💻 COMPUTING · LESSON 2 OF 5

Introduction to Python with Karel

Commanding a dog-robot to solve puzzles — your first steps in programming!

🐕 Meet Karel

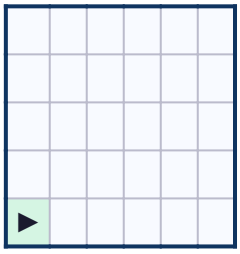
Who is Karel? Karel is a friendly robot dog that lives inside a grid world. You are the programmer — you give Karel instructions and Karel follows them exactly. If you make a mistake, Karel gets confused! That's why we must write our code carefully.

Karel's world is a **grid of squares**. Karel can:

- Move forward one square
- Turn left (90°)
- Pick up and place tennis balls
- Follow functions you define

The ► symbol shows where Karel is and which direction Karel faces. Karel always starts at the **bottom-left** corner facing **East** (right).

START: KAREL FACES RIGHT



► = Karel | ● = Ball | ■ = Wall

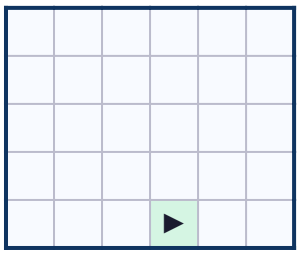
📋 Karel's Basic Commands

Important rule: All Python commands must be written *exactly* right — correct spelling, correct brackets `()`, and no extra spaces. Python is case-sensitive: `Move()` is NOT the same as `move()`!

	Command	What it does	Think of it as…
🚶	`move()`	Karel moves one square forward	Take one step
↺	`turn_left()`	Karel turns 90° to the left	Spin left like a dancer
🎾	`put_ball()`	Karel places a tennis ball on the current square	Drop a ball here
🤏	`take_ball()`	Karel picks up a ball from the current square	Pick it up
↩	`turn_around()`	Karel turns 180° — faces the opposite direction	Do a U-turn

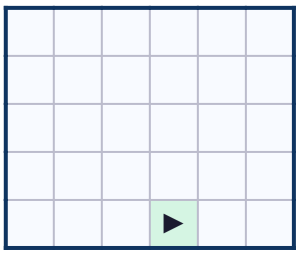
🧪 Example 1 — Moving Karel Across the Grid

AFTER 3 × MOVE()



→

AFTER 3 × MOVE()



solution_example1.py

```
# Move Karel 3 squares to the right
move()    # step 1 → Karel moves to column 1
move()    # step 2 → Karel moves to column 2
move()    # step 3 → Karel moves to column 3
```

Key idea: Each `move()` is one instruction. The computer runs them **one at a time, top to bottom**. This is called *sequential execution*.

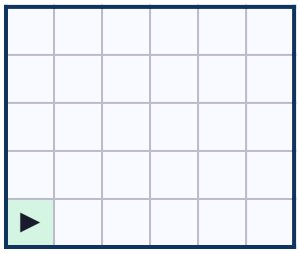
✏️ **Try it yourself!**

What code would move Karel **5 squares** to the right? Write it out on paper before typing it!

🎾 Example 2 — Placing Tennis Balls

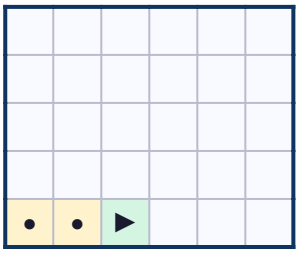
Challenge: Karel starts facing right at the left side. Place a tennis ball on **each of the first 3 squares** Karel visits.

START: KAREL FACES RIGHT



→

AFTER PLACING 3 BALLS



solution_example2.py

```
# Place a ball, move, repeat...
put_ball()  # drop ball on square 0
move()      # step to square 1
put_ball()  # drop ball on square 1
move()      # step to square 2
put_ball()  # drop ball on square 2
```

usually a smarter way — called a *loop*. We'll learn loops in Lesson 5! 🔥

✏️ **Try it yourself!**

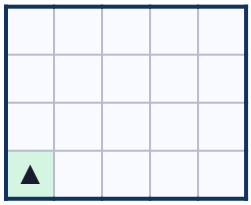
How would you pick up all 3 balls Karel just placed? Hint: use `take_ball()` and `move()`.

⚙️ Creating Your Own Commands — Functions

Problem: Karel can turn *left* with `turn_left()`, but there is **no** `turn_right()` command! How do we make Karel turn right?

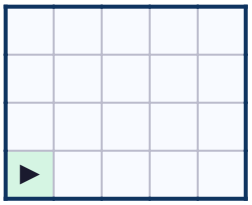
Maths trick: Turning left 3 times = turning right once! (3 × 90° left = 270° left = 90° right)

KAREL FACES NORTH (UP)



→

AFTER TURN_RIGHT(): FACES EAST



turn_right_function.py

```
# Define a new command called turn_right
def turn_right():
    turn_left()  # 1st left turn (90°)
    turn_left()  # 2nd left turn (180°)
    turn_left()  # 3rd left turn (270° = turn right!)

# ── Now we can USE our new command ──
move()       # Karel steps forward
turn_right() # Karel turns right using our function
move()       # Karel steps forward again
```

Rules for writing a function:

1. Start with the keyword `def`
2. Write the function name (no spaces!)
3. Add brackets `()` and a colon `:`
4. Indent the body by 4 spaces — Python is strict about this!
5. Call the function anywhere in your code by writing its name followed by `()`

📝 Summary

✅ What you learned today

- Karel is a robot dog that lives in a grid — you control it with Python commands
- `move()` moves Karel one square forward in the direction it faces
- `turn_left()` rotates Karel 90° to the left
- `put_ball()` and `take_ball()` interact with tennis balls
- Python runs instructions **one line at a time**, top to bottom
- You can create **your own commands** using `def function_name():`
- Indentation (4 spaces) inside a function is **not optional** — Python needs it!
- Turning left 3 times = turning right once (a clever trick!)

🎬 Watch & Learn — Recommended Videos

1. Karel the Dog — Introduction (CS106A Stanford)
https://www.youtube.com/watch?v=d8RRE2rDiEg

2. Python with Karel — CodeHS Tutorial
https://codehs.com/course/intro_python_karel

3. Interactive Karel Exercises
https://stanford.edu/~cpiech/karel/ide.html

Ask a parent or teacher to help you open these links on a computer. Try the interactive Karel IDE — you can run your code and watch Karel move!