

2 Sequences: Lists and Tuples

Note: Throughout the Instructor Solutions Manual, solutions are *not provided* for project, research and challenge exercises, many of which are substantial and appropriate for term projects, directed-study projects, capstone-course projects and thesis topics. **Before assigning a particular exercise for homework, instructors should check the IRC to be sure the solution is available.** These Instructor Solutions Manual PDFs contain **only answers to short-answer exercises and any discussion questions asked in other exercises.** Code corresponding to programming exercises can be found in the `solutions` folder of chapter-specific subfolders, e.g., `ch01` for Chapter 1, `ch02` for Chapter 2, etc. Code generally is provided both in **Python source-code files** (`.py`) and **Jupyter Notebooks** (`.ipynb`).

Exercises

Use IPython sessions for each exercise where practical.

5.1 (*What's Wrong with This Code?*) What, if anything, is wrong with each of the following code segments?

a) `day, high_temperature = ('Monday', 87, 65)`

Answer: There are too few variables to unpack the tuple. The statement should be:

`day, high_temperature, low_temperature = ('Monday', 87, 65)`

b) `numbers = [1, 2, 3, 4, 5]`
`numbers[10]`

Answer: The index 10 is not within the list bounds. In this case, valid indices are in the range 0-4 because the list contains five items.

c) `name = 'amanda'`
`name[0] = 'A'`

Answer: Strings are immutable so you cannot assign `'A'` to the elements in a string sequence.

d) `numbers = [1, 2, 3, 4, 5]`
`numbers[3.4]`

Answer: Indices must be integer values or slices.

e) `student_tuple = ('Amanda', 'Blue', [98, 75, 87])`
`student_tuple[0] = 'Ariana'`

Answer: Tuples are immutable. Once a tuple is created, you cannot assign new elements to it.

f) `('Monday', 87, 65) + 'Tuesday'`

Answer: The `+` operator works only with sequences of the same type, two lists, two tuples or two strings.

g) `'A' += ('B', 'C')`

Answer: The `+=` operator must have a variable as its left operand, not a string literal.

h) `x = 7`
`del x`
`print(x)`

Answer: Once you delete the variable `x`, it no longer exists, so you cannot print it. The `print` statement must appear before the `del` statement.

i) `numbers = [1, 2, 3, 4, 5]`
`numbers.index(10)`