PAOLO PERROTTA

B3844 - PYTHON FOR BUSINESS LAB

2025/26

DAY 1

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ABOUT THIS COURSE

- **Lessons every Friday at 13**
- **5 lessons, 3 hours each including breaks**
- **Lessons 2 to 5 should be in Via Zamboni, 34 (but check!)**
- **You’ll probably need to exercise at home**
- **My mail:** `paolo.perrotta2@unibo.it`

ABOUT ME

- Programmer, author, teacher

ABOUT YOU

- You don’t need to know anything about programming
- You *do* need to feel confident using a computer

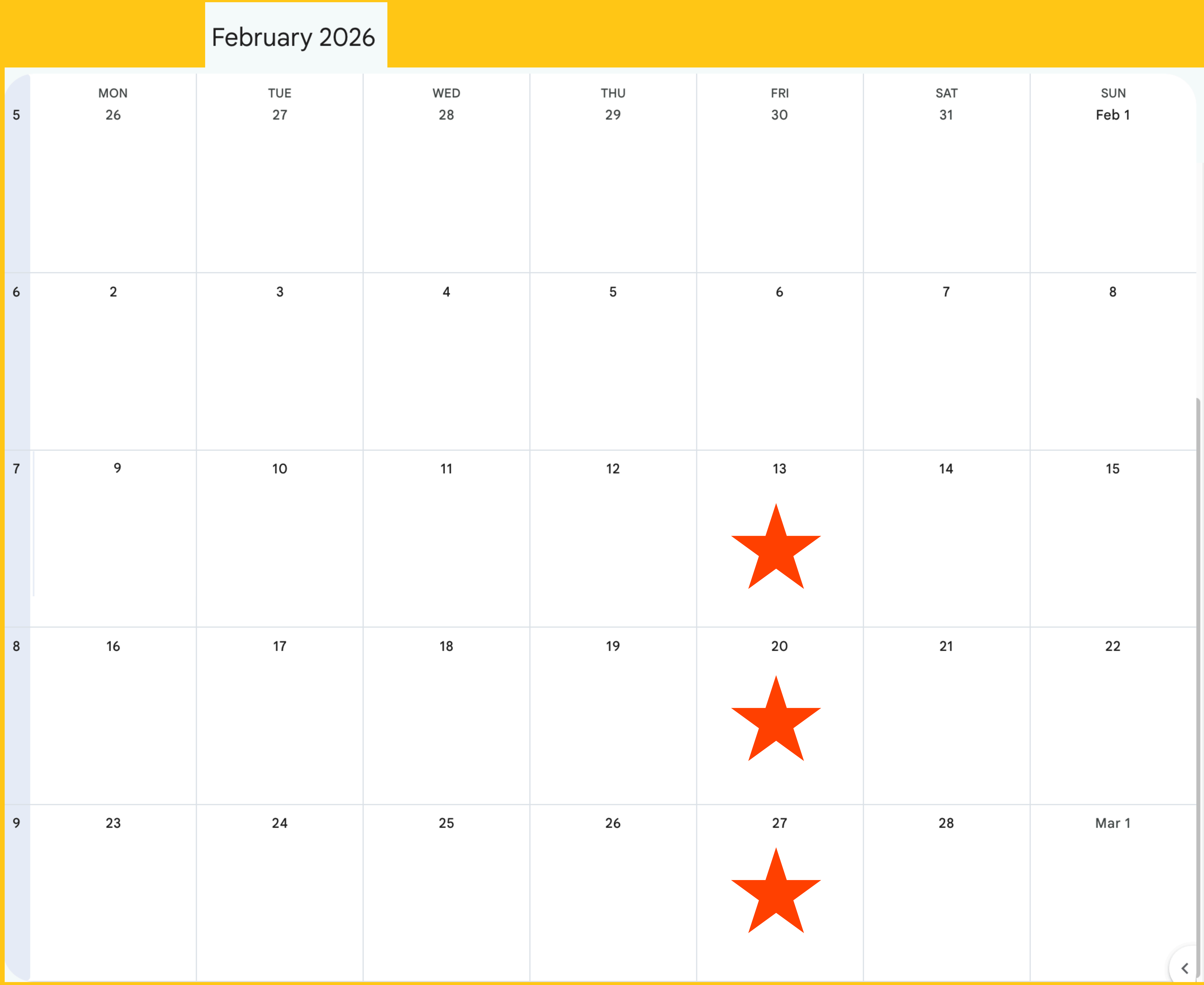
ABOUT THE EXAM

- **Multiple choice questions, pass/fail**
- **Sessions: March 30 / May 25 / August 31**
- **For details and subscriptions, check *AlmaEsami***

→

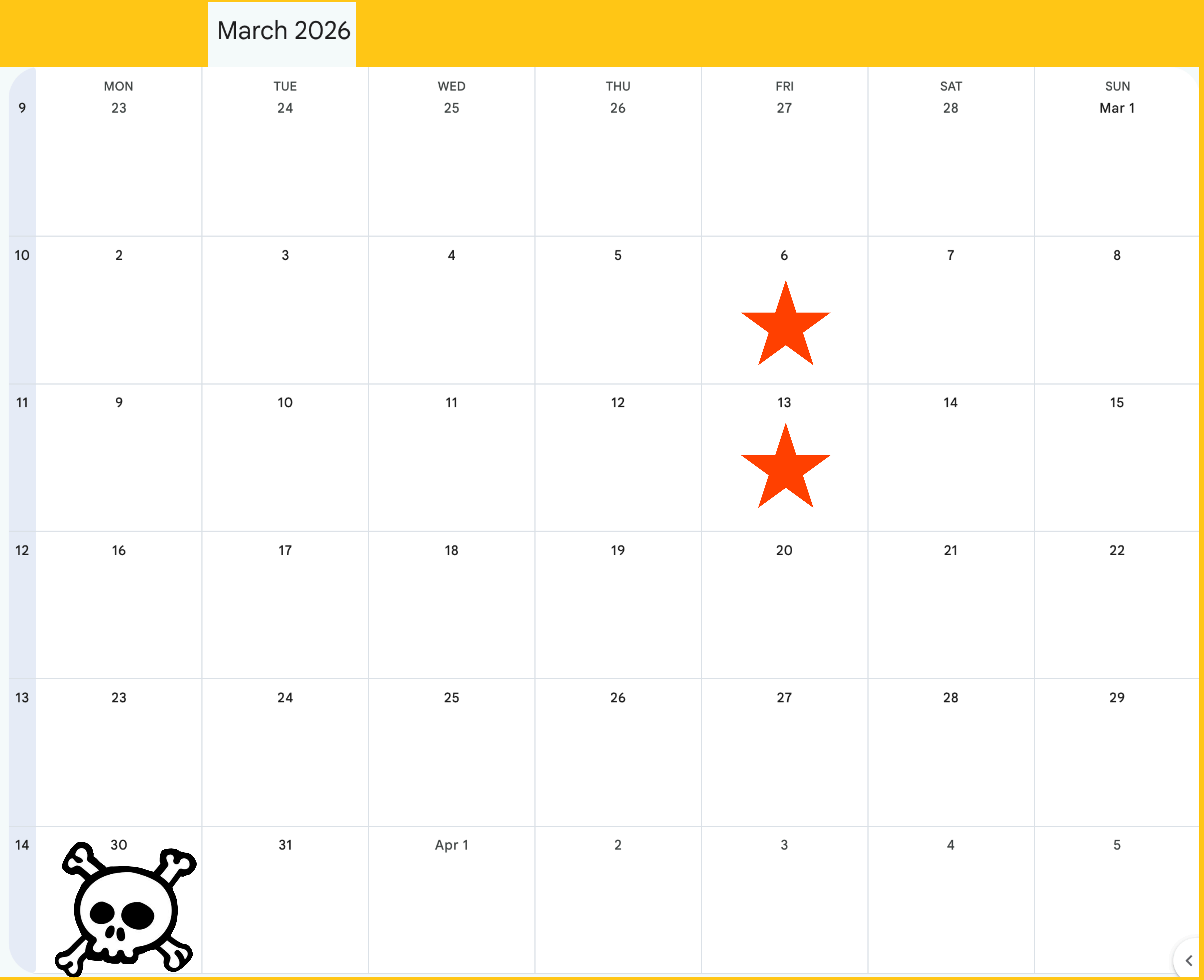
COURSE CALENDAR

aaaa



February 2026

	MON	TUE	WED	THU	FRI	SAT	SUN
5	26	27	28	29	30	31	Feb 1
6	2	3	4	5	6	7	8
7	9	10	11	12	13 ★	14	15
8	16	17	18	19	20 ★	21	22
9	23	24	25	26	27 ★	28	Mar 1



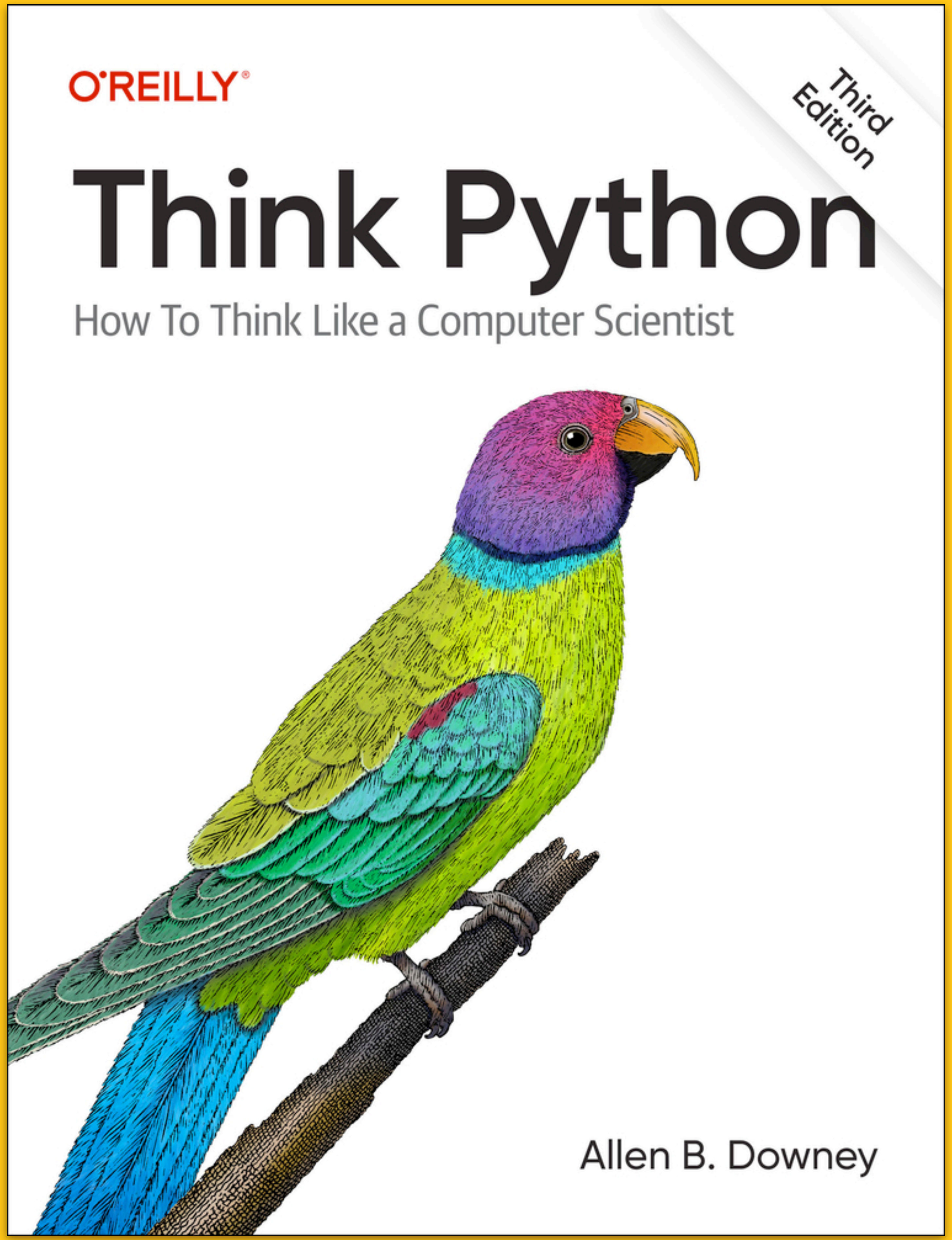
March 2026

	MON	TUE	WED	THU	FRI	SAT	SUN
9	23	24	25	26	27	28	Mar 1
10	2	3	4	5	6 ★	7	8
11	9	10	11	12	13 ★	14	15
12	16	17	18	19	20	21	22
13	23	24	25	26	27	28	29
14	30 ☠	31	Apr 1	2	3	4	5

LINKS

- **Notebooks repository:** `https://github.com/nusco/python-unibo`
- **Calendar, program, etc.:** `http://tiny.cc/unibo-python`
- **Course material on Virtuale:** `https://virtuale.unibo.it`
- **Exams:** `https://almaesami.unibo.it`

AN (OPTIONAL) TEXTBOOK



O'REILLY

Third Edition

Think Python

How To Think Like a Computer Scientist

Allen B. Downey

ABOUT THE TEXTBOOK

- The slides aren’t detailed enough to study on your own
- You can find your own online Python resources
- If you want a book, download *“Think Python”* for free (1st Edition)
- If you want a dead trees book, buy *“Think Python”* on paper

THINGS YOU SHOULD DO

- Attend the lessons
- Listen and (maybe) follow along
- Complete activities
- Help each other
- Ask questions, especially if you’re stuck on an exercise
- Exercise at home
- Have fun

INTRODUCTION: ABOUT PYTHON

RUNNING PYTHON

- The REPL
- Computational notebooks
- Text editors / Integrated Development Environments

CHECK OUT THE REPL

- **Log in to your computer**
- **Launch the Anaconda prompt**
- `python --version`

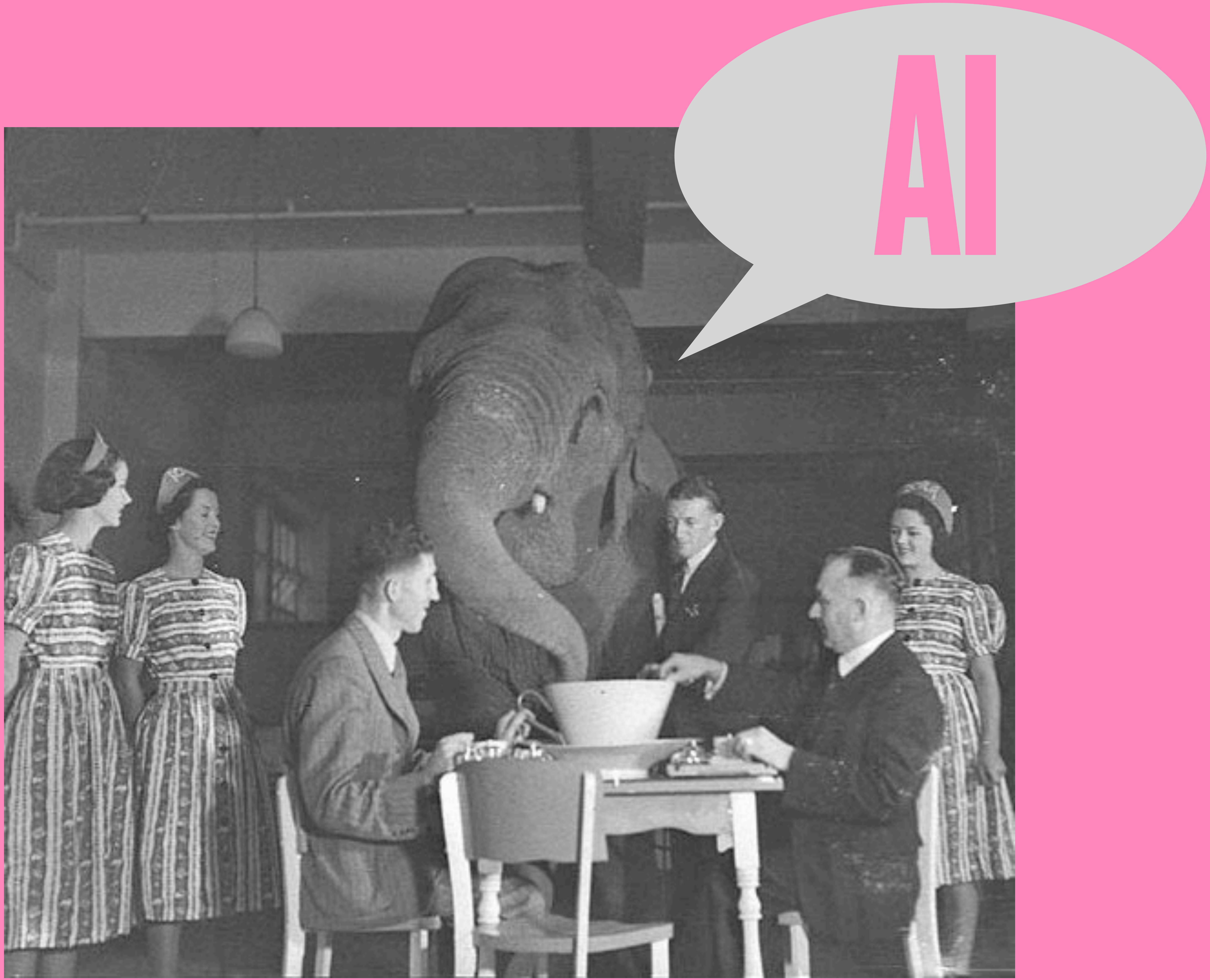
NOTEBOOK: EXAMPLE NOTEBOOK

10’

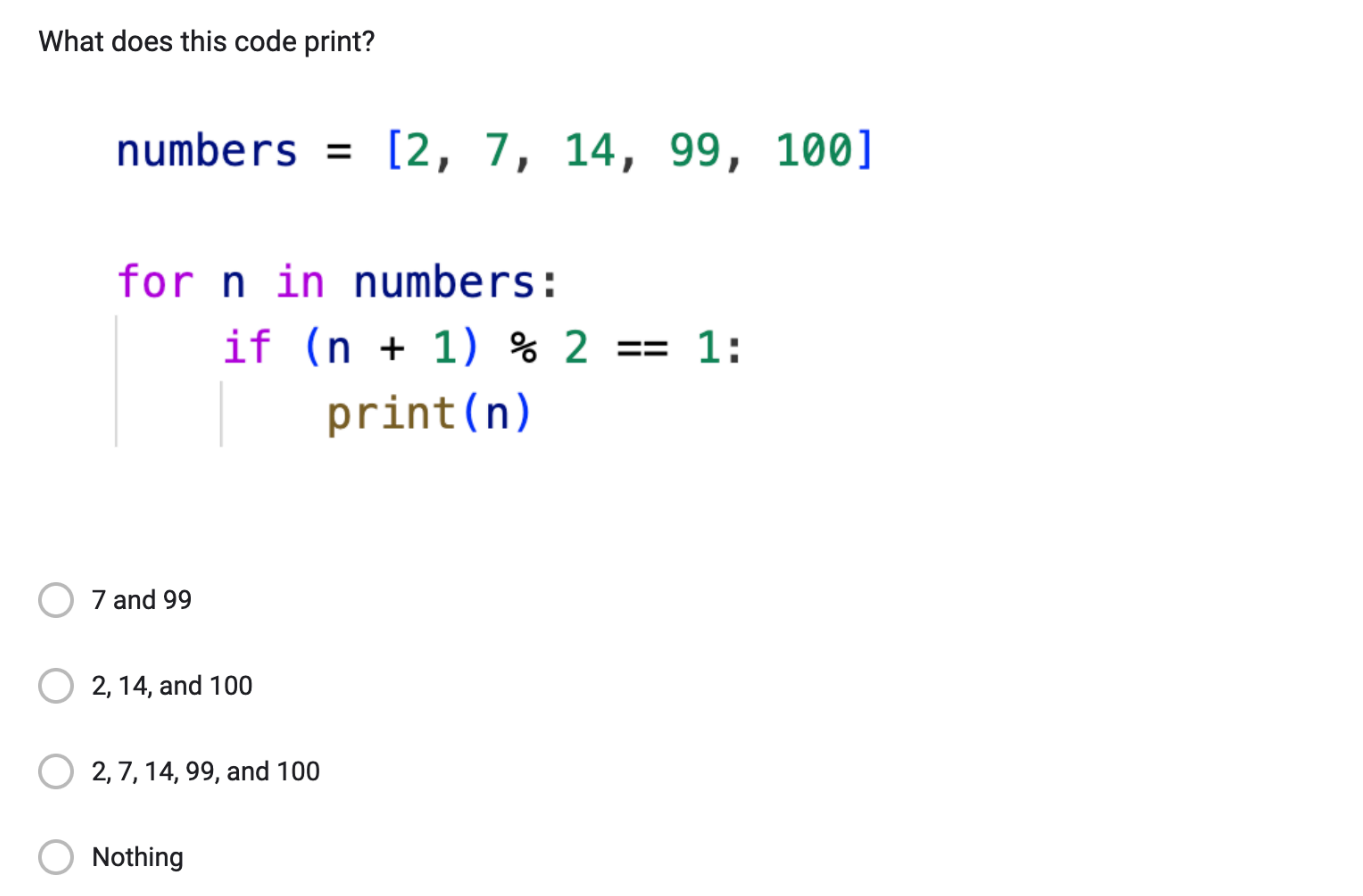
- **Open** `https://github.com/nusco/python-unibo/`
- **Run through the “Example Notebook” notebook**
- **Don’t bother with the AI functionalities**

“PROGRAMMING”

- What programming is about
- What “Python for Business” means



By Sam Hood - State Library of New South Wales. Home and Away - 19020., Public Domain, https://commons.wikimedia.org/w/index.php?curid=41531205



What does this code print?

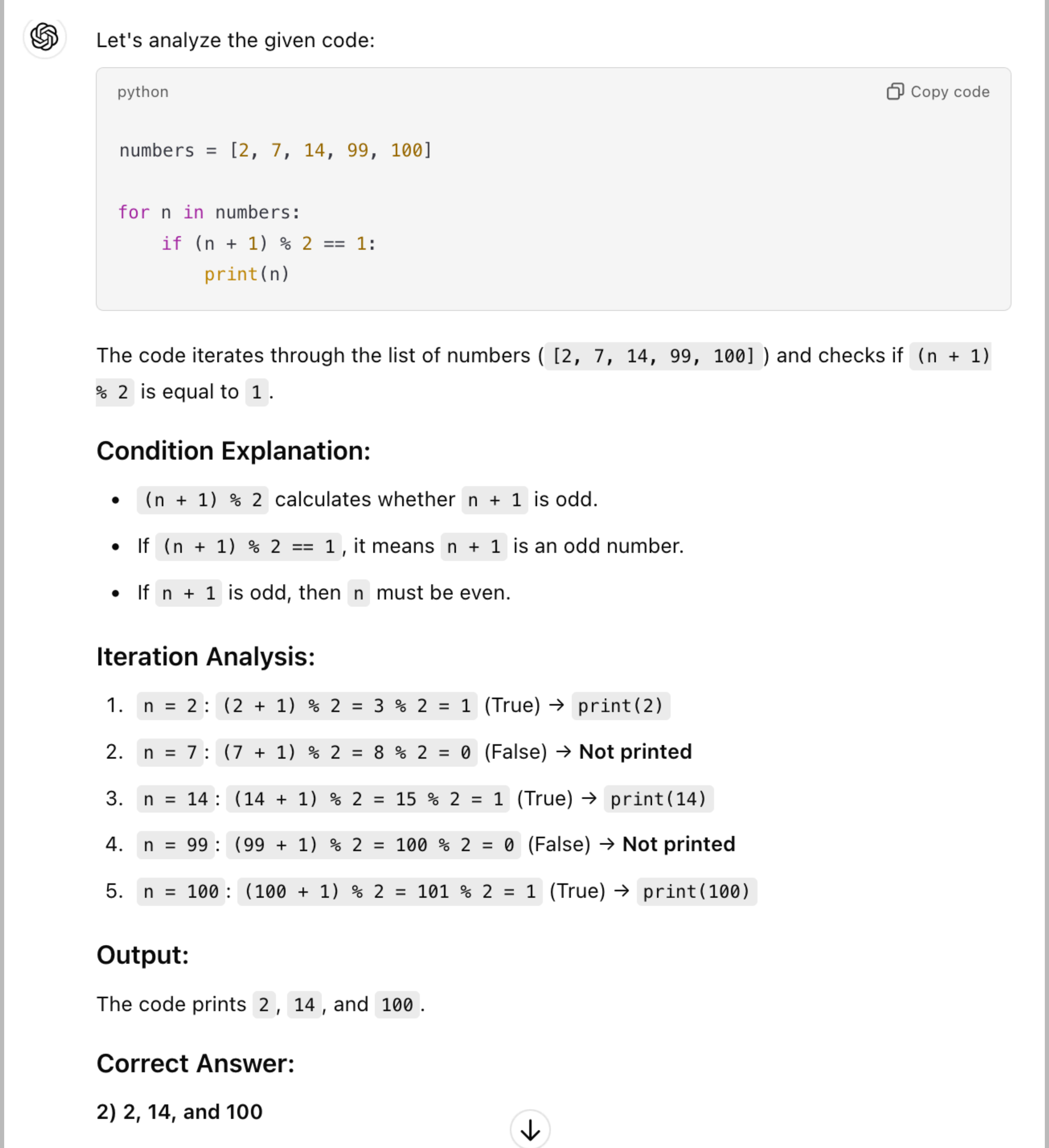
```
numbers = [2, 7, 14, 99, 100]

for n in numbers:
    if (n + 1) % 2 == 1:
        print(n)
```

○ 7 and 99

○ 2, 14, and 100

○ 2, 7, 14, 99, and 100

○ Nothing



Let's analyze the given code:

```python
numbers = [2, 7, 14, 99, 100]

for n in numbers:
    if (n + 1) % 2 == 1:
        print(n)
```

The code iterates through the list of numbers (`[2, 7, 14, 99, 100]`) and checks if `(n + 1) % 2` is equal to `1`.

Condition Explanation:

- `(n + 1) % 2` calculates whether `n + 1` is odd.
- If `(n + 1) % 2 == 1`, it means `n + 1` is an odd number.
- If `n + 1` is odd, then `n` must be even.

Iteration Analysis:

1. `n = 2`: `(2 + 1) % 2 = 3 % 2 = 1` (True) → `print(2)`
2. `n = 7`: `(7 + 1) % 2 = 8 % 2 = 0` (False) → **Not printed**
3. `n = 14`: `(14 + 1) % 2 = 15 % 2 = 1` (True) → `print(14)`
4. `n = 99`: `(99 + 1) % 2 = 100 % 2 = 0` (False) → **Not printed**
5. `n = 100`: `(100 + 1) % 2 = 101 % 2 = 1` (True) → `print(100)`

Output:

The code prints `2`, `14`, and `100`.

Correct Answer:

2) 2, 14, and 100

🐮💩

“Programming is dead.”

A lot of people

> “AI won't take your job. It's somebody using AI that will take your job.”
>
> Richard Baldwin

TWO MORE THINGS

YOU HAVE TO BE PICKY

- Only one of these work:
  - `print("Hello")`
  - `Print("Hello")`
  - `print "Hello"`
  - `PRINT("Hello")`
  - `print ("Hello")`

PRINTING TO THE SCREEN

- In a standalone program
- In the REPL
- In a notebook

1. NUMERIC TYPES

INTS AND FLOATS

- What they are
- Conventions

NOTEBOOK: NUMERIC TYPES

15’

- **Open** `https://github.com/nusco/python-unibo/`
- **Run through the “Numeric Types” notebook**

MORE ABOUT FLOATS

- When would you use floats?
- When would you *not* use floats?

2. STRINGS

STRINGS

- What strings are
- Escaping
- Concatenation and repetition

NOTEBOOK: STRINGS

10’

- **Open** `https://github.com/nusco/python-unibo/`
- **Run through the “Strings” notebook**