University Mostafa Benboulaid Batna 2 **3rd Year Computer Security Engineer**
Faculty of Mathematics and Computer Science **Academic year 2025/2026**
Computer Science Department

Practical Python Exercise — Built-In Data Types

You're given a series of short tasks. Each one is designed to help you experiment and master a different Python built-in data type. Work through them step by step, writing and testing your code as you go. Don't just copy—try to predict each output first!

1. Numeric, String, and Boolean Basics
   - Create variables **a**, **b**, and **c** for types **int**, **float**, and **complex**. Print their types.
   - Make a boolean variable that checks if a is greater than **b**. Print the result.
   - Make a string with your **name** and another string for your **favorite AI tool**. Concatenate them with a space in between.
2. Work With Sequences
   - Create a **list** with five numbers and a **tuple** with three words. Print both and their types.
   - Use **range()** to generate a list of numbers from 0 to 9 (hint: use list(range(10))).
3. Dictionary Manipulation
   - Make a dictionary mapping three AI concepts (e.g., 'ML', 'DL', 'NLP') to a short explanation. Print the value associated with 'DL'.
   - Update one value in the dictionary and print the entire dictionary.
4. Set Operations
   - Create a set from the list of numbers in step 2. Add a duplicate number and print the set to see what happens.
   - Convert the set to a frozenset and show its type.
5. Practice with Binary Types
   - Create a bytes object from the string "Artificial Intelligence".
   - Create a bytearray from the same string, then modify the first byte to a new value (hint: bytearray(b"...")).
6. Explore NoneType
   - Assign None to a variable called answer. Print its type.
   - Check if answer is None using an if-statement. Print 'No answer' if so.

Challenge (optional)

Pick a small dataset (e.g., a list of numbers, a short string, or a dictionary), and for each data type above, write a snippet that transforms or processes it (e.g., count elements, sum values, find unique items).

When you finish each task, explain to yourself—or a peer—what the data type does, and why it’s useful for AI work.