Lesson 4: Strings and User Input in Python

Marcin Kurzyna

Lecture Goals

This lecture introduces:

- How to read input from the user.
- How strings work and how to manipulate them.
- Important string methods and formatting techniques.

Strings are one of the most commonly used data types in programming, especially when working with user input, files, or text data.

1 User Input with `input()`

The `input()` function reads text typed by the user:

```
name = input("Enter your name: ")
print("Hello,", name)
```

Converting Input to Numbers

`input()` always returns a `str`. To work with numbers, convert using `int()` or `float()`:

```
age = int(input("Enter your age: "))
height = float(input("Enter your height: "))

print("Age:", age, "Height:", height)
```

2 String Indexing and Slicing

Strings are sequences of characters and can be indexed like lists.

```
word = "Python"
print(word[0])     # P
print(word[-1])    # n
print(word[1:4])   # yth
```

3 String Methods

Useful built-in methods:

Method	Effect
`lower()`	convert to lowercase
`upper()`	convert to uppercase
`strip()`	remove surrounding whitespace
`replace(a, b)`	replace text
`split()`	split into a list of words
`join(list)`	join elements into a string

Example:

```python
text = "   Hello World   "
clean = text.strip().upper()
print(clean)  # HELLO WORLD
```

4 String Concatenation and Formatting

4.1 Concatenation

```python
first = "Python"
second = "Programming"
combined = first + " " + second
print(combined)
```

4.2 Formatted Strings (f-strings)

```python
name = "Alice"
age = 20
print(f"{name} is {age} years old.")
```

5 Example: Greeting Program

```python
name = input("Enter your name: ").strip()
print(f"Hello, {name}! Nice to meet you.")
```

Summary

This lecture introduced:

- The `input()` function for reading user input.
- String indexing and slicing.
- Useful string methods (`lower()`, `upper()`, `strip()`, `replace()`, `split()`, `join()`).
- String formatting using f-strings.

6 Exercises

1. Write a program that asks the user for their first and last name and prints them in reverse order (last, first).
2. Ask the user for a sentence and print:
   - the sentence in uppercase,
   - the sentence in lowercase,
   - the length of the sentence.
3. Ask the user for a word and print the first and last character.
4. Write a program that counts how many spaces are in a user-entered sentence.
5. Ask the user for a list of words separated by spaces, then print each word on a separate line.
6. Write a function `reverse_string(s)` that returns the string reversed.
7. Write a program that checks whether a user-entered word is a palindrome.
8. Challenge: Write a program that asks for a full name and formats it as:

   LastName, FirstName MiddleName