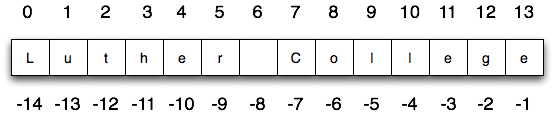
**Collection Data Types**

* Strings contain characters <https://docs.python.org/2.4/lib/string-methods.html>
* Operations (follow arithmetic sequence)
  + Concatenation: + operator, join end-to-end
  + Repetition: \* operator
  + Slicing: string[start : stop+1 : step]
  + Length: len(string)
  + Cannot assign a new value to string[0] (immutable)
  + in & not in: tests presence of substring (‘’ is substring of everything)
* Indexing: use square brackets
  + 
  + m = school[2] #creates new string called m
* Dot notation
* Comparison
  + Equality operator
  + Lexicographical order (uppercase first, and alphabetical order)
    - Apple>banana>apple
    - Ordinal value of given character: ord
    - Convert integer value to char: chr
* Iteration
  + for idx in range(len(“apple”)):
  + for i, char in enumerate(iterable, start=0):

| **Method** | **Parameters** | **Description** |
| --- | --- | --- |
| upper | none | Returns a string in all uppercase |
| lower | none | Returns a string in all lowercase |
| capitalize | none | Returns a string with first character capitalized, the rest lower |
| strip | none | Returns a string with the leading and trailing whitespace removed |
| lstrip | none | Returns a string with the leading whitespace removed |
| rstrip | none | Returns a string with the trailing whitespace removed |
| split | separator | Returns a list of strings that have been separated |
| count | item | Returns the number of occurrences of item |
| replace | old, new | Replaces all occurrences of old substring with new |
| center | width | Returns a string centered in a field of width spaces |
| ljust | width | Returns a string left justified in a field of width spaces |
| rjust | width | Returns a string right justified in a field of width spaces |
| find | item | Returns the leftmost index where the substring item is found, or -1 if not found |
| rfind | item | Returns the rightmost index where the substring item is found, or -1 if not found |
| index | item | Like find except causes a runtime error if item is not found |
| rindex | item | Like rfind except causes a runtime error if item is not found |
| format | substitutions | Involved! See [String Format Method](https://runestone.academy/runestone/books/published/thinkcspy/Strings/StringMethods.html#format-strings), below |

**Formatting**

* {field\_name:conversion} is a placeholder
* Conversions s(string) d(decimal int base 10) f(float) c(char) b(binary)
  + print("{}'s temperature is {:.2f} Celsius.".format("Oka", 36.732))
  + print("Every {3} should know the use of {2} {1} programming and {0}".format("programmer", "Open", "Source", "Operating Systems"))
* Text-align: by default, left-align for strings, right-align for numbers, ^(center) >(right) <(left)
  + print(“It is {0:5} degrees outside”.format(40))

**L-systems**

<https://runestone.academy/runestone/books/published/thinkcspy/Strings/TurtlesandStringsandLSystems.html>

**Files**

* finding a file: create a relative file path starting from the folder that contains the python program and follows a computer’s file hierarchy.
  + Parent folder: use .. (unique parent folder for each file)

| **Method Name** |  | **Use** | **Explanation** |
| --- | --- | --- | --- |
| open |  | open(filename,'r') | Open a file called filename and use it for reading. This will return a reference to a file object. |
| open |  | open(filename,'w') | Open a file called filename and use it for writing. This will also return a reference to a file object. |
| close |  | filevariable.close() | File use is complete. |
| write |  | Filevar.write(astring) | Add a string to the end of the file. filevar must refer to a file that has been opened for writing. |
| read(n) |  | Filevar.read(n) | Reads and returns a string of n characters, or the entire file as a single string if n is not provided. |
| readline(n) |  | filevar.readline(n) | Returns the next line of the file with all text up to and including the newline character. If n is provided as a parameter than only n characters will be returned if the line is longer than n. |
| readlines(n) |  | filevar.readlines(n) | Returns a list of strings, each representing a single line of the file. If n is not provided then all lines of the file are returned. If n is provided then n characters are read but n is rounded up so that an entire line is returned. |

* File Reading Methods: need to reopen the file before each read in order to start from beginning to move the marker (indicating current read position) back to the start
* File Writing Methods
  + need to close both the outfile and infile
  + add newline after each outfile.write(dataline + '\n')

infile = open("qbdata.txt", "r")  
line = infile.readline() *# priming read*  
while line:  
 values = line.split()  
 print('QB ', values[0], values[1], 'had a rating of ', values[10] )  
 line = infile.readline() *# reassign line variable to next line of file*  
infile.close()

**With**

with open('mydata.txt') as md:  
 line = md.readline()  
 while line:  
 print(line)  
 line = md.readline()  
print(md)

The context manager automates the process of doing common operations at the start of some task, as well as automating certain operations at the end of some task. In the context of reading and writing a file, the normal operation is to open the file and assign it to a variable. At the end of working with a file the common operation is to make sure that file is closed.

Open methods:

* r >> open file for reading (default)
* w >> open file for writing, creates new file if non-existent , truncates the file if existent
* a >> open file for appending without truncating it, creates new file if non-existent,
* r+ >> for reading and writing
* a+ >> for reading and appending, creates new file if non-existent