Meeting 7 Exercises

Info 206

14 September 2017

Write individual scripts to solve the following exercises.

1. Create two functions (script name arithmetic_[lastname].py).

a. Write the functions (3 points)

- The first should be called "multiply". It should take in a list as a parameter, then output the product of all numbers in that list.
- The second should be called "addify". It should take in a list as a parameter, then output the sum of all numbers in that list.

Use a "doc string" to create help documentation for your function.

Sample Output:

```
print(multiply([1,3,5,8]))
print(addify([1,3,5,8]))
120
17
```

```
def multiply(ListNum):
    return ans

def addify(ListNum):
    return ans
```

Return the following in your script.

```
print(multiply([1,3,5,8]))
print(addify([1,3,5,8]))
```

Also, this should print in your script and return a useful message:

```
help(multiply)
```

b. Add a default parameter to your function, set to "None". Update your functions so that you are given a custom message if the user forgets to enter an argument. (1 points)

```
print(multiply())
print(addify())
Please provide a list to multiply:
Please provide a list to add:
```

c. [extra credit] Now for a fun trick. As you learned in the course videas, functions are really just objects! Prove this to yourself by printing the type of your functions. (1 point)

```
print(type(multiply))
print(type(addify))
<class 'function'>
<class 'function'>
```

---

2. Nested ("Wrapped") Functions (submit your script as wrapped_[lastname].py)

a. Write a function, `sum_digits`, which takes an `int` and returns the sum of its (positive value) digits. (2 points)

b. Write another function that "wraps" `sum_digits` in that it calls `sum_digits` from within it. Call this function `diff_sum_digits`. Use that function to compute the input number minus the sum of digits of input number. (2 points)

c. Write another function that "wraps" `diff_sum_digits`. If `diff_sum_digits` returns a result that has more than one digit (either negative or positive) have your new function replace the result with the sum of the digits of the result. Do this repeatedly until the result has just one digit, then display it. (2 points)

Below you'll find an example of what we mean when we say "wraps" - this is not a decorator function or anything like that, just a function that calls another function inside of it.

```
def example_base_func(x):
    "Returns the value of the input * -1"
    return -1 * x

def wraps_example_base_func(x):
    temp_val = example_base_func(x)
    if temp_val < 0:
        return "trivial example"

print(example_base_func(5))
print(wraps_example_base_func(5))
```

You can use the sample output in this cell to sanity check your work.

```
print(sum_digits(54321))
print(sum_digits(-54321))
print(diff_sum_digits(54321))
print('final value: ' + str(wraps_diff_sum_digits(54321)))

15
15
54306
final value: 9
```