# Whirlwind Introduction To Types and Functions

Python directly recognizes a variety of types of data. Here are a few:

Numbers: 3, 6, -7, 1.25

Character strings: 'hello', 'The answer is: '

Lists of objects of any type: [1, 2, 3, 4], ['yes', 'no', 'maybe']

A special datum meaning nothing: None

Python has large collection of built-in functions that operate on different kinds of data to produce all kinds of results. To make a function do its action, parentheses are required. These parentheses surround the parameter or parameters, as in a function in algebra class.

The general syntax to execute a function is

functionName ( parameters )

One function is called type, and it returns the type of any object. The Python Shell will evaluate functions. In the Shell the last line should look like

>>>

Continuing on the same line enter

type(7)

Always remember to end with the Enter key. After the Shell responds, you should see something like

**>>>** type(7)

<class 'int'>

>>>

In the result, int is short for integer. The word class is basically a synonym for type in Python.

Note that the line with the value produced by the shell does not start with >>> and appears at the left margin. Hence you can distinguish what you type (after the >>> prompt) from what the computer responds.

At the end you see a further prompt where you can enter your next line....

For the rest of this section, at the >>> prompt in the Python Shell, individually enter each line below that is set off in typewriter font. So next enter

type(1.25)

Note the name in the last result is float, not real or decimal, coming from the term “floating point”, for reasons that will be explained later, in [Floats, Division, Mixed Types](http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/float.html#floats-division-mixed).

Enter

type('hello')

In your last result you see another abbreviation: str rather than string. Enter

type([1, 2, 3])

Strings and lists are both sequences of parts (characters or elements). We can find the length of that sequence with another function with the abbreviated name len. Try both of the following, separately, in the Shell:

len([2, 4, 6])

len('abcd')

Some functions have no parameters, so nothing goes between the parentheses. For example, some types serve as no-parameter functions to create a simple value of their type. Try

list()

You see the way an empty list is displayed.

Functions may also take more than one parameter. Try

max(5, 11, 2)

Above, max is short for maximum.

Some of the names of types serve as conversion functions (where there is an obvious meaning for the conversion). Try each of the following, one at a time, in the Shell:

str(23)

int('125')

An often handy Shell feature: an earlier Shell line may to copied and edited by clicking anywhere in the previously displayed line and then pressing theEnter key. For instance you should have entered several lines starting with len. click on any one, press Enter, and edit the line for a different test.

## The list Type

Lists are ordered sequences of arbitrary data. Lists are the first kind of data discussed so far that are mutable: the length of the sequence can be changed and elements substituted. We will delay the discussion of changes to lists until a further introduction to objects. Lists can be written explicitly.Read the following examples

['red', 'green', 'blue']

[1, 3, 5, 7, 9, 11]

['silly', 57, 'mixed', -23, 'example']

[] *# the empty list*

The basic format is a square-bracket-enclosed, comma-separated list of arbitrary data.

## The range Function

There is a built-in function range, that can be used to automatically generate regular arithmetic sequences. Try the following in the Shell:

list(range(4))

list(range(10))

The general pattern for use is

range(sizeOfSequence)

This syntax will generate the integers, one at a time, as needed [[2]](http://anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/loops.html#lazy). If you want to see all the results at once as a list, you can convert to a list as in the examples above. The resulting sequence starts at 0 and ends before the parameter. We will see there are good reasons to start from 0 in Python. One important property of sequences generated by range(n) is that the total number of elements is n. The sequence omits the number n itself, but includes 0 instead.