In this module we will be covering the topics of string operations, string slicing, and testing, searching and manipulating strings.

String Operations

PYTHON has several ways to access individual characters in a string, and has methods that let you perform operations on them.

Accessing Individual Characters in a String

One of the easiest ways to access individual characters within a string is to use the *for loop.*

for variable in string:

statement

statement

etc.

You can also access individual characters in a string with an index. Each string character is treated like an element in a list starting with position "0".

String Concatenation

You can append one string to another using the "+" operation.

String are immutable - once created they CANNOT be changed.

String Slicing

Slicing expressions can be used to select a range of characters from a string.

string[start : end]

Testing, Searching and Manipulating Strings

There are operators and methods for testing strings, searching the contents of strings, and getting modified copies of strings.

in And not in Operators

string 1 in string 2

String Methods

stringvar.method(arguments)

isalnum() returns true if string contains only alphabetic letters or digits and is at least 1 character in length - returns false otherwise

isalpha() returns true if the string contains only alphabetic letters and is at least 1 character in length - returns false otherwise

isdigit() returns true if the string contains only numeric digits and is at least 1 character in length - returns false

otherwise

islower() returns true if all the alphabetic letters are lowercase, and contains at least 1 alphabetic characters - returns false otherwise

isspace() returns true if the string contains only whitespace characters and is at least 1 character in length - returns false otherwise

(whitespace characters are: spaces, newlines (\n), and tabs (\t))

isupper() returns true if all of the alphabetic letters are uppercase, and the string has at least 1 alphabetic letter - returns false otherwise

Modification Methods

Lower() returns copy with all characters converted to lowercase

lstrip() returns copy with all whitespace characters removed

lstrip(char) char argument is a string containing a characters - returns copy of the string with all instances of char that appear at the beginning

of the string removed

rstrip() returns copy with all whitespace characters at the end removed

rstrip(char) same as lstrip(char) but at the end

strip() returns a copy of string with all leading and trailing whitespace characters removed

strip(char) returns a copy of string with char characters at beginning and end removed

upper() returns a coy of the string with all alphabetic letters converted to uppercase.

Search and Replace Methods

endswith(substring) substring argument is a string - returns true if the string ends with substring

find(substring) substring argument is a string - returns lowest index in the string where substring is found

replace(old, new) old and new arguments are both strings - returns a copy of the string with all instances of old replaced by new

startswith(substring) substring is a string - returns true if the string starts with the substring

Repetition Operator

string_to_copy * *n*