Usando el

Robot Educativo Programable

Germán Osella Massa
german.osella@nexo.unnoba.edu.ar

Agenda

- Python, el lenguaje de programación
- Python en la actualidad
- Instalación del software del REP
- Arquitecturas soportadas
- Uso y programación del REP
- Uso de Administradores
- ¿Preguntas?

Python

Características

- Lenguaje de programación claro, simple y minimalista.
- Uso de la indentación para definir bloques.
- Imperativo (procedural y orientado a objetos).
- Fuertemente tipado, dinámico e interpretado.
- Permite identificadores con acentos y eñes.
- Sesiones interactivas (uso de un REPL).

¡Ver DEMO!

Python en la actualidad

- Ubicado entre los primeros 5 lenguajes de programación más populares en el mundo (según índices TIOBE y PYPL).
- En el ámbito académico, se ha convertido en el lenguaje introductorio más utilizado en las universidades más prestigiosas de Estados Unidos.
- Permite la exploración y descubrimiento del lenguaje como de su biblioteca estándar a partir de la ejecución de sentencias o breves fragmentos de código en su REPL.

Robot Educativo Programable

Instalación del software

- Se requiere un intérprete de Python 3.
- Preferiblemente CPython 3.5 o 3.4.
- Se recomienda instalarlo con WinPython:

https://winpython.github.io/

- El software del robot se instala desde una consola ejecutando el comando:

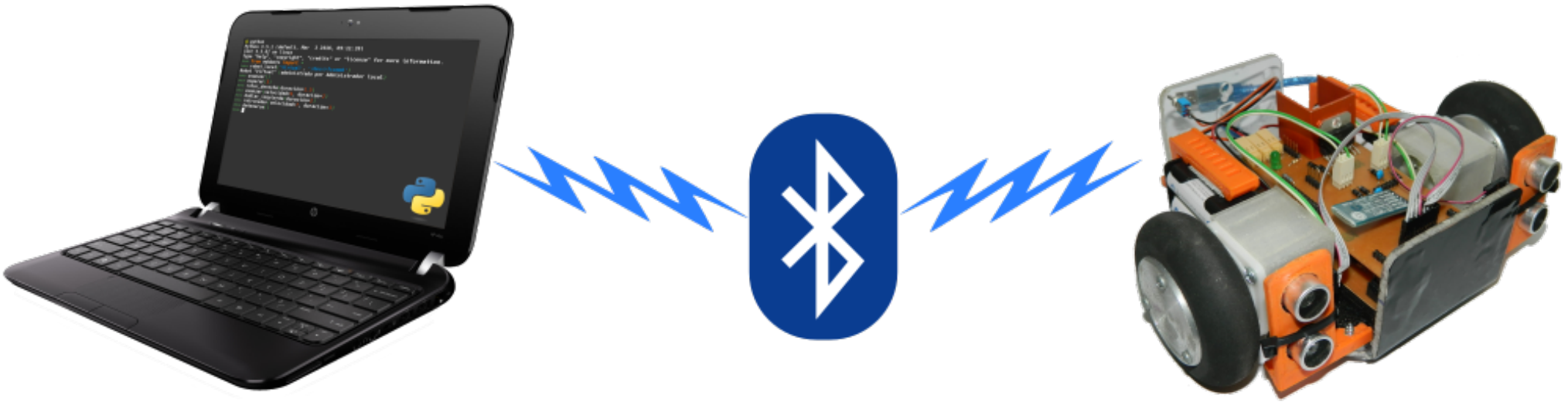
```
> pip install edubots
```

Arquitectura

- Se consideró importante simplificar al máximo la interacción con el robot.
- El robot no es autónomo.
- Se lo comanda desde una computadora utilizando órdenes simples en un lenguaje de muy alto nivel.
- Para la comunicación se utiliza Bluetooth
- Se admiten varios esquemas de uso.

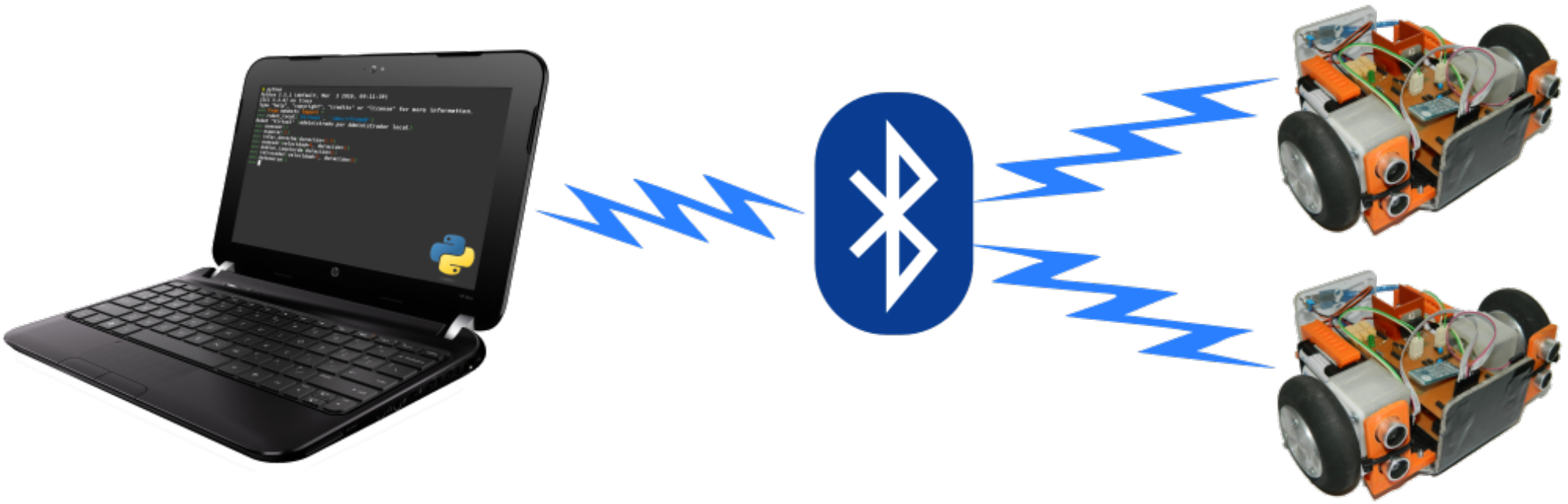
Arquitectura

El esquema más simple:



Arquitectura

Puede expandirse:



(hasta 6 robots por adaptador bluetooth)

Arquitectura

Si se usan varias máquinas en red:

Arquitectura

- En un **Administrador** es necesario ejecutar el **servidor** de los robots.
- Debe **vincularse a cada robot** con el Administrador de la misma forma que se asocia cualquier dispositivo bluetooth.
- Cada robot tendrá así asignado un puerto de comunicaciones **`COM:`** o **`/dev/rfcomm`**.
- Debe indicarse el nombre de cada robot junto con su puerto asociado en un archivo **`edubots.ini`**.

Arquitectura

Ejemplo de un archivo **edubots.ini**:

```
[Server]
interface = 192.168.1.10
```

Dirección IP del Administrador
(puede omitirse)

```
[Robots]
Pablo  = COM4
Tyrone = COM5
Uniqua = COM6
```

Robots reconocidos por el Administrador

Los nombres de los robots deben darse de la forma:
Nombre = Puerto

Arquitectura

El servidor se ejecuta mediante:

```
> python -m edubots.server
```

Buscará el archivo **edubots.ini** primero en el directorio actual y, si no lo encuentra, verá si existe en el directorio del usuario.

También puede configurarse mediante parámetros:

```
> python -m edubots.server 192.168.1.2 Tasha COM8
```

Arquitectura

¡Y pueden haber varios administradores!

Programación

- El robot se comanda vía las funciones provistas por la biblioteca **edubots**.
- Se requiere el “encantamiento mágico”:

```
>>> from edubots import *
```

Luego de ejecutar la línea anterior, todas las funciones del robot estarán disponibles para usarse como si estuvieran incluidas en la propia biblioteca estándar de Python.

Programación

- Si no se utiliza un Administrador, se debe asociar al robot con un puerto serie de comunicaciones y luego utilizar la función **conocer_robot()** para comenzar a usarlo:

```
>>> conocer_robot('Austin', 'COM8:')
Robot "Austin" (administrado por Administrador local)
>>>
```

Programación

- Un robot entiende tres tipos de comandos:
  - De **administración**: Permiten conocer información del robot, suspenderlo, reanudar su funcionamiento o dejar de usarlo.
  - De **movimiento**: Indican al robot que realice algún tipo de desplazamiento o que se detenga.
  - De **sensores**: Obtienen información del estado de los distintos tipos de sensores.

Programación

- Los comandos de administración son:
  - **`robots()`**: Devuelve una lista con los robots conocidos.
  - **`robot(nombre)`**: Establece que robot será el que reciba los siguientes comandos.
  - **`olvidar_robot(nombre)`**: Termina la conexión con el robot que tenga el nombre dado, impidiendo que se lo pueda seguir comandando.

Programación

- Los comandos de administración son:
  - **`nombre()`**: Devuelve el nombre del robot actualmente comandado.
  - **`suspender()`**: Suspende al robot actual, haciendo que se detenga y comience a ignorar todos los comandos recibidos.
  - **`reanudar()`**: Reactiva al robot que fue previamente suspendido.

Programación

- Los comandos de movimiento son muchos y variados pero tienen la misma forma:

```
acción(velocidad, duración)
```

Las posibles **acciones** son:

- **avanzar**
- **retroceder**
- **rotar_izquierda**
- **rotar_derecha**
- **doblar_izquierda**
- **doblar_derecha**
- **doblar_izquierda_avanzando**
- **doblar_izquierda_retrocediendo**
- **doblar_derecha_avanzando**
- **doblar_derecha_retrocediendo**

Programación

- La **velocidad** indica con que rapidez se realizará la acción y se la expresa como un valor real entre **0** y **1** (ambos incluidos), donde **0** indica detenido y **1** indica la velocidad máxima.
- Esto permite trabajar con valores con parte decimal (por ejemplo, **0.5**) o como fracción (por ejemplo, **1/2**) para señalar que se desea lograr un desplazamiento **“a media máquina”**.

Programación

- La **duración** de la acción se la expresa en segundos y son admisibles valores entre **0** y **60** segundos inclusive, permitiendo además valores reales con parte decimal, para indicar fracciones de segundo.
- La ejecución de la **acción** **detendrá la ejecución del programa** hasta que hubiera trascurrido la **duración** indicada.

Programación

- Si no se indica la **velocidad**, se asume un desplazamiento a velocidad media (**1/2**).
- Si no se indica la **duración**, se asume que la acción se realizará **indefinidamente**. La consecuencia de esto es que la **ejecución del programa continuará inmediatamente después de haber iniciado la acción**, permitiendo tomar decisiones al mismo tiempo que el robot se mueve.

Programación

- Existen dos comandos más que no siguen el patrón anterior:

  moverse(izquierda,derecha,duración)

  que permite establecer por separado la velocidad con la que gira cada una de las ruedas, admitiendo valores reales entre **-1** y **1** (incluidos), donde un valor negativo señala retroceso en lugar de avance.

  detenerse()

  Detiene al robot, dejándolo inmóvil.

Programación

- Los comandos de sensores permiten obtener mediciones a través de los sensores de distancia (ultrasónicos):

```
sensor_distancia(ubicación)
```

ubicación puede ser **'izquierda'** o **'derecha'**, indicando que se desea obtener la medición únicamente con el sensor del lado señalado.

Programación

- Los comandos de sensores permiten obtener mediciones a través de los sensores de distancia (ultrasónicos):

```
sensores_distancia()
```

Devuelve una lista con las mediciones obtenidas usando cada uno de los sensores de distancia disponibles.

Programación

- También se pueden realizar mediciones con los sensores del detector de líneas:

```
sensores_línea()
```

Devuelve una lista con las mediciones obtenidas usando cada uno de los sensores infrarrojos disponibles.

Programación

- Se incluyen otros comandos útiles:
  - **`esperar(duración)`**: Suspende la ejecución del programa durante la duración señalada.
  - **`tiempo()`**: Devuelve el tiempo actual en segundos, expresado en un número real.
  - **`al_azar(probabilidad)`**: Devuelve un valor verdadero o falso al azar según la probabilidad indicada.

¡Ver DEMO!

Administradores

Administradores

- Comandos para usar administradores:
  - **`conocer_administrador(dirección)`**: Pone como disponibles a todos los robots que posee el administrador indicado.
  - **`olvidar_administrador(dirección)`**: Deja de trabajar con el administrador indicado, haciendo desaparecer todos los robots que éste poseía.
  - **`administradores()`**: Devuelve una lista con todos los administradores en uso.

Administradores

- Al comenzar a trabajar con un nuevo administrador, automáticamente estarán disponibles para controlar todos los robots que posea, los que pueden verse mediante la función **`robots()`**.
- De igual manera, la dejar de trabajar con un administrador, se perderá el acceso a todos los robots de dicho administrador.

¿Preguntas?

Referencias:

- Sitio oficial de Python:
  https://www.python.org/
- Comunidad de Python Argentina:
  http://www.python.org.ar/
- Sitio de WinPython:
  https://winpython.github.io/
- Desarrollo de la biblioteca **`edubots`**:
  https://github.com/gosella/edubots