Università Politecnica delle Marche

Eng. Lorenzo Cesaretti

Dipartimento di Ingegneria dell’Informazione (DII)

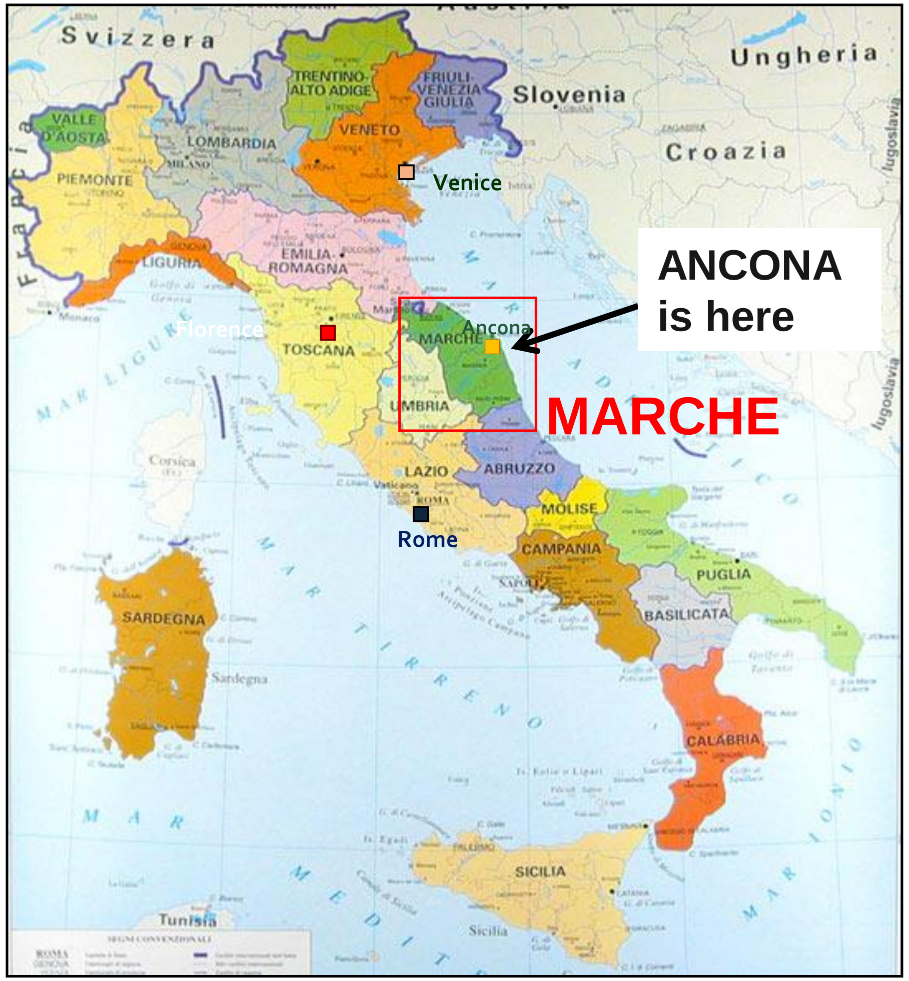
Facoltà di Ingegneria

Lorenzo Cesaretti

PhD Student @ Univpm
CTO TALENT srl

l.cesaretti@pm.univpm.it
lorenzo.cesaretti@weturtle.org

My research papers:
https://www.researchgate.net/profile/Lorenzo_Cesaretti/research



LAND of:

- Poets (Leopardi)
- Teachers (Montessori)
- Painters (Raffaello Sanzio)
- Shoes & Fashion (Tod's)
- Food & Wine (Verdicchio)
- Saints

Laboratory of Modelling, Analysis and Control of Dynamical systems (LabMACS)

Marine Robotics

Competences, skills

- Development of NGC, hw and sw for robotics
- Data acquisition and processing
- 2D, 3D documentation and reconstruction

Equipment for underwater activities

- ROVs, ASV (Deep Ocean PhantomS2, Prometeo Reloaded, VideoRay Pro4)
- High definition 3D cameras, FullHD DV video cameras
- USBL positioning system, Imaging sonars
- Multiparametric probe for water analysis

Educational Robotics

Competences, skills:

- Development of Robotic Tools for Lesson Plans (**Educational Kits**)
- **Real time monitoring** and analysis of students' programming during ER activities (patent application)

Aims:

- incorporating **Robotics in school's curricula** since an early stage of education
- introducing children to STREM and **eSTREM** (environmental Science Technology Robotics Engineering Maths)

My research project @Univpm

Analysis and development of advanced mechatronic devices for Educational Robotics

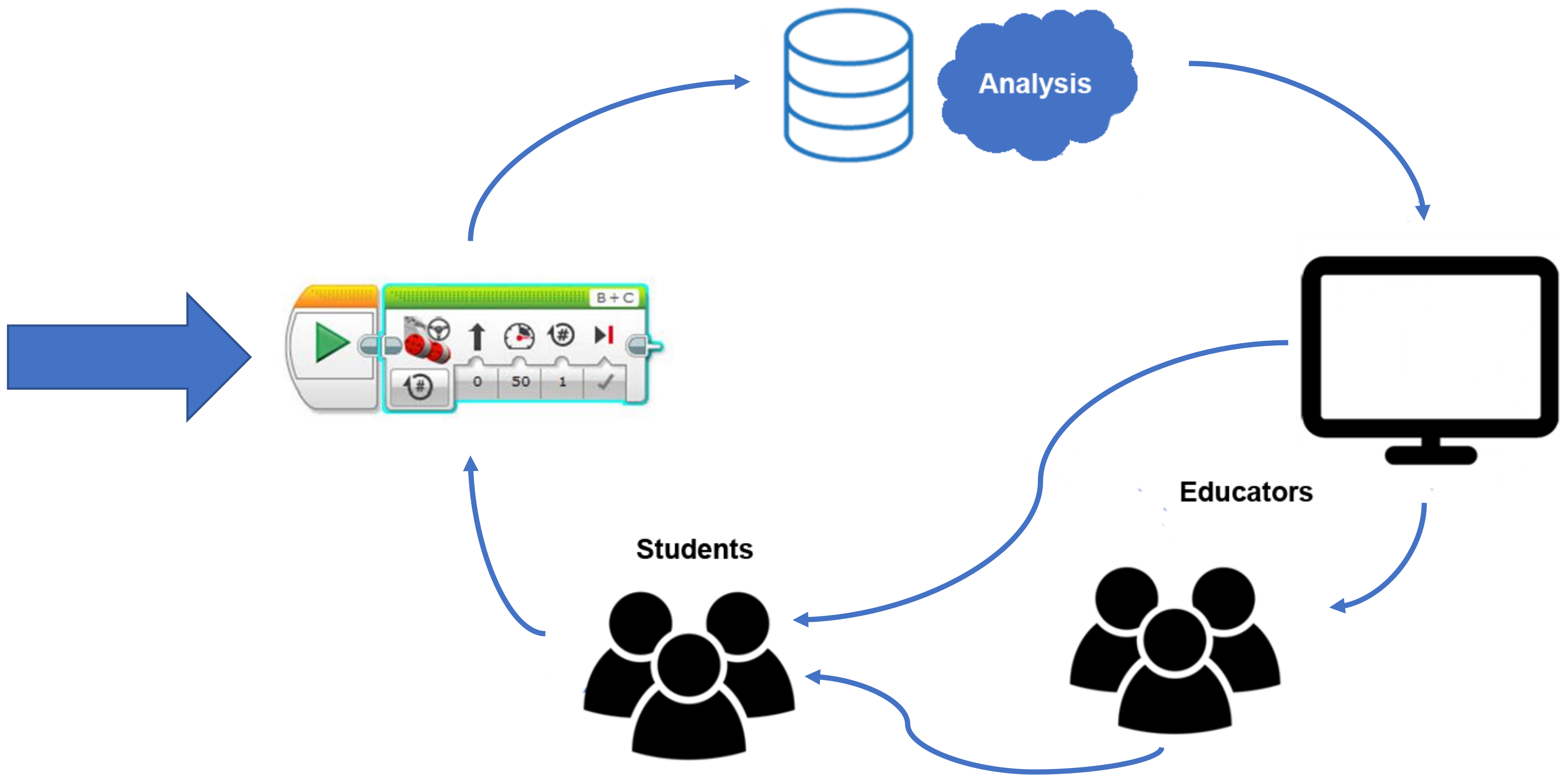
2016/2017

A criticism emerges within the robotics community in recent years claiming that there is **a clear lack of quantitative research** on how robotics can increase learning achievements in students.

- Evaluating the **efficacy** of commercial kits and custom tools (IoT sensors, pattern recognition).
- Assessing students and teachers, **collecting data automatically**, using a statistical approach (IoT sensors, pattern recognition).

My research project @Univpm

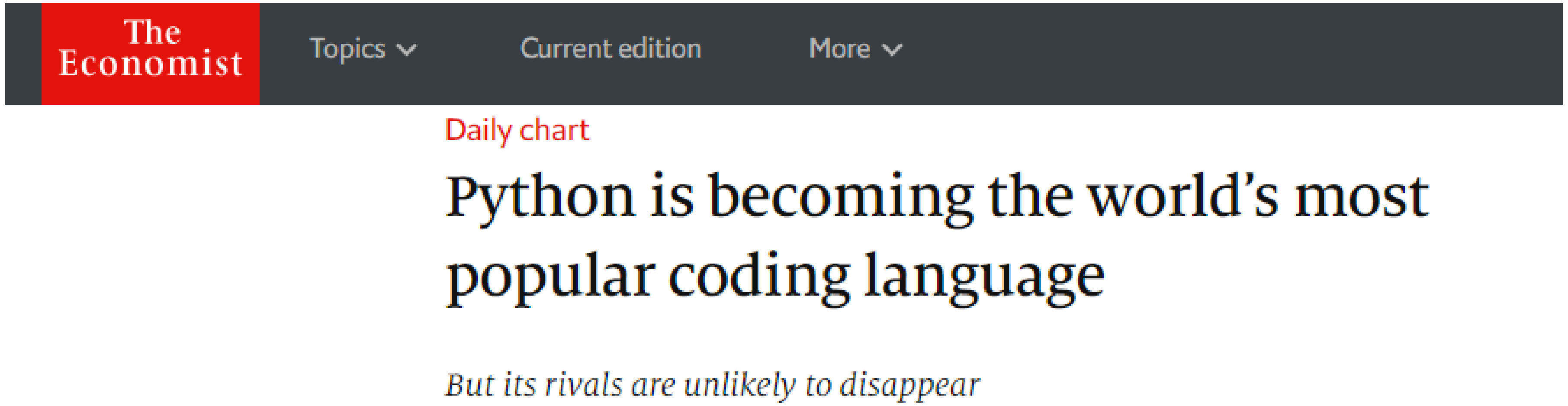
Real time monitoring of students' activity (especially design / programming)



Extract meaningful information from the collected data, using **Machine Learning approach**

Programming the EV3 robot using Python

Why Python?



Daily chart

Python is becoming the world's most popular coding language

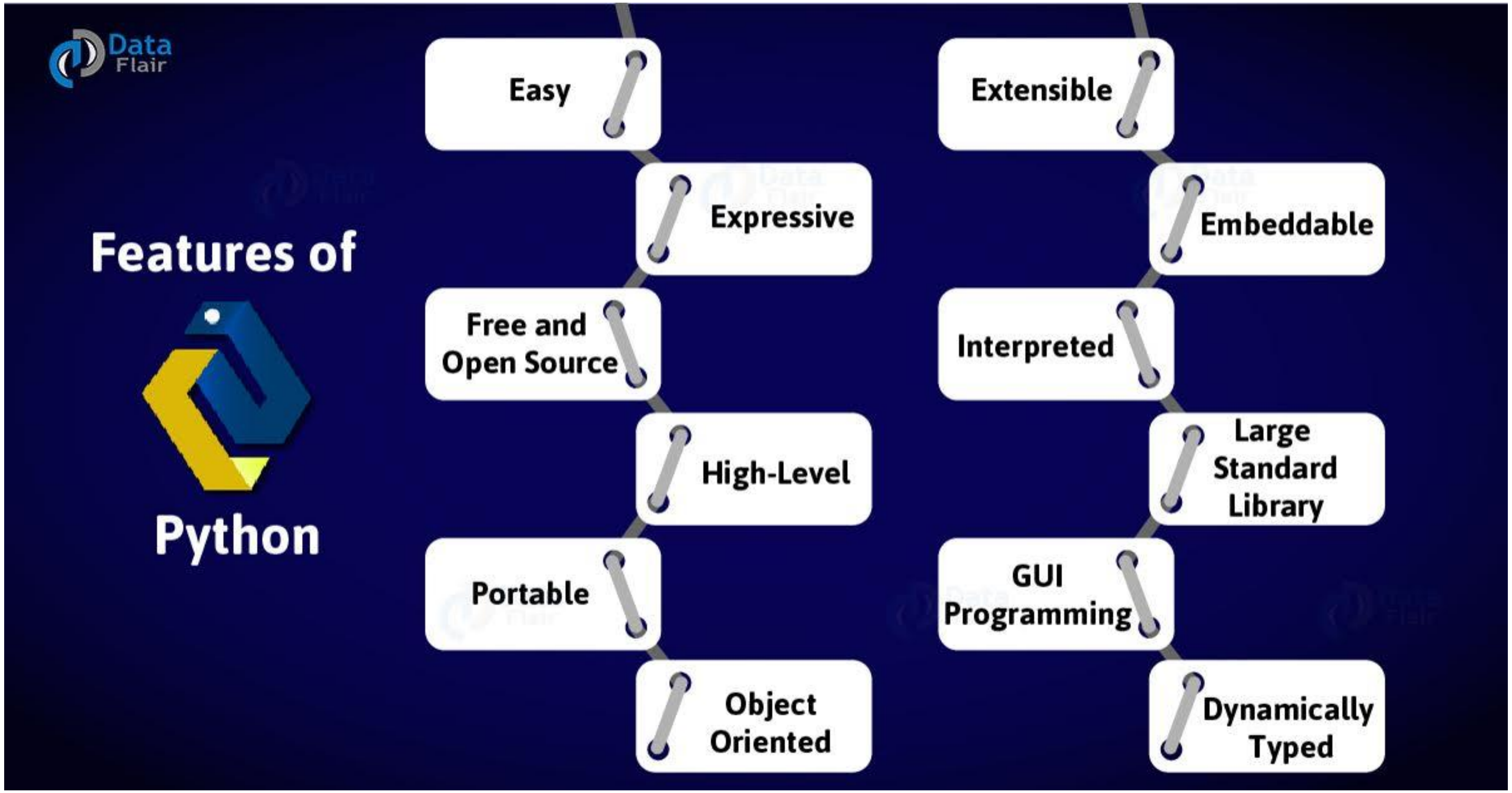
But its rivals are unlikely to disappear

Python is the future of **AI and Machine Learning**! (scikit-learn, Keras, TensorFlow)

https://www.economist.com/graphic-detail/2018/07/26/python-is-becoming-the-worlds-most-popular-coding-language

Programming the EV3 robot using Python

What is Python? Some features!



https://www.quora.com/What-are-the-features-of-Python

Programming the EV3 robot using Python

Setting up Visual Studio Code for EV3

Follow the tutorial at:
https://sites.google.com/site/ev3devpython/setting-up-vs-code

Or open the document in your moodle: **setting-up-ev3python.pdf**

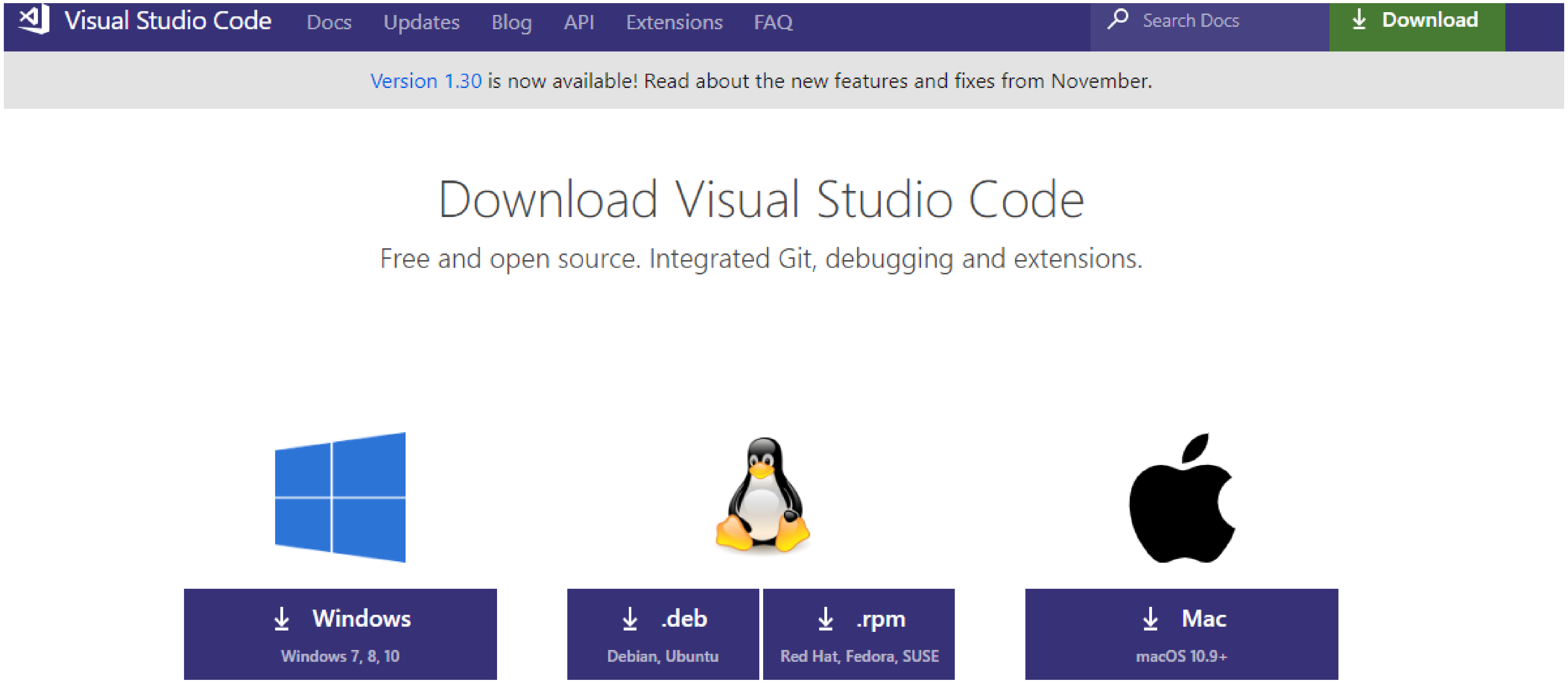
START FROM POINT 6 (“Download and install Microsoft Visual Studio Code” **)**

Programming the EV3 robot using Python

Download and install **Microsoft Visual Studio Code** (VS Code).

This is a free multiplatform code editor, compatible with Windows, Mac OS and Linux.

https://code.visualstudio.com/Download



Version 1.30 is now available! Read about the new features and fixes from November.

Download Visual Studio Code

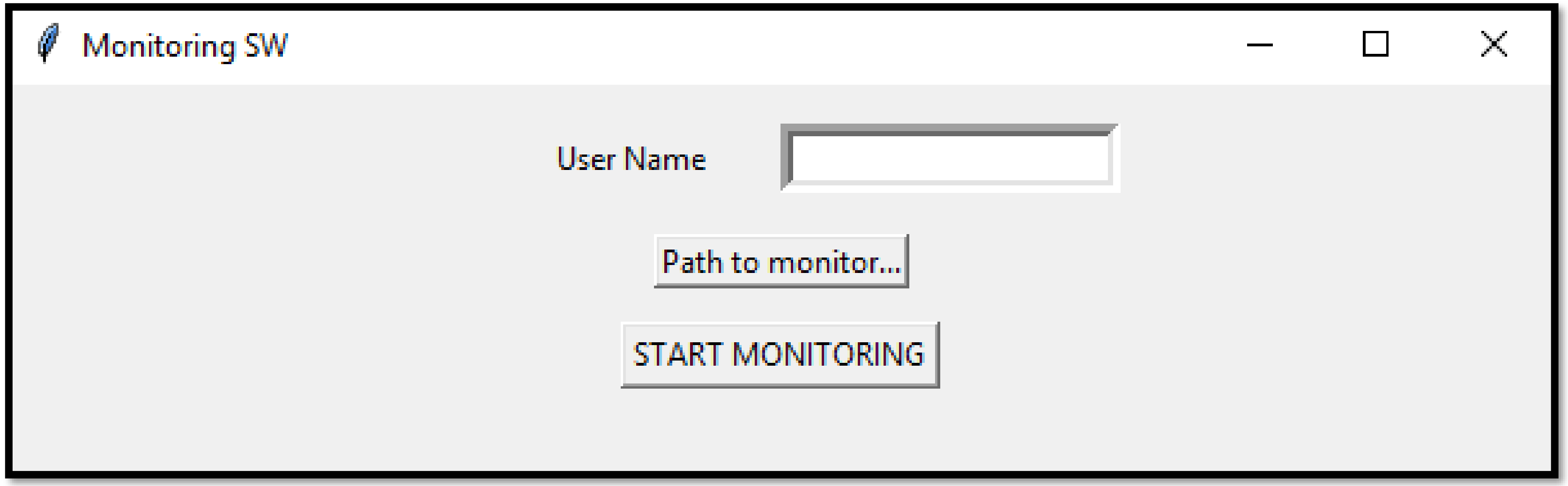
Free and open source. Integrated Git, debugging and extensions.

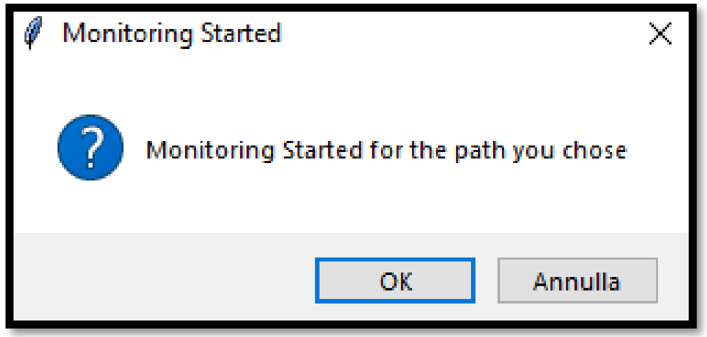
Programming the EV3 robot using Python

Starting the monitoring script

For our research project is **VERY IMPORTANT** that you start the script before any programming activities (of this week!)

1. Insert your user name
2. Click on START MONITORING and select the path where you will save all your programming files
3. Click OK and start to program in Visual Studio Code!





Monitoring Started for the path you chose

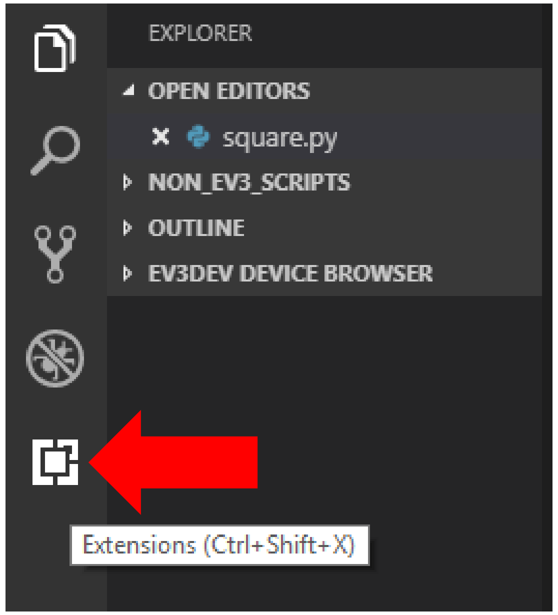
Programming the EV3 robot using Python

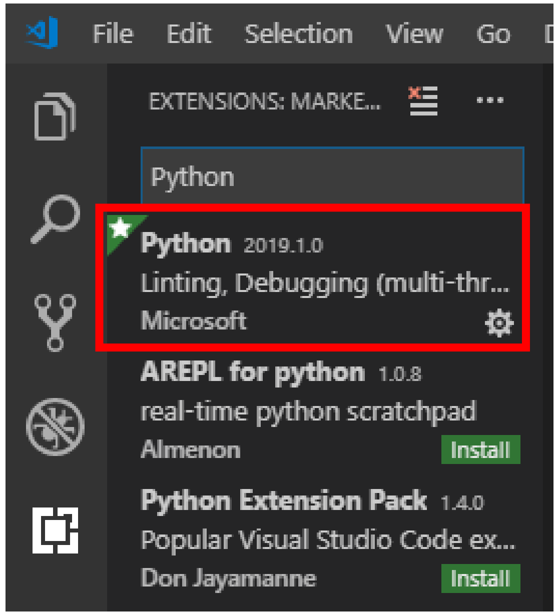
Download and unzip the starter EV3 python project

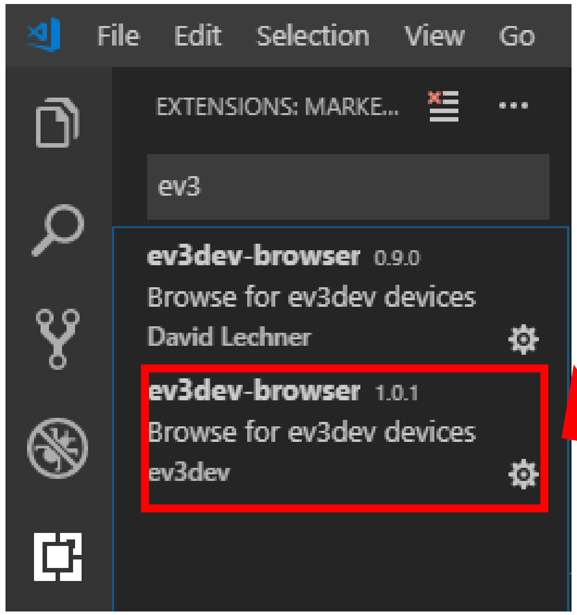
In order to be able to run your EV3 Python scripts on the EV3 from VS Code you MUST have open in VS Code a folder that contains your script and certain other files. **Download** the starter project called **vscode-hello-python:**

https://github.com/ev3dev/vscode-hello-python

Download Python and EV3dev browser extension





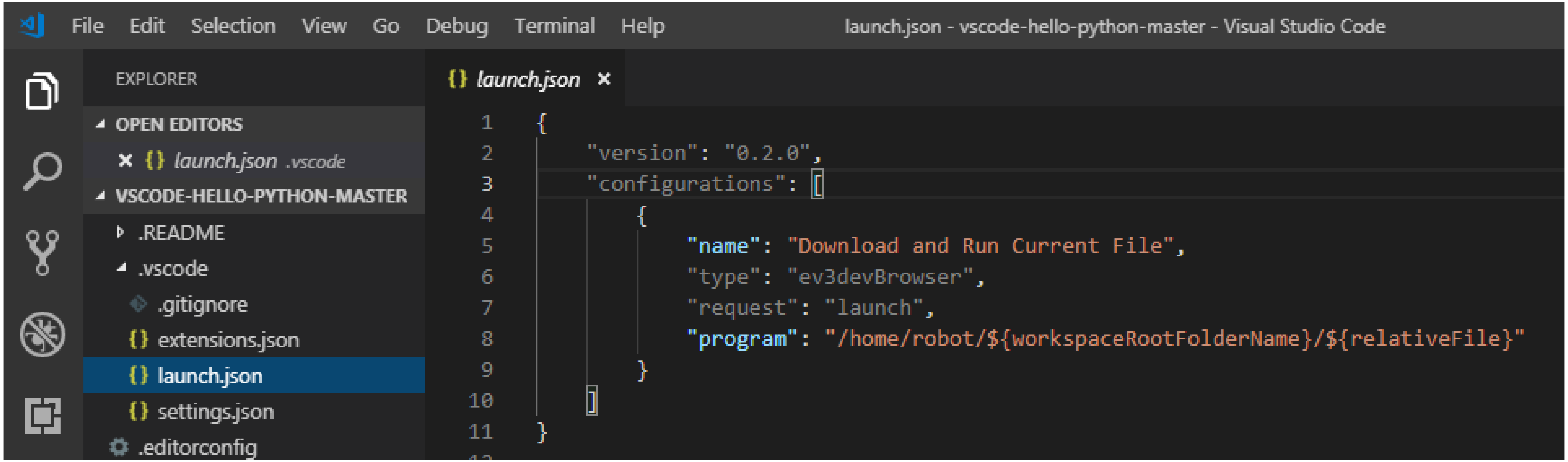


Last version of the extension

Programming the EV3 robot using Python

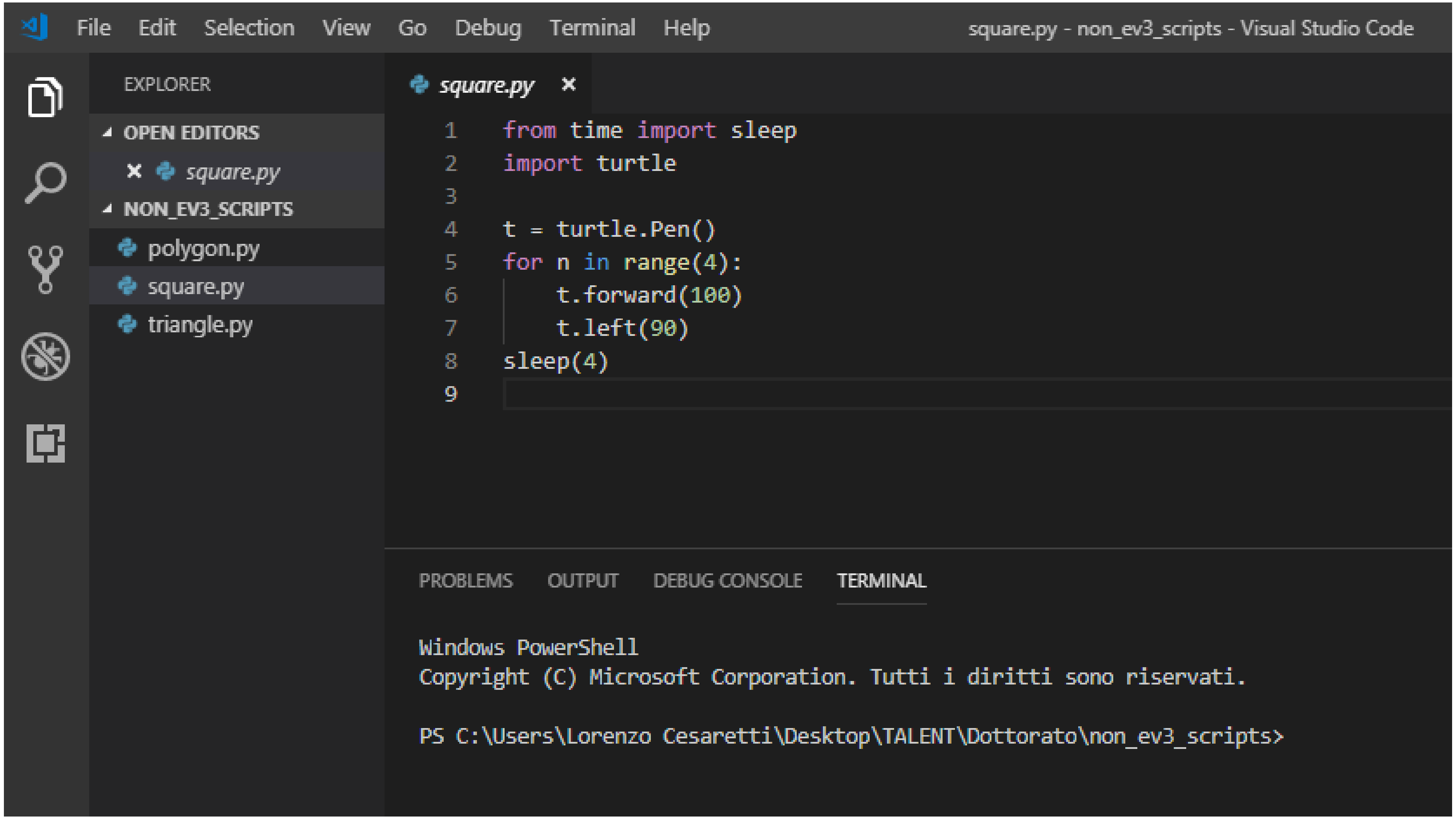
Last step

Open **launch.json** (in the .vscode) and check the code: it must be like this!



```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Download and Run Current File",
            "type": "ev3devBrowser",
            "request": "launch",
            "program": "/home/robot/${workspaceRootFolderName}/${relativeFile}"
        }
    ]
}
```

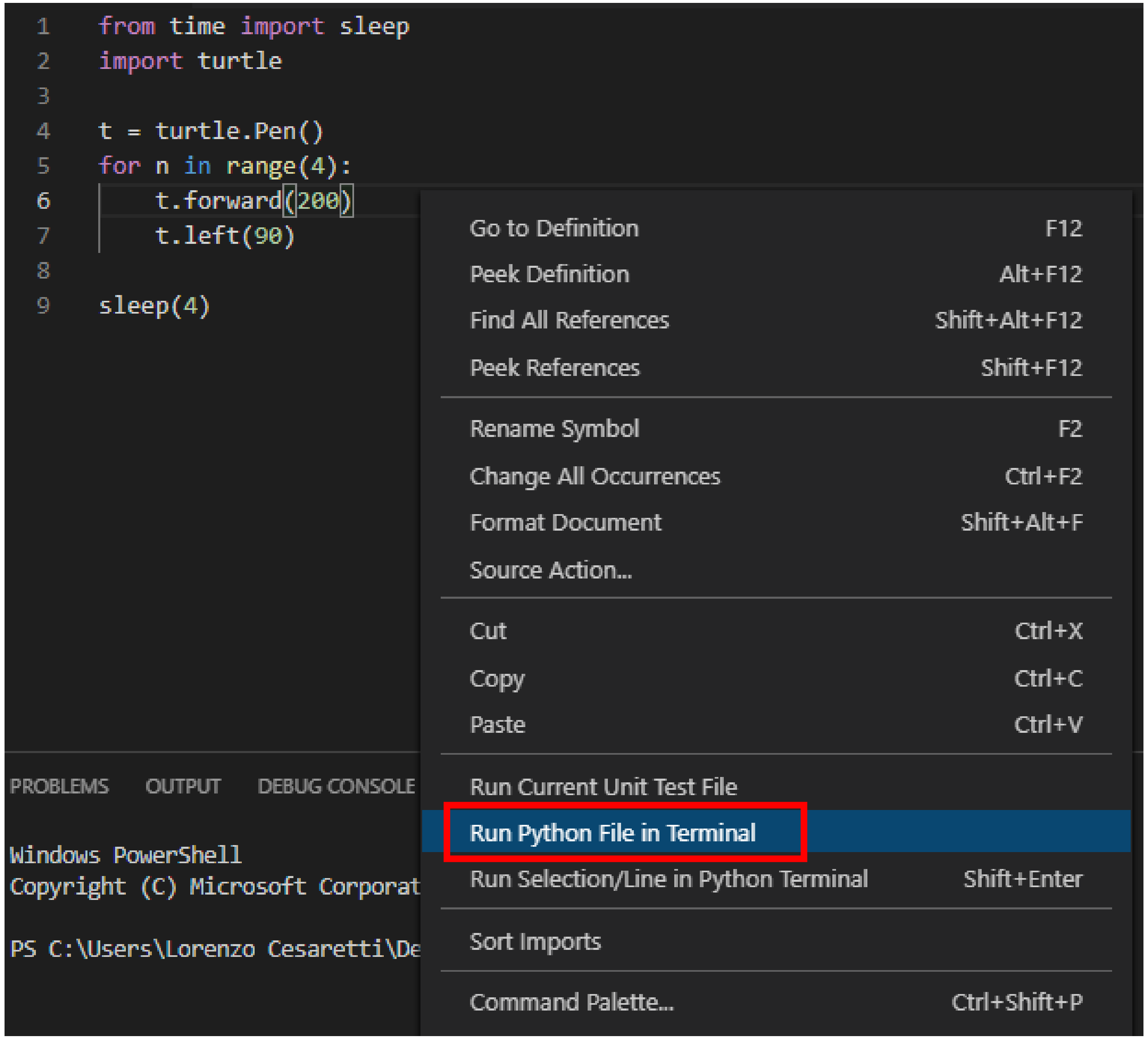
Microsoft Visual Studio Code – A first Python example



```python
from time import sleep
import turtle

t = turtle.Pen()
for n in range(4):
    t.forward(100)
    t.left(90)
sleep(4)

```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tutti i diritti sono riservati.

PS C:\Users\Lorenzo Cesaretti\Desktop\TALENT\Dottorato\non_ev3_scripts>
```

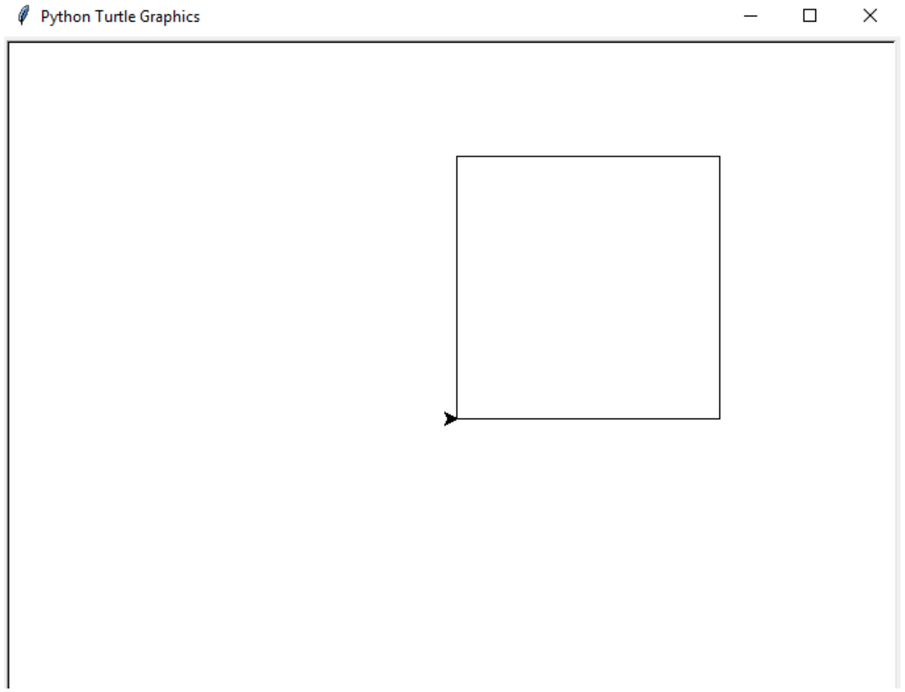
Microsoft Visual Studio Code – A first Python example



```
from time import sleep
import turtle

t = turtle.Pen()
for n in range(4):
    t.forward(200)
    t.left(90)

sleep(4)
```

Go to Definition F12
Peek Definition Alt+F12
Find All References Shift+Alt+F12
Peek References Shift+F12
Rename Symbol F2
Change All Occurrences Ctrl+F2
Format Document Shift+Alt+F
Source Action...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Run Current Unit Test File
Run Python File in Terminal
Run Selection/Line in Python Terminal Shift+Enter
Sort Imports
Command Palette... Ctrl+Shift+P
PROBLEMS
OUTPUT
DEBUG CONSOLE

```
Windows PowerShell
Copyright (C) Microsoft Corporat

PS C:\Users\Lorenzo Cesaretti\De
```



Python flow control

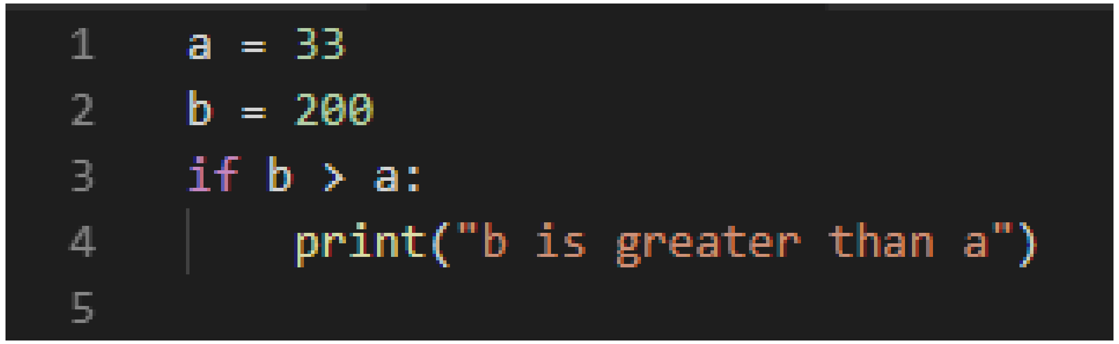
If – statement

```
if condition :
  instructions
  …
```

```
if condition :
  instructions
  …
elif condition :
  instructions
  …
```

```
if condition :
  instructions
  …
elif condition :
  instructions
  …
else:
  instructions
  …
```

An example



```
a = 33
b = 200
if b > a:
    print("b is greater than a")

```

https://docs.python.org/3/tutorial/controlflow.html

Python flow control

For – Statement

```
for x in range():
  instructions
  …
```

```
string = “Hello world”
for x in string:
  instructions
  …
```

```
collection = ['hey', 5, 'd’]
for x in collection:
  instructions
  …
```

As the ***for*** loop in Python is so **powerful**, ***while*** is **rarely used**

While

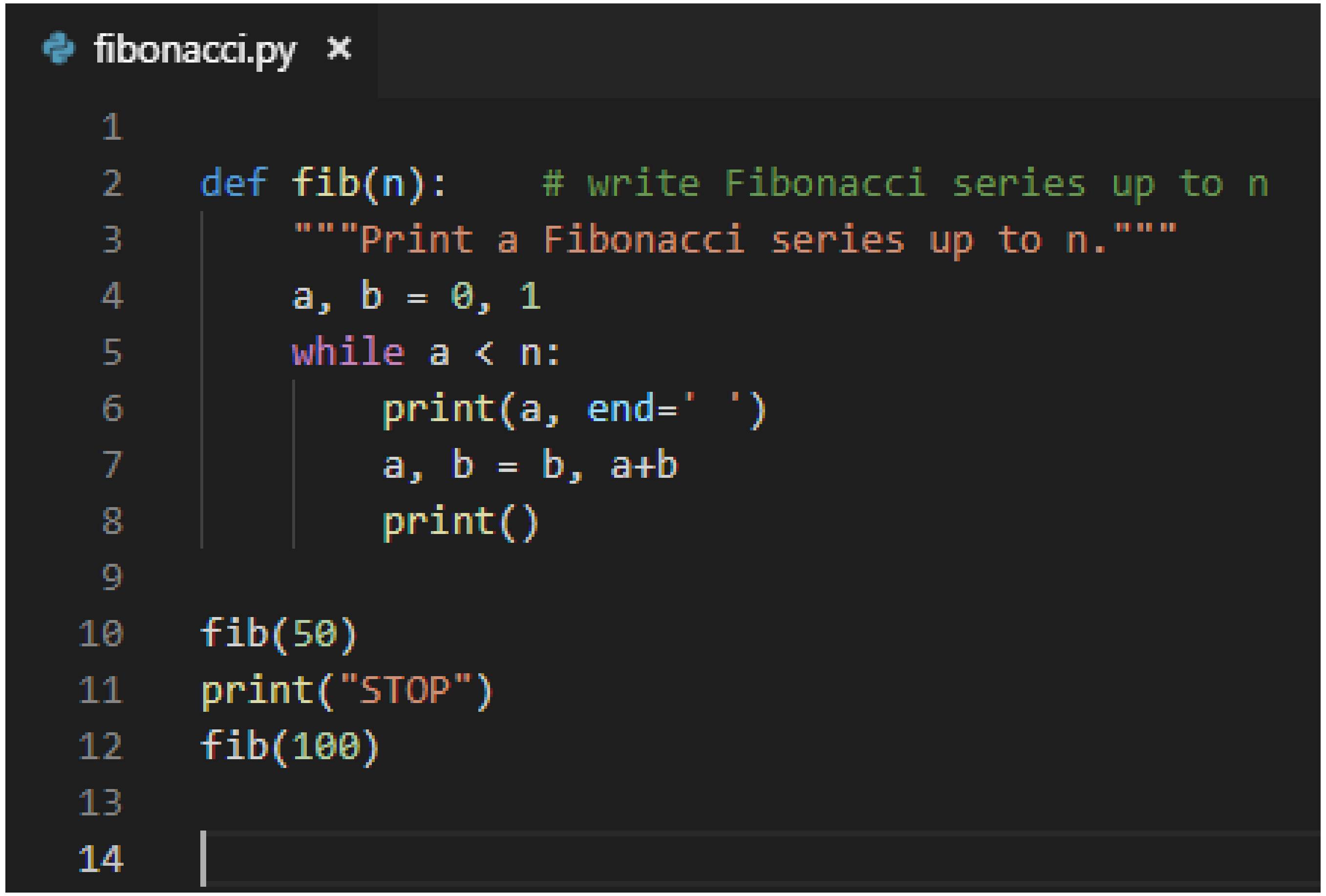
```
x = 1
while True:
    print ("To infinity and beyond! We're getting close, on %d now!" % (x))
    x += 1
```

https://docs.python.org/3/tutorial/controlflow.html

Python flow control

Function

Defining and calling function fib()



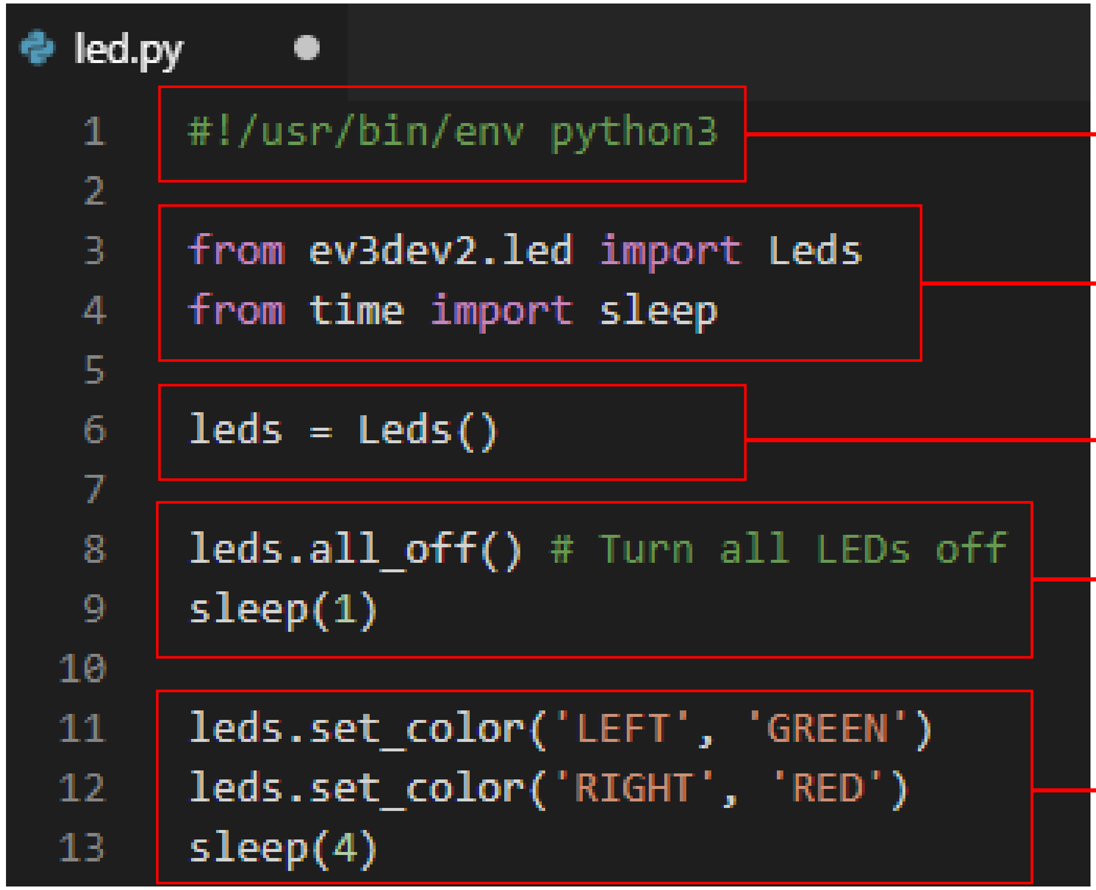
fibonacci.py

```python

def fib(n):    # write Fibonacci series up to n
    """Print a Fibonacci series up to n."""
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
        print()

fib(50)
print("STOP")
fib(100)


```

https://docs.python.org/3/tutorial/controlflow.html

Python with Lego Mindstorms EV3

First example - Leds



```python
#!/usr/bin/env python3

from ev3dev2.led import Leds
from time import sleep

leds = Leds()

leds.all_off() # Turn all LEDs off
sleep(1)

leds.set_color('LEFT', 'GREEN')
leds.set_color('RIGHT', 'RED')
sleep(4)
```

Shebang line: it defines where the interpreter is located.

These lines import Leds and sleep from 2 modules (ev3dev2 and time)

Class instantiation: it creates a new instance of the class Leds and assigns this object to the local variable leds.

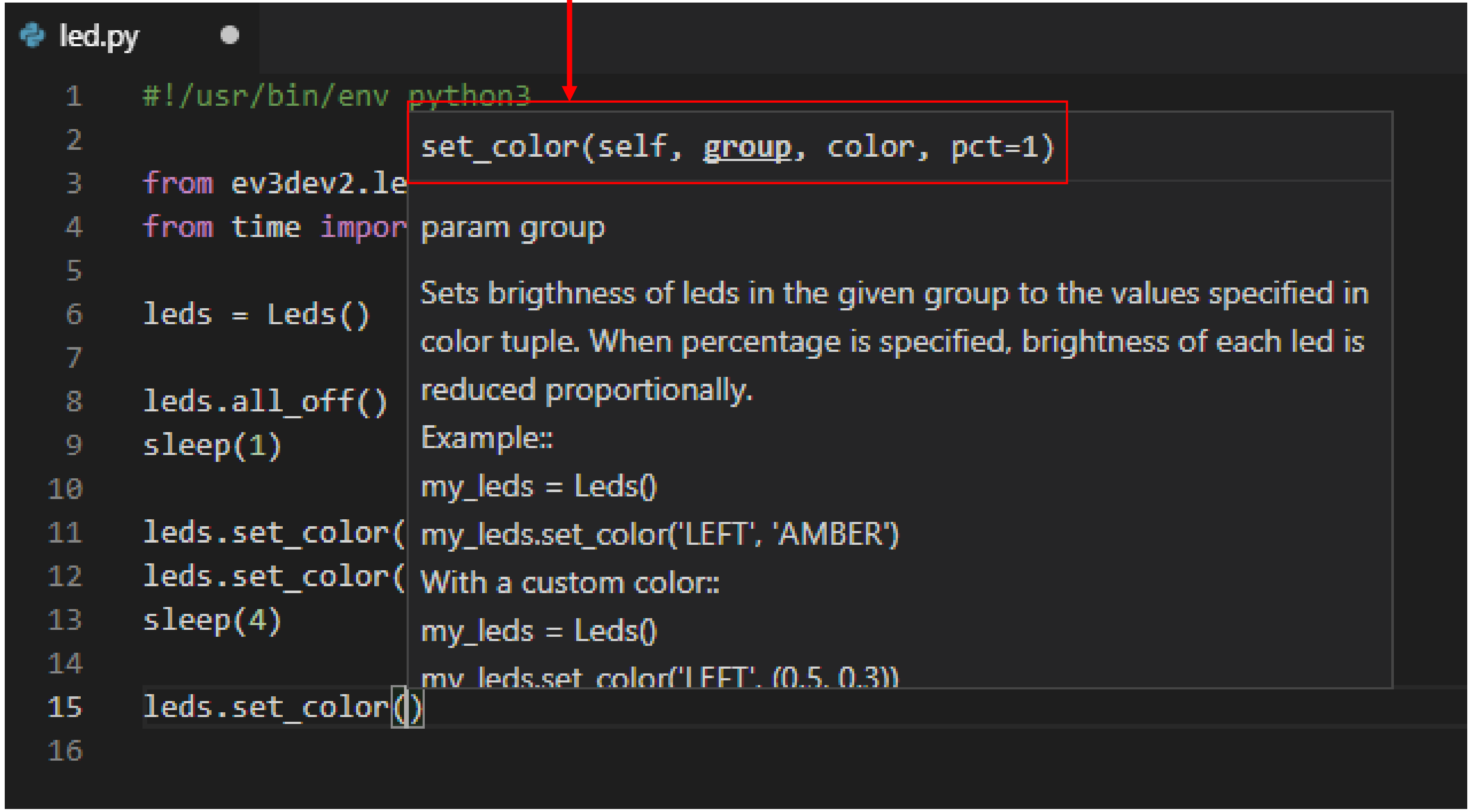
Calling the method "all_off": it is a function that "belongs to" the object Leds; then there is a pause for the CPU (1 s)

Calling the method "set_color": it is a function that "belongs to" the object Leds.

https://sites.google.com/site/ev3devpython/learn_ev3_python/leds

Python with Lego Mindstorms EV3

Visual Studio Code suggests you the **parameters** (and other information) for each function!



led.py

```
#!/usr/bin/env python3

from ev3dev2.le
from time impor

leds = Leds()

leds.all_off()
sleep(1)

leds.set_color(
leds.set_color(
sleep(4)

leds.set_color()
```

```
set_color(self, group, color, pct=1)
```

param group

Sets brigthness of leds in the given group to the values specified in color tuple. When percentage is specified, brightness of each led is reduced proportionally.

Example::

my_leds = Leds()

my_leds.set_color('LEFT', 'AMBER')

With a custom color::

my_leds = Leds()

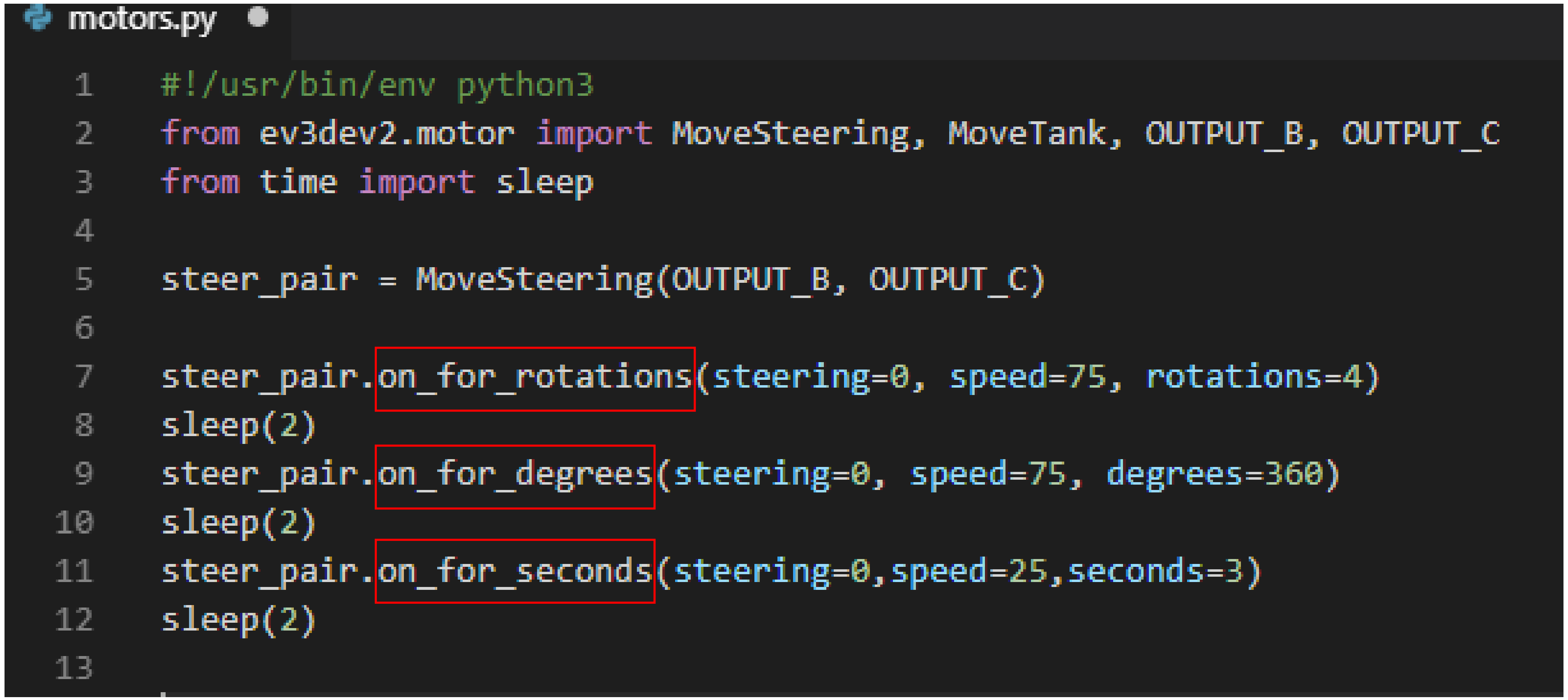
my_leds.set_color('LEFT', (0.5, 0.3))

https://docs.python.org/3/tutorial/controlflow.html

Python with Lego Mindstorms EV3

Motors

How to turn on the motors with the MoveSteering class (3 different methods!)



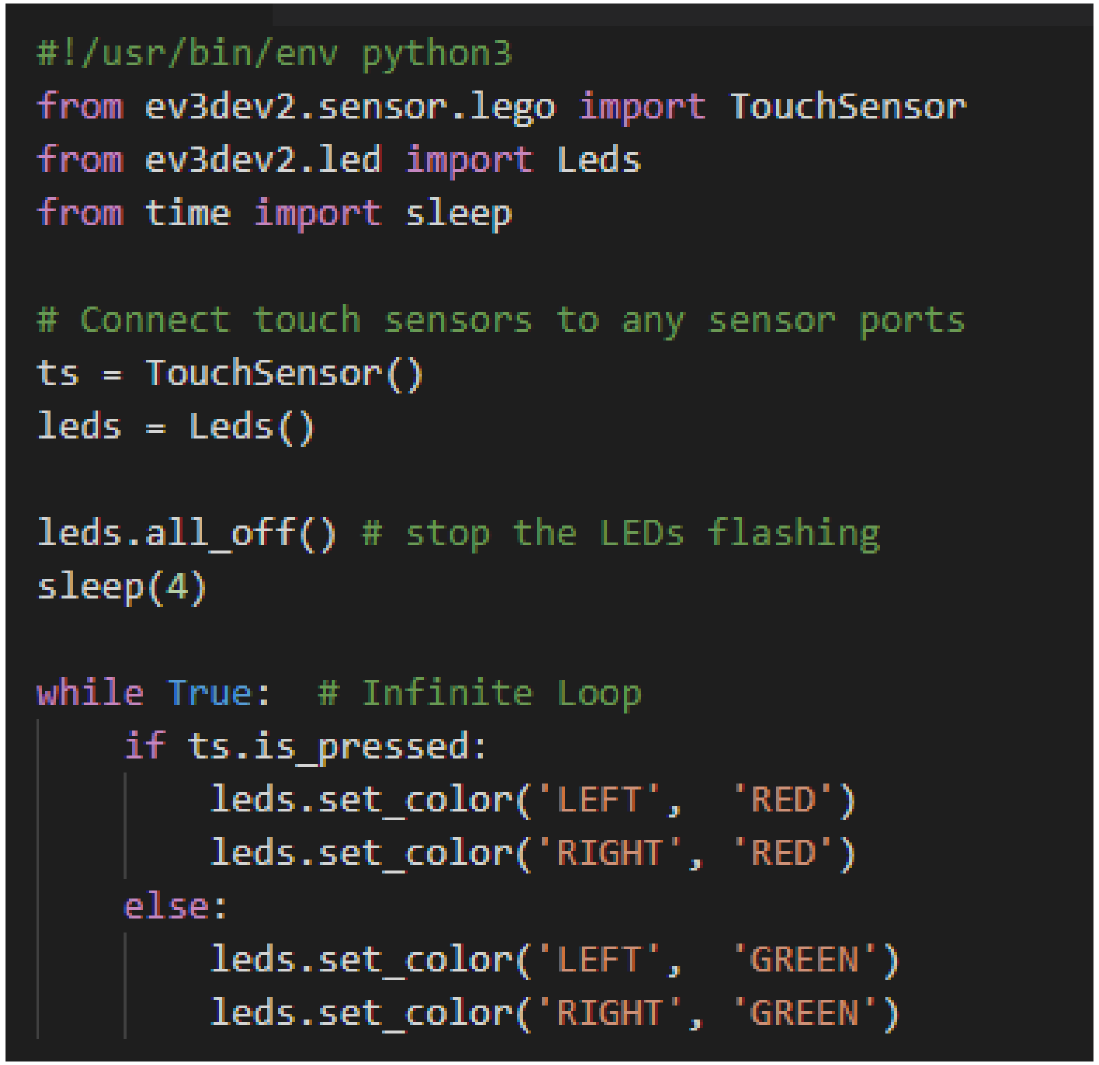
motors.py

```python
#!/usr/bin/env python3
from ev3dev2.motor import MoveSteering, MoveTank, OUTPUT_B, OUTPUT_C
from time import sleep

steer_pair = MoveSteering(OUTPUT_B, OUTPUT_C)

steer_pair.on_for_rotations(steering=0, speed=75, rotations=4)
sleep(2)
steer_pair.on_for_degrees(steering=0, speed=75, degrees=360)
sleep(2)
steer_pair.on_for_seconds(steering=0,speed=25,seconds=3)
sleep(2)

```

https://sites.google.com/site/ev3devpython/learn_ev3_python/using-motors

Python with Lego Mindstorms EV3

Sensors (touch sensor)



```python
#!/usr/bin/env python3
from ev3dev2.sensor.lego import TouchSensor
from ev3dev2.led import Leds
from time import sleep

# Connect touch sensors to any sensor ports
ts = TouchSensor()
leds = Leds()

leds.all_off() # stop the LEDs flashing
sleep(4)

while True:  # Infinite Loop
    if ts.is_pressed:
        leds.set_color('LEFT',  'RED')
        leds.set_color('RIGHT', 'RED')
    else:
        leds.set_color('LEFT',  'GREEN')
        leds.set_color('RIGHT', 'GREEN')
```

https://sites.google.com/site/ev3devpython/learn_ev3_python/using-sensors