COMP 10261: Python Strings

Sam Scott, Mohawk College, 2021

Strings are a lot like lists, except that they are **immutable**. This means it is not possible to change the contents of a string once it's created.

STRING OPERATORS

Operators are symbols or keywords in the language that change or combine the contents of variables. Here are some important string operators:

`s + b`	Concatenates `a` and `b` into a new string
`s * n`	Creates a new string by repeating (`n` times) the contents of the string `a`
`s[i]`	Returns a new single character string from location `i` in string `s`
`s[i:j]`	Returns a new string slice from string `s` (both `i` and `j` are optional)
`e in s`	Returns `True` if `e` is a substring of `s`
`e not in s`	Returns `True` if `e` is not a substring of `s`

STRING FUNCTIONS

The following are built in global functions that operate on strings.

`len(s)`	Returns the length of `s`
`str(x)`	Convert x to a string

STRING METHODS

In Python, strings are objects with a number of built-in methods.

`s.strip()`	Returns a copy of s with leading and trailing whitespace removed
`s.lower()`	Returns lowercase version of s
`s.upper()`	Returns uppercase version of s
`s.find(e)`	Returns the index of the substring `e` (or -1 if not found)
`s.startswith(e)`	Returns true if the string starts with `e`
`s.endsswith(e)`	Returns true if the string ends with `e`
`s.count(e)`	Returns a count of the occurrences of substring `e`
`s.split()`	Returns a list of each token ("word") in the string.

PROCESSING STRINGS

To process a string is to "visit" (print, change, etc.) every character of the list. You can use the same **for loop** patterns as you would for lists.