Python Lists Cheat Sheet

Python Lists

Introduction to lists

```
# Creating lists
list1 = []                  # empty list
list2 = list()              # empty list
list3 = list('Python')    # => ['P', 'y', 't', 'h', 'o', 'n'] -> creates a list from a string
list4 = ['Python', 'Go', 2018, 4.5, [1,2.3, 'abc']]    # a list stores any type of object
len(list4)                   # => 5 -> returns the number of elements in the list

# Lists are indexed like strings
list4 = ['Python', 'Go', 2018, 4.5, [1,2.3, 'abc']]
list4[0]       # => 'Python'
list4[-1]      # => [1, 2.3, 'abc']
list4[4][1]    # => 2.3
#list4[10]    # Raises an IndexError (out of bounds index)

# A list is a mutable object and can be modified
list4[0] = 'Rust'    # =>['Rust', 'Go', 2018, 4.5, [1, 2.3, 'abc']]

# Lists are sliced like strings. Slicing returns a new list
# General syntax: list[start:stop:step]
# start is included, stop is excluded and step is by default 1
numbers = [1, 2, 3, 4, 5]
numbers[1:4]                  # => [2, 3, 4]
numbers[1:40]                 # => [2, 3, 4, 5]  -> out of bound slicing doesn't return error
numbers[:3]                   # => [1, 2, 3]  -> by default start is zero
numbers[2:]                   # => [3, 4, 5]  -> by default stop is the end of the list
numbers[::]                   # => [1, 2, 3, 4, 5]  -> returns the entire list
numbers[1:5:3]                # => [2, 5]  -> from 2 included to 5 excluded in steps of 3
numbers[4:1:-2]               # => [5, 3]
```

```
numbers[0:2] = ['a', 'b']       # => ['a', 'b', 3, 4, 5]   -> slicing modifies a list
numbers[0:2] = ['x', 'y', 'z']  # => ['x', 'y', 'z', 3, 4, 5]

l1 = [1, 2, 3]
l2 = l1             # l1 and l2 reference the same object, l2 IS NOT a copy of l1
l1 is l2            # => True
l1 == l2            # => True
l1.append(4)    # here I've modified both l1 and l2, they are still the same list
l1 is l2            # => True
l1 == l2           # => True

l3 = l1.copy()    # l2 is a copy of l1, they don't reference the same object
l1 == l2            # => True
l1 is l2            # => False
l3.remove(1)
l1 == l3            # => False
l1 is l3            # => False

l1 = [1, 2]
id(l1)              # => 139875652516360 (you'll get another value)
l1 += [3, 4]        # => [1, 2, 3, 4]  -> concatenating a new list to l1 - equivalent to using extend()
id(l1)              # => 139875652516360  -> it's the same list

l1 = l1 + [5, 6]    # => [1, 2, 3, 4, 5, 6] -> concatenating a new list to l1
id(l1)              # => 139875654318792    -> l1 is a new list
```

Iterating over a list

```
ip_list = ['192.168.0.1', '192.168.0.2', '10.0.0.1']
for ip in ip_list:
    print(f'Connecting to {ip} ...')
```

List Membership

in and not in operators test list membership

```
'10.0.0.1' in ip_list      # => returns True
'192' not in ip_list       # => returns True
'192' in ip_list          # => returns False
```

List Methods

```
dir(list)      # returns a list will all list methods

# list.clear() removes all items from list
```

```
l1 = ['a', 'b', 'c']
l1.clear()

list1 = [1, 2.2, 'abc']
len(list1 )     # => 3

# list.append() adds a single element to the end of the list
list1.append(5)          # => [1, 2.2, 'abc', 5]
# list1.append(6, 7)    # TypeError: append() takes exactly one argument (2 given)
list1.append([6, 7])     # => [1, 2.2, 'abc', 5, [6, 7]]

# list.extend() extends the list with elements of an iterable object
list1.extend([5.2])          # => [1, 2.2, 'abc', 5, [6, 7], 5.2]
#list1.extend(5.2)          # TypeError: 'float' object is not iterable
list1.extend(['x', 'y'])      # => [1, 2.2, 'abc', 5, [6, 7], 5.2, 'x', 'y']

# list.insert() Inserts an item at a given index
list1.insert(2, 'T')               # => [1, 2.2, 'T', 'abc', 5, [6, 7], 5.2, 'x', 'y']

# Insert on the last position
list1.insert(len(list1), 'Q')    # => [1, 2.2, 'T', 'abc', 5, [6, 7], 5.2, 'x', 'y', 'Q']

# list.pop() removes and returns an element of the list
list1 = [1, 2.2, 'T', 'abc', 5, [6, 7], 5.2, 'x', 'y', 'Q']
print(list1)          # => [1, 2.2, 'T', 'abc', 5, [6, 7], 5.2, 'x', 'y', 'Q']
list1.pop()           # => 'Q'
list1.pop(2)          #=> 'T'
print(list1)          # => [1, 2.2, 'abc', 5, [6, 7], 5.2, 'x', 'y']
#list1.pop(50)    # IndexError: pop index out of range

# list.remove() removes the first occurance and doesn't return an item of the list
print(list1)               #[1, 2.2, 'abc', 5, [6, 7], 5.2, 'x', 'y']
list1.remove('abc')    # => [1, 2.2, 5, [6, 7], 5.2, 'x', 'y']
#list1.remove('a')      # ValueError: list.remove(x): x not in list

# list.index() returns the index of an item
letters = list('abcabcabc')
letters.index('b')         # => 1
letters.index('b', 3)      # => 4 -> it starts from index 3
letters.index('b', 3, 6)   # => 4 -> it searches from index 3 to index 6

# list.count() returns the no. of occurrences of an item in a list
letters.count('a')    # => 3

# Sort a list
```

```
# list.sort() and sorted(list)
nums = [6, -1, 55, 2.3]
sorted(nums)    # => [-1, 2.3, 6, 55] -> returns a NEW sorted list
print(nums) # => [6, -1, 55, 2.3]

nums.sort()    # sorts the list in-place
print(nums)    # => [-1, 2.3, 6, 55]
max(nums)    # => 55
min(nums)    # => -1
sum(nums)   # => 62.3

# These methods return an error if the list is not sortable
nums.append('5.5')
#nums.sort()    TypeError: '<' not supported between instances of 'str' and 'int'
#nums.max()     AttributeError: 'list' object has no attribute 'max'

# Converting a list into a string and a string into a list
ip_list = ['192.168.0.1', '192.168.0.2', '10.0.0.1']
# str.join() returns a string from a list
ip_str = ':'.join(ip_list)       # => ip_str is equal to '192.168.0.1:192.168.0.2:10.0.0.1'

# str.split() returns a list from a string
ip_list = ip_str.split(':')    # => ip_list is equal to ['192.168.0.1', '192.168.0.2', '10.0.0.1']
```

List Comprehension

Syntax: list = [expression for item in iterable if condition]

```
s = [x for x in range(10)]
print(s)    # => [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

evens = [x for x in s if x % 2 == 0]
print(evens)    # => [0, 2, 4, 6, 8]


word_list = 'I learn Python programming!'.split()
info = [[w.upper(), w.lower(), len(w)] for w in word_list]
print(info)

# Celsius to Fahrenheit
celsius = [7.12, 10.1, 14.15, 22.5, 29.4, 32.9]
fahrenheit = [ 1.8 * x + 32 for x in celsius ]
print(fahrenheit)    # => [44.816, 50.18, 57.47, 72.5, 84.92, 91.22]
```

```
miles = [12, 10, 26, 80]
# 1 mile = 1.609 km
km = [m * 1.609 for m in miles]
print(km)    # => [19.308, 16.09, 41.834, 128.72]
```