

## Adding to Lists in Python

### Textbook

## Adding to Lists in Python



### Add to a List

There are several ways to add to a list in python. One of these methods is `append()`.

```
1 goats = ["Billy", "Frannie", "Leslie", "Barbara", "Scott"]
2
3 goats.append("Molly")
4
5 print(goats)
```

Try it!

This adds the value of `"Molly"` to the end of the list.

## Combining Lists

You can combine lists using the method `extend()`.

```
1 goats = ["Billy", "Frannie", "Leslie", "Barbara", "Scott"]
2
3 goats.extend(["Janie", "Boulder", "Penelope", "Frank"])
4
5 print(goats)
```

Try it!

This adds the new list to the end of the original list.

See what happens when you extend just a string:

```
1 goats = ["Billy", "Frannie", "Leslie", "Barbara", "Scott"]
2
3 goats.extend("Curly")
4
5 print(goats)
```

Try it!

## Adding an Item to a List in a Specific Place

Sometimes you might want to add an item to a list in a specific place. This is done with `insert()`.

```
1 goats = ["Billy", "Frannie", "Leslie", "Barbara", "Scott"]
2
3 goats.insert(2, "Curly")
4
5 print(goats)
```

Try it!

The first number in the parenthesis is the index value where you would like to insert the value. In this case, it will insert `"Curly"` at index value `2`.

## Challenge Help

To create a list of strings, use the following code:

```
my_list = input().split()
```

Remember that inputs are automatically accepted as strings.

The input should look like words separated by spaces. No commas, parentheses, or brackets are needed. For example: `apple banana strawberry` would be an acceptable input.

This would create a python list: `["apple", "banana", "strawberry"]`

These challenges may require putting a bunch of numbers as input into a list. To do this, use this code:

```
my_list = [int(n) for n in input().split()]
```

This code creates a list called `my_list` and the `input().split()` command breaks up the input into each individual integer. It uses a for loop to assign all the individual inputs to their index in the list.

For this code, put your input question right into the `input()` above. For example:

```
my_list = [int(n) for n in input("Input a list of numbers").split()]
```

The input should look like numbers separated by spaces. No commas, parenthesis, or brackets are needed. For example: `2 6 8 33 24 2` would be an acceptable input.

This would create a python list: `[2, 6, 8, 33, 24, 2]`

## Variable Aliasing

Aliasing occurs when two or more different variables (or names) in a program refer to the same underlying memory location. This means that if you modify the value using one variable, the change will be reflected when you access the value using any of the other aliased variables, because they are all looking at the same data. Think of it like having two different nicknames for the same person; if you talk to "John" and tell him something, "Johnny" (the nickname) also knows it because they are the same individual.

## Why Same Value Alone Doesn't Create Aliasing

It's crucial to understand that simply giving multiple variables the **same value does not automatically create aliasing**. Let's look at an example:

```
1 a = 10
2 b = 10
3
```

In this case, `a` and `b` both hold the value `10`. However, they are typically stored in **separate memory locations**. If you were to change `a` to `20`, `b` would still be `10`. They are independent variables that just happen to have the same content. They are like two separate books that both contain the same story. If you edit one book, the other remains unchanged.

## How to Achieve Aliasing

To achieve aliasing, you need to make the variables **refer to the same memory location**. The exact way to do this depends on the programming language.

### Example in Python

In Python, when you assign one variable to another, you are often creating an alias, especially for mutable objects (like lists or dictionaries). For immutable objects (like numbers or strings), reassigning a variable creates a new object rather than modifying the existing one in place, so aliasing with immutable types generally refers to both variables pointing to the *same* immutable object in memory, but if you reassign one, it just points to a *new* immutable object.

```
1 list1 = [1, 2, 3]
2 list2 = list1 # list2 now aliases list1
3
```

Here, `list1` and `list2` both point to the **exact same list object** in memory. If you modify `list2`:

```
1 list2.append(4)
2 print(list1) # Output: [1, 2, 3, 4]
3
```

You'll see that `list1` also reflects the change because `list1` and `list2` are aliases pointing to the same data.

## Checkpoint

---

### Adding to Lists in Python

Consider the following list:

```
flowers = ["rose", "tulip", "lilac", "sunflower"]
```

Use what you learned in the lesson to do the following:

1. **Append** an item to the end of your list.
2. **Extend** your list with a list that has 2 items in it.
3. **Insert** an item into a middle section of your list.

### Requirements:

- Append to your list.
- Extend your list with another list
- Insert into a middle section of your list.

## Questions (12)

---

**1. Which of the following options could be used to add one or more items to a list in Python? Select all that apply.**

SELECT MULTIPLE

**Select all that apply:**

- A. `extend()`
- B. `insert()`
- C. `append()`
- D. `change()`

## 2. Which method can join two lists together?

MULTIPLE CHOICE

Choose the correct answer:

- A. pop()
- B. append()
- C. extend()
- D. insert()

## 3. Edit the text box below to debug (fix) the code:

DEBUG CODE

Code to Debug:

```
1 fish.insert("Bubbles", 2)
```

## 4. What will the following code print out?

MULTIPLE CHOICE

```
chickens = ["Clucky", "Barbara", "Nancy", "Fran", "Rhode"] chickens.append("Daphne") print(chickens)
```

Choose the correct answer:

- A. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Daphne']
- B. [ 'Daphne', 'Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode']
- C. chickens
- D. [ 'Clucky', 'Barbara', 'Daphne', 'Nancy', 'Fran', 'Rhode']

## 5. When you append an item to a list, where will it go?

MULTIPLE CHOICE

Choose the correct answer:

- A. At the beginning of the list.
- B. In the middle of the list
- C. At the end of the list.
- D. Randomly

## 6. What happens when you extend a string?

Choose the correct answer:

- A. It adds space on either side of the string.
- B. It attaches it to the last declared string.
- C. It separates the characters of the string and puts them in a list.
- D. It creates a copy of the string.

## 7. What will the following code print out?

```
chickens = ["Clucky", "Barbara", "Nancy", "Fran", "Rhode"] chickens.insert(2, "Daphne") print(chickens)
```

Choose the correct answer:

- A. ['Clucky', 'Barbara', 'Daphne', 'Nancy', 'Fran', 'Rhode']
- B. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Daphne']
- C. ['Daphne', 'Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode']
- D. ['Clucky', 'Daphne', 'Barbara', 'Nancy', 'Fran', 'Rhode']

## 8. What will the following code print out?

```
chickens = ["Clucky", "Fran", "Rhode"] chickens.extend("Fran") print(chickens)
```

Choose the correct answer:

- A. ['Clucky', 'Fran', 'Rhode', 'F', 'r', 'a', 'n']
- B. ['C', 'l', 'u', 'c', 'k', 'y', 'F', 'r', 'a', 'n', 'R', 'h', 'o', 'd', 'e']
- C. ['Clucky', 'Rhode', 'F', 'r', 'a', 'n']
- D. ['Clucky', 'F', 'r', 'a', 'n'] 'Rhode'

## 9. What will the following code print out?

```
chickens = ["Clucky", "Barbara", "Nancy", "Fran", "Rhode"] chickens.extend(["Charlotte", "Crystal"]) print(chickens)
```

Choose the correct answer:

- A. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Charlotte', 'Crystal']
- B. ['Charlotte', 'Crystal', 'Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode']
- C. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Charlotte', 'C', 'r', 'y', 's', 't', 'a', 'l']
- D. ['Charlotte', 'Crystal']

**10. Edit the text box below to debug (fix) the code:**

DEBUG CODE

**Code to Debug:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid", "wavy"]
2
3 balloons.append("happy birthday")
4
5 print(balloons)
```

**11. Edit the text box below to debug (fix) the code:**

DEBUG CODE

**Code to Debug:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid"]
2
3 balloons.extend(["happy birthday", "wavy", "checkers"])
4
5 print(balloons)
```

**12. Debug the following code: Assume you are trying to put "polka dots" at the beginning of the list.**

DEBUG CODE

**Code to Debug:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid"]
2
3 balloons.insert(1, "polka dots")
4
5 print(balloons)
```



## Challenges (5)

### 1. Last Minute Treats

You are throwing an awesome pool party! You know that the best part of any party is the food, so you made a list of all the things you want to get for your pool party. On your way to the store, you remembered two more things and want to add them to your list.

Consider the following list:

```
treats = ["popcorn", "popsicles", "soda", "chips", "cookies"]
```

1. Create a program that takes in **two inputs**.
2. The inputs will be other treats you'd like to add to the list.
3. Print out the list with the **two inputs added to it at the end**.

For example:

Inputs: `juice` , `crackers`

Output: `['popcorn', 'popsicles', 'soda', 'chips', 'cookies', 'juice', 'crackers']`

Another example:

Inputs: `cake` , `smoothies`

Output: `['popcorn', 'popsicles', 'soda', 'chips', 'cookies', 'cake', 'smoothies']`

### 2. Olympic Events

The Olympic games bring together countries from all over the world.

1. Create a **list of events** that happen at the Olympic games.
2. Then, **create another list** of new events that have been added recently to the Olympic games.
3. Attach the list of new events to the original list of Olympic games using `extend()` .
4. Print the final list.

#### Requirements:

- Combine the list of new events to the original list of Olympic events.
- Print the final list.



### 3. Spelling Bee

Have you ever been in a spelling bee? Often, the participant needs to say the word, say how it's spelled, and then say the word again.

Create a program that will do this for you!

The program will take an **input of one word**. It will then print out a **list of the word**, how the word is **spelled**, and the **word again**.

For example:

Input: `handkerchief`

Output: `['handkerchief', 'h', 'a', 'n', 'd', 'k', 'e', 'r', 'c', 'h', 'i', 'e', 'f', 'handkerchief']`

Another example:

Input: `storm`

Output: `['storm', 's', 't', 'o', 'r', 'm', 'storm']`

Hint: Try using `append`, `extend`, then `append` again.

### 4. More Siblings!

Your parents decided to have a new foster kid join your family! His name is Mitch and you aren't yet sure how old he is.

There are **4 kids** in the family already. **Amanda is 15. Levi is 13. Nicole is 10. Lilly is 6.**

Consider the following list of kids in order from oldest to youngest:

```
family = ["Amanda", "Levi", "Nicole", "Lilly"]
```

Depending on age, where would the new kid fall on the list?

Create a program that will put `Mitch` into the family where he goes according to age.

For example:

Input: `11`

Output: `["Amanda", "Levi", "Mitch", "Nicole", "Lilly"]`

Another example:

Input: `3`

Output: `['Amanda', 'Levi', 'Nicole', 'Lilly', 'Mitch']`

If Mitch and a family member are the same age, place Mitch before the sibling.

## 5. Donation

You're way excited because your favorite store is doing a promotion right now. Every time you spend **\$5 or more**, they will **donate 10 percent** of your order to a local charity.

1. Create a program that will keep track of how much donation money your purchases have generated.
2. Create an input that will generate a list of numbers.
3. For every number in that list that's **5 or more**, add **10 percent** of that purchase to a new list.
4. Make sure to **round each item to one decimal point**.
5. Print the new list.

For example:

Input: 2 2 5 9 7 3 10

Output: [0.5, 0.9, 0.7, 1.0]

Another example:

Input: 1 3 3 4 5 6 7 8 9 10

Output: [0.5, 0.6, 0.7, 0.8, 0.9, 1.0]

**Hint:** Create a variable and assign it to an empty list. Then you can add things into that empty list.

## Answer Keys & Solutions

### Checkpoint Solutions

#### Adding to Lists in Python

```
1 flowers = ["rose", "tulip", "lilac", "sunflower"]
2
3 flowers.append("daffodil")
4
5 print(flowers)
6
7 flowers.extend(["iris", "buttercup"])
8
9 print(flowers)
10
11 flowers.insert(3, "violet")
12
13 print(flowers)
```

### Questions

1. Which of the following options could be used to add one or more items to a list in Python? Select all that apply.

SELECT MULTIPLE

Correct Answers:

- |             |             |
|-------------|-------------|
| A. extend() | ✓ Correct   |
| B. insert() | ✓ Correct   |
| C. append() | ✓ Correct   |
| D. change() | ✗ Incorrect |

#### Explanation:

extend() adds lists together insert() adds into the list append() adds to the end of the list

2. Which method can join two lists together?

MULTIPLE CHOICE

Correct Answer:

A. pop()

✗ Incorrect

B. append()

✗ Incorrect

C. extend()

✓ Correct

D. insert()

✗ Incorrect

#### Explanation:

pop() pulls the last item off the list insert() adds into the list append() adds to the end of the list

### 3. Edit the text box below to debug (fix) the code:

DEBUG CODE

#### Incorrect Code:

```
1 fish.insert("Bubbles", 2)
```

#### Correct Solution:

```
1 fish.insert(2, "Bubbles")
```

#### Explanation:

The number comes first inside the parentheses

### 4. What will the following code print out?

MULTIPLE CHOICE

#### Correct Answer:

A. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Daphne']

✓ Correct

B. ['Daphne', 'Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode']

✗ Incorrect

C. chickens

✗ Incorrect

D. ['Clucky', 'Barbara', 'Daphne', 'Nancy', 'Fran', 'Rhode']

✗ Incorrect

#### Explanation:

.append adds the value to the end.

### 5. When you append an item to a list, where will it go?

MULTIPLE CHOICE

#### Correct Answer:

A. At the beginning of the list.

✗ Incorrect

B. In the middle of the list

✗ Incorrect

C. At the end of the list.

✓ Correct

D. Randomly

✗ Incorrect

#### Explanation:

`.append` adds the value to the end of the list.

## 6. What happens when you extend a string?

MULTIPLE CHOICE

### Correct Answer:

A. It adds space on either side of the string.

✗ Incorrect

B. It attaches it to the last declared string.

✗ Incorrect

C. It separates the characters of the string and puts them in a list.

✓ Correct

D. It creates a copy of the string.

✗ Incorrect

#### Explanation:

the string "winter" would look like this: `["w", "i", "n", "t", "e", "r"]`

## 7. What will the following code print out?

MULTIPLE CHOICE

### Correct Answer:

A. `['Clucky', 'Barbara', 'Daphne', 'Nancy', 'Fran', 'Rhode']`

✓ Correct

B. `['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Daphne']`

✗ Incorrect

C. `[Daphne', 'Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode']`

✗ Incorrect

D. `['Clucky', 'Daphne', 'Barbara', 'Nancy', 'Fran', 'Rhode']`

✗ Incorrect

#### Explanation:

This will insert the value of "Daphne" at index value of 2

## 8. What will the following code print out?

Correct Answer:

- |  |             |
|--|-------------|
| A. ['Clucky', 'Fran', 'Rhode', 'F', 'r', 'a', 'n']                             | ✓ Correct   |
| B. ['C', 'l', 'u', 'c', 'k', 'y', 'F', 'r', 'a', 'n', 'R', 'h', 'o', 'd', 'e'] | ✗ Incorrect |
| C. ['Clucky', 'Rhode', 'F', 'r', 'a', 'n']                                     | ✗ Incorrect |
| D. ['Clucky', 'F', 'r', 'a', 'n'] 'Rhode'                                      | ✗ Incorrect |

## 9. What will the following code print out?

Correct Answer:

- |  |             |
|--|-------------|
| A. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Charlotte', 'Crystal']                         | ✓ Correct   |
| B. ['Charlotte', 'Crystal', 'Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode']                         | ✗ Incorrect |
| C. ['Clucky', 'Barbara', 'Nancy', 'Fran', 'Rhode', 'Charlotte', 'C', 'r', 'y', 's', 't', 'a', 'l'] | ✗ Incorrect |
| D. ['Charlotte', 'Crystal']  | ✗ Incorrect |

### Explanation:

Extending a list adds the values to the end of another list.

## 10. Edit the text box below to debug (fix) the code:

Incorrect Code:

```
1 balloons = ["stripes", "spots", "zig zags", "solid", "wavy"]
2
3 balloons.append("happy birthday")
4
5 print(baloons)
```

Correct Solution:

```
1 balloons = ["stripes", "spots", "zig zags", "solid", "wavy"]
2
3 balloons.append("happy birthday")
4
5 print(balloons)
```

**Explanation:**

Check the spelling

**11. Edit the text box below to debug (fix) the code:**[DEBUG CODE](#)**Incorrect Code:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid"]
2
3 balloons.extend(["happy birthday", "wavy", "checkers"])
4
5 print(balloons)
```

**Correct Solution:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid"]
2
3 balloons.extend(["happy birthday", "wavy", "checkers"])
4
5 print(balloons)
```

**Explanation:**

There is a missing square bracket

**12. Debug the following code: Assume you are trying to put "polka dots" at the beginning of the list.**[DEBUG CODE](#)**Incorrect Code:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid"]
2
3 balloons.insert(1, "polka dots")
4
5 print(balloons)
```

**Correct Solution:**

```
1 balloons = ["stripes", "spots", "zig zags", "solid"]
2
3 balloons.insert(0, "polka dots")
4
5 print(balloons)
```

**Explanation:**



Remember that indexing begins by counting at 0

## Challenges

### 1. Last Minute Treats

Solution:

```
1 first = input("What other item do you want?")
2 second = input("What final item do you want?")
3
4 treats = ["popcorn", "popsicles", "soda", "chips", "cookies"]
5
6 treats.append(first)
7 treats.append(second)
8
9 print(treats)
```

### 2. Olympic Events

Solution:

```
1 olympics = ["running", "gymnastics", "swimming", "volleyball", "basketball"]
2
3 new_olympics = ["karate", "surfing", "baseball", "skateboarding", "sport climbing"]
4
5 olympics.extend(new_olympics)
6
7 print(olympics)
```

### 3. Spelling Bee

Solution:

```
1 spelling = []
2
3 word = input("What word do you want spelled out?")
4
5 spelling.append(word)
6
7 spelling.extend(word)
8
9 spelling.append(word)
10
11 print(spelling)
```

## 4. More Siblings!

Solution:

```
1 mitch = int(input("How old is Mitch?"))
2
3 family = ["Amanda", "Levi", "Nicole", "Lilly"]
4
5 if mitch >= 15:
6     family.insert(0, "Mitch")
7 elif 15 > mitch >= 13:
8     family.insert(1, "Mitch")
9 elif 13 > mitch >= 10:
10    family.insert(2, "Mitch")
11 elif 10 > mitch >= 6:
12    family.insert(3, "Mitch")
13 else:
14    family.insert(4, "Mitch")
15
16 print(family)
```

## 5. Donation

Solution:

```
1 my_list = [int(n) for n in input("Input a list of numbers").split()]
2
3 cash_back = []
4
5 for x in my_list:
6     if x >= 5:
7         x = x * .1
8         cash_back.append(round(x,1))
9
10 print(cash_back)
```