

# Python List Operations Cheat Sheet

## Creating a List

```
my_list = [1, 2, 3]
empty_list = []
```

## Adding Elements

```
append(x): Adds x to end
    my_list.append(4)
insert(i, x): Inserts x at index i
    my_list.insert(1, 9)
extend(iterable): Appends multiple items
    my_list.extend([5,6])
+= : Same as extend
    my_list += [7,8]
```

## Removing Elements

```
remove(x): Removes first x
    my_list.remove(3)
pop([i]): Removes & returns item
    my_list.pop()
clear(): Removes all items
    my_list.clear()
```

## Accessing Elements

```
Indexing: my_list[0], my_list[-1]
Slicing: my_list[1:3], my_list[:], my_list[::-1]
```

## Searching & Counting

```
in: Checks presence
    if 2 in my_list
index(x): Returns first index of x
    my_list.index(2)
count(x): Number of times x appears
    my_list.count(2)
```

## Sorting & Reversing

```
sort(): In-place sort
    my_list.sort()
sorted(): New sorted list
    sorted(my_list)
reverse(): Reverse in-place
    my_list.reverse()
reversed(): Reverse iterator
    list(reversed(my_list))
```

# Python List Operations Cheat Sheet

## Copying a List

```
copy(): Shallow copy
    new = my_list.copy()
list(): Using constructor
    new = list(my_list)
[:]: Slice copy
    new = my_list[:]
```

## Looping

```
for item in my_list:
    print(item)
for i, item in enumerate(my_list):
    print(i, item)
```

## List Comprehension

```
[x*x for x in range(5)]
```

## Length & Other Ops

```
len(my_list): Length
[1, 2] + [3, 4]: Concatenation
[0]*5: Replication
list('abc'): Conversion to list
```

## Nested Lists

```
matrix = [[1, 2], [3, 4]]
print(matrix[0][1]) # 2
```