

Lecture 8: Strings and File I/O

CS 51P

September 28, 2022

strings are sequences

```
string = "Hello world!"
```

- can:

- loop over contents

```
for char in string:  
    print(char)
```

- check membership

```
check = "!" in string
```


- get the length

```
length = len(string)
```

- index into them

```
char = string[3]  
char2 = string[12]
```

**IndexError: range object
index out of range**



Example

- Define a function `str_even` that takes one parameter `s` (a string) and returns a string comprised of only the even characters of `s`

Exercise

- Define a function `findchar` that takes two parameters, a string `s` and a character `c` and returns the index of the first instance of that character. If that character does not appear in the string, it returns `-1`
- `findchar("hello", "h") == 0`
- `findchar("hello", "l") == 2`
- `findchar("hello", "a") == -1`

slicing (1)

- For extracting part of a sequence

```
s[:]  
s[start:]  
s[:end]  
s[start:end]
```

```
>>> s = "Hello world!\n\n"  
>>> s[6]  
      'w'  
>>> s[2:7]  
      'llo w'  
>>> s[5:]  
      ' world!\n\n'  
>>> s[:3]  
      'Hel'
```

slicing (2)

- For extracting part of a sequence

```
s[:]  
s[start:]  
s[:end]  
s[start:end]  
  
s[start::step]  
s[:end:step]  
s[start:end:step]
```

```
>>> s = "Hello world!\n\n"  
>>> s[2::2]  
      'lowrd\n'  
>>> s[1:10:3]  
      'eoo'  
>>> s[:5:2]  
      'Hlo'  
>>> s[-3:-10:-1]  
      '!dlrow'
```

Exercise

- Evaluate the following expressions.

```
test = "This is a string"
```

- `test[10]`
- `test[0:2]`
- `test[:5]`
- `test[::2]`

Example

- Define a function `str_even` that takes one parameter `s` (a string) and returns a string comprised of only the even characters of `s`

Exercise

- Define a function `half` that takes one parameter `s` (a string) and returns the first half of that string.

String methods

```
from string import *

test = " This is a string"

1 test.startswith(" Thi")
2 test.startswith(" this")
3 test.endswith("a")
4 test.endswith("string")
5 test.lower()
6 test.strip()
7 test.strip("g")
8 test.strip(" g")
9 test.find("i")
10 test.find(" is")
11 test.find("banana")
12 test.replace("s", "S")
13 test.replace("is", "si")
```

What is the value of
test at the end?

ranges are also sequences

- range is a type the same way str or int or float is a type

```
x = range(12,21,2)
```

- can:

- loop over the contents

```
for i in x:  
    print(i)
```

- check membership

```
check = (18 in x)
```


- get the length

```
len(x)
```

- index into them

```
x[3]  
x[5]
```

IndexError: range object
index out of range



files are also sequences

- a file is a sequence of strings
- ... so we can use the keyword `in` to loop through the lines of a file!

```
file = open("filename.txt", "r")
for line in file:
    print(line)
file.close()
```

Reading and Writing Files

- reading from a file:

```
file_in = open("filename.txt", "r")
for line in file_in:
    # do something with each line
file_in.close()
```

- can use `file_in.readline()` to get one line at a time
- reading from a file keeps going from where it was
- writing to a file:

```
file_out = open("filename.txt", "w")
char1 = file_out.write("Line 1 in the file\n")
char2 = file_out.write("Next line in the file\n")
file_out.close()
```

Remember to close your files!

Example

- Define a function `count_chars` that takes on parameter `filename` and returns the number of characters in that file.

Exercise

- Define a function `combine_files` that takes three arguments (`infile1`, `infile2`, `outfile`), all of which are strings, and creates a new file named `outfile` whose contents are the contents of the file named `infile1` followed by the contents of the file named `infile2`.

```
def example1(filename):  
    s = 0  
  
    file = open(filename, "r")  
    for line in file:  
        print(line)  
    file.close()  
  
    print(s)
```

- what if the file doesn't exist?
- what if it does exist but you don't have access permissions?

aside on lab/assignment

- Read before lab: “All I Really Need to Know about Pair Programming I Learned in Kindergarten”:
<https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF>
- Think carefully about whether you want to work with a partner on this assignment. Since you can only work on the code when both of you are in the same physical location, you should think about when you’ll want to work on the assignment and how much time you’ll want to put in. Working with a partner is a commitment. Once you’ve checked in this lab, you may not change your mind about who you’re working with, or whether you’re working alone or in a pair.
- If you want to work in a pair, please try to find a partner for this assignment. Let the professors know if you’d like help finding someone to work with.