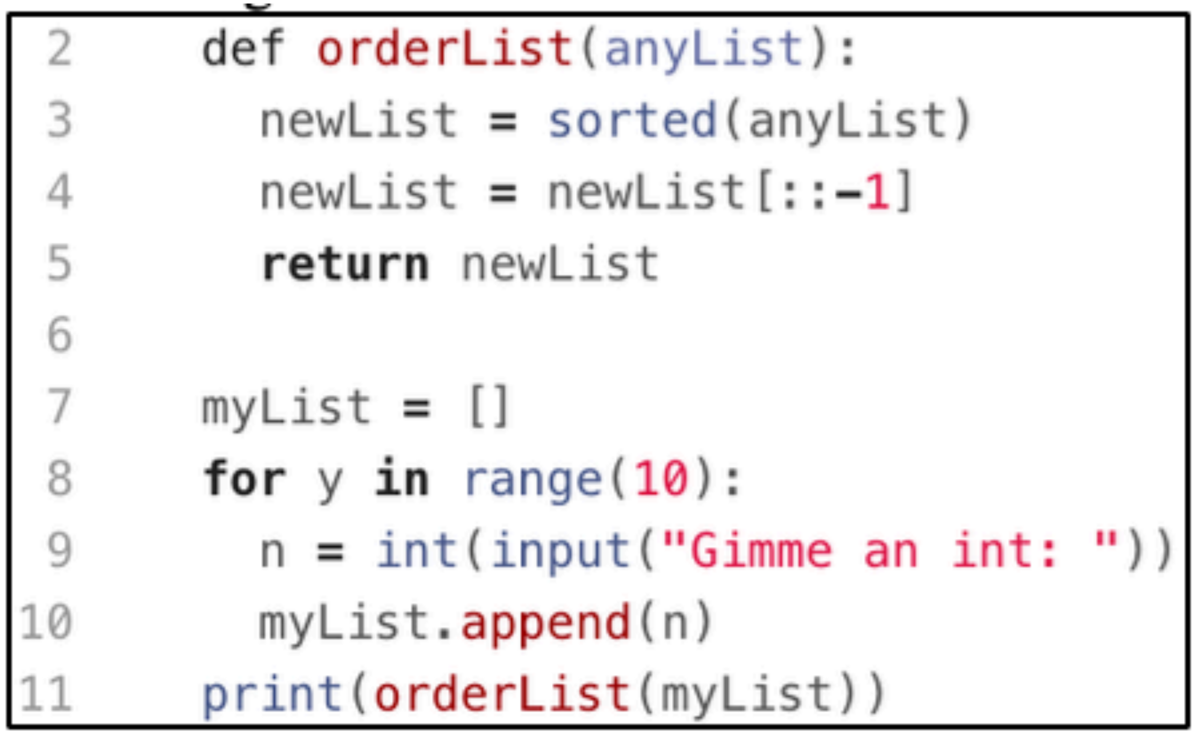
POGIL – Activity 18: More Strings & Lists

- We know how to do some things with strings, but there's more!

- Look at Python Activity 18, Questions 1-6
- Find a partner and talk through the questions together

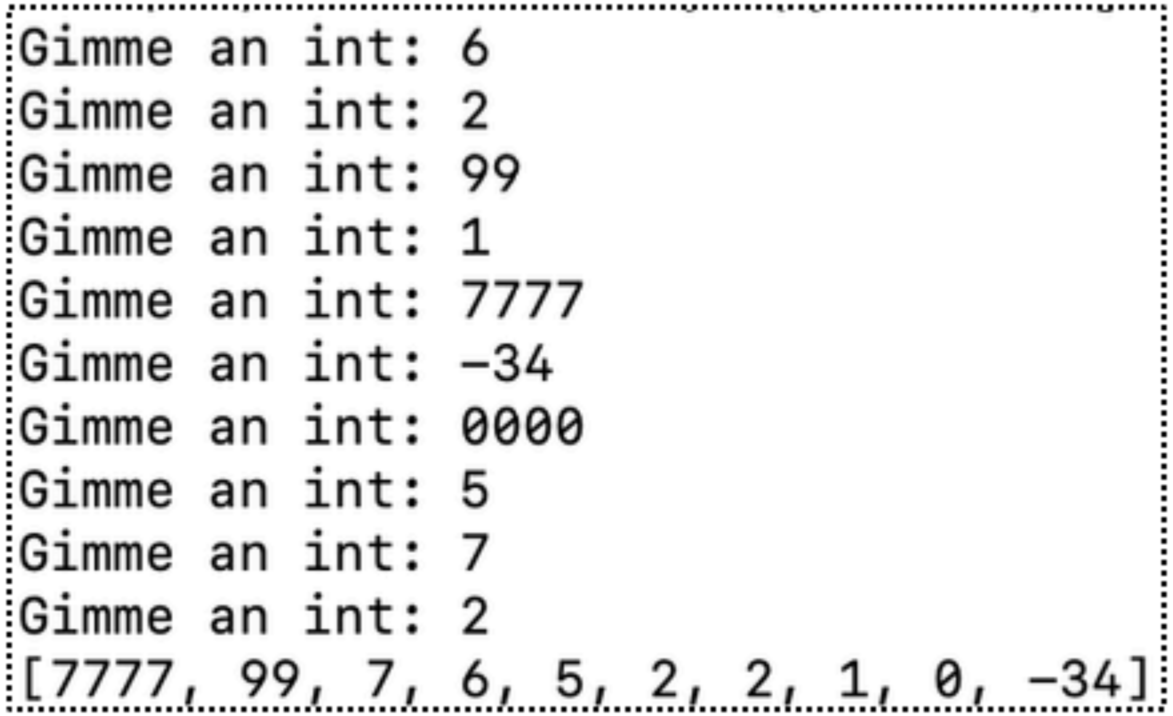


POGIL – Activity 18: Question 1



```python
    def orderList(anyList):
        newList = sorted(anyList)
        newList = newList[::-1]
        return newList

    myList = []
    for y in range(10):
        n = int(input("Gimme an int: "))
        myList.append(n)
    print(orderList(myList))
```



```
Gimme an int: 6
Gimme an int: 2
Gimme an int: 99
Gimme an int: 1
Gimme an int: 7777
Gimme an int: -34
Gimme an int: 0000
Gimme an int: 5
Gimme an int: 7
Gimme an int: 2
[7777, 99, 7, 6, 5, 2, 2, 1, 0, -34]
```

a. What is the name of the function defined in this program? ____________________

b. What does the function do? ____

c. What might `newList = sorted(anyList)` do? ______________________________

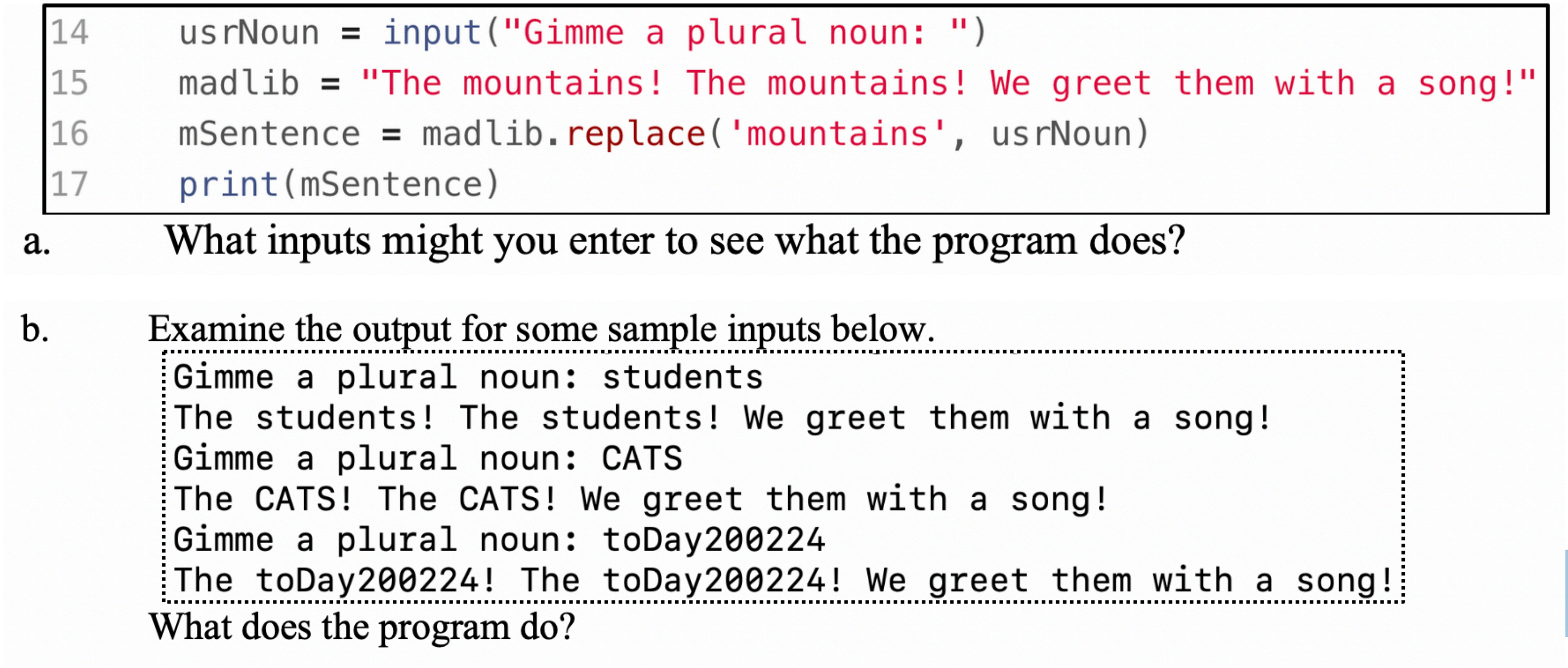
d. What might `newList[:-1]` do? _______________________________________

sorted(lst)

- Sorts a sequence of objects
  - `>>> lst = ['hello','goodbye','goodmorning']`
  - `>>> sorted(lst)`
  - `['goodbye', 'goodmorning', 'hello’]`
- Including strings (sequence of characters)
  - `>>> s = 'hello'`
  - `>>> sorted(s)`
  - `['e', 'h', 'l', 'l', 'o']`

What happened to our string?

POGIL – Activity 18: Question 2



```python
usrNoun = input("Gimme a plural noun: ")
madlib = "The mountains! The mountains! We greet them with a song!"
mSentence = madlib.replace('mountains', usrNoun)
print(mSentence)
```

a. What inputs might you enter to see what the program does?

b. Examine the output for some sample inputs below.

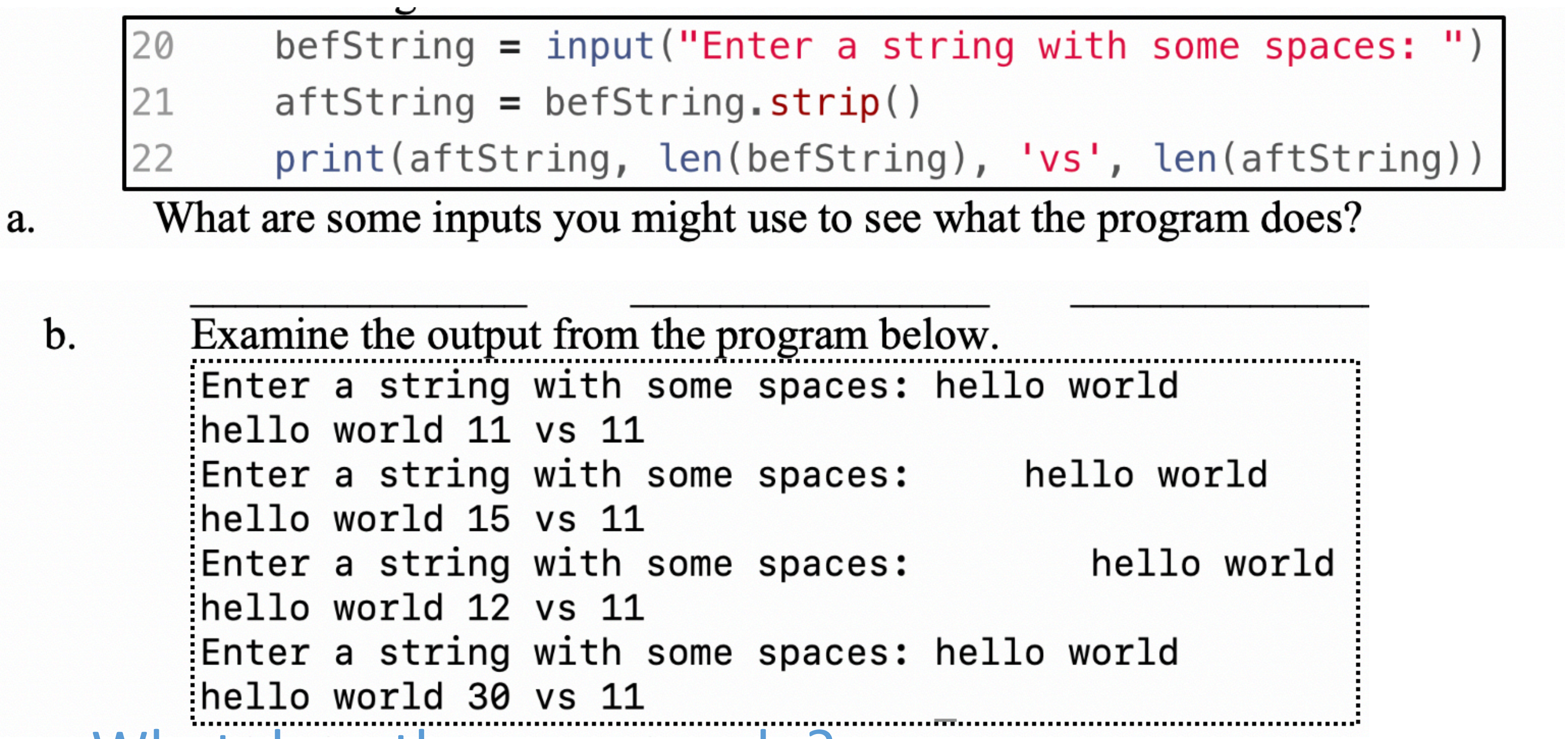
```
Gimme a plural noun: students
The students! The students! We greet them with a song!
Gimme a plural noun: CATS
The CATS! The CATS! We greet them with a song!
Gimme a plural noun: toDay200224
The toDay200224! The toDay200224! We greet them with a song!
```

What does the program do?

What is **replace()**?

What are **replace()**'s parameters?

POGIL – Activity 18: Question 3



```
    befString = input("Enter a string with some spaces: ")
    aftString = befString.strip()
    print(aftString, len(befString), 'vs', len(aftString))
```

a. What are some inputs you might use to see what the program does?

_______________ _______________ _______________

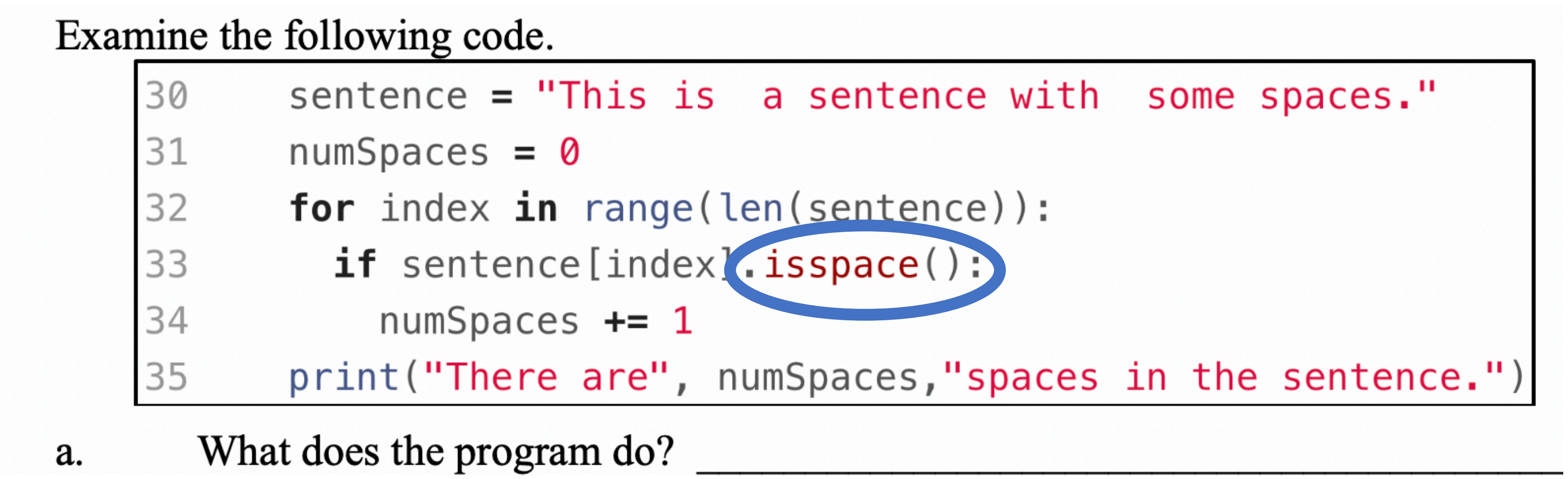
b. Examine the output from the program below.

```
Enter a string with some spaces: hello world
hello world 11 vs 11
Enter a string with some spaces:   hello world
hello world 15 vs 11
Enter a string with some spaces:    hello world
hello world 12 vs 11
Enter a string with some spaces: hello world
hello world 30 vs 11
```

What does the program do?

What does **strip()** do?

POGIL – Activity 18: Question 4



Examine the following code.

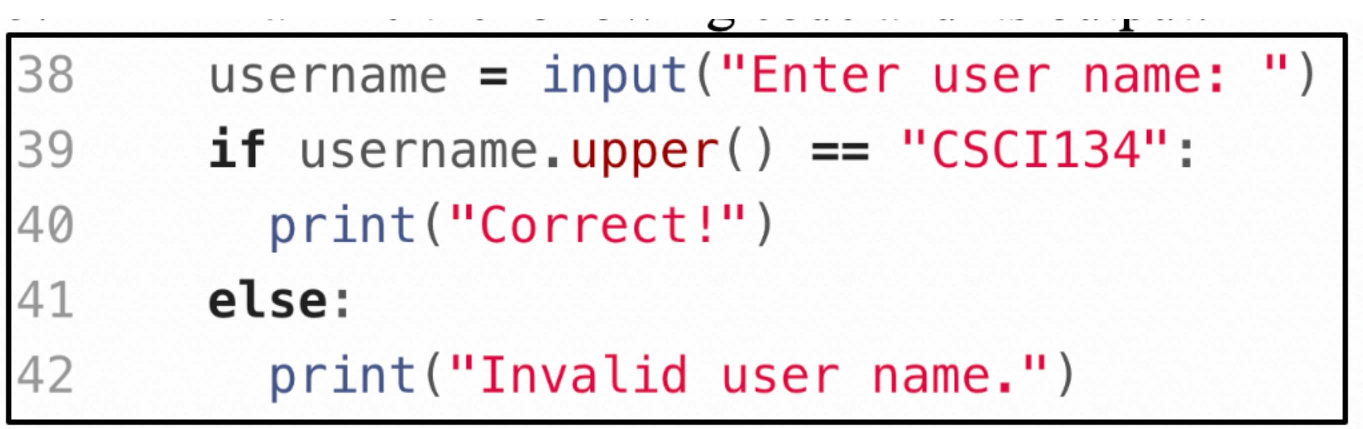
```
sentence = "This is  a sentence with  some spaces."
numSpaces = 0
for index in range(len(sentence)):
  if sentence[index].isspace():
    numSpaces += 1
print("There are", numSpaces,"spaces in the sentence.")
```

a. What does the program do? ______________________________

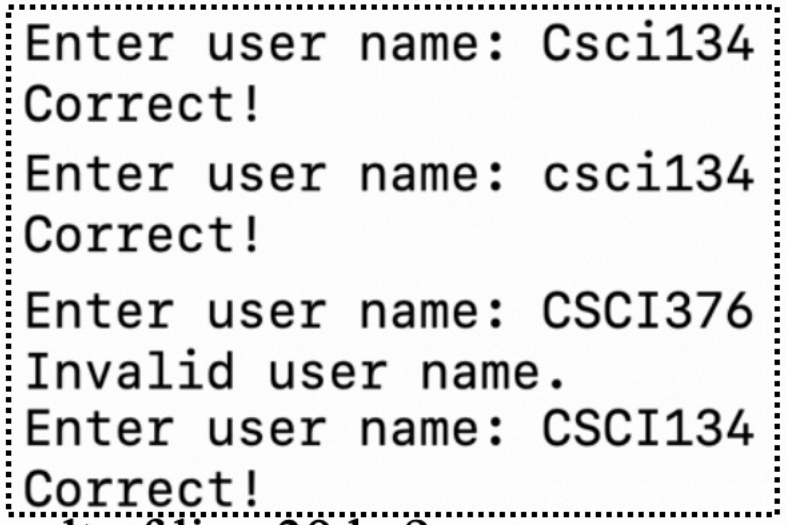
```
There are 8 spaces in the sentence.
```

What does **isspace()** do?
What might **isalpha()** be?
isdigit()?

POGIL – Activity 18: Question 5

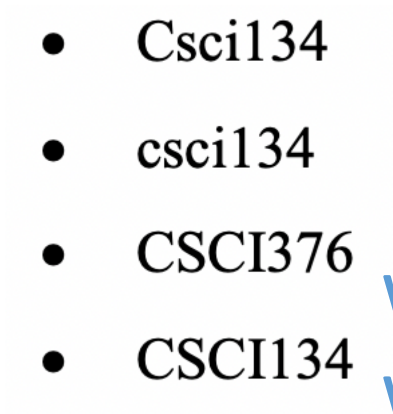


```python
username = input("Enter user name: ")
if username.upper() == "CSCI134":
  print("Correct!")
else:
  print("Invalid user name.")
```



```
Enter user name: Csci134
Correct!
Enter user name: csci134
Correct!
Enter user name: CSCI376
Invalid user name.
Enter user name: CSCI134
Correct!
```

```
if username.upper()=='CSCI134'
```



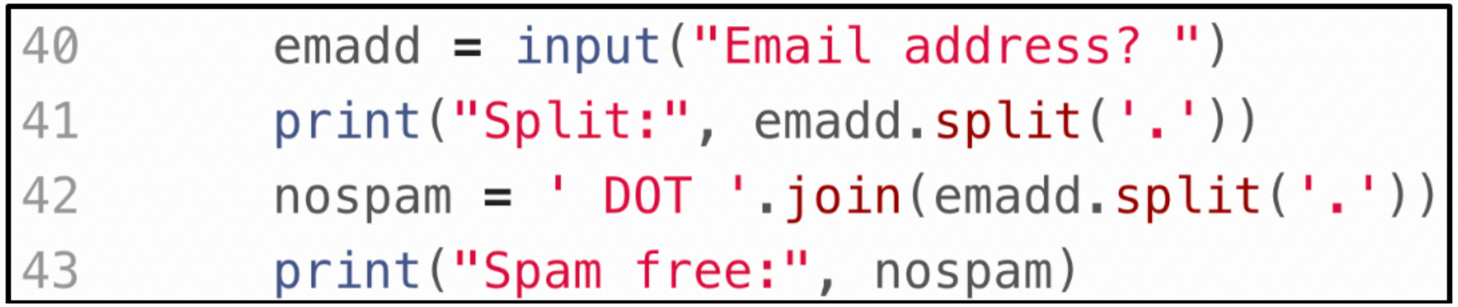
- Csci134
- csci134
- CSCI376
- CSCI134

What does **upper()** do?

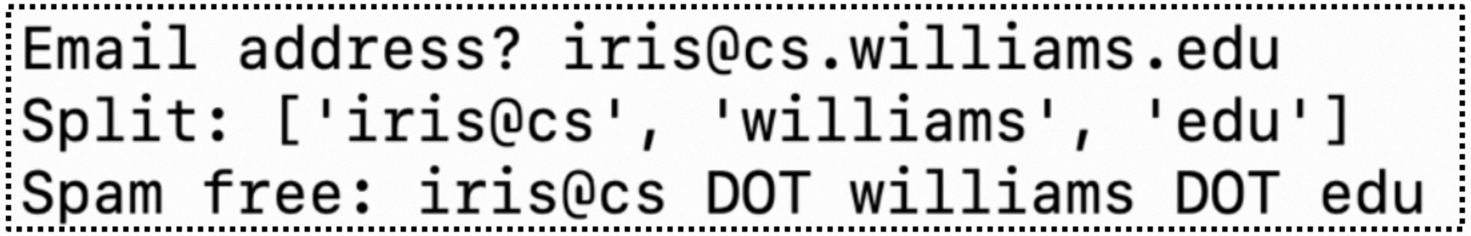
What might **lower()** be?

How would we check this in interactive python?

POGIL – Activity 18: Question 6



```
        emadd = input("Email address? ")
        print("Split:", emadd.split('.'))
        nospam = ' DOT '.join(emadd.split('.'))
        print("Spam free:", nospam)
```



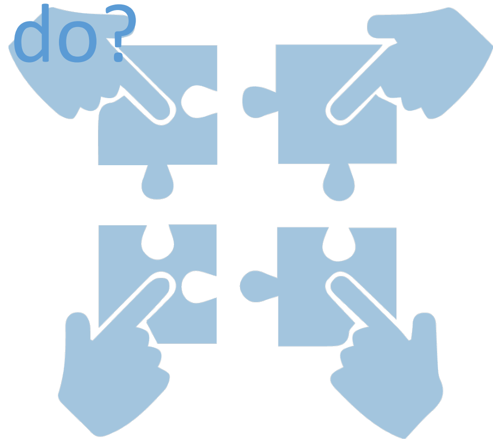
```
Email address? iris@cs.williams.edu
Split: ['iris@cs', 'williams', 'edu']
Spam free: iris@cs DOT williams DOT edu
```

What does **split()** do?

What might `print('dog,cat,mouse,cheese'.split(','))` do?

What is **join()** doing?

How would we check this in interactive python?



`s.join(lst)`

- Converting a list into a string by joining with a specified character

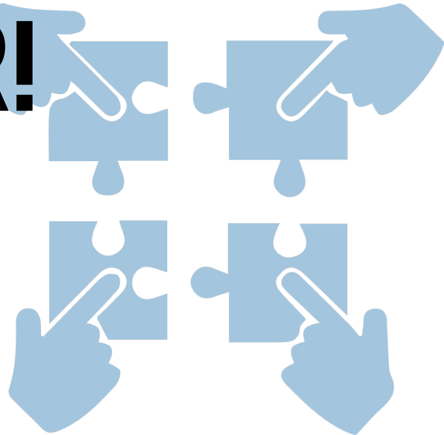
- `>>> s = ’hello'`
- `>>> t = ''.join(sorted(s))`
- `>>> t`
    - `‘ehllo’`

This character says what to join the list with

YOU SHOULD COMPLETE THE REST OF ALL POGILS OUTSIDE OF CLASS.

BEST DONE WITH A PARTNER OR STUDY GROUP.

CHECK YOUR ANSWERS ON A COMPUTER!



Strings

Cleaning and processing text data helps with lots of interesting tasks, as we’ll see in lab today.

A Tale of Two Sequences

Accessing elements by index

- `l = [18,20,5,16]`
- `l[1]`
  - `20`
- `l[-1]`
  - `16`

- `s = ‘hello’`
- `s[1]`
  - `’e’`
- `s[-1]`
  - `‘o’`

A Tale of Two Sequences

Lengths of sequences

- `l = [18,20,5,16]`
- `len(l)`
  - 4

- `s = ‘hello’`
- `len(s)`
  - 5

A Tale of Two Sequences

Slice notation for lists and strings

- `l = [18,20,5,16,'d']`
- `l[2:] [5,16,'d']`
- `l[:2] [18,20]`
- `l[:-2] [18,20,5]`
- `l[-2:] [16,'d']`
- `l[2:-2] [5]`

- `h = "Hello"`
- `h[2:] 'llo'`
- `h[:2] 'He'`
- `h[:-2] 'Hel'`
- `h[-2:] 'lo'`
- `h[2:-2] 'l'`

We call this ‘slice notation’ or ‘string slicing’

A Tale of Two Sequences

Slice notation - step

- `l = [18,20,5,16]`
- `l[::-1]`
  - `[16,5,20,18]`

- `s = ‘hello’`
- `s[::-1]`
  - `’olleh’`
- `s[::2]`
  - `‘hlo’`

Slice Notation

- `s[start:end:step]` # start through not past end, by step

- `s[start:end]` # items start through end-1
- `s[start:]` # items start through the rest of the list
- `s[:end]` # items from the beginning through end-1
- `s[:]` # a copy of the whole list

A Tale of Two Sequences

Sorting lists and strings

- `l = [18,20,5,16]`
- `m = sorted(l)`
- `l`
  - `[18,20,5,16]`
- `m`
  - `[5,15,18,20]`

- `s = ‘hello’`
- `t = sorted(s)`
- `s`
  - `‘hello’`
- `t`
  - ['e', 'h', 'l', 'l', 'o']

???

If you want to use these modified strings & lists, you need to attach them to a variable!

A Tale of Two Sequences

- What do we do with `t=['e', 'h', 'l', 'l', 'o']`?
  - We can turn it back into a string!

- `‘’.join(t)`
  - `‘ehllo’`

`';'.join(sorted('hello'))`

`'e;h;l;l;o'`

We now have the tools to alphabetize a string.
How do we do that?

A Tale of Two Sequences

- `l = [18,20,5,16]`
- `l.append(’dogge’)`

- `l`
  - `[18,20,5,16,‘dogge’]`

- `s = ‘hello’`
- `s.append(‘world’)`

- `ERROR`
- `s += ”!”`
  - `’hello!’`

A String is not a List!

String Functions

`s = ‘ Csci 134 ’`

- `s.lower()`
  - `‘ csci 134 ’`
- `s.upper()`
  - `‘ CSCI 134 ’`

- `>>> s`
- `‘ Csci 134 ’`

- `s.strip()`
  - `‘Csci 134’`
- `s.replace(‘ ‘, ‘’)`
  - `‘Csci134’`

If you want to use these modified strings, you need to attach them to a variable!

String Functions

More built-in string functions described in python.org documentation:

https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str

Canon Forms of Strings

WHEN COMPARING TWO STRINGS TAKING THE CANONICAL FORMS AND COMPARING IS SOMETIMES HELPFUL!

1. Are ‘`Night`’ and ‘`Thing  `’ anagrams of each other?
2. Remove spaces: ‘`Night`’ vs. ‘`Thing`’
3. Make lowercase: ‘`night`’ vs. ‘`thing`’
4. Alphabetize: ‘`ghint`’ vs. ‘`ghint`’
5. Compare → Yes! They are anagrams

Is the best ordering?
Different situations might require different approaches

Canon Forms of Strings

2. Remove spaces

- `>>> h = ’ Hello   '`
- `>>> spaceH = h.strip()`
- `>>> spaceH`
- `‘Hello’`

Canon Forms of Strings

3. Lowercase

- `>>> spaceH = ’Hello'`
- `>>> lowH = spaceH.lower()`
- `>>> lowH`
- `‘hello’`

Canonical Form

4. Alphabetize

- `>>> lowH = ’hello'`
- `>>> alpH = ''.join(sorted(lowH))`
- `>>> alpH`
- `‘ehllo’`

This character says what to join the list with

Canon Forms of Strings

`canon()`

WILL BE VERY USEFUL IN THIS WEEK'S LAB.

CAN YOU FIGURE OUT WAYS TO USE IT?

In labs, when we ask you to write a function, we usually want you to use that function in some way!

QUESTIONS?

Leftover Slides

Format Printing

```
print("{} was born on {}/{}/{}".format("Pixel",5,16, 2018))
```

Pixel was born on 5/16/2018

This will print the same exact text:

```
name = "Pixel"
month = 5
day = 16
year = 2018
print("{} was born on {}/{}/{}".format(name,month,day, year))
```

A Tale of Two Sortings…

- `l = [18,20,5,16]`
- `l.sort()`
- `l`
  - `[5,16,18,20]`

.sort() sorts the list itself

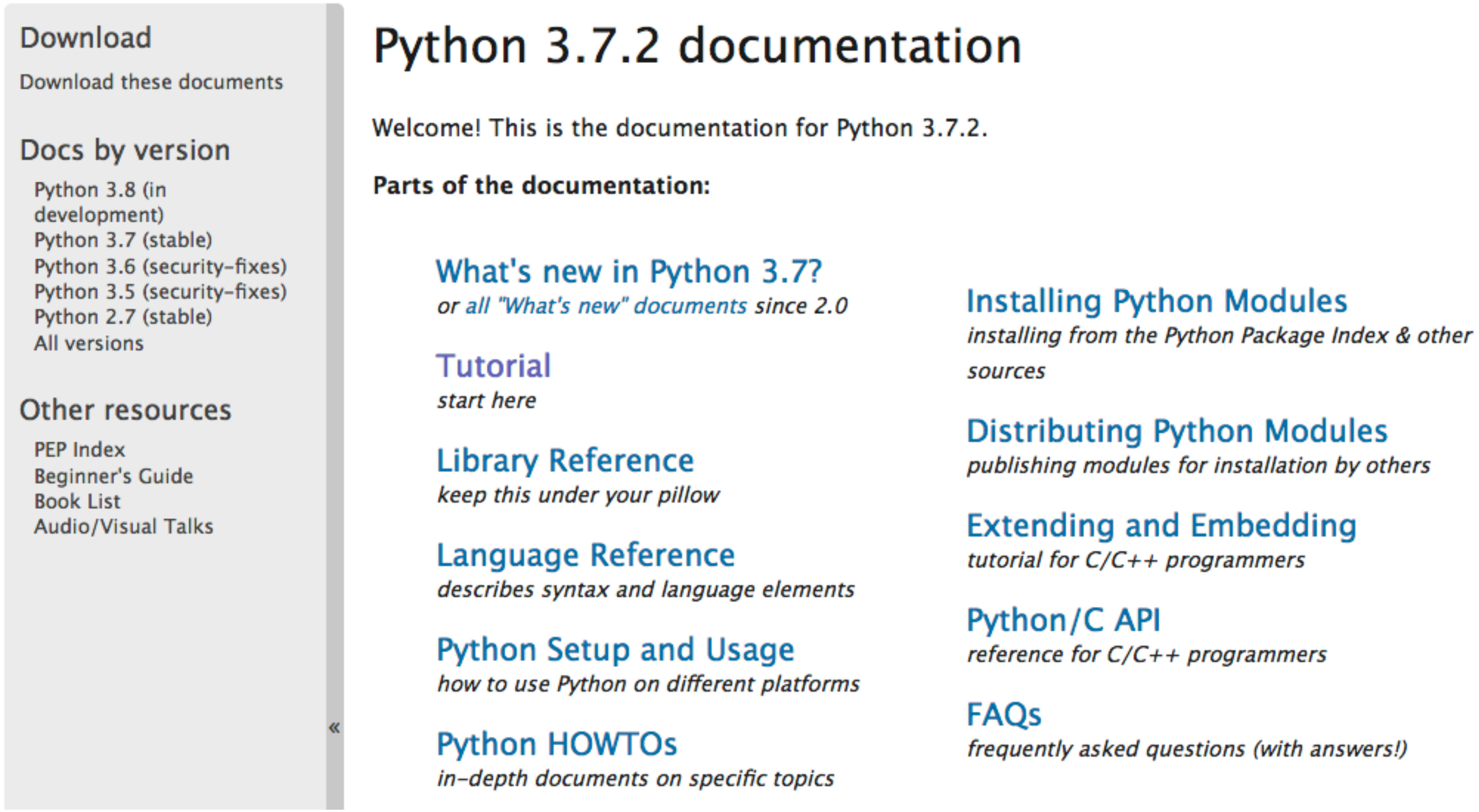
- `m = [18,20,5,16]`
- `sorted(m)`
- `m`
  - `[18,20,5,16]`

sorted() returns a copy of the sorted list

Python Documentation

- `> pydoc3 list`
- `> pydoc3 string`
- You need to be at the Terminal, not in interactive python
  - Interactive python starts with this: '>>>'
- Also, python.org documentation:
  - https://docs.python.org/3/index.html

Python Documentation



Download

Download these documents

Docs by version

- Python 3.8 (in development)
- Python 3.7 (stable)
- Python 3.6 (security-fixes)
- Python 3.5 (security-fixes)
- Python 2.7 (stable)
- All versions

Other resources

- PEP Index
- Beginner's Guide
- Book List
- Audio/Visual Talks

«

Python 3.7.2 documentation

Welcome! This is the documentation for Python 3.7.2.

Parts of the documentation:

What's new in Python 3.7?
or all "What's new" documents since 2.0

Tutorial
start here

Library Reference
keep this under your pillow

Language Reference
describes syntax and language elements

Python Setup and Usage
how to use Python on different platforms

Python HOWTOs
in-depth documents on specific topics

Installing Python Modules
installing from the Python Package Index & other sources

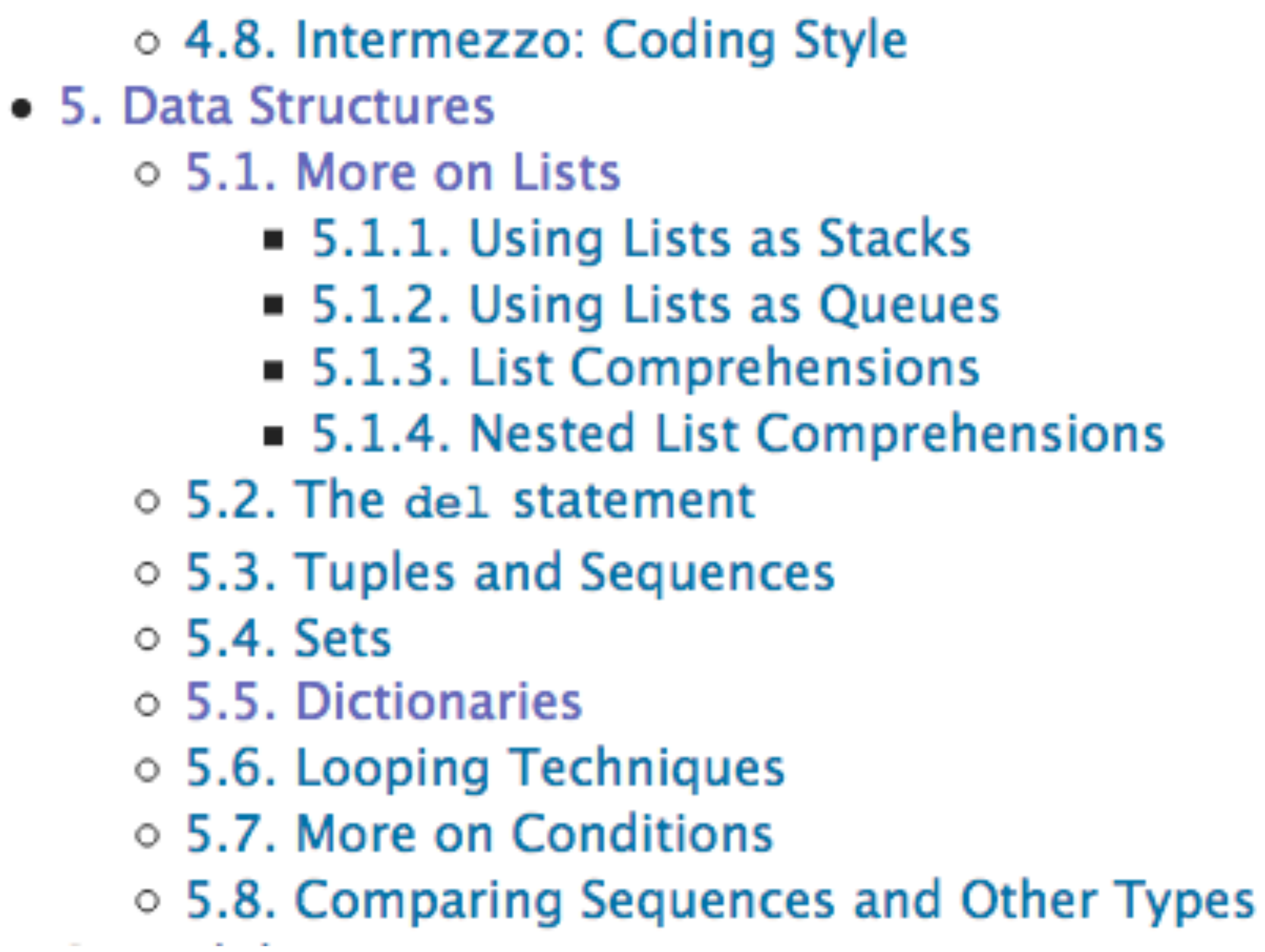
Distributing Python Modules
publishing modules for installation by others

Extending and Embedding
tutorial for C/C++ programmers

Python/C API
reference for C/C++ programmers

FAQs
frequently asked questions (with answers!)

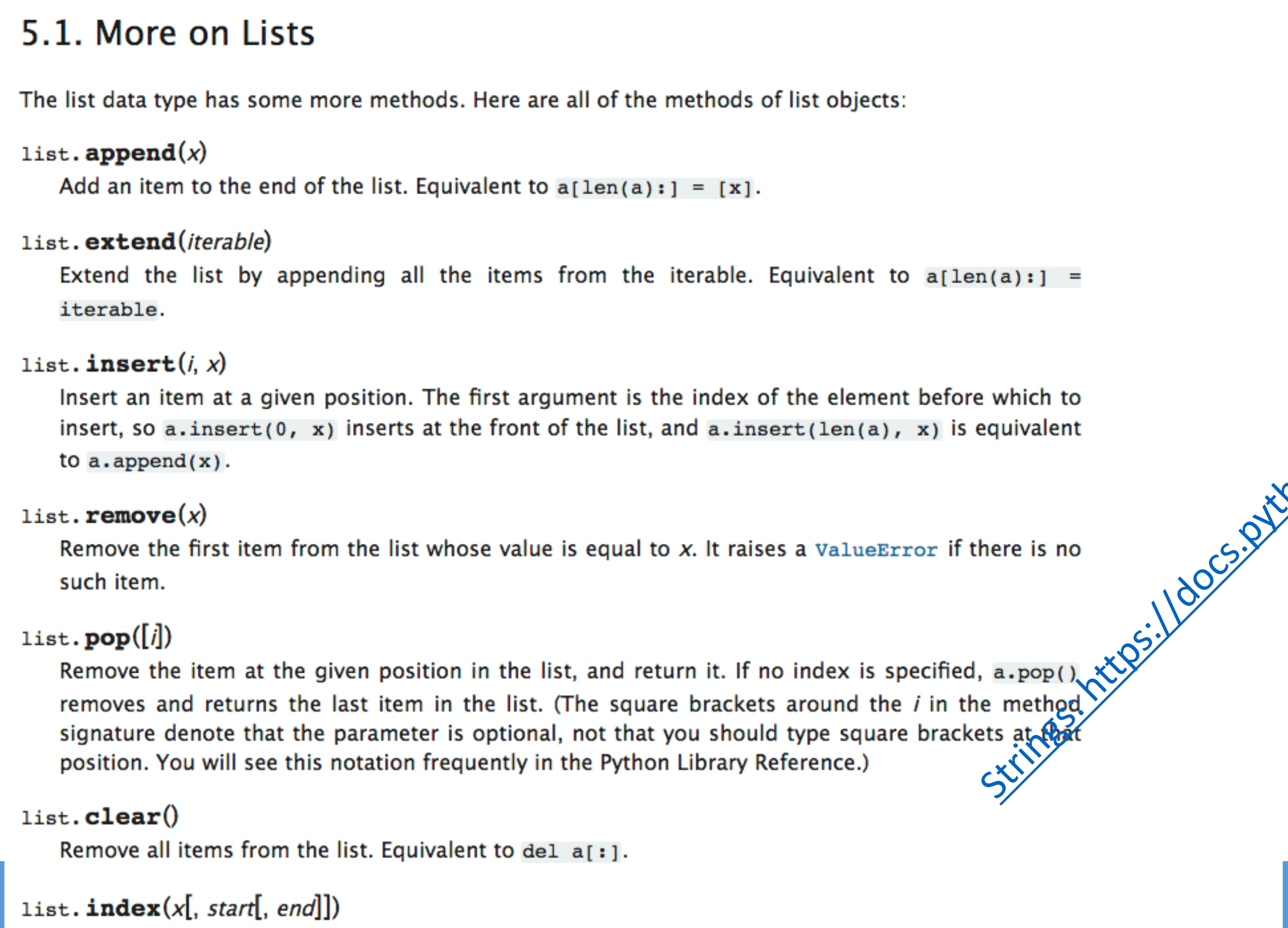
Python Documentation



Python Documentation

https://docs.python.org/3/tutorial/datastructures.html#more-on-lists

Strings: https://docs.python.org/3/library/stdtypes.html#text-sequence-type-str



5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.`**`append`**`(`*x*`)`
Add an item to the end of the list. Equivalent to `a[len(a):] = [x]`.

`list.`**`extend`**`(`*iterable*`)`
Extend the list by appending all the items from the iterable. Equivalent to `a[len(a):] = iterable`.

`list.`**`insert`**`(`*i*, *x*`)`
Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x)`.

`list.`**`remove`**`(`*x*`)`
Remove the first item from the list whose value is equal to *x*. It raises a `ValueError` if there is no such item.

`list.`**`pop`**`([`*i*`])`
Remove the item at the given position in the list, and return it. If no index is specified, `a.pop()` removes and returns the last item in the list. (The square brackets around the *i* in the method signature denote that the parameter is optional, not that you should type square brackets at that position. You will see this notation frequently in the Python Library Reference.)

`list.`**`clear`**`()`
Remove all items from the list. Equivalent to `del a[:]`.

`list.`**`index`**`(`*x*`[`, *start*`[`, *end*`]])`