

## Writing Python Scripts – Useful Commands

### **Print:**

This is a command which you will probably use a lot when writing python scripts. By entering print followed by a variable or a string of text, the script will display that variable or string of text in the terminal window which you have ran the script from. For example, for a value if you had defined the variable angle as 45 you would enter:

```
print angle
```

This should appear in your Python script as:

```
print angle
```

This should produce the following output in your terminal window when the script is run:

```
45
```

If you wanted to do this for a string of text, for example Hello World, you would enter:

```
print "Hello World"
```

This should appear in your Python script as:

```
print "Hello World"
```

This should produce the following output in your terminal window when the script is run:

```
Hello World
```

### **Note:**

If you want to enter text in a Python script that the computer will recognize as text and not a variable, you must enter it in speech marks.

### **Comment:**

If you want to enter a comment in a Python script to explain what a part of the script does but is ignored by the computer when running the script, simply enter a hash symbol at the start of the line of text. For example:

```
# This is an example of a comment
```

This should appear in your Python script as:

```
# This is an example of a python script
```

### **Wait:**

This is a command which you will probably use a lot when writing python scripts. Before you can use this command you must type the following before you start to type the main body of your script :

```
import time
```

This should appear in your Python script as:

```
import time
```

Now by entering `time.sleep` immediately followed by a set of brackets with a decimal value in, you can tell the computer to wait the amount of time in seconds expressed by the decimal value before running the next line of code. For example, if you wanted the computer to wait 5 seconds before executing the next line of code you would enter:

```
time.sleep(5)
```

This should appear in your Python script as:

```
time.sleep(5)
```

### **Get Current Gripper Position:**

This is a set of commands that allows you to get the current position of the left or right arm gripper, you can do this by entering the following commands:

```
left = baxter_interface.Limb('left')
pose = left.endpoint_pose()
x = pose["position"].x
y = pose["position"].y
z = pose["position"].z
```

This should appear in your Python script as:

```
left = baxter_interface.Limb('left')
pose = left.endpoint_pose()
x = pose["position"].x
y = pose["position"].y
z = pose["position"].z
```

**Note:** to get the right arm version just change left to right everywhere it says left.

## **Get Current Gripper Orientation:**

This is a set of commands that allows you to get the current orientation of the left or right arm gripper, you can do this by entering the following commands:

```
left = baxter_interface.Limb('left')
pose = left.endpoint_pose()
curXo = pose["orientation"].x
curYo = pose["orientation"].y
curZo = pose["orientation"].z
curWo = pose["orientation"].w
```

This should appear in your Python script as:

```
left = baxter_interface.Limb('left')
pose = left.endpoint_pose()
curXo = pose["orientation"].x
curYo = pose["orientation"].y
curZo = pose["orientation"].z
curWo = pose["orientation"].w
```

**Note:** to get the right arm version just change left to right in the everywhere it says left

**Note:** these are in the Quaternion coordinate system and are not the Euler angle coordinates