**Step 1: Import the re module**

import re

* The re module in Python is used for working with **regular expressions** (regex).
* Regular expressions help **find and replace patterns** in a string.

**Step 2: Define the Input String**

string = "C O D E WOR LD"

* The variable string contains **multiple spaces** between letters.
* Our goal is to **remove all spaces** from this string.

**Step 3: Create a Regular Expression Pattern**

spaces = re.compile(r'\s+')

* The re.compile(r'\s+') function **creates a regex pattern** to match **one or more whitespace characters (\s+)**.
* **\s** → Matches any whitespace (space, tab, newline).
* **+** → Matches **one or more** occurrences of whitespace.

**Step 4: Use re.sub() to Replace Spaces**

result = re.sub(spaces, "", string)

* **re.sub(pattern, replacement, string)** replaces all matches of pattern with replacement in string.
* Here, spaces is our pattern (\s+ for spaces).
* "" (empty string) is the replacement → **removes spaces**.
* The modified string is stored in result.

**Step 5: Print the Result**

print(f"The revised result is: {result}")

* This prints the cleaned-up string **without spaces**.

**Final Output**

The revised result is: CODEWORLD

**Execution Flow**

1. **Input String**: "C O D E WOR LD"
2. **Regex \s+ finds spaces** → "C O D E WOR LD"
3. **Replace spaces with ""** → "CODEWORLD"
4. **Print the cleaned string**.

**Key Takeaways**

✔ **Regular expressions (re)** provide a powerful way to manipulate text.  
✔ **\s+ matches one or more spaces**, allowing us to remove them efficiently.  
✔ **re.sub() replaces matched patterns** with a new value.  
✔ The final result is **a clean string without spaces**.

This method can be **extended** to remove tabs (\t) and newlines (\n) as well.