

Python- List

Q1. Create an empty list.

You can create an empty list by using empty square brackets or using the list() function without any values. Create an empty list by using the list() function without arguments.

Q2. Create a list and add an element to the end of this list.

If we want to add an element at the end of a list, we should use append . It is faster and direct. If we want to add an element somewhere within a list, we should use insert

Q3. For the above created list print the last element.

To get the last element of the list using the naive method in [Python](#). There can be 2-naive methods to get the last element of the list. Iterating the whole list and getting, the second last element and reversing the list, and printing the first element.

Q4. Reverse the above created list.

Using reversed() we can reverse the list and a list_reverseiterator object is created, from which we can create a list using list() type casting. Or, we can also use the list reverse() function to reverse list in place.

Q5. Sort the above created list.

The sorted() function will create a new sorted list with list elements sorted in the desired order.

Q6. Create a list of your top three favorite movies, then print the second movie title.

1. Implement the Analyzer class so that its solver method returns an array of up to the top three movie titles that have been set as the favorite ones by friends of a given user. You can assume that the function input structures are properly populated.
2. If the solver method doesn't return top three movies, it should return the best N number of movies up to 3, otherwise, it should return an empty list.
3. Movies of the same count should be ordered alphabetically.

Q7. Create a list of your favorite animals, then add a new animal to the list and print the updated list.

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# In[1]:
```

```
# 4-2. Animals: Think of at least three different animals that have a common characteristic.
```

```
# Store the names of these animals in a list, and then use a for loop to
```

```
# print out the name of each animal.
```

```
# • Modify your program to print a statement about each animal, such as
```

```
# A dog would make a great pet.
```

```
# • Add a line at the end of your program stating what these animals have in
```

```
# common. You could print a sentence such as Any of these animals would
```

```
# make a great pet!
```

```
animals = ["cat", "dog", "hamster"]
```

```
for animal in animals:
```

```
    print(animal + " will make a great pet!")
```

Q8. Create a list of your favorite cities, then use the index() method to find the position of a specific city on the list and print it.

```
# list of items
```

```
list2 = ['dubai', 'colombo', 'mumbai', 'kandy', 'pune']
```

```
# Will print the index of 'kandy' in list2
```

```
print(list2.index('kandy'))
```