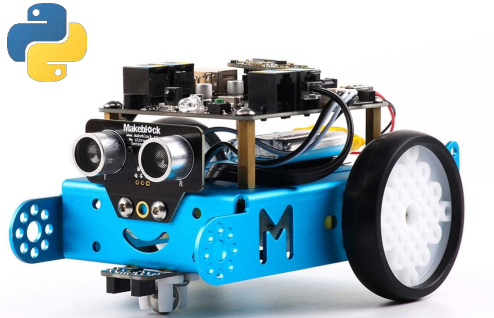




Ready to program your own robot?

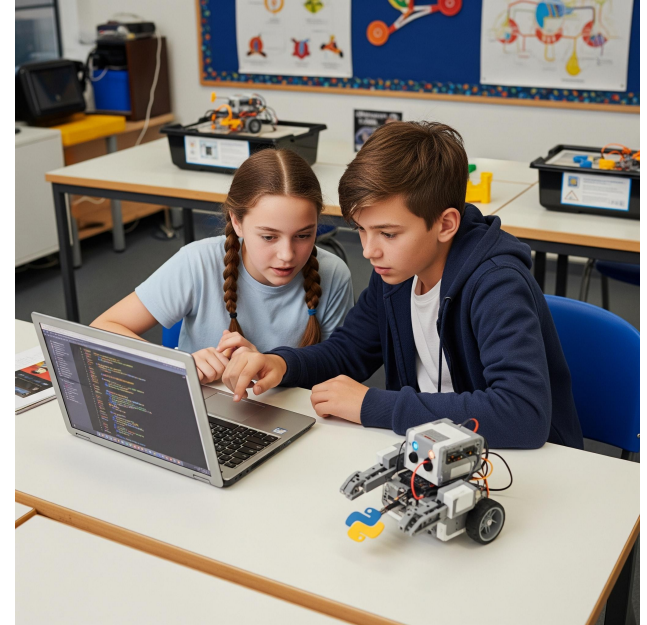
1 robot per pair
(work together)



1 robot per pair

Pair programming - work together

Two heads are better than one



Agenda

- First install some tools together to control a robot
 - Installation tools
 - Python
 - Professional development environment IDE
- Python basics
- Assignments to Program the robot

Python Basics

Basic language constructs are the fundamental **building blocks** that you use in almost any programming language to **write instructions for a computer**.

Think of them as the **grammar rules and basic vocabulary** you need to **form sentences** in a language like **Python**.



1. Variables and Data Types

A variable is like a labeled box where you can store information. You give the box a name and put a value inside it. The data type is the type of information you're storing.

Analogy: Imagine you have storage boxes labeled "Toys," "Books," and "Socks." The label is the variable name, and the items inside are the value.

The most common data types are:

- **String:** Plain text, like a name or a message. You put it in quotes.

```
player_name = "Alex"
```

- **Integer:** A whole number.

```
score = 100
```

- **Float:** A number with a decimal point.

```
pi = 3.14
```

- **Boolean:** Represents one of two values: True or False. It's perfect for tracking things that can be on or off.

```
is_game_over = False
```

2. Operators

Operators are symbols that perform an operation on your variables and values. You already know many of them from math class.

- **Arithmetic Operators:** Used for math.
 - **+** (addition), **-** (subtraction), ***** (multiplication), **/** (division).
Example: `total_score = 100 + 50`
- **Comparison Operators:** Used to compare two values.
The **result** is always a **Boolean** (True or False).
 - **==** (equal to), **!=** (not equal to), **>** (greater than), **<** (less than).
Example: `has_high_score = (score > 1000)`

3. Conditional Statements (If/Else)

Conditionals allow your program to make decisions. They let you run different pieces of code depending on whether a **condition** is **True** or **False**.

- **Analogy:** It's like a fork in the road. **If** the weather is sunny, you go to the park. **Else**, you stay home and watch a movie.

It uses the keywords **if**, **elif** (else if), and **else**.

```
# Check if the player has enough coins to buy an item
if coins >= 50:
    print("You can buy the new sword!")
elif coins >= 20:
    print("You can afford a shield.")
else:
    print("You don't have enough coins.")
```

4. Loops (For/While)

Loops are used to repeat a block of code multiple times, so you don't have to write it out over and over again.

- **Analogy:** It's like telling someone to "take 5 steps forward." You don't say "step, step, step, step, step." You just specify the action and how many times to do it.

There are two main types:

- **for loop:** Repeats code a **specific number of times**.

```
# Prints the numbers 0 through 4
for i in range(5):
    print(i)
```

- **while loop:** Repeats code as long as a certain **condition is True**.

```
# A simple countdown
health = 3
while health > 0:
    print("Player has " + str(health) + " health left.")
    health = health - 1 # Decrease health by 1 each time

print("Game Over!")
```


5. Functions

A **function** is a named, reusable block of code that performs a specific task. You define it once and can then use it (or "call" it) whenever you need it.

- **Analogy:** A function is like a recipe. You write the recipe (the function) once for "Bake a Cake." Then, anytime you want a cake, you just follow the recipe (call the function) without having to figure it out from scratch.
- ```
Define a function to greet a player
def greet_player(name):
 print("Hello, " + name + "! Welcome back.")

Call the function for different players
greet_player("Alex")
greet_player("Bella")
```

# Assignments to Program the robot

Start with the assignments