Physics Without Frontiers

ICTP The Abdus Salam International Centre for Theoretical Physics

Practical Programming
in Python

Inspired by 'Practical Programming' by Paul Gries, Jennifer Campbell, Jason Montojo

Lecture 4: Summary & Exercises
Working with Text

Strings, String Operations, Printing Information, Reading from the Keyboard

> *"Nothing is so obvious that it's obvious… The use of the word "obvious" indicates the absence of a logical argument."*
>
> – Errol Morris

Lecture 4: Summary

In this lecture you learned the following:

- Python uses type `str` to represent text as sequences of characters.
- Strings are created by placing pairs of single or double quotes around the text. Multiline strings can be created using matching pairs of triple quotes.
- Special characters like newline and tab are represented using escape sequences that begin with a backslash.

Escape Sequences

Escape Sequence	Description
`\'`	single quote
`\"`	double quote
`\\`	backslash
`\t`	tab
`\n`	newline
`\r`	carriage return

- Values can be printed using built-in function `print`, and input can be provided by the user using built-in function `input`.

Lecture 4: Exercises

When writing code, only use Python concepts that have been introduced in the lectures already.

Exercise 1:

What value does each of the following expressions evaluate to? Verify your answers by typing the expressions into the Python shell.

a. `'Computer' + 'Science'`

b. `'Darwin\'s'`

c. `'H2O' * 3`

d. `'CO2' * 0`

Exercise 2:

Express each of the following phrases as Python strings using the appropriate type of quotation marks (single, double, or triple) and, if necessary, escape sequences. There is more than one correct answer for each of these phrases.

a. They’ll hibernate during the winter.

b. "Absolutely not," he said.

c. "He said, ’Absolutely not,’" recalled Mel.

d. hydrogen sulfide

e. left\right

Exercise 3:

Rewrite the following string using single or double quotes instead of triple quotes:

```
"""A
B
C"""
```

Exercise 4:

Use built-in function `len` to find the length of the empty string.

Exercise 5:

Given variables `x` and `y`, which refer to values `3` and `12.5`, respectively, use function `print` to print the following messages. When numbers appear in the messages, variables `x` and `y` should be used.

a. The rabbit is 3.

b. The rabbit is 3 years old.

c. 12.5 is average.

d. 12.5 * 3

e. 12.5 * 3 is 37.5

Exercise 6:

Consider this code:

```
>>> first = 'John'
>>> last = 'Doe'
>>> print(last + ', ' + first)
```

What is printed by the code above?

Exercise 7:

Use `input` to prompt the user for a number, store the number entered as a `float` in a variable named `num`, and then print the contents of `num`.

Exercise 8:

Complete the examples in the docstring and then write the body of the following function:

```
def repeat(s, n):
    """
    Return s repeated n times.

    Return the string s repeated n times. If n is
    negative, return the empty string.

    Examples:

        >>> repeat('yes', 4)
        'yesyesyesyes'
        >>> repeat('no', 0)

        >>> repeat('no', -3)

        >>> repeat('yesnomaybe', 2)

    """
```

Exercise 9:

Complete the examples in the docstring and then write the body of the following function:

```
def total_length(s1, s2):
    """
    Return the sum of the lengths of s1 and s2.

    Examples:

        >>> total_length('yes', 'no')
        5
        >>> total_length('yes', '')

        >>> total_length('YES!!!!', 'Noooooo')

    """
```