

Python Reference Sheet

Note: This list is not exhaustive. See the source links below for more details.

[PDF Version](#)

Common Sequence Methods

Source: <https://docs.python.org/3/library/stdtypes.html#typeseq-common>

s.count(x)

Return total number of occurrences of *x* in *s*.

s.index(x[, i[, j]])

Return the index of the first occurrence of *x* in *s* (at or after index *i* and before index *j*).

Mutable Sequence Methods

Source: <https://docs.python.org/3/library/stdtypes.html#typeseq-mutable>

s.append(x)

Append *x* to the end of the sequence.

s.insert(i, x)

Insert *x* into *s* at the index given by *i*.

s.pop(i=-1)

Retrieve the item at *i* and also remove it from *s*.

s.remove(x)

Remove the first item from *s* where `s[i]` is equal to *x*.

Dictionary Methods

Source: <https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>

dict.get(key, default=None)

Return the value for *key* if *key* is in the dictionary, else *default*. If *default* is not given, it defaults to `None`, so that this method never raises a `KeyError`.

dict.items()

Return a new view of the dictionary's items ((*key*, *value*) pairs).

dict.keys()

Return a new view of the dictionary's keys.

dict.pop(key[, default])

If *key* is in the dictionary, remove it and return its value, else return *default*. If *default* is not given and *key* is not in the dictionary, a `KeyError` is raised.

dict.values()

Return a new view of the dictionary's values.

File Methods

Source: <https://docs.python.org/3/library/io.html#io.TextIOBase>

file.close()

Flush and close this stream. This method has no effect if the file is already closed. Once the file is closed, any operation on the file (e.g. reading or writing) will raise a `ValueError`.

file.flush()

Flush the write buffers of the stream if applicable. This does nothing for read-only and non-blocking streams.

file.read(size=-1, /)

Read and return at most *size* characters from the stream as a single str. If *size* is negative or `None`, reads until EOF.

file.readline(size=-1, /)

Read until newline or EOF and return a single str. If the stream is already at EOF, an empty string is returned. If *size* is specified, at most *size* characters will be read.

file.readlines(hint=-1, /)

Read and return a list of lines from the stream. *hint* can be specified to control the number of lines read: no more lines will be read if the total size (in bytes/characters) of all lines so far exceeds *hint*.

file.write(s, /)

Write the string *s* to the stream and return the number of characters written.

file.writelines(lines, /)

Write a list of lines to the stream. Line separators are not added, so it is usual for each of the lines provided to have a line separator at the end.

String Methods

Source: <https://docs.python.org/3/library/stdtypes.html#string-methods>

`str.capitalize()`

Return a copy of the string with its first character capitalized and the rest lowercased.

`str.endswith(suffix[, start[, end]])`

Return `True` if the string ends with the specified `suffix`, otherwise return `False`. `suffix` can also be a tuple of suffixes to look for. With optional `start`, test beginning at that position. With optional `end`, stop comparing at that position.

`str.find(sub[, start[, end]])`

Return the lowest index in the string where substring `sub` is found within the slice `s[start:end]`. Optional arguments `start` and `end` are interpreted as in slice notation. Return `-1` if `sub` is not found.

`str.isalpha()`

Return `True` if all characters in the string are alphabetic and there is at least one character, `False` otherwise.

`str.isdigit()`

Return `True` if all characters in the string are digits and there is at least one character, `False` otherwise.

`str.islower()`

Return `True` if all cased characters in the string are lowercase and there is at least one cased character, `False` otherwise.

`str.isspace()`

Return `True` if there are only whitespace characters in the string and there is at least one character, `False` otherwise.

`str.istitle()`

Return `True` if the string is a titlecased string and there is at least one character, for example uppercase characters may only follow uncased characters and lowercase characters only cased ones. Return `False` otherwise.

`str.isupper()`

Return `True` if all cased characters in the string are uppercase and there is at least one cased character, `False` otherwise.

`str.join(iterable)`

Return a string which is the concatenation of the strings in `iterable`. A `TypeError` will be raised if there are any non-string values in `iterable`, including bytes objects. The separator between elements is the string providing this method.

`str.lower()`

Return a copy of the string with all the cased characters converted to lowercase.

`str.replace(old, new, count=-1)`

Return a copy of the string with all occurrences of substring `old` replaced by `new`. If `count` is given, only the first `count` occurrences are replaced. If `count` is not specified or `-1`, then all occurrences are replaced.

`str.split(sep=None, maxsplit=-1)`

Return a list of the words in the string, using `sep` as the delimiter string. If `maxsplit` is given, at most `maxsplit` splits are done (thus, the list will have at most `maxsplit+1` elements). If `maxsplit` is not specified or `-1`, then there is no limit on the number of splits (all possible splits are made).

`str.splitlines(keepends=False)`

Return a list of the lines in the string, breaking at line boundaries. Line breaks are not included in the resulting list unless `keepends` is given and true.

`str.startswith(prefix[, start[, end]])`

Return `True` if string starts with the `prefix`, otherwise return `False`. `prefix` can also be a tuple of prefixes to look for. With optional `start`, test string beginning at that position. With optional `end`, stop comparing string at that position.

`str.strip([chars])`

Return a copy of the string with the leading and trailing characters removed. The `chars` argument is a string specifying the set of characters to be removed. If omitted or `None`, the `chars` argument defaults to removing whitespace.

`str.title()`

Return a titlecased version of the string where words start with an uppercase character and the remaining characters are lowercase.

`str.upper()`

Return a copy of the string with all the cased characters converted to uppercase.