TMTplus Introduction to Scientific Programming

Mahdi Farnaghi, Mahdi Khodadadzadeh & Robert Ohuru

March 2021

Answers 2

Functions

30

Ex 2.1

Ex 2.3

The script assigns a text string to the variable `x`. Then it creates variable `n` with the calculated number of characters from string `x` using function `len()`, converted into a string using function `str()`. At last, it prints the variable `n` and the string `" - "` and also the variable `x`.

Ex 2.4

Syntax (syntactic) errors are invalid code that the compiler doesn't understand. More specifically, they are violations against the rules of the programming language. For example incorrect indentation can confuse the interpreter, and it will report about this.

In contrast, *semantic* errors are typically valid code, in the sense that they have correct syntax, however, the program when run does not do what it was intended to do. (It will typically do nothing useful, or present results that are incorrect.)

Ex 2.7

```
def print_with_brackets(name):
    print ("[[[␣", name , "␣]]]␣")

def print_three_names(name1,name2,name3):
    printWithBrackets(name1)
    printWithBrackets(name2)
    printWithBrackets(name3)
```

Ex 2.8

The + operator does string concatenation and not the sum, this operator only does the sum if we use numbers only. The * operator does multiplication and works also with strings. Another observation is that in the print statement each argument must have its own data type. I cannot mix data types for example 2

does not work because we are concatenating a string with a number but 3 works fine.

Ex 2.9

```
import math

def area_circle(radius):
'''This function calculates the area of a circle based on a
   given radius'''

    # define a function within the function area_circle
    def squared(x):
        return x**2
    return squared(radius)*math.pi

#print the area of circle with radius 5
print(area_circle(5))

# print the docstring without the comment
print(area_circle.__doc__)
```