CS131B

12 Functions

Key terms: def return

Reading: Severance 4

Exercise: Write a program that dies with a RecursionError.

Compartmentalizing procedures with defined interfaces is nearly as old as programming itself; it was proposed by Grace Murray Hopper in 1952. Functions, or methods, are in Python declared using the keyword def and a collection of arguments. All calls are by reference and these arguments are normally untyped. Functions generally finish work by returning some object.

```
# Defining a temperature conversion function:
def c2f(c):
  return 32+9*c/5
# Defining its inverse as an inline lambda function:
f2c = lambda f: 5*(f-32)/9
```

Since version 3.5, Python supports statically typed function definitions.¹⁶ As an example, we can guarantee that a conversion function receive only numbers:

```
# Demonstrate conversion that rejects nonnumeric arguments:
def c2f(c:float):
  return 32+9*c/5
```

Arguments have to either have names or else occur in a specific position. Positional ones come in order, resembling a tuple; named arguments are more like a dict. In either case, default values (with =) makes it more possible for a function to serve both basic and advanced uses.

```
# Define and employ a helper function for annotating log entries.
# Note the combined positional and named arguments.
def annotate ( message, subsystem='unspecified' ):
  return '{}: {}.'.format(subsystem,message)
print ( annotate ( message="It is midnight" ) )
print ( annotate ( message='Irrigation commences',
  subsystem='garden' ) )
```

Testing is the process of verifying that functions and programs operate properly; that for a given input, they produce the expected output. Usually, we would want to test many cases, and have them all be correct; this is called unit testing. In the absence of a formal testing framework the programmer might still want to make some checks:

```
# Verify that f2c and c2f agree about certain temperatures:
print(f2c(32)==0)
print(c2f(100)==212)
print(f2c(-40)==c2f(-40))
```

Functions have their own scope, so names defined there do not exist elsewhere.

Advanced function techniques like generators, decorators, and execution pools are covered in CS231, Advanced Python Programming.

Next, we will address regular expressions.

¹⁶https://docs.python.org/3/library/typing.html