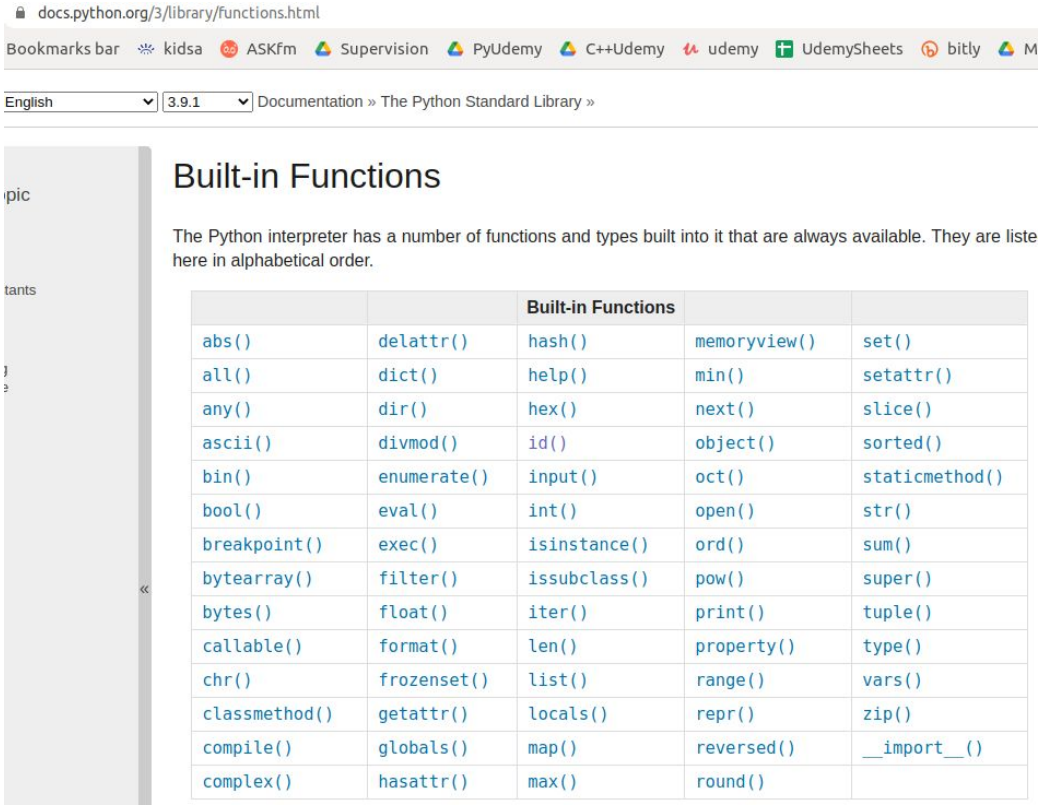
Python Programming

List with Functions

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Built-in Functions



		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

- We already know: *abs, min, id, enumerate, input, bool, int, str, sum, pow, float, print, len, type, range, globals, map, max, round*
- Some of them have relations with list

Flexible Reading

```
# .split() return list of strings
my_list = input().split()

for item in my_list:
    print(item, end=' ')
print()

# now list of integers
my_list = list(map(int, input().split()))

print(type(my_list), type(my_list[0]))  # list, int

for item in my_list:
    print(item, end=' ')
print()

# very helpful to read variable number of items on same line

```

sum, min, max, help functions

```
2
3   my_list = [4, 5, 7, 4, 5, 4, 8]
4
5   print(sum(my_list))      # 37
6
7   print(min(my_list), max(my_list))   # 3 8
8
9   my_list = ['ali', 'ziad', 'mostafa']
10
11  print(min(my_list), max(my_list))   # ali ziad
12
13  help(my_list.count)      # without () : passing a function
14
15  """
16  Help on built-in function count:
17
18  count(value, /) method of builtins.list instance
19      Return number of occurrences of value.
20  """
```

enumerate function

```
my_list = [1, 'mostafa', 4]
for idx, item in enumerate(my_list):
    print(idx, item)
    idx = -100  # no effect
"""
0 1
1 mostafa
2 4
"""

# NOTE: this creates a complete list in memory
# Slow for a huge range
lst = list(enumerate(range(5, 9)))

for item in lst:
    print(item)
"""
(0, 5)
(1, 6)
(2, 7)
(3, 8)
"""
```

all and functions

```
# all: Return True if all elements of the iterable are true
lst = [10, 20, -12, 'Mostafa']

print(all(lst))      # True
print(all([]))       # True

# items cause Faulse
print(all([False]))      # False
print(all(['']))         # False
print(all([0]))          # False

print(all([10, 0, 2]))          # False

# Return True if any element of the iterable is true
lst = [10, 20, 0, 'Mostafa']

print(all(lst))      # False
print(any([]))       # True
```

all and functions: Docs

all(*iterable*)

Return `True` if all elements of the *iterable* are true (or if the iterable is empty). Equivalent to:

```
def all(iterable):
    for element in iterable:
        if not element:
            return False
    return True
```

any(*iterable*)

Return `True` if any element of the *iterable* is true. If the iterable is empty, return `False`. Equivalent to:

```
def any(iterable):
    for element in iterable:
        if element:
            return True
    return False
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."