Problem Link:

https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/

Problem Description:

Given two strings `needle` and `haystack`, return the index of the first occurrence of `needle` in `haystack`, or `-1` if `needle` is not part of `haystack`.

Problem Approach:

Well, there is a brute force approach to iterate through both the strings and check whether the needle is present in the haystack, and return the first index of it. Which is an O(n*m) solution, not that bad. Easy to understand! There is also an O(n) solution which is python specific.

Sample Test Case:

Input: haystack = "sadbutsad", needle = "sad"

Output: 0

Solution:

We start by iterating through the haystack, checking the first occurrence of the needle's first character. If found, we check whether the rest of the characters of the needle are present in the haystack starting from index i where we found the needle's first character. If yes, we return i, and we return -1 if no index is found which starts with the needle's first character.

Code (Python):

```
def strStr(self, haystack: str, needle: str) -> int:
        for i in range(len(haystack)-len(needle)+1):
            if haystack[i] == needle[0]:
             # Slicing to find the subString
                if haystack[i:len(needle)+i] == needle:
                    return i
        return -1
```

In the following solution, we're doing the same thing, but with the help of python's inbuilt method **.index()** to find the first index of the needle in the haystack.

```
def strStr(self, haystack: str, needle: str) -> int:
        return haystack.index(needle) if needle in haystack else -1
```