Strings and Lists

We've already seen numbers and strings, but Python also has variable types that can hold more than one piece of data at a time. The simplest such variable is a list.

You can assign data to a list in the following way: `list_name = [item_1, item_2, ..., item_n]`. The items of the list can be of any other type: integer, float, string. You even explore your inner Zen and make lists of lists!

Any item in a list can be accessed by its index, or the number that indicates its place in the list. For example, try running the following code:

```
tea_party = ['March Hare', 'Hatter', 'Dormouse', 'Alice']
print (tea_party[2])
```

Your output should be:

```
Dormouse
```

Note that the output was *not* `Hatter`, as you might have guessed. This is because in Python, indexing begins with 0, not 1. This property is called 0-based numbering, and it's shared by many programming languages.

You can easily change existing list items by reassigning them. Try running the following:

```
tea_party[1] = 'Cheshire Cat'
print (tea_party)
```

This code should output the list with "Hatter" replaced by "Cheshire Cat":

```
March Hare, Cheshire Cat, Dormouse, Alice
```

You can also add items to the end of an existing list by using the function `append()`:

```
tea_party.append('Jabberwocky')
print (tea_party)
```

This code outputs the following:

```
March Hare, Cheshire Cat, Dormouse, Alice, Jabberwocky
```

If you need to obtain only some of a list, you can use the notation `list_name[a:b]` to get only those from index `a` up to but *not* including index `b`. For example, `tea_party[1:3]` returns `Cheshire Cat, Dormouse`, not `Cheshire Cat, Dormouse, Alice`. This process is called "list slicing".

If the first index of the slice is unspecified, then Python assumes that the slice begins with the beginning of the list (i.e., index 0); if the second index of the slice is unspecified, then you will obtain the items at the end of the list. For example, `tea_party[:2]` returns `March Hare, Cheshire Cat` and `tea_party[3:]` returns `Alice, Jabberwocky`.

You can also use negative indices to count items backtracking from the end of the list. So `tea_party[-2:]` returns the same output as `tea_party[3:]`: `Alice, Jabberwocky`.

Finally, Python equips you with the magic ability to slice strings the same way that you slice lists. A string can be considered as a list of characters, each of which having its own index starting from 0. For example, try running the following code:

```
a = 'flimsy'
b = 'miserable'
c = b[0:1] + a[2:]
print (c)
```

This code will output the string formed by the first character of `miserable` and the last four characters of `flimsy`:

```
mimsy
```