

## Python Basics 3

### 1 Lists

To understand what we are talking about today, we must first go over a list. A list is a series of numbers held together by `[]`. A initialization of a list would like this:

```
l = [1, 2, 3, 10]
```

You can access a element in a list by using it's *index*, or location in the list by typing `l[0]`. Notice that the index starts with 0, which accesses the first element. All elements start from zero and go up. This means that element two can be accessed by `l[1]`, and element three by `l[3]`, and so on and so forth. Look at the code below for a better explanation

```
l = [1, 2, 3, 10]
print l[0]
print l[2]
```

The output for this code is 1, 3

### 2 for loops

No that we have explained lists, we will go over `for` loops. A `for` loop will iterate over a list. It will take each element from the list and give it to you in a temporary variable. In the example below, `x` is the variable, and `l` is the list:

```
l = [1, 2, 3, 10]
for x in l:
    print x
```

The output to this function is 1, 2, 3, 10 as the `for` loop iterates throw each element. Note the use of the `in` keyword. This needs to be in every `for` loop. This tells Python to pull elements from the list.

### 3 Control flow: while and if/else

Program execution proceeds from top to bottom, unless you change the *flow* of execution. Today, you will do this in three ways, `while`, `if/else`, and function calls. Let's make a while loop:

```
i = 0
while i < 10:
    print i
    i = i + 1
```

We define the *scope* of while loop by indenting the code to the right. Whitespace matters in Python. We *only use spaces* for this, and we *use four spaces for each level of indent*.

```
i = 0
while True:
    print i
    i = i + 1
```

How to stop this thing? Reset the robot. Just a short tap on the button will do. The use Alt+pt0 reconnect.

Well, `while` was a lot of fun. Let's add an `if` statement to our program:

```
i = 0
while i < 10:
    if i > 5:
        print i
    else:
        print -i
    i = i + 1
```

Let's look at some logical operators:

```
this = False
that = False
```

```
i = 0
while i < 4:
    print i
    if this:
        print 'this'
    if that:
        print 'that'
    if this and that:
        print 'this and that'
    if this or that:
        print 'this or that'
    i = i + 1

    if (not this) and (not that):
        this = False
        that = True
    elif (not this) and (that):
        this = True
        that = False
    elif (this) and (not that):
        this = True
        that = True
    else:
        this = False
        that = False
```