Useful Python Constructs - Shortcut Sheet v1

The following functions are defined in terms of the *types of the argument(s)* and the *type of the result* if a type isn't specified for (e.g. x), any type can be supplied but the result might not be defined.

e.g.

len(x) → int	Take an object x and returns an integer
float(x) → float	Takes an object x and will return a floating point number (if possible)

The square brackets [ ] around a item means it can be omitted. e.g. *prompt* can be omitted here:

input(**[** prompt **]**) → str	Read a string from user.

Below, **str** means string, type **bool** (Boolean) type has values **True** and **False,** and ***Iterable*** means **list-like.**

Functions:	
input([prompt]) → str	Read a string from standard input (usually the keyboard)
abs(x) → number	Return the absolute value of x.
chr(x) → str	Returns the string value of x
ord(x) → int	Returns the ASCII value of x
int(x) → int	Convert x to an integer, if possible.
float(x) → float	Convert x to a float value, if possible.
len(x) → int	Return the length of x which can be a string, tuple or list or dictionary
max(iterable) → object	With a single iterable argument, return largest item.
max(a, b, c, ...) → object	Return the largest of two or more arguments.
min(iterable) → object	With a single iterable argument, return smallest item.
min(a, b, c, ...) → object	Return the smallest of two or more arguments.
open(name[, mode]) → file open for reading, writing	Open a text file, and return file handle Legal modes are "r","rt" (read), "w", "wt" (write)
range([start], stop, [step]) → list-like-object of int	Return the integers starting with *start* and ending with *stop-1* with *step* specifying the amount to increment (or decrement).
dict:	
x in D → bool	Returns True if x is a key in D
D[k] → object	Produce the value associated with the key k in D.
del D[k]	Remove D[k] from D.
D.clear()	Sets D to empty dictionary
D.copy()	Returns a copy of D
D.get(k) → object	Return D[k] if k in D, otherwise return None.
D.keys() → list-like-object of object	Return the keys of D.
D.values() → list-like-object of object	Return the values associated with the keys of D.
D.items() → list-like-object of (object, object)	Return the (key, value) pairs of D, as 2-tuples.
files	
open(filename, mode) → F (a file handle)	open a file, return the file handle
with open(filename, mode) as F:	open a file within a *with* context
F.close() → NoneType	Close the file.
F.read() → str	Read until EOF (End Of File) is reached, and return as a string.

F.readline() → str	Read & return the next line from the file as a string. Retain newline. Return an empty string at EOF
F.readline**s**() → list of str	Return a list of the lines from the file. Each string ends in a newline.
F.writeline(s)	Write the string s to the file.
F.writeline**s**(list of str)	Write a sequence (e.g. a list) of strings to the file. writelines() does **NOT** add line separators.
list:	
x in L → bool	Produce True if x is in L and False otherwise.
L.append(x) → NoneType	Append x to the end of the list L.
L.count(x)	Returns the number of occurrences of x in L
L.insert(index, x) → NoneType	Remove the first occurrence of value from L.
L.remove(value) → NoneType	Reverse *IN PLACE*.
L.reverse() → NoneType	Sort the list in ascending order.
L.sort() → NoneType	
str:	
x in S → bool	Produce True if and only if x is in S.
str(x) → str	Convert an object into its string representation, if possible.
S.capitalize() → str	Return a copy of the string S, capitalised.
S.endswith(suffix)	Return True if the string ends with the specified suffix, otherwise return False
S.find(sub[, i]) → int	Return the lowest index in S (starting at S[i], if *i* is given) where the string sub is found or -1 if *sub* does not occur in S.
S.isdigit() → bool	Return True if all characters in S are digits and False otherwise.
S.islower()	Return True if all characters in S are lower case and False otherwise.
S.isupper()	Return True if all characters in S are uppercase and False otherwise.
S.lower() → str	Return a copy of the string S converted to lowercase.
S.replace(old, new) → str	Return a copy of string S with all occurrences of the string old replaced with the string new.
S.split ([sep]) → list of str	Return a list of the words in S, using string sep as the separator, If sep is not specified, use contiguous whitespace as the separator.
S.startswith(prefix)	Return True if string starts with the prefix, otherwise return False.
S.strip () → str	Return a copy of S with leading and trailing whitespace removed.
S.title() → str	Return a copy of the string S with the first letter of each word capitalised.
S.upper() → str	Return a copy of string S converted to uppercase.