# 14 - Longest Common Prefix

## Dificulty - Easy

You are given an array of strings `strs`. Return the longest common prefix of all the strings.

If there is no longest common prefix, return an empty string `""`.

Example 1:

`Input: strs = ["bat","bag","bank","band"]`

`Output: "ba"`

Example 2:

`Input: strs = ["dance","dag","danger","damage"]`

`Output: "da"`

Example 3:

`Input: strs = ["neet","feet"]`

`Output: ""`

# Code Solution:

```
class Solution:
    def longestCommonPrefix(self, strs: List[str]) -> str:
        res = ""
        for i in range(len(strs[0])):
            for s in strs:
                if  i == len(s) or s[i] != strs[0][i]:
                    return res
            res += strs[0][i]
        return res
```

- Time O(n·m) where n = number of strings and m = length of the first (or min length, since you stop at first mismatch/shortest).
- Space O(1) extra beyond the output; including the returned prefix it's O(k) where k is the prefix length.

## Explanation:

- The algorithm builds the longest common prefix by comparing characters position-by-position across all strings.
- It uses the first string as a candidate template: iterate over its indices. That sets an upper bound; if any other string ends first or differs, you stop there.
- At each index `i`, scan every string: if any string is too short `(i == len(s))` or has a different character `(s[i] != strs[0][i])`, the prefix ends immediately, so return what you've accumulated.
- If all strings match at `i`, append that character to the result and continue.
- This works because the first mismatch (or shortest string boundary) defines the maximal shared prefix; going further would break equality for at least one string.