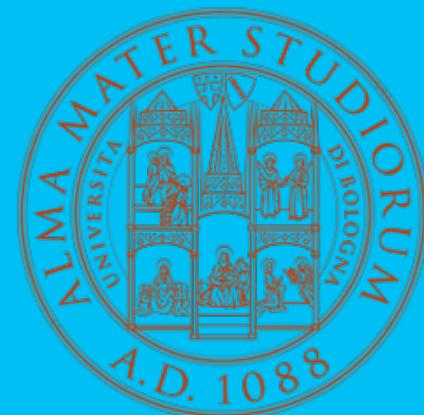


PAOLO PERROTTA

B3844 - PYTHON FOR BUSINESS LAB

2024/25

DAY 1



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ABOUT THIS COURSE

- **Lessons every Friday, same room, same time (but please check)**
- **5 lessons, 3 hours each, with breaks**
- **You'll probably need to exercise at home**
- **Please subscribe on Virtuale: <https://virtuale.unibo.it>**
- **My mail: paolo.perrotta2@unibo.it**



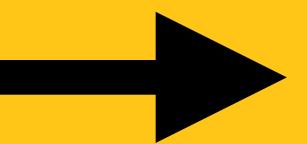
ABOUT THIS COURSE

February 2025

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

March 2025

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



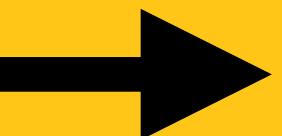
ABOUT ME

- **Programmer, author, teacher**



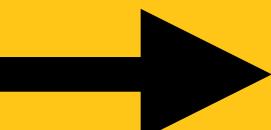
ABOUT YOU

- 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**
- **1st session: March, 28**
- **2nd session: June, 12**
- **3rd session: September, 11**
- **For details (hours, place) and subscriptions, check *AlmaEsami***



ABOUT THE EXAM

aaaa

February 2025

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

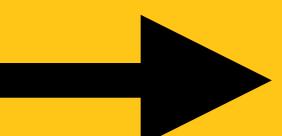
March 2025

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

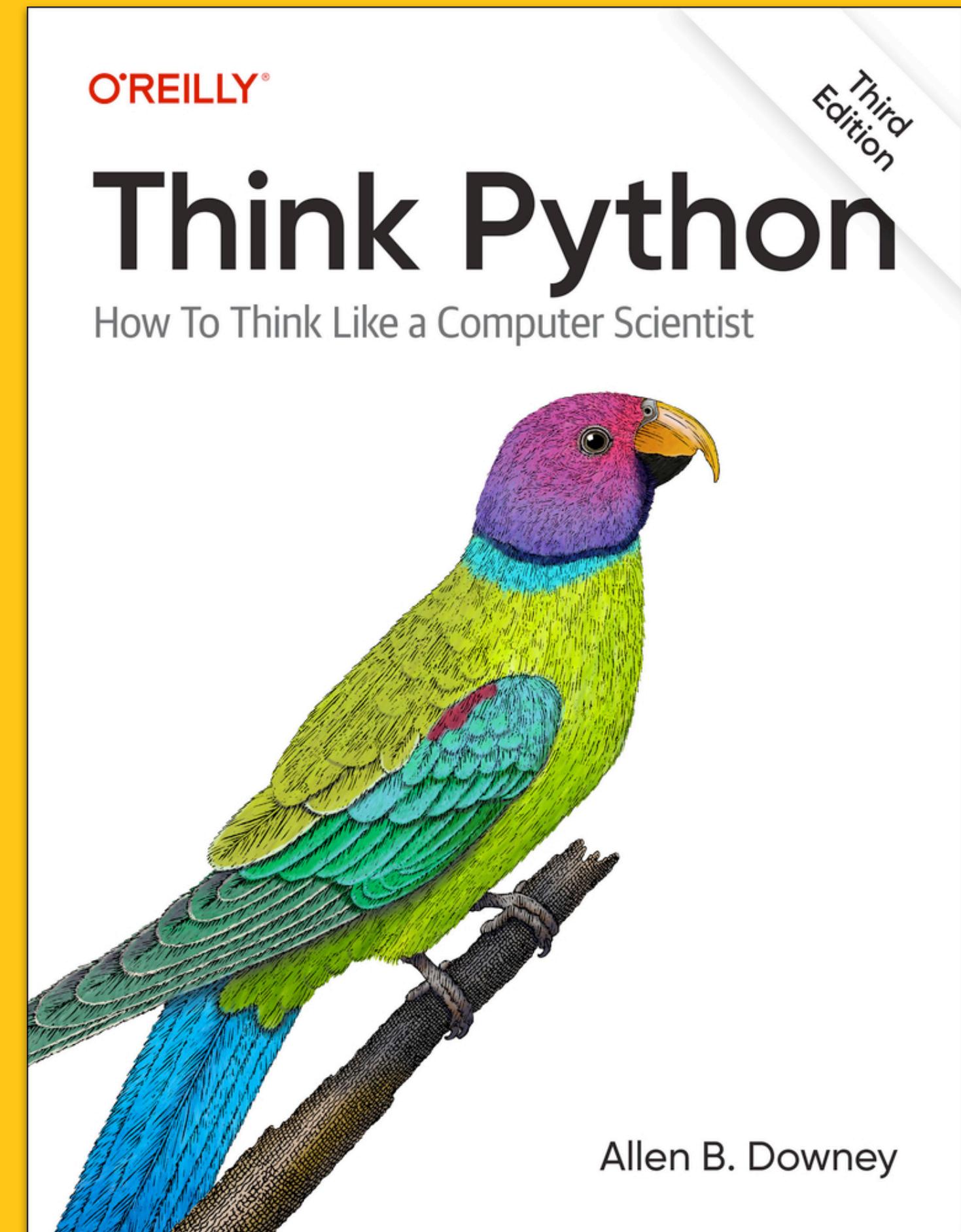


LINKS TO SAVE

- **Calendar, program, etc.**: <https://tinyurl.com/unibo-python>
- **Course material on Virtuale**: <https://virtuale.unibo.it>
- **Notebooks repository**: <https://github.com/nusco/python-unibo>
- **Exams**: <https://almaesami.unibo.it>

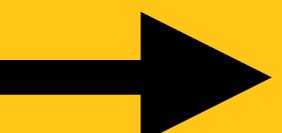


AN (OPTIONAL) TEXTBOOK



ABOUT THE TEXTBOOK

- You can just use online Python resources
- If you want a book, download “*Think Python*” for free
- If you want a dead trees book, buy “*Think Python*” on paper



5 THINGS YOU SHOULD DO

- Listen and follow along
- Complete activities
- Help each other
- Ask questions
- Exercise at home



QUESTIONS?



LOG IN TO YOUR GOOGLE ACCOUNT

- **Create it if you don't have one**
- **Otherwise be ready to help others**
- **Remember to log out at the end of the lesson**

INTRODUCTION: ABOUT PYTHON

RUNNING PYTHON

- **The REPL**



CHECK OUT THE REPL

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



RUNNING PYTHON

- The REPL
- Computational notebooks





NOTEBOOK: EXAMPLE NOTEBOOK

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

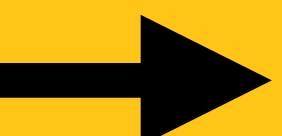
RUNNING PYTHON

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

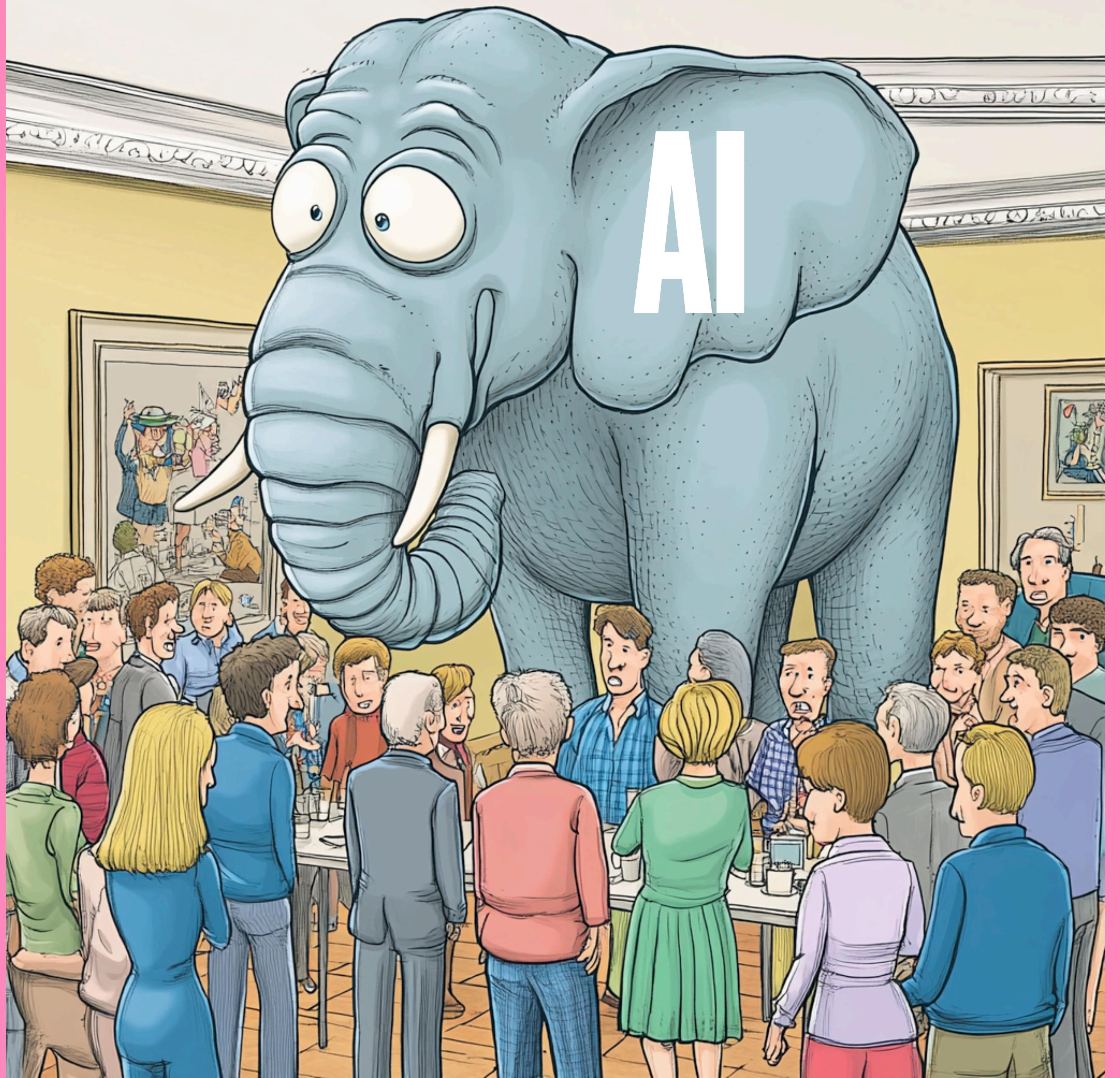


“PROGRAMMING”

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



WHY DO YOU WANT TO LEARN TO PROGRAM?



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

Copy code

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

Output:

The code prints 2, 14, and 100.

Correct Answer:

2) 2, 14, and 100



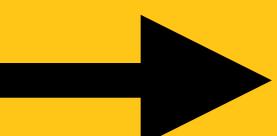
SO... WHAT'S THE POINT?

**“AI WON’T TAKE YOUR JOB. IT’S SOMEBODY
USING AI THAT WILL TAKE YOUR JOB.”**

RICHARD BALDWIN

YOU HAVE TO BE PICKY

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

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

MORE ABOUT FLOATS

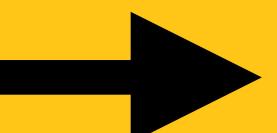
- When would you use floats?
- When would you *not* use floats?



2. STRINGS

STRINGS

- **What strings are**
- **Escaping**
- **Concatenation and repetition**





NOTEBOOK: STRINGS

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

QUESTIONS?