Clase 3: Usando los sensores

Grupo de Desarrollo Lihuen

Año 2012



Clase 3

- Repaso de estructuras básicas de Python
- Los sensores del Robot
- Guía práctica

Primer curso de programación usando robots y Python

Objetivos

El objetivo de esta clase es afianzar el uso del lenguaje Python utilizando los sensores del robot

Python

Repasando la sintaxis

¿Qué vimos hasta ahora?

- Sentencias condicionales usando la instrucción if o sus variantes if-elif-else
- Estructuras iterativas (sentencias while y for)
- Definimos funciones usando el constructor def
- Escribimos módulos en Python

Sentencia if

Usamos la sentencia if para ejecutar instrucciones en forma condicional

if condicion:
 sentencias
elif condicion:
 sentencias
else:
 sentencias



¿ Se acuerdan qué hace la función raw input?

```
print '''Indicanos hacia qué direcci
    ón te gustaría mover el robot:
1.- Avanzar"
2.- Retroceder'''
opcion=raw_input("Opción:")
if opcion=='l':
    robot.forward(100,1)
elif opcion=='2':
    robot.backward(100,1)
else:
    robot.stop()
```

Sentencia while

Permite ejecutar repetidamente un conjunto de instrucciones mientras se cumpla una condición usamos la sentencia while

while condicion:
 sentencial
 sentencia2

Sentencia while

Permite ejecutar repetidamente un conjunto de instrucciones mientras se cumpla una condición usamos la sentencia while

while condicion:
 sentencial
 sentencia2

```
print('''Acción para mover el robot:
       1.- Girar a la derecha
       2.- Girar a la izquierda
       4. - Salir'')
opcion=raw_input ("Opción:")
while (opcion! = '4'):
  if opcion=='1':
    robot.turnRight(100,1)
  elif opcion=='2':
    robot.turnLeft(100,1)
    robot.forward(100,1)
  else:
    print "Ingresaste una opción no
        válida."
  opcion = raw input ("Opción: ")
```

Sentencia for

Cuando podemos saber exactamente las veces que queremos repetir una serie de instrucciones podemos usar la sentencia for

Sentencia for

for var in lista:
 sentencial
 sentencia2

ejemplo

```
for i in [1,2,3]:
  robot.forward(50, 0.5)
  robot.bakckward(50, 0.5
  robot.turnRight(35, 1)
  robot.turnLeft(35, 1)
  wait(1)
```

Sentencia for

Cuando podemos saber exactamente las veces que queremos repetir una serie de instrucciones podemos usar la sentencia for

Sentencia for

for var in lista:
 sentencial
 sentencia2

ejemplo

```
for i in [1,2,3]:
  robot.forward(50, 0.5)
  robot.bakckward(50, 0.5)
  robot.turnRight(35, 1)
  robot.turnLeft(35, 1)
  wait(1)
```

Función range()

La función range () permite generar listas en forma dinámica

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(3, 8)
[3, 4, 5, 6, 7]
>>> range(0, 10, 2)
[0, 2, 4, 6, 8]
>>> range(7, 3, -1)
[7, 6 5, 4]
>>> range(-10, -100, -30)
[-10, -40, -70]
```

Función range()

Veamos un par de ejemplos:

Ejemplo¹

```
for i in range(1,4):
  robot.forward(50, 0.5)
  robot.bakckward(50, 0.5)
  robot.turnRight(35, 1)
  robot.turnLeft(35, 1)
  wait(1)
```

įy esto?

```
sum=0
for i in range(1,100, 2):
   sum=sum+i
print sum
```

Función range()

Veamos un par de ejemplos:

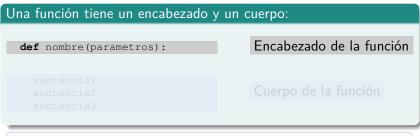
Ejemplo

```
for i in range(1,4):
  robot.forward(50, 0.5)
  robot.bakckward(50, 0.5)
  robot.turnRight(35, 1)
  robot.turnLeft(35, 1)
  wait(1)
```

¿y esto?

```
sum=0
for i in range(1,100, 2):
   sum=sum+i
print sum
```

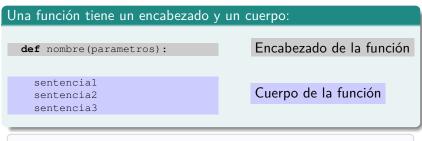
Funciones





CUIDADO CON LA INDENTACION!!!!!

Funciones





CUIDADO CON LA INDENTACION!!!!!

Funciones

Una función tiene un encabezado y un cuerpo: def nombre(parametros): Encabezado de la función sentencia1 sentencia2 sentencia3 Cuerpo de la función



CUIDADO CON LA INDENTACION!!!!!

Funciones

```
def mi_funcion(arg1, arg2);
  sentencia1
  sentencia2
```

Para ejecutar un función se la debe invocar

```
mi_funcion(20,10)
asociando:
arg1 con el valor 20, y
arg2 con el valor 10.
```

Funciones

```
def mi_funcion(arg1, arg2);
  sentencia1
  sentencia2
```

• Para ejecutar un función se la debe invocar:

```
mi_funcion(20,10) asociando:
```

- arg1 con el valor 20, y
- arg2 con el valor 10.

Funciones

Las funciones pueden tener argumentos que tomen valores por defecto:

```
def mi_funcion(arg1=1, arg2=3):
    sentencia1
    sentencia2
```

Pudiendo invocar

- mi_funcion(): Si no envío argumentos, se asumen los valores por defecto
- mi_funcion(2): En cuyo caso se asocia arg1 al valor 2 y arg2 se asocia con el valor por defecto
- mi_funcion(arg2=2, arg1=1000): En este caso, la asociación queda explícita nombrando los parámetros

Funciones

Las funciones pueden tener argumentos que tomen valores por defecto:

```
def mi_funcion(arg1=1, arg2=3):
    sentencia1
    sentencia2
```

Pudiendo invocar:

- mi_funcion(): Si no envío argumentos, se asumen los valores por defecto
- mi_funcion(2): En cuyo caso se asocia arg1 al valor 2 y arg2 se asocia con el valor por defecto
- mi_funcion(arg2=2, arg1=1000): En este caso, la asociación queda explícita nombrando los parámetros

Ejemplo

```
def hago_figuras(r):
  tiempo = 2
  1ado = 30
  esperar = 2
  cant figuras = input ("Ingrese la cantidad de figuras: ")
  for i in range(cant_figuras):
    cant_lados = input("Ingrese el numero de lados de la
        figura "+str(i+1)+":")
    vuelta = input ("Según la figura entre la vuelta de la
        figura "+str(i+1)+":")
    for j in range(cant_lados):
      r.forward(lado, tiempo)
      r.turnRight(vuelta, 2)
      wait (esperar)
```

Programa - Módulo

- Usando el IDE, guardamos nuestro código en un archivo
- Extensión .py.
- Minúsculas y mayúsculas!!

Mi programa

print "Hola mundo"

Lo ejecutamos en la terminal o desde Geany **python** hola.py

Trabajando con los sensores

Volviendo al robot



- Sensor de distancia
- Sensor de línea

Trabajando con los sensores

Volviendo al robot



- Sensor de distancia
- Sensor de línea

Trabajando con los sensores

¿Cómo obtenemos los valores de los sensores?

- senses(robot) Muestra en pantalla continuamente el estado.
- robot.ping() Devuelve la distancia en centímetros al objeto frente al robot.
- robot.getObstacle() Devuelve True si hay un obstáculo a menos de 10 centímetros.

Trabajando con los sensores

¿Cómo obtenemos los valores de los sensores?

- senses(robot) Muestra en pantalla continuamente el estado.
- robot.ping() Devuelve la distancia en centímetros al objeto frente al robot.
- robot.getObstacle() Devuelve True si hay un obstáculo a menos de 10 centímetros.

Trabajando con los sensores

¿Cómo obtenemos los valores de los sensores?

- senses(robot) Muestra en pantalla continuamente el estado.
- robot.ping() Devuelve la distancia en centímetros al objeto frente al robot.
- robot.getObstacle() Devuelve True si hay un obstáculo a menos de 10 centímetros.

Trabajando con los sensores

Vista de senses(robot)



Trabajando con los sensores

Ejemplo con ping()

```
def hay_obstaculo(robot):
   if robot.ping() <15:
      return True
   else:
      return False</pre>
```

Usando la función hay_obstaculo()

```
robot.forward()
while not hay_obstaculo(robot):
    pass
robot.stop()
```

Trabajando con los sensores

Ejemplo con ping()

```
def hay_obstaculo(robot):
   if robot.ping() <15:
      return True
   else:
      return False</pre>
```

Usando la función hay_obstaculo()

```
robot.forward()
while not hay_obstaculo(robot):
    pass
robot.stop()
```

Ejemplos con getObstacle()

Sin parámetros

```
robot.forward()
while not robot.getObstacle():
    pass
robot.stop()
```

Avanza hasta que encuentra un obstáculo a 10 centímetros.

Con parámetros

```
robot.forward()
while not robot.getObstacle(30):
    pass
robot.stop()
```

Avanza hasta que encuentra un obstáculo a 30 centímetros.

Ejemplos con getObstacle()

Sin parámetros

```
robot.forward()
while not robot.getObstacle():
    pass
robot.stop()
```

Avanza hasta que encuentra un obstáculo a 10 centímetros.

Con parámetros

```
robot.forward()
while not robot.getObstacle(30):
    pass
robot.stop()
```

Avanza hasta que encuentra un obstáculo a 30 centímetros.

Sensores de líneas

Sensor de contraste

```
robot.getLine()
izq ,der = robot.getLine()
```



Si queremos guardar la información debemos usar una tupla izq, der = robot.getLine()

Primer curso de programación usando robots y Python Ahora si..

A trabajar!! ..