Chapter 4: Text Data, Files, and Exceptions

Resource: Introduction to Computing Using Python by Ljubomir Perkovic

Topics to cover:

- 4.1 Strings, Revisited
- 4.2 Formatted Output
- 4.3 Files
- 4.4 Errors and Exceptions

Strings

Don't forgot the quotes and match sure they match!

Switch the quotes if you are using quotes in your string

Or use the *escape sequence* **(\)** if both types of quotes are used to indicate that a quote is not a string delimiter but part of the string value

For multi-line text use (1) triple quotes or (2) use new line characters **(\n)**

Indexing Operator

Indexing operator: **s[i]** - takes index *i* and returns single-character string at index *i*

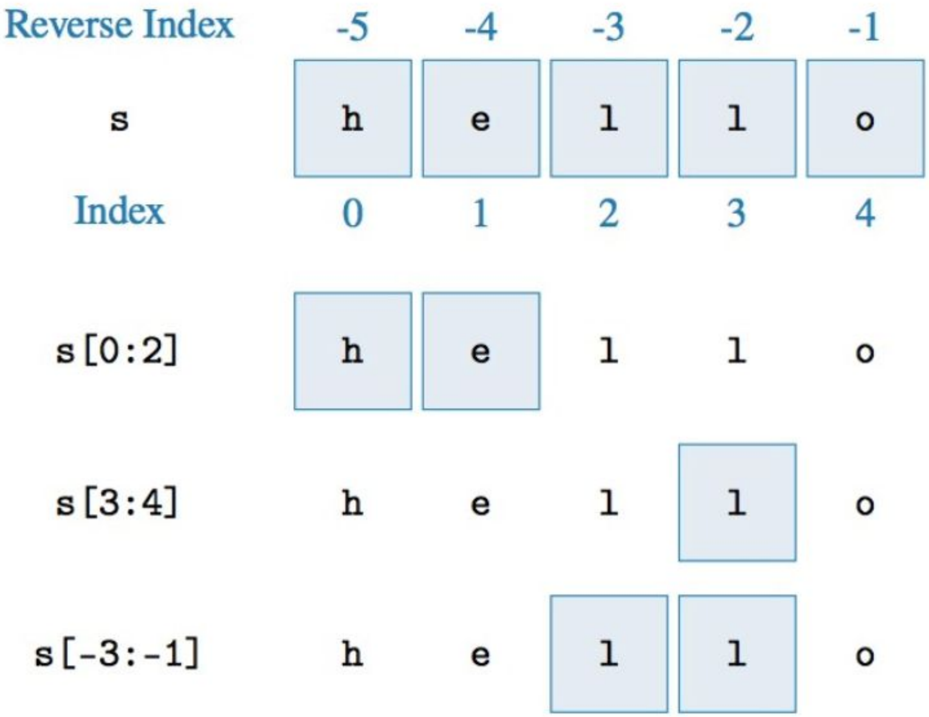
Slicing: **s[i,j]** - substring of string *s* (slice) that starts at index *i* and ends at index *j* - 1 (before *j*)

If the slice we want starts at the first character of string, we can drop the first index

If the slice we want ends with the last character of string, we must drop the second index

Can also get slices using negative indexes

Indexing Operator



Reverse Index	-5	-4	-3	-2	-1
s	h	e	l	l	o
Index	0	1	2	3	4
s[0:2]	h	e	l	l	o
s[3:4]	h	e	l	l	o
s[-3:-1]	h	e	l	l	o

Indexing operator - lists

Can also be used to obtain a slice of a list **s[i:j]**

Slice of a list **s[i:j]** is a list, returns a list when indexing operator is applied to a list with two arguments

Note: different from case when indexing operator is applied to a list with only one argument, **s[i]**, the ***item*** of the list is returned

String Methods

find() - called by string and checks if input is a substring of string, returns index of first character of first occurrence else returns -1

count() - called by string and counts occurrences of input, returns count

replace() - called by string and takes 2 string inputs, *old string* and *new string*, outputs a ***copy*** of string with old string replaced by new string (string are immutable!)

String Methods cont.

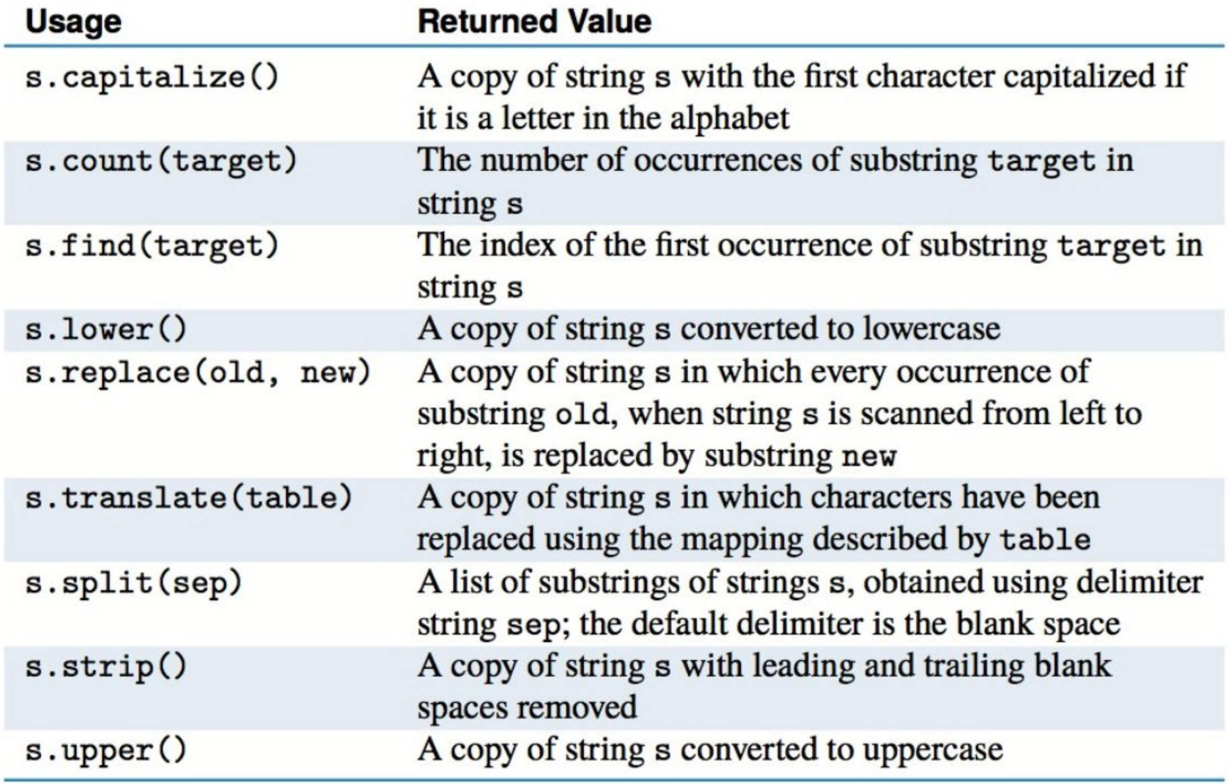
capitalize() - called by string and makes the first character of string uppercase

upper() - called by string and makes all letters uppercase

split() - called on string to get a list of words in string, splits on input given (acts as delimiter)

help()

String Methods cont.



Usage	Returned Value
`s.capitalize()`	A copy of string `s` with the first character capitalized if it is a letter in the alphabet
`s.count(target)`	The number of occurrences of substring `target` in string `s`
`s.find(target)`	The index of the first occurrence of substring `target` in string `s`
`s.lower()`	A copy of string `s` converted to lowercase
`s.replace(old, new)`	A copy of string `s` in which every occurrence of substring `old`, when string `s` is scanned from left to right, is replaced by substring `new`
`s.translate(table)`	A copy of string `s` in which characters have been replaced using the mapping described by `table`
`s.split(sep)`	A list of substrings of strings `s`, obtained using delimiter string `sep`; the default delimiter is the blank space
`s.strip()`	A copy of string `s` with leading and trailing blank spaces removed
`s.upper()`	A copy of string `s` converted to uppercase

Formatted Output - print

Used to print values onto screen

Takes arbitrary number of input objects (does not need to be same type)

Separator argument: default print separator are blanks **(‘ ‘)** but can change the separator with ***sep*** argument

End argument: ending for each string, use ***end*** argument

Formatted Output - format vs f strings

format() - using empty placeholders, can use explicit placeholders or default placeholders

f strings (formatted string literals) - using better named placeholders, remember **f** in front

Files