\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Python Standard Library functions and methods**

**FEEL FREE TO USE THIS FOR YOUR CA**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Any methods here are **instance methods** unless otherwise stated.

**Tk()**– creates a root-window component for a GUI-based application, into which other GUI components can be added. Needed for any GUI-based application that uses the likes of text-areas, labels, buttons etc. It is available from the **tkinter** module.

**title(***text***)**– allows the title-bar text of the window to be set to the value of the string *text*

**Text(***parent-window,* **width=***width,* **height=***height,* **font=***font***)** – creates a text-area component whose parent will be *parent-window*, whose width and height in pixels will be *width* and *height* respectively and whose font will be font. The font used can be specified using parentheses. For example, font=(“courier”,12) indicates that the font to be used for the text-area will have a “courier” style and a point-size of 12. It is available from the **tkinter** module.

**pack()** – this method decides how the component will be added to a parent window.

**insert(***location***,** *text***)** – this method decides exactly where some text will be written to a text-area. For example, the value **“1.0”** for *location* indicates that the text should be inserted on the first row and first column of the text-area i.e. in the top-left corner of it. *location* is the string that decides the location of the *text* to be written to the text-area. In order to just add onto the end of existing text in the text-area, the value of location should be **“end”**.

**delete**(*start location*, *end location*) – this method allows the user to remove text from a text-area, starting at *start location* and ending at *end location*. So, for example, to remove all the text on the text-area from the 3rd row onwards, you could use delete(“3.0”,”end”)

**range(***startnum*,*endnum*,*step***)** – this built-in function is used to generate and return a list that contains a set of whole numbers from *startnum* up to, but not including, *endnum.* Argument *step* decides the gap between the numbers. It is commonly used with for loops in Python.

**random()** – this function returns a pseudo-random floating-point value between 0 and up to, but **not including**, 1. It is defined in the random module.

**randint(***startnum*,*endnum***)** – this function returns a pseudo-random integer value between *startnum* up to **and including**, *endnum*. It is defined in the random module.

**pow(***x, y***)** – this function returns the value of *x* raised to the power of *y* where *x* and *y* are expected to be numeric. It is defined in the math module, but there is also a built-in function by the same name that does an equivalent job.

**sqrt(***x***)** – this function returns the square root of *x*, which is expected to be numeric. It is defined in the math module.

**sin(***x***)** – this function returns the trigonometric sine of *x*, where *x* is an angle in radians and is numeric. It is defined in the math module.

**cos(***x***)** – this function returns the trigonometric cosine of *x*, where *x* is an angle in radians and is numeric. It is defined in the math module.

**tan(***x***)** – this function returns the trigonometric tangent of *x*, where *x* is an angle in radians and is numeric. It is defined in the math module.

**float(***value***)** – this method returns a float whose value has been converted from the data *value*, if possible.

**int(***value***)** – this method returns an int whose value has been converted from the data *value*, if possible.

**str(***value***)** - this method returns a string whose value has been converted from the data *value*.

**len(***value***)** – this built-in function is used to determine and return the number of characters in a string *value.*

**ord(***value***)** – this built-in function is used to determine and return the number of ASCII code of a **single character** string *value.*

**startswith(***str***)** – method that can be called on a string object to determine whether or not the string begins with the letters specified by the argument *str*. If it does, it returns the bool value True, otherwise it returns False.

**endswith(***str***)** – method that can be called on a string object to determine whether or not the string ends with the letters specified by the argument *str*. If it does, it returns the bool value True, otherwise it returns False.

**count(***str***)** – method that can be called on a string object to determine and return the number of times the string *str* appears within it.

**find(***str***)** – method that can be called on a string object to determine whether or not it contains the string *str*. If it does, it returns the location of the first character of the string as an integer. If it doesn’t, it returns the value -1. This version examines the entire string.

**find(***str,startnum,endnum***)** – method that can be called on a string object to determine whether or not it contains the string *str*. If it does, it returns the location of the first character of the string as an integer. If it doesn’t, it returns the value -1. This version only examines the region of the string from location *startnum* to location *endnum*. The character at location *startnum* is included in the check but the one at location *endnum* is not.

**isdigit()** – method which checks to see whether the string object on which it is called only contains digit characters in the range ‘0’ to ‘9’ inclusive. If it does, it will return the bool value True, False otherwise. If the string is empty, it will return False.

**isalpha()** – method which checks to see whether the string object on which it is called only contains letters in the range ‘a’ to ‘z’ or ‘A’ to ‘Z’ inclusive. If it does, it will return the bool value True, False otherwise. If the string is empty, it will return False.

**isalnum()** – method which checks to see whether the string object on which it is called only contains letters in the range ‘a’ to ‘z’ or ‘A’ to ‘Z’ inclusive or digits. If it does, it will return the bool value True, False otherwise. If the string is empty, it will return False.

**isupper()** – method which checks to see whether the alphabetical characters in the string object on which it is called are all uppercase letters in the range ‘A’ to ‘Z’ inclusive. If they are, it will return the bool value True, False otherwise. If the string is empty, it will return False. It ignores all non-alphabetical characters in the string.

**islower()** – method which checks to see whether the alphabetical characters in the string object on which it is called are all lowercase letters in the range ‘a’ to ‘z’ inclusive. If they are, it will return the bool value True, False otherwise. If the string is empty, it will return False. It ignores all non-alphabetical characters in the string.

**lower()** – method which converts the string object on which it is called to its lowercase equivalent and returns this value

**upper()** – method which converts the string object on which it is called to its uppercase equivalent and returns this value

**askstring(***title-bar text***,** *message***)** – a **static** method of the class simpledialog that allows a small dialog box to appear which has a text-box in it allowing input to be supplied. The first (string) argument *title-bar text* will specify the title-bar text to be displayed in the title-bar, while the second (string) argument *message* specifies the message you will display to the user on the main window of the dialog. simpledialog is available from the **tkinter** package.

**showinfo(***title-bar text, message***)** – a **static** method of the class messagebox that allows a small dialog box to appear. The first argument *title-bar text* will be a string which gives the dialog some text in its title-bar while the second one (also a string) will be the *message* you wish to issue to the user. messagebox is available from the **tkinter** package.