lists_exercises_pdf

January 31, 2019

1 An Introduction to Python

1.1 Lists

1.1.1 This notebook contains the programming exercises for An Introduction to Python: Lists.

1.1.2 This is the PDF version of the Jupyter Notebook, provided only for convenience. It is recommended that you download the Jupyter Notebook(.ipynb) and interactively code your answers.

References: - Fundamentals of Python Programming. Halterman, Richard. Southern Adventist Univesity. 2018.

2 Exercises

2.1 Do the following problems.

2.1.1 Jupyter Notebook Shortcuts:

There are two modes when a cell is highlighted.

Command Mode: Press ESC to activate. The cell has a blue border if this mode is active. In this mode, you can add, delete, create, copy and paste cells.

- create a new cell above the current cell: a
- create a new cell below the current cell: b
- delete the current cell: dd
- change the current cell's type to "Code": y
- change the current cell's type to "Markdown": m

Edit Mode: Press ENTER to activate. The cell has a green border if this mode is active. In this mode, you can edit and type text into the cell.

- execute the current cell and create a new cell: SHIFT + ENTER

Create a list of 5 string objects representing different types of foods. Use a for loop to iterate over the list and print the objects separated by spaces.

Use the list created in the previous problem and demonstrate the use of append, remove and pop by adding to, removing from and popping elements from the list.

Given the list `lst = [20, 1, -34, 40, -8, 60, 1, 3]` evaluate the following expressions. Write your answers in markdown first then use code to verify your answer.

- lst[0:3]
- lst[4:8]
- lst[-5:-3]
- lst[4:]
- lst[:]
- lst[:4]
- lst[1:5]
- -34 in lst
- -34 not in lst
- len(lst)

Create the following list: [1,2,3,4,5,6,7,8]. Use slicing to obtain the following sublists:

- [2,3,4,5]
- [1,3,5,7]
- [1,2,3,4,5,6] (use negative indices for this problem)
- [1,2,3,4,5,6,7,8]
- [8,7,6,5,4,3,2,1]
- [6,5,4,3]

Create an empty list.

1) Then use a for loop to append to the list the multiples of 3 from 12 to 2340. Print out the length of this list.

2) Redo part 1) of this problem but use list comprehensions instead of a for loop.

3) Use list comprehension as in part 2) but add in the additional condition that the numbers need to be a multiple of 7.

Use the built-in function `list()` and the `range()` function to create a list of even numbers from 2 to 50.

Use `sum` to find the sum of a list created in the previous problem.

Write the list represented by each of the following list comprehension expressions. Write your answers in markdown then check with code.

- [x + 1 for x in [2, 4, 6, 8]]
- [10*x for x in range(5, 10)]
- [x for x in range(10, 21) if x % 3 == 0]

Suppose lst = [-2,4,6,-4,-5,12,-7,5,3]. Use list comprehension to extract the positive numbers from the list.

Write a segment of code that asks the user to input a positive integer `n`. Use list comprehension to create a list `factors` which is a list of factors of `n`.

Without running the following code, can you tell what it does for a given integer `n`? (Pretty cool!)

```
[x for x in range(2, n) if n % x == 0] == []
```

Create a list where each element of the list is another list of a student name(string) and a gpa(float). Your list should have 4 student-gpa pairs. Then use a nested loop to print out each student name and gpa, each on a different line. One sample output:

```
Mike 2.3
John 3.5
Michele 3.7
Carol 3.2
```