

# LISTS

**Authors:** Jordi Masip and Oriol Lanuza.

**Date:** 02/03/2014

## INTRODUCTION

Lists are a datatype you can use to store a collection of different pieces of information as a sequence under a single variable name.

You should be careful to avoid using the word **list** as a variable, that's a reserved word.

You can assign items to a list with an expression of the form:

```
list_name = [item_1, item_2] #create a list of 2 items.
```

## ACCESS BY INDEX

You can access an individual item on the list by its index. An index is like an address that identifies the item's place in the list. The index appears directly after the list name, in between brackets, like this:

```
print list_name[index] #prints the item assigned to the index.
```

List indices begin with 0. You access the first item in a list like this:

```
print list_name[0] #prints the first item in the list.
```

A list index behaves like any other variable name. It can be used to access as well as assign values. For example:

```
list_name[3] = "Hello!" #this changes the item in index 3 to "Hello!"
```

## LIST LENGTH

A list doesn't have to have a fixed length. You can add items to the end of a list any time you like.

You can add new items with `.append()` function:

```
list_name.append(new_item) #adds new_item to the list
```

You can find the number of items in the list with the function `len(list_name)`:

```
print len(list_name) #prints the list's length
```

## LIST SLICING

If you only want to see a portion of a list, you can use `list_name[index1:index2]`:

```
letters = ['a', 'b', 'c', 'd', 'e']  
slice = letters[1:3] #slices become a list of the 2nd and 3rd elements
```

If your list slice includes the very first or last item in a list (or a string), the index for that item doesn't have to be included:

```
my_list[:2]           # Grabs the first two items
my_list[3:]           # Grabs the fourth through last items
```

We start at the index before the colon and continue up to but not including the index after the colon.

You can slice a string exactly like a list. In fact, you can think of strings as lists of characters: each character is a sequential item in the list, starting from index 0.

## SEARCH, INSERT AND ORDER

Sometimes you need to search for an item in a list. You can do this with `.index()` function:

We can also insert items into a list in order with `.insert()`, for example:

```
list_name.insert(1, 'Hello') #this will put the item 'Hello'
# in index 1 and move the following items on +1 index.
```

We can order a list with function `.sort()`:

```
animals = ["cat", "ant", "bat"]
animals.sort()

for animal in animals:
    print animal    #this will print 'ant','bat','cat'; in alphabetical order.
```

## REMOVING ELEMENTS

You can use `.remove(item)` to remove an item from the list, `.pop(index)` to remove the item at index from the list and return this one to you, or `del(list_name[index])` to remove that item like `.pop` without returning it:

```
n = [1, 3, 5]
n.pop(1)           # Returns and deletes 3 (the item at index 1)
n.remove(1)         # Removes 1 from the list
del(n[1])           # Removes 1 and doesn't return anything
```