

# Longest Common Prefix

Find the longest string that is a prefix of ALL strings

● String

■ EASY

## What You Will Learn

- The vertical scanning approach — column by column
- How to use the first string as a reference
- Early exit when mismatch is found

## Intuition First

Imagine stacking all words on top of each other. Read column by column from left to right. As long as every word has the same letter in that column, it's part of the common prefix. The moment ANY word differs — stop.

```
Words: ['flower', 'flow', 'flight']
```

```
Column 0: f, f, f → all same ✓ prefix = 'f'
```

```
Column 1: l, l, l → all same ✓ prefix = 'fl'
```

```
Column 2: o, o, i → MISMATCH ✗ stop!
```

```
Answer: 'fl'
```

## Java Code

```
1  class Solution {
2      public String longestCommonPrefix(String[] strs) {
3          if (strs == null || strs.length == 0) return "";
4          String prefix = strs[0]; // Start with first word
5
6          for (int i = 1; i < strs.length; i++) {
7              // Shrink prefix until strs[i] starts with it
8              while (!strs[i].startsWith(prefix)) {
9                  prefix = prefix.substring(0, prefix.length() - 1);
10             if (prefix.isEmpty()) return "";
11         }
12     }
13     return prefix;
14 }
15 }
```

## Complexity

| Type  | Value  | Meaning                                                                    |
|-------|--------|----------------------------------------------------------------------------|
| Time  | $O(S)$ | S = total chars across all strings. Worst case: all strings are identical. |
| Space | $O(1)$ | Only a few string variables                                                |

## Common Mistakes

### Mistake

Comparing all pairs instead of using first string as anchor. The common prefix can never be longer than the shortest string.

## Quick Revision

|   |                                                       |
|---|-------------------------------------------------------|
| ✓ | Take first string as initial prefix                   |
| ✓ | For each other string, shrink prefix until it matches |
| ✓ | Use <code>startsWith()</code> to check efficiently    |
| ✓ | Return empty string if no common prefix found         |
| ✓ | Time $O(S)$ where S = total characters                |