


# EVE

# tutorials

**Mersul**  **înainte**

MicroPython

By Sanjay and Arvind Seshan



## BEGINNER PROGRAMMING LESSON

# OBIECTIVELE LECȚIEI

1. Să învățăm cum să facem robotul să meargă înainte și înapoi.
2. Să învățăm cum să folosim clasa DriveBase.

## Cerințe:

1. Cunoștințe de bază în Python (e.g. Ce sunt variabilele, cum scriem expresii)

# SĂ ÎNCEPEM

1

```
#!/usr/bin/env pybricks-micropython
```

2

```
from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor,
                                  InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color,
                                  SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase
```

**Un program tipic de ev3 python începe cu linii ca cele de mai sus. Programul de mai sus este generat automat cu fiecare program nou.**

**Linia (1) spune EV3-ul să utilizeze micropython pentru a rula acest cod.**

**Liniile marcate (2) spun micropython-ului să încarce părți specifice cum ar fi codul pybricks pe care îl vei folosi în acest program.**

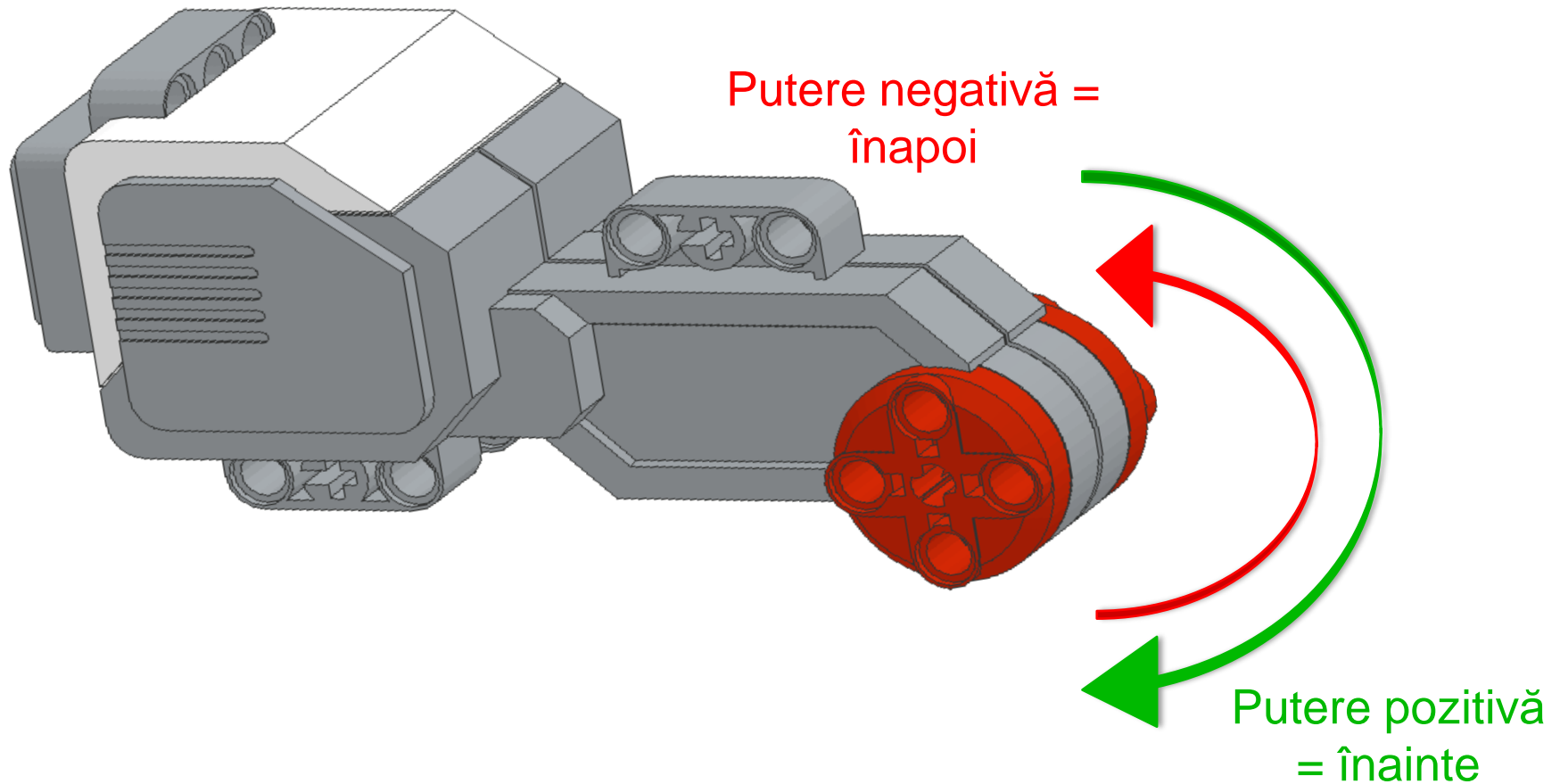
# MODURI DIFERITE DE MIȘCARE

La fel ca în mediul de programare graphic EV3, sunt diferite moduri de a face robotul să se miște. Această lecție acoperă două modalități:

**1) DriveBase:** Utilizarea clasei DriveBase îți oferă posibilitatea de a comanda ambele motoare simultan și fac ca robotul să se întoarcă în același timp. Este mai mult un fel de block de programare EV3-G “move steering” și “move tank”.

**2) Comenzile de motoare:** Utilizarea clasei motoarelor îți permit să comanzi fiecare motor de tracțiune independent. Aceasta seamănă mai mult ca block-urile de programare din Ev3-G “large motor” și “medium motor”. Aceasta va fi acoperit de o altă lecție.

# PUTERE NEGATIVĂ & POZITIVĂ: ÎNAPOI & ÎNAINTE



# CREAREA UNUI DRIVEBASE

1

```
# Initialize two motors with default settings on Port B and Port C.
```

```
left_motor = Motor(Port.B)  
right_motor = Motor(Port.C)
```

```
# The wheel diameter of the Robot Educator is 56 millimeters.  
# The distance between wheels (axle_track) is 114 millimeters.
```

```
wheel_diameter = 56  
axle_track = 114
```

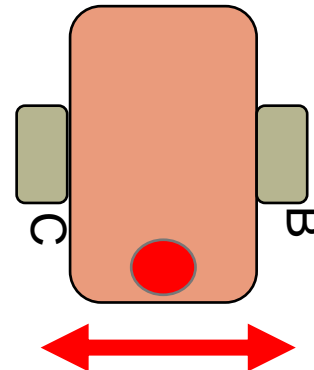
```
# Create a DriveBase object. The wheel_diameter and axle_track values are needed to move robot correct  
speed/distance when you give drive commands.
```

2

```
robot = DriveBase(left_motor, right_motor, wheel_diameter, axle_track)
```

Înainte să poți utiliza comenzi simple, trebuie să creezi un obiect DriveBase care folosește două motoare, diametru roții și lungimea osiei.

(1) Setează unele variabile pentru a stoca parametrii și (2) creează DriveBase.



Osia  
(distanța dintre roți  
în mm)



Diametru roții  
(fie măsoară sau  
citești înscrisurile  
în mm).

# CUM MERGEM ÎNAINTE?

```
# This drives at 100 mm/sec straight  
robot.drive (100, 0)  
# This drives straight backwards at 500 mm/sec for 2 seconds  
robot.drive_time(-500, 0, 2000)
```

Cu un obiect DriveBase, poți face robotul să meargă în diferite moduri.

`drive(speed, steering)` → merge cu viteza `speed` mm/sec cât `steering` degrees/sec până când programul se oprește sau primește o altă comandă.

`drive_time(speed, steering, time)` → merge cu viteza `speed` mm/sec în timp ce `steering` degrees/sec for `time` milliseconds

Cum facem să se miște o anumită distanță? Rotații? Trebuie să învățăm cum să folosim senzorul de rotație.

# CUM NE OPRIM?

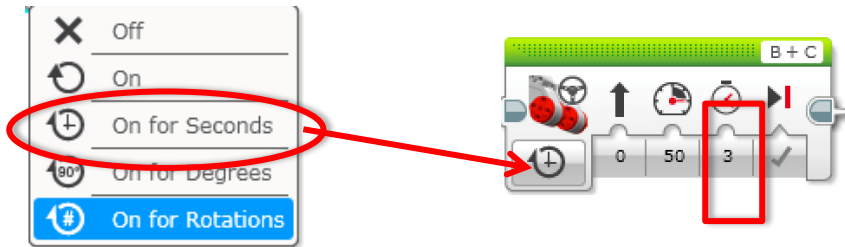
```
# this stops any active movement and actively brakes the motor  
robot.stop(Stop.BRAKE)  
  
# this stops any active movement and leaves the motors on coast  
robot.stop(Stop.COAST)
```

`drive_time()` se va opri după trece perioada de timp specificată. Cu toate acestea, motorul va continua să meargă după ce se îndeplinește condiția de ieșire. Dacă dorești să se oprească motoarele, trebuie să utilizezi comanda de `stop()`.

Toate cele de mai sus corespund comezilor de frână bruscă și oprire din software-ul EV3-G.



# PROVOCAREA 1: MERGI ÎNAINTE (3 SECUNDE)



Ținta este să scriem un program python care mișcă robotul pentru 3 secunde la 500mm/sec și încearcă să se oprească precis la 1500mm (1.5m).

Această comandă este similară cu block-ul verde din stânga. Observă puterea block-ului verde nu este în mm/sec ci în \*10 grade/sec, 50 putere este 500 grade/secundă.

# PROVOCAREA 1 SOLUȚIA: MERGI ÎNAINTE (3 SECONDS)

1

```
#!/usr/bin/env pybricks-micropython
from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color, SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase
```

```
# Initialize two motors with default settings on Port B and Port C.
```

```
left_motor = Motor(Port.B)
```

```
right_motor = Motor(Port.C)
```

```
# setup wheel diameter and axle_track
```

```
wheel_diameter = 56
```

```
axle_track = 114
```

```
# setup DriveBase
```

```
robot = DriveBase(left_motor, right_motor, wheel_diameter, axle_track)
```

```
# This drives straight at 500 mm/sec for 3 seconds
```

```
robot.drive_time(500, 0, 3000)
```

```
# This stops the motor and brakes for accuracy
```

```
robot.stop(Stop.BRAKE)
```

2

3

- 1) Deasupra se află de fapt un cod cadru pentru ce am descris anterior.  
Este necesar pentru a seta programul.
- 2) Motorul pornește pentru 3 secrete la 500mm/sec
- 3) Oprește robotul și frânează.

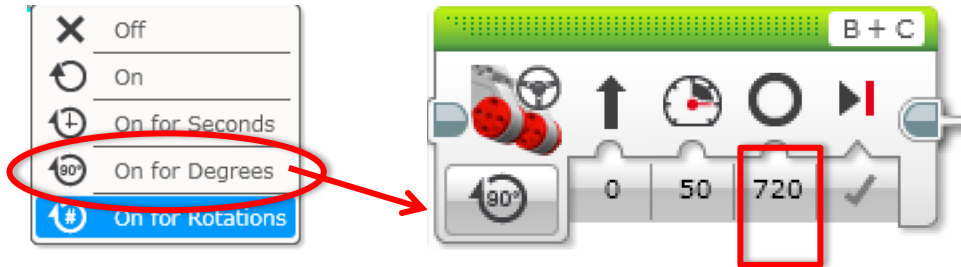
# CUM SĂ MERGEM O ANUMITĂ DISTANȚĂ?

```
# this will reset the rotation sensor on the left motor
left_motor.reset_angle(0)
# this command reads the left motor's rotation sensor
left_motor.angle()
```

`drive_time()` va opri mișcarea după ce trece perioada de timp specificată. Poți folosi timpul și viteza (în mm/sec) pentru a estima distanța parcursă.

Dacă îți dorești ca motorul să se rotească un anumit număr de grade, va fi nevoie să citești senzorul de rotație și să folosești comenzi suplimentare python pentru asta.

# PROVOCAREA 2: MERGI ÎNAINTE (720 DE GRADE)



Ținta este să scriem un program python, care să facă robotul să meargă înainte pentru 720 de grade înainte și se va opri cu precizie.

Această comandă este similară cu block-ul verde din stânga. Observă puterea block-ului verde nu este în mm/sec ci în \*10 grade/sec, 50 putere este 500 grade/secundă.

# PROVOCAREA 2 SOLUȚIA: MERGI ÎNAINTE (720 DE GRADE)

1

```
#!/usr/bin/env pybricks-micropython
from pybricks import ev3brick as brick
from pybricks.ev3devices import (Motor, TouchSensor, ColorSensor, InfraredSensor, UltrasonicSensor, GyroSensor)
from pybricks.parameters import (Port, Stop, Direction, Button, Color, SoundFile, ImageFile, Align)
from pybricks.tools import print, wait, StopWatch
from pybricks.robotics import DriveBase
```

```
# Initialize two motors with default settings on Port B and Port C.
```

```
left_motor = Motor(Port.B)
right_motor = Motor(Port.C)
# setup wheel diameter and axle_track
wheel_diameter = 56
axle_track = 114
```

```
# setup DriveBase
robot = DriveBase(left_motor, right_motor, wheel_diameter, axle_track)
```

2

```
# first reset the rotation sensor
```

```
left_motor.reset_angle(0)
```

```
# start the robot driving
robot.drive(200, 0)
```

3

```
# use a loop to wait for rotation sensor to reach 720
```

```
while left_motor.angle() < 720:
```

```
    pass
```

```
# stop the motor
```

```
robot.stop(Stop.BRAKE)
```

- 1) Acesta este același cadru ca în codul utilizat mai devreme.
- 2) Resetează senzorul de rotație și începe mișcarea. Observă că spre deosebire de EV3, nu poți face asta într-o ordine inversă.
- 3) Așteaptă ca rotația să ajungă la 720 de grade și apoi se oprește. Observă că această așteptare este implementată în buclă.

# NOTĂ

- Nu poți reseta un senzor de rotație în timp ce robotul se mișcă utilizând clasa „drivebase”. De exemplu, schimbarea resetării senzorului de rotație cu comanda `robot.drive()` în provocarea 2 va face ca robotul să se miște la infinit .
- DriveBase nu este complet echivalentul block-urilor verzi din EV3-G. Îi lipsește:
  - Sincronizarea roților. Dacă ții o roată cu mâna cealaltă continuă să se rotească.
  - Accelerare/Decelerare. Block-uri verzi includ accelerație/decelerație pentru Move for degrees/rotations pentru a îmbunătăți acuratețea. Este încă neclar dacă DriveBase face asta.

# CREDITS

Această lecție de Mindstorms a fost realizată de Sanjay Seshan și Arvind Seshan.

Mai multe lecții sunt disponibile pe [ev3lessons.com](http://ev3lessons.com)

Această lecție a fost tradusă în limba română de echipa de robotică FTC – ROSOPHIA #21455 RO20.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).