|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| is | Evaluates to true if the variables on either side of the operator point to the same object and false otherwise. | x is y, here **is** results in 1 if id(x) equals id(y). |
| is not | Evaluates to false if the variables on either side of the operator point to the same object and true otherwise. | x is not y, here **is not** results in 1 if id(x) is not equal to id(y). |
| in | Evaluates to true if it finds a variable in the specified sequence and false otherwise. | x in y, here in results in a 1 if x is a member of sequence y. |
| not in | Evaluates to true if it does not finds a variable in the specified sequence and false otherwise. | x not in y, here not in results in a 1 if x is not a member of sequence y. |

Mathematical Functions

|  |  |
| --- | --- |
| **Function** | **Returns ( description )** |
| [**abs(x)**](https://www.tutorialspoint.com/python/number_abs.htm) | The absolute value of x: the (positive) distance between x and zero. |
| [cmp(x, y)](https://www.tutorialspoint.com/python/number_cmp.htm) | -1 if x < y, 0 if x == y, or 1 if x > y |
| [exp(x)](https://www.tutorialspoint.com/python/number_exp.htm) | The exponential of x: ex |
| [fabs(x)](https://www.tutorialspoint.com/python/number_fabs.htm) | The absolute value of x. |
| [max(x1, x2,...)](https://www.tutorialspoint.com/python/number_max.htm) | The largest of its arguments: the value closest to positive infinity |
| [min(x1, x2,...)](https://www.tutorialspoint.com/python/number_min.htm) | The smallest of its arguments: the value closest to negative infinity |
| [pow(x, y)](https://www.tutorialspoint.com/python/number_pow.htm) | The value of x\*\*y. |
| [round(x [,n])](https://www.tutorialspoint.com/python/number_round.htm) | x rounded to n digits from the decimal point. Python rounds away from zero as a tie-breaker: round(0.5) is 1.0 and round(-0.5) is -1.0. |
| [sqrt(x)](https://www.tutorialspoint.com/python/number_sqrt.htm) | The square root of x for x > 0 |
| [choice(seq)](https://www.tutorialspoint.com/python/number_choice.htm) | A random item from a list, tuple, or string. |

## String Special Operators

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Concatenation - Adds values on either side of the operator | a + b will give HelloPython |
| \* | Repetition - Creates new strings, concatenating multiple copies of the same string | a\*2 will give -HelloHello |
| [] | Slice - Gives the character from the given index | a[1] will give e |
| [ : ] | Range Slice - Gives the characters from the given range | a[1:4] will give ell |
| in | Membership - Returns true if a character exists in the given string | H in a will give 1 |
| not in | Membership - Returns true if a character does not exist in the given string | M not in a will give 1 |
| r/R | Raw String - Suppresses actual meaning of Escape characters. The syntax for raw strings is exactly the same as for normal strings with the exception of the raw string operator, the letter "r," which precedes the quotation marks. The "r" can be lowercase (r) or uppercase (R) and must be placed immediately preceding the first quote mark. | print r'\n' prints \n and print R'\n'prints \n |
| % | Format - Performs String formatting | See at next section |

print("{:.2f}".format(3.1415926));

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Format** | **Output** | **Description** |
| 3.1415926 | {:.2f} | 3.14 | 2 decimal places |
| 3.1415926 | {:+.2f} | +3.14 | 2 decimal places with sign |
| -1 | {:+.2f} | -1.00 | 2 decimal places with sign |
| 2.71828 | {:.0f} | 3 | No decimal places |
| 5 | {:0>2d} | 05 | Pad number with zeros (left padding, width 2) |
| 5 | {:x<4d} | 5xxx | Pad number with x's (right padding, width 4) |
| 10 | {:x<4d} | 10xx | Pad number with x's (right padding, width 4) |
| 1000000 | {:,} | 1,000,000 | Number format with comma separator |
| 0.25 | {:.2%} | 25.00% | Format percentage |
| 1000000000 | {:.2e} | 1.00e+09 | Exponent notation |
| 13 | {:10d} | 13 | Right aligned (default, width 10) |
| 13 | {:<10d} | 13 | Left aligned (width 10) |
| 13 | {:^10d} | 13 | Center aligned (width 10) |

string.**ascii\_letters**

The concatenation of the **[ascii\_lowercase](https://docs.python.org/2/library/string.html" \l "string.ascii_lowercase" \o "string.ascii_lowercase)** and **[ascii\_uppercase](https://docs.python.org/2/library/string.html" \l "string.ascii_uppercase" \o "string.ascii_uppercase)**constants described below. This value is not locale-dependent.

string.**ascii\_lowercase**

The lowercase letters 'abcdefghijklmnopqrstuvwxyz'. This value is not locale-dependent and will not change.

string.**ascii\_uppercase**

The uppercase letters 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'. This value is not locale-dependent and will not change.

string.**digits**

The string '0123456789'.

string.**hexdigits**

The string '0123456789abcdefABCDEF'.

string.**letters**

The concatenation of the strings [**lowercase**](https://docs.python.org/2/library/string.html#string.lowercase) and **[uppercase](https://docs.python.org/2/library/string.html" \l "string.uppercase" \o "string.uppercase)**described below. The specific value is locale-dependent, and will be updated when **[locale.setlocale()](https://docs.python.org/2/library/locale.html" \l "locale.setlocale" \o "locale.setlocale)** is called.

string.**lowercase**

A string containing all the characters that are considered lowercase letters. On most systems this is the string 'abcdefghijklmnopqrstuvwxyz'. The specific value is locale-dependent, and will be updated when **[locale.setlocale()](https://docs.python.org/2/library/locale.html" \l "locale.setlocale" \o "locale.setlocale)** is called.

string.**uppercase**

A string containing all the characters that are considered uppercase letters. On most systems this is the string 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'. The specific value is locale-dependent, and will be updated when **[locale.setlocale()](https://docs.python.org/2/library/locale.html" \l "locale.setlocale" \o "locale.setlocale)** is called.

|  |  |
| --- | --- |
| **SN** | **Methods with Description** |
| 1 | [**capitalize()**](https://www.tutorialspoint.com/python/string_capitalize.htm) Capitalizes first letter of string |
| 2 | [**center(width, fillchar)**](https://www.tutorialspoint.com/python/string_center.htm)  Returns a space-padded string with the original string centered to a total of width columns. |
| 3 | [**count(str, beg= 0,end=len(string))**](https://www.tutorialspoint.com/python/string_count.htm) Counts how many times str occurs in string or in a substring of string if starting index beg and ending index end are given. |
| 6 | [**endswith(suffix, beg=0, end=len(string))**](https://www.tutorialspoint.com/python/string_endswith.htm) Determines if string or a substring of string (if starting index beg and ending index end are given) ends with suffix; returns true if so and false otherwise. |
| 8 | [**find(str, beg=0 end=len(string))**](https://www.tutorialspoint.com/python/string_find.htm) Determine if str occurs in string or in a substring of string if starting index beg and ending index end are given returns index if found and -1 otherwise. |
| 9 | [**index(str, beg=0, end=len(string))**](https://www.tutorialspoint.com/python/string_index.htm) Same as find(), but raises an exception if str not found. |
| 10 | [**isalnum()**](https://www.tutorialspoint.com/python/string_isalnum.htm) Returns true if string has at least 1 character and all characters are alphanumeric and false otherwise. |
| 11 | [**isalpha()**](https://www.tutorialspoint.com/python/string_isalpha.htm) Returns true if string has at least 1 character and all characters are alphabetic and false otherwise. |
| 12 | [**isdigit()**](https://www.tutorialspoint.com/python/string_isdigit.htm) Returns true if string contains only digits and false otherwise. |
| 13 | [**islower()**](https://www.tutorialspoint.com/python/string_islower.htm) Returns true if string has at least 1 cased character and all cased characters are in lowercase and false otherwise. |
| 14 | [**isnumeric()**](https://www.tutorialspoint.com/python/string_isnumeric.htm) Returns true if a unicode string contains only numeric characters and false otherwise. |
| 15 | [**isspace()**](https://www.tutorialspoint.com/python/string_isspace.htm) Returns true if string contains only whitespace characters and false otherwise. |
| 17 | [**isupper()**](https://www.tutorialspoint.com/python/string_isupper.htm) Returns true if string has at least one cased character and all cased characters are in uppercase and false otherwise. |
| 18 | [**join(seq)**](https://www.tutorialspoint.com/python/string_join.htm) Merges (concatenates) the string representations of elements in sequence seq into a string, with separator string. |
| 19 | [**len(string)**](https://www.tutorialspoint.com/python/string_len.htm) Returns the length of the string |
| 20 | [**ljust(width[, fillchar])**](https://www.tutorialspoint.com/python/string_ljust.htm) Returns a space-padded string with the original string left-justified to a total of width columns. |
| 21 | [**lower()**](https://www.tutorialspoint.com/python/string_lower.htm) Converts all uppercase letters in string to lowercase. |
| 22 | [**lstrip()**](https://www.tutorialspoint.com/python/string_lstrip.htm) Removes all leading whitespace in string. |
| 24 | [**max(str)**](https://www.tutorialspoint.com/python/string_max.htm) Returns the max alphabetical character from the string str. |
| 25 | [**min(str)**](https://www.tutorialspoint.com/python/string_min.htm) Returns the min alphabetical character from the string str. |
| 26 | [**replace(old, new [, max])**](https://www.tutorialspoint.com/python/string_replace.htm) Replaces all occurrences of old in string with new or at most max occurrences if max given. |
| 27 | [**rfind(str, beg=0,end=len(string))**](https://www.tutorialspoint.com/python/string_rfind.htm) Same as find(), but search backwards in string. |
| 28 | [**rindex( str, beg=0, end=len(string))**](https://www.tutorialspoint.com/python/string_rindex.htm) Same as index(), but search backwards in string. |
| 29 | [**rjust(width,[, fillchar])**](https://www.tutorialspoint.com/python/string_rjust.htm) Returns a space-padded string with the original string right-justified to a total of width columns. |
| 30 | [**rstrip()**](https://www.tutorialspoint.com/python/string_rstrip.htm) Removes all trailing whitespace of string. |
| 31 | [**split(str="", num=string.count(str))**](https://www.tutorialspoint.com/python/string_split.htm) Splits string according to delimiter str (space if not provided) and returns list of substrings; split into at most num substrings if given. |
| 32 | [**splitlines( num=string.count('\n'))**](https://www.tutorialspoint.com/python/string_splitlines.htm) Splits string at all (or num) NEWLINEs and returns a list of each line with NEWLINEs removed. |
| 33 | [**startswith(str, beg=0,end=len(string))**](https://www.tutorialspoint.com/python/string_startswith.htm) Determines if string or a substring of string (if starting index beg and ending index end are given) starts with substring str; returns true if so and false otherwise. |
| 34 | [**strip([chars])**](https://www.tutorialspoint.com/python/string_strip.htm) Performs both lstrip() and rstrip() on string |
| 35 | [**swapcase()**](https://www.tutorialspoint.com/python/string_swapcase.htm) Inverts case for all letters in string. |
| 36 | [**title()**](https://www.tutorialspoint.com/python/string_title.htm) Returns "titlecased" version of string, that is, all words begin with uppercase and the rest are lowercase. |
| 37 | [**translate(table, deletechars="")**](https://www.tutorialspoint.com/python/string_translate.htm) Translates string according to translation table str(256 chars), removing those in the del string. |
| 38 | [**upper()**](https://www.tutorialspoint.com/python/string_upper.htm) Converts lowercase letters in string to uppercase. |
| 39 | [**zfill (width)**](https://www.tutorialspoint.com/python/string_zfill.htm) Returns original string leftpadded with zeros to a total of width characters; intended for numbers, zfill() retains any sign given (less one zero). |

LISTS

| **Operation** | **Result** | **Notes** |
| --- | --- | --- |
| s[i] = x | item *i* of *s* is replaced by *x* |  |
| s[i:j] = t | slice of *s* from *i* to *j* is replaced by the contents of the iterable *t* |  |
| del s[i:j] | same as s[i:j] = [] |  |
| s[i:j:k] = t | the elements of s[i:j:k] are replaced by those of *t* | (1) |
| del s[i:j:k] | removes the elements of s[i:j:k] from the list |  |
| s.append(x) | appends *x* to the end of the sequence (same ass[len(s):len(s)] = [x]) |  |
| s.clear() | removes all items from s (same as del s[:]) | (5) |
| s.copy() | creates a shallow copy of s (same as s[:]) | (5) |
| s.extend(t) or s += t | extends *s* with the contents of *t* (for the most part the same as s[len(s):len(s)] = t) |  |
| s \*= n | updates *s* with its contents repeated *n* times | (6) |
| s.insert(i, x) | inserts *x* into *s* at the index given by *i* (same as s[i:i]= [x]) |  |
| s.pop([i]) | retrieves the item at *i* and also removes it from *s* | (2) |
| s.remove(x) | remove the first item from *s* where s[i] == x | (3) |
| s.reverse() | reverses the items of *s* in place |  |