

UTE**C**

UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA

**INGENIERÍA
ELECTRÓNICA**

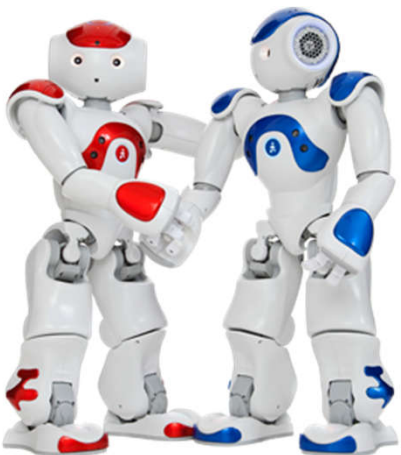


OpenDay 

- Día 2: Parte 3 - Introducción a Python

Prof. Oscar E. Ramos, Ph.D.

Universidad de Ingeniería y Tecnología (UTE)
Departamento de Ingeniería Electrónica
14 de febrero del 2018



Introducción a Python

- ¿Qué es Python?
 - Es un lenguaje de programación de tipo script
- Ejemplo de función en Python



```
def op1(a, b):  
    c = 2*a + 3*b  
    return c  
  
print op1(5, 6)
```

} Definición de la función

← Prueba de la función

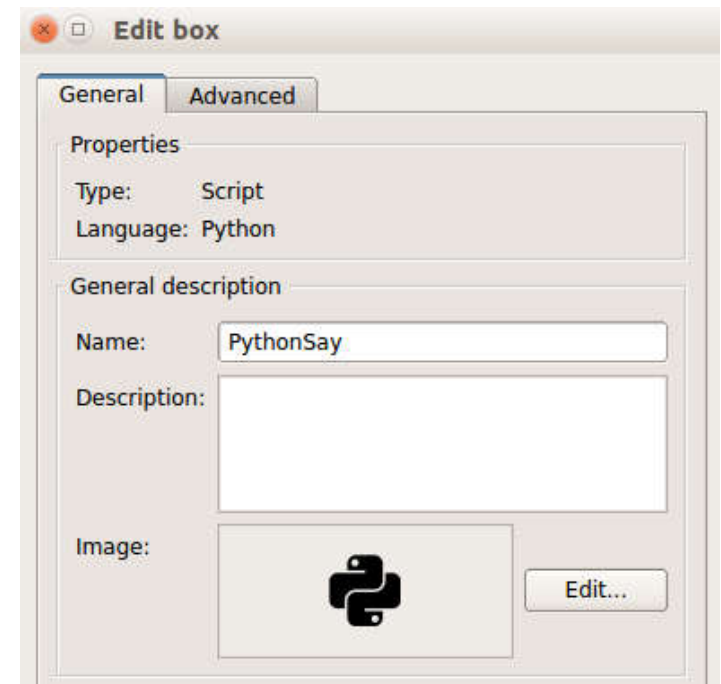
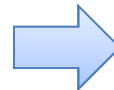
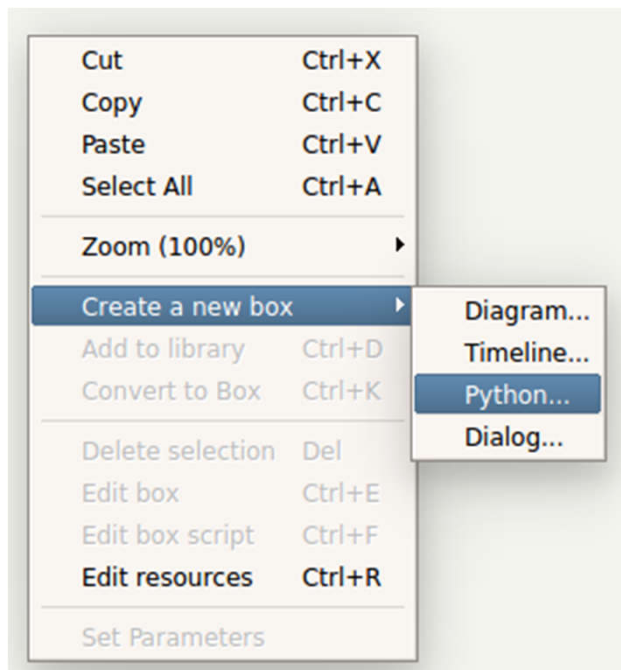
- Ejemplo de clase en Python

```
class MyClass:  
    def __init__(self):  
        pass  
  
    def set_values(self, a, b):  
        self.a = a  
        self.b = b  
  
    def op1(self):  
        c = 2*self.a + 3*self.b  
        return c
```

```
test = MyClass()  
test.set_values(5, 6)  
test.op1()
```

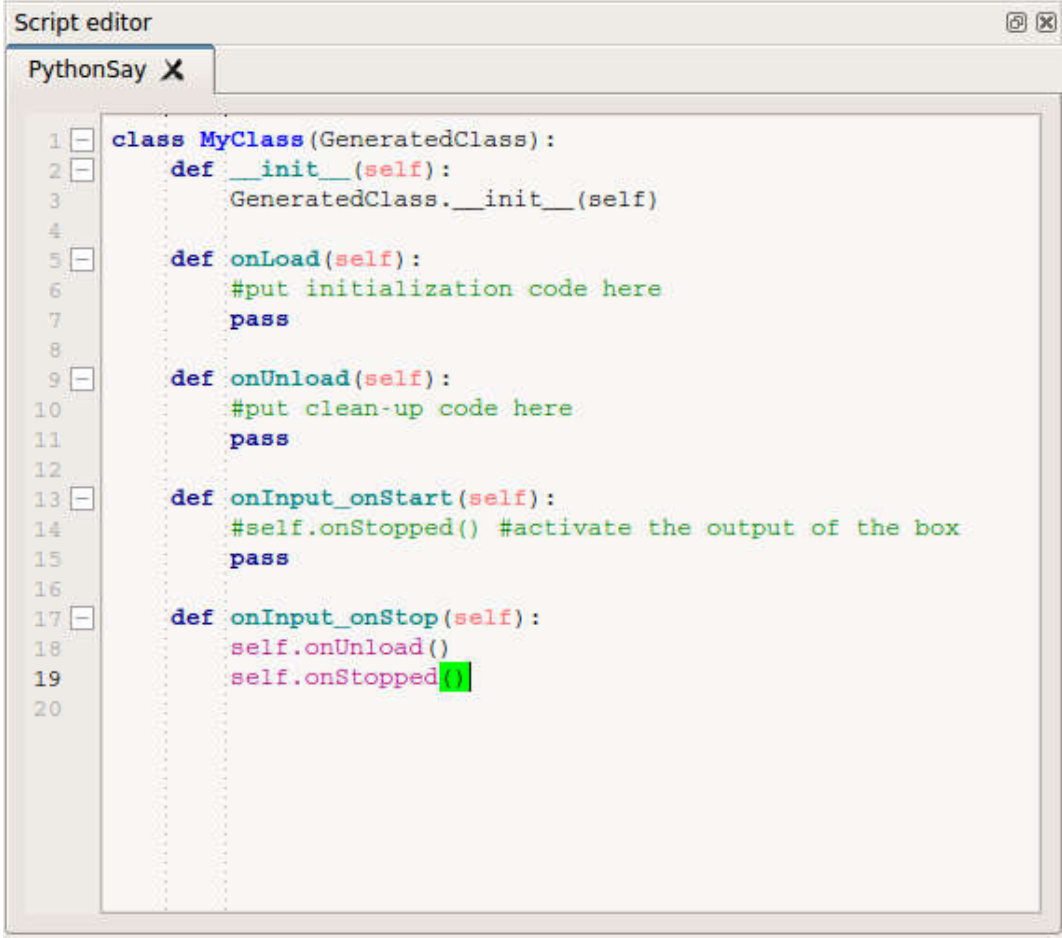
Robot Hablando con Python

- Hacer click derecho en el espacio de trabajo (espacio plomo), seleccionar “Create a new box” y luego “Python”
- Cambiar el nombre a “Python Say” y aceptar



Robot Hablando con Python

- Hacer doble click al bloque generado



```
Script editor
PythonSay X
1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4
5     def onLoad(self):
6         #put initialization code here
7         pass
8
9     def onUnload(self):
10        #put clean-up code here
11        pass
12
13    def onInput_onStart(self):
14        #self.onStopped() #activate the output of the box
15        pass
16
17    def onInput_onStop(self):
18        self.onUnload()
19        self.onStopped()
20
```

Robot Hablando con Python

- En la función `onInput_onStart` escribir:

```
ttsProxy = ALProxy("ALTextToSpeech")  
ttsProxy.say("Hello world!")
```

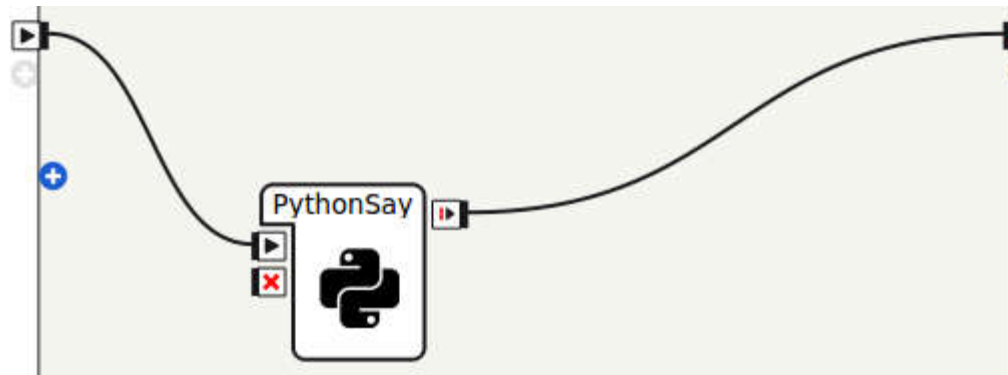
```
class MyClass(GeneratedClass):  
    def __init__(self):  
        GeneratedClass.__init__(self)  
  
    def onLoad(self):  
        #put initialization code here  
        pass  
  
    def onUnload(self):  
        #put clean-up code here  
        pass  
  
    def onInput_onStart(self):  
        ttsProxy = ALProxy("ALTextToSpeech")  
        ttsProxy.say("Hola a todos")  
  
    def onInput_onStop(self):  
        self.onUnload()  
        self.onStopped()
```

Nota:

El espaciado (indentación) es importante en Python
Escribir las líneas usando la indentación adecuada

Robot Hablando con Python

- Conectar el bloque en Python y ejecutar
 - El robot debe decir lo indicado



Caminar usando Python

- Crear un bloque de Python llamado PythonWalk y conectarlo:



- Dicho bloque realizará lo siguiente:
 - Hacer que el robot se levante
 - Hacer que el robot camine hacia adelante 0.20 m
 - Hacer que el robot camine hacia atrás 0.20 m
 - Hacer que el robot se siente
 - Apagar los motores

Caminar usando Python

- Hacer doble click al bloque e ingresar las siguientes líneas de código en la función “onInput_onStart”

Nota: las líneas que comienzan con # son comentarios y no necesitan escribirse

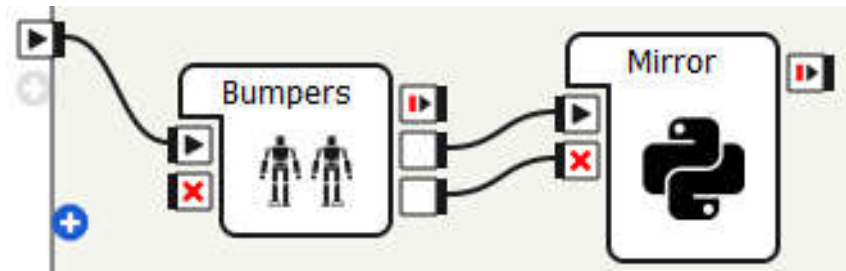
```
def onInput_onStart(self):  
    # Proxy para posiciones predeterminadas  
    postureProxy = ALProxy("ALRobotPosture")  
    # Proxy para el movimiento  
    motionProxy = ALProxy("ALMotion")  
    # 1) Levantarse  
    postureProxy.goToPosture("Stand", 0.8)  
    # 2) Caminar hacia adelante  
    motionProxy.moveTo(0.2, 0.0, 0.0)  
    # 3) Caminar hacia atras  
    motionProxy.moveTo(-0.2, 0.0, 0.0)  
    # 4) Sentarse  
    postureProxy.goToPosture("Sit", 0.8)  
    # 5) Apagar los motores  
    motionProxy.post.stiffnessInterpolation("Body", 0, 0.1)  
    time.sleep(0.1)  
    self.onStopped()
```


Espejo de un Brazo

- **Objetivo:**

Al mover el brazo izquierdo del robot, el brazo derecho realiza el mismo movimiento a modo de “espejo”

- Construir el siguiente diagrama, donde **Mirror** es un bloque de Python



- Agregar el siguiente código en el bloque de Python

Espejo de un Brazo

```

1 class MyClass(GeneratedClass):
2     def __init__(self):
3         GeneratedClass.__init__(self)
4         self.motion = ALProxy("ALMotion")
5         self.done = False
6         self.stiff = ['RShoulderPitch', 'RShoulderRoll', 'RElbowYaw', 'RElbowRoll']
7         self.unstiff = ['LShoulderPitch', 'LShoulderRoll', 'LElbowYaw', 'LElbowRoll']
8         self.origStiffness = self.motion.getStiffnesses(self.unstiff)
9
10    def onLoad(self):
11        #put initialization code here
12        pass
13
14    def onUnload(self):
15        #put clean-up code here
16        self.done = True
17        self.motion.setStiffnesses(self.unstiff, self.origStiffness)
18        pass
19
20    def onInput_onStart(self):
21        #self.onStopped() #activate the output of the box
22        self.motion.setStiffnesses(self.unstiff, 0.0)
23        self.done = False
24        while not self.done:
25            vals = self.motion.getAngles(self.unstiff, True)
26            vals[1] = -vals[1]
27            vals[2] = -vals[2]
28            vals[3] = -vals[3]
29            self.motion.setAngles(self.stiff, vals, 0.5)
30            time.sleep(0.1)
31        pass
32
33    def onInput_onStop(self):
34        self.onUnload() #it is recommended to reuse the clean-up as the box is stopped
35        self.onStopped() #activate the output of the box
  
```

Ejercicio

- Hacer que al mover el brazo, el robot mire su mano
 - Deshabilitar la rigidez del robot
 - Obtener la posición de la cabeza y de la mano con la función **getPosition** (de ALMotion)
 - Usando la función **setAngles**, hacer que la cabeza del robot mire en la dirección del vector de la cabeza a las manos.
- Parte 2:
 - Cambiar la mano que mira el robot al presionar alguno de sus sensores