# Python Programming: Lists

# Lesson Objectives

*After this lesson, you will be able to…*

- Create lists in Python.
- Print out specific elements in a list.
- Perform common list operations.

# What is a List?

Variables hold one item.

```
my_color = "red"

my_peer = "Brandi"
```

**Lists** hold multiple items - and lists can hold anything.

```
# Declaring lists

colors = ["red", "yellow", "green"]

my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha"]


# Strings

colors = ["red", "yellow", "green"]


# Numbers

my_nums = [4, 7, 9, 1, 4]


# Both!
```

# Accessing Elements

**List Index** means the location of something (an *element*) in the list.

List indexes start counting at 0!

| List | “Brandi” | “Zoe” | “Steve” | “Aleksander” | “Dasha” |
|---|---|---|---|---|---|
| Index | 0 | 1 | 2 | 3 | 4 |

```
my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha"]

print(my_class[0]) # Prints "Brandi"

print(my_class[1]) # Prints "Zoe"

print(my_class[4]) # Prints "Dasha"
```

run ▶

Not sure what to do? Run some examples (start typing to dismiss)

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
```

# List Operations - Length

`len()`:

- A built in `list` operation.
- How long is the list?

```
# length_variable = len(your_list)


my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha"]

num_students = len(my_class)

print("There are", num_students, "students in the class")

# => 5
```

# Adding Elements: Append

`.append()`:

- A built in `list` operation.
- Adds to the end of the list.
- Takes any element.

```
# your_list.append(item)
```

```
my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha"]

my_class.append("Sonyl")

print(my_class)

# => ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha", "Sonyl"]
```

# Adding Elements: Insert

`.insert()`:

- A built in `list` operation.
- Adds to any point in the list
- Takes any element and an index.

```
# your_list.insert(index, item)
```

```
my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha", "Sonyl"]

my_class.insert(1, "Sanju")

print(my_class)

# => ["Brandi", "Sanju", "Zoe", "Steve", "Aleksander", "Dasha", "Sonyl"]
```

# Removing elements - Pop

`.pop()`:

- A built in `list` operation.
- Removes an item from the end of the list.

```
# your_list.pop()


my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha", "Sonyl"]
student_that_left = my_class.pop()
print("The student", student_that_left, "has left the class.")
# => "Sonyl"
print(my_class)
# => ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha"]
```

# Removing elements - Pop(index)

`.pop(index)`:

- A built in `list` operation.
- Removes an item from the list.
- Can take an index.

```
# your_list.pop(index)
```

```
my_class = ["Brandi", "Zoe", "Steve", "Aleksander", "Dasha", "Sonyl"]

student_that_left = my_class.pop(2) # Remember to count from 0!

print("The student", student_that_left, "has left the class.")

# => "Steve"

print(my_class)

# => ["Brandi", "Zoe", "Aleksander", "Dasha", "Sonyl"]
```

run ▶

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
```

run ▶

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
```

# 1. DECLARE A LIST WITH THE NAMES OF YOUR CLASSMATES

my_class = [“Brandi”, “Zoe”, “Steve”, “Aleksander”, “Dasha”, “Sonyl”]

# 2. PRINT OUT THE LENGTH OF THAT LIST

```
print(len(my_class))
```

# 3. PRINT THE 3RD NAME ON THE LIST

```
print(my_class[2])
```

# 4. DELETE THE FIRST NAME ON THE LIST

```
deleted_classmate = my_class.pop(0)
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### !! List Mutation: Warning !!

This won't work as expected - don't do this!

```
colors = ["red", "yellow", "green"]
print colors.append("blue")
#    => None
```

This will work - do this!

```
colors = ["red", "yellow", "green"]
colors.append("blue")
print colors
#    => ["red", "yellow", "green", "blue"]
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

**Quick Review: Basic List Operations**

```
# List Creation
my_list = ["red", 7, "yellow", 1]

# List Length
list_length = len(my_list) # 4

# List Index
print(my_list[0]) # red

# List Append
my_list.append("Yi") # ["red", 7, "yellow", 1, "Yi"]

# List Insert at Index
my_list.insert(1, "Sanju") # ["red", "Sanju", 7, "yellow", 1, "Yi"]

# List Delete
student_that_left = my_list.pop() # "Yi"; ["red", "Sanju", 7, "yellow", 1]

# List Delete at Index
student_that_left = my_list.pop(2) # 7; ["red", "Sanju", "yellow", 1]
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### Numerical List Operations - Sum

Some actions can only be performed on lists with numbers.

`sum()`:

- A built in `list` operation.
- Adds the list together.
- Only works on lists with numbers!

```
# sum(your_numeric_list)


team_batting_avgs = [.328, .299, .208, .301, .275, .226, .253, .232, .287]
sum_avgs = sum(team_batting_avgs)
print("The total of all the batting averages is", sum_avgs)
# => 2.409
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### List Operations - Max/Min

`max()` or `min()`:

- Built in `list` operations.
- Finds highest, or lowest, in the list.

```
# max(your_numeric_list)
# min(your_numeric_list)

team_batting_avgs = [.328, .299, .208, .301, .275, .226, .253, .232, .287]
print("The highest batting average is", max(team_batting_avgs))
# => 0.328
print("The lowest batting average is", min(team_batting_avgs))
# => 0.208
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### You Do: Lists

On your local computer, create a `.py` file named `list_practice.py`. In it:

1. Save a list with the numbers `2`, `4`, `6`, and `8` into a variable called `numbers`.
2. Print the max of `numbers`.
3. Pop the last element in `numbers` off; re-insert it at index `2`.
4. Pop the second number in `numbers` off.
5. Append `3` to `numbers`.
6. Print out the average number (divide the sum of `numbers` by the length).
7. Print `numbers`.

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### Summary and Q&A

We accomplished quite a bit!

```
# List Creation
my_list = ["red", 7, "yellow", 1]
# List Length
list_length = len(my_list) # 4
# List Index
print(my_list[0]) # red
# List Append
my_list.append("Yi") # ["red", 7, "yellow", 1, "Yi"]
# List Insert at Index
my_list.insert(1, "Sanju") # ["red", "Sanju", 7, "yellow", 1, "Yi"]
# List Delete
student_that_left = my_list.pop() # "Yi"; ["red", "Sanju", 7, "yellow", 1]
# List Delete at Index
student_that_left = my_list.pop(2) # 7; ["red", "Sanju", "yellow", 1]
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### Summary and Q&A

And for numerical lists only…

```
# Sum all numbers in list
sum_avgs = sum(team_batting_avgs)
# Find minimum value of list
min(team_batting_avgs)
# Find maximum value of list
max(team_batting_avgs)
```

# 5. RE-ADD THE NAME YOU DELETED TO THE END OF THE LIST

my_class.append(deleted_classmate)

print(my_class)

### Additional Resources

- Python Lists - Khan Academy Video
- Google For Education: Python Lists
- Python-Lists