

Python programming

This week's lab will focus on your first programs in Python. As in previous weeks, you will work in teams of two: one of you will be the *driver* and the other will be the *navigator*. The *driver* is the person who will perform entry on the system. The *navigator* is the person who tells the driver what to enter. You must alternate roles throughout the lab. Take advantage of this situation – bounce ideas off each other and discuss things when they do not seem clear. If you get stuck, raise your hand and your TA will help you as soon as they are free.

We started this course discussing algorithms. Now we will take problems, create a plan, and program it. Do not hesitate to create pseudo-code or flowcharts to better illustrate your plan before you code it.

Here are some Python programming hints:

1. To define a function:

```
def <functionName>(input1, input2, ... , inputn):
    indent body → <create commands that make up function>
```

2. A function can return a value (only one). To return a value, use:

```
return <value, expression or identifier name>
```

3. Remember to indent in blocks to identify a body of lines that must be executed. for example:

```
def countdown(start): #define a function countdown
    while start > 0: # perform the following statements (indented)
        print start
        start = start - 1
x = 4 #no longer in function - see how indent changes
countdown(x) # we have assigned 4 to x now we call
            # countdown and pass the value in x to
            # the function
```

Now you must open the IDLE integrated development environment and create the following functions:

Step 1

Create a function called `addThree` which accepts three values, adds them, and prints the following:
The sum of <value1> plus <value2> plus <value3> is <sum>

Where `value1`, `value2`, and `value3` are the values that were accepted, and `sum` is the sum of the three values.

So if you called function with the values 4, 2, and 7, it would print:

The sum of 4 plus 2 plus 7 is 13

Save your function in a file (`q1.py`) and test it. Call it using the values 4, 2, and 7. Assign these values to identifiers `a`, `b`, and `c` and call your function again using the identifiers.

Python programming

Step 2

Now modify your function so that it no longer prints anything. Instead, it returns the sum of the values (using the return statement). Remember a function can only return a single value.

Test the function calling it with three values. How can you be sure that it worked? Try printing the result of the function:

```
print addThree(4, 2, 7)
```

Nothing but the sum should be printed.

Step 3

Write a function that accepts a list of strings or numbers and prints them in reverse order.

Hint: `len(<listIdentifier>)` gives you the number of items in list `listIdentifier`.

How could you test this function? (with both strings and numbers?)

Step 4

Write a function that accepts a string, and a character, and returns the number of times that the character occurs in the string.

Hint: `len(<StringIdentifier>)` gives you the number of characters in string `StringIdentifier`.

We did this in pseudocode in class. You can use the length to limit how many times you go through the loop.

syntax hint if statement:

```
# here we use a equals b as our test condition,
# we could have used other conditions.
if a == b: # we use == to test if a equals b
    do this line # not really code - just a place holder
    do this line # maybe we want to print or add something here
    do this line # in these lines if the condition is true
else: # we execute the next block if the condition is false
    do this line
    do this line
    do this line
```

Note that the `else:` part of the statement is optional and only necessary if there is something to be done if the condition is false.

Step 5

Before you leave today – you may want to email these solutions to yourself and your partner so that you can use them to study from for future tests and assignments.