Lists

Table of contents

Lists are a super important type of data structure in Python. The below problems test your understanding of some list principles (the sub-bullets describe which ones are relevant to each problem), and we hope you find them useful!

Get first element

Fill out the function `get_first_element(lst)` which takes in a list `lst` as a parameter and prints the first element in the list. The list is guaranteed to be non-empty.

You can change the items in the `SAMPLE_LIST` list to test your code, but we've also included some DocTests which you can run by pressing the Check button in the bottom right corner!

Get last element

Fill out the function `get_last_element(lst)` which takes in a list `lst` as a parameter and prints the last element in the list. The list is guaranteed to be non-empty, but there are no guarantees on its length.

Get list

Write a program in `get_list.py` which continuously asks the user to enter values which are added one by one into a list. When the user presses enter without typing anything, print the list.

Here's a sample run (user input in bold italics):

```
$ python get_list.py
Enter a value: 1
Enter a value: 2
Enter a value: 3
Enter a value:
Here's the list:
['1', '2', '3']
```

Shorten

Fill out the function `shorten(lst, max_len)` which removes elements *from the end of* `lst`, which is a list, and prints each item it removes until `lst` is `max_len` items long. If `lst` is already shorter than `max_len` you should leave it unchanged. We've written a `main()` function for you which passes the `SAMPLE_LIST` and `MAX_LENGTH` constants into your function once you run the program; you can change the values of these constants to play around with different inputs!

Count even

Fill out the function `count_even(lst)` which takes in a list of integers and prints the number of even numbers in the list. We've added some DocTests you can use to check your code, just press the Check button in the bottom right corner. We've also written a `main()` function for you which inputs the constant `SAMPLE_LIST`; you can change this to whatever input you'd like, or add more DocTests!

Where's Karel?

Fill in the function `find_karel(roster)` which takes in a list, `roster`, and prints "Karel is here!" if 'karel' is an element in `roster`, or prints "Karel isn't here." if 'karel' isn't an element in `roster`. You can assume all elements in `roster` will be lowercase.

We've written some DocTests for you (press the Check button to run them, and note that DocTests are very picky about output -- make sure you're printing exactly "Karel is here!" or "Karel isn't here.") and also written a `main()` function which passes the `SAMPLE_ROSTER` constant into `find_karel(roster)` . You can change `SAMPLE_ROSTER` or add DocTests to test your code with different inputs.