**Data Types & Methods**

Data Types

Data types are the classification or categorization of data items. It represents the kind of value that tells what operations can be performed on a particular data. Since everything is an object in Python programming, data types are actually classes and variables are instance (object) of these classes.

Python has the following data types built-in by default, in these categories:

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | dict |
| Set Types: | set, frozenset |
| Boolean Type: | bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

You can get the data type of any object by using the type() function.

Strings

In Python, [Strings](https://www.geeksforgeeks.org/python-strings/) are arrays of bytes representing Unicode characters. A string is a collection of one or more characters put in a single quote, double-quote or triple quote. In python there is no character data type, a character is a string of length one. It is represented by str class.



Methods

capitalize()

Converts the first character to upper case

casefold()

Converts string into lower case

center()

Returns a centered string

count()

Returns the number of times a specified value occurs in a string

encode()

Returns an encoded version of the string

endswith()

Returns true if the string ends with the specified value

expandtabs()

Sets the tab size of the string

find()

Searches the string for a specified value and returns the position of where it was found

format()

Formats specified values in a string

format\_map()

Formats specified values in a string

index()

Searches the string for a specified value and returns the position of where it was found

isalnum()

Returns True if all characters in the string are alphanumeric

isalpha()

Returns True if all characters in the string are in the alphabet

isdecimal()

Returns True if all characters in the string are decimals

isdigit()

Returns True if all characters in the string are digits

isidentifier()

Returns True if the string is an identifier

islower()

Returns True if all characters in the string are lower case

isnumeric()

Returns True if all characters in the string are numeric

isprintable()

Returns True if all characters in the string are printable

isspace()

Returns True if all characters in the string are whitespaces

istitle()

Returns True if the string follows the rules of a title

isupper()

Returns True if all characters in the string are upper case

join()

Joins the elements of an iterable to the end of the string

ljust()

Returns a left justified version of the string

lower()

Converts a string into lower case

lstrip()

Returns a left trim version of the string

maketrans()

Returns a translation table to be used in translations

partition()

Returns a tuple where the string is parted into three parts

replace()

Returns a string where a specified value is replaced with a specified value

rfind()

Searches the string for a specified value and returns the last position of where it was found

rindex()

Searches the string for a specified value and returns the last position of where it was found

rjust()

Returns a right justified version of the string

rpartition()

Returns a tuple where the string is parted into three parts

rsplit()

Splits the string at the specified separator, and returns a list

rstrip()

Returns a right trim version of the string

split()

Splits the string at the specified separator, and returns a list

splitlines()

Splits the string at line breaks and returns a list

startswith()

Returns true if the string starts with the specified value

strip()

Returns a trimmed version of the string

Numeric

In Python, numeric data type represent the data which has numeric value. Numeric value can be integer, floating number or even complex numbers. These values are defined as int, float and complex class in Python.

**Integers**–

This value is represented by int class. It contains positive or negative whole numbers (without fraction or decimal). In Python there is no limit to how long an integer value can be.

**Float**–

This value is represented by float class. It is a real number with floating point representation. It is specified by a decimal point. Optionally, the character e or E followed by a positive or negative integer may be appended to specify scientific notation.

**Complex Numbers** –

Complex number is represented by complex class. It is specified as *(real part) + (imaginary part)j*. For example – 2+3j



Sequence

In Python, sequence is the ordered collection of similar or different data types. Sequences allows to store multiple values in an organized and efficient fashion.

List

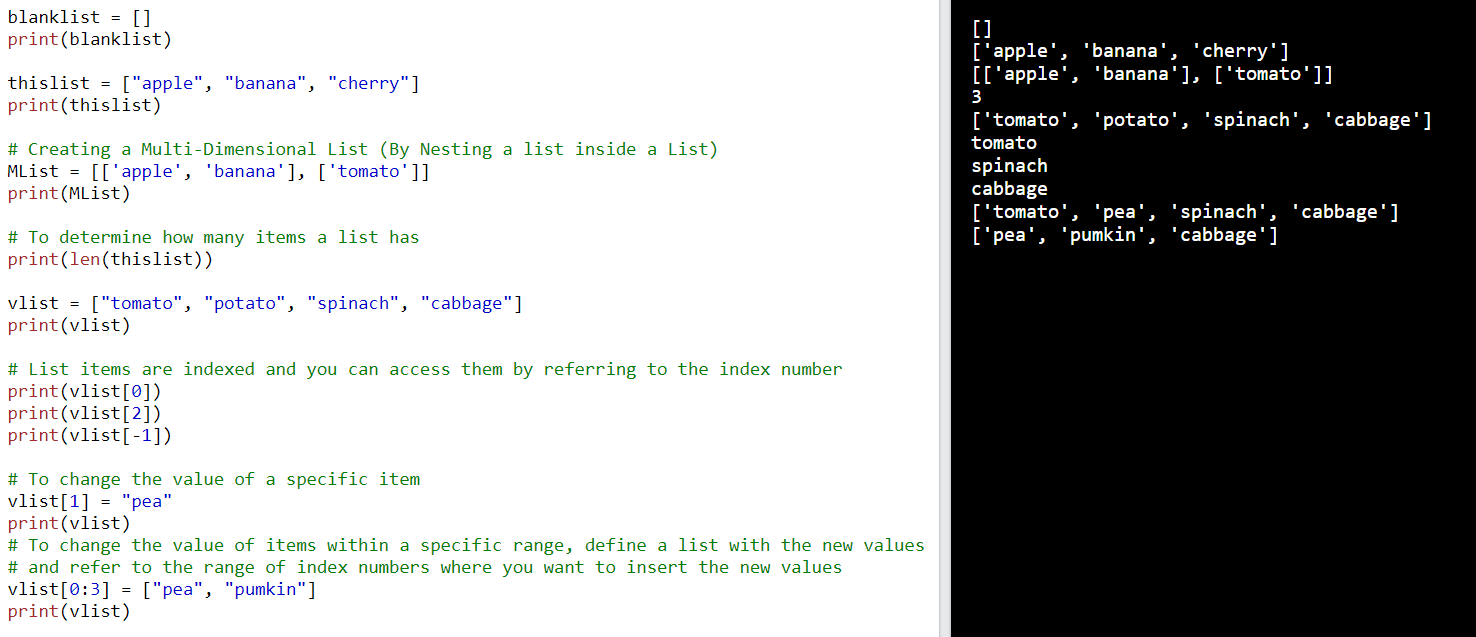
[Lists](https://www.geeksforgeeks.org/python-list/) are just like the arrays, declared in other languages which is a ordered collection of data.

It is very flexible as the items in a list do not need to be of the same type.

Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are [Tuple](https://www.w3schools.com/python/python_tuples.asp), [Set](https://www.w3schools.com/python/python_sets.asp), and [Dictionary](https://www.w3schools.com/python/python_dictionaries.asp), all with different qualities and usage.

Lists are created using square brackets.



Methods

append()

Adds an element at the end of the list

clear()

Removes all the elements from the list

copy()

Returns a copy of the list

count()

Returns the number of elements with the specified value

extend()

Add the elements of a list (or any iterable), to the end of the current list

index()

Returns the index of the first element with the specified value

insert()

Adds an element at the specified position

pop()

Removes the element at the specified position

remove()

Removes the item with the specified value

reverse()

Reverses the order of the list

sort()

Sorts the list