Functions, Part 1

March 8, 2021

Administrative Notes

Exam 1 grades should be available

- The grades were pretty good
- It was an easy test, but I'm still really pleased at the class performance
- Question 12 - remember that " ".join only works with lists of strings, not with lists of ints - you should have cast the elements of the list to be strings, e.g. in a for loop, prior to invoking join - I let that slide; normally that would be worth two points
- Question 13 and the "error" that wasn't an error - we'll walk through that (there's an explanation on Discord, but let's walk through)
- Question 17, the second programming question - the most common error was not writing a loop to generate the Fibonacci numbers - it was called out in the instructions

More administrative notes

Next week is Spring Break

There's no homework this week, both because of Spring Break and because the majors' sections are working on a take-home exam all week.

There IS a lab tomorrow/this week.

- It's on functions, which is the topic today and Wednesday
- You should be able to start working on it tomorrow; we'll at least have introduced all the concepts. Kev and Mariah can help tomorrow, too.

After Spring Break:

- Homework 5 (of 6, total) the week we come back
- Program project #1 right after that.

Now, on to Functions

The fourth type of program control flow:

- Sequential
- Conditional
- Iterative
- Functions - jump to another part of the program; execute code there; and then return to wherever you were before you jumped

What is a function?

In math, it's a unique mapping between each input and one output

$f(x) = x^2 + 2x + 1$

Put in a value for x, you'll get back one output.

In python, it's sort of the same idea, but not exactly

A block of code, called by its name, that optionally takes input and returns a specific output

An example of a function in Python

```
def calculate_days (years, months, days):
    num_days = 365 * years + 30 * months + days
    if years > 4:
        num_days += 1
    return num_days
```

This is the name of the function. It must be a legal Python variable name

These are the parameters.

The word “def” means you’re defining a function

This code is the body of the function

The “return” statement is a way of passing a value back to where the function was called from

Calling a function

To call the function the previous slide, calculate_days:

```
yrs = 23

mos = 3

days = 22

length_of_time = calculate_days(yrs, mos, days)
```

These are arguments. They will be matched up with parameters, in order

If the function returns a value, you have to store it somewhere if you want to use it

Call the function by using its name, with appropriate arguments

Why use functions?

Use a function whenever your program is going to do the same thing multiple times

- No, not like in a loop. In different *parts* of the program.
- You ***could*** rewrite the code each time, but you're likely to do it differently somewhere
- You could just copy and paste the same code each time, but what if you're just copying and pasting a bug?

Using functions simplifies the program and makes it easier to get the program correct

Functions vs. methods

calculate_days() is a *function*; str.split() is a *method*. What's the difference?

A ***method*** is explicitly tied to ***one object***; the object on which it is invoked.

A ***function*** is not tied to anything; it operates on the values passed to its parameters (if there are any).

Hopefully this becomes clear over the next couple of days!

Built-in functions and user-defined functions

Python has a number of built-in functions that you've already used:

```
print("Hello, world")   # print is a function; "Hello, world" is its argument.

len(num_list)           # len is a function; num_list is its argument. It returns the number of elements in num_list
```

You can add to the built_in functions with functions that you will define yourself. Like calculate_days, a few slides ago.

Where do you define functions?

Usually, above the main program - but that's *NOT* required.

Start with your header comment

```
LABELS = ['A', 'B', 'Others']
```

Now define your constants

```
def calculate_days(days, months, years)
```

Then define your functions

```
If __name__ == "__main__":
```

And finally your main program

Where do you call functions?

Anywhere other than on the left hand side of an equals sign

The call to a function can be anywhere in the program where the function is needed

Cān a function call another function?

- YES

Can a function call itself?

- YES; that's called 'recursion' and we'll get to that

Can a function call the main program?

- NO; it can return values to the main program but that's all

Does a function have to be defined before it can be called?

NO!!!! You can call a function before it’s defined!!! Python can handle this.

If Python sees something that looks like a function call, it will simply wait for that function to be defined.

Python3 homework6.py ...

Matching arguments and parameters

I said that parameters and arguments are matched in order. What does that mean?

```
def subtract(x, y):
   print(x,”-”,y,”=”,str(x-y))
if __name__ == “__main__”:
   x = 3
   y = 4
   subtract(y, x)
```

What happens if the arguments and parameters don’t match?

```
def subtract(x, y):
   print(x,”-”,y,”=”,str(x-y))
if __name__ == “__main__”:
   x = 3
   y = 4
   subtract(y)
```

Scope of variables

Scope means where a variable can be directly seen and used

Variables defined in the main program can be seen everywhere in the program. BUT - you should only directly use them in the main program. If you need their values in a function, pass them as arguments. If you use a main program variable directly in a function, without passing it as an argument, that's called a "global variable" and it will cost you major points!!!

Variables defined in functions can only be seen and used in the functions where they are defined. These are called "local variables."

Now some examples

The rest of the class period will be taken up with some examples of programming using functions. More explanations on Wednesday.