Interview Cake

Write a function to reverse a string in-place↴ .

> Since strings in C# are **immutable**↴ , first convert the string into *an array of characters*, do the in-place reversal on that array, and re-join that array into a string before returning it. This isn't technically "in-place" and the array of characters will cost $O(n)$ additional space, but it's a reasonable way to stay within the spirit of the challenge. If you're comfortable coding in a language with mutable strings, that'd be even better!

Breakdown

In general, an " in-place↴ " algorithm will require swapping elements.

Solution

We swap the first and last characters, then the second and second-to-last characters, and so on until we reach the middle.

C# (beta) ▼

```
public string Reverse(string str)
{
    char[] strChars = str.ToCharArray();

    int startIndex = 0;
    int endIndex = strChars.Length - 1;

    while (startIndex < endIndex)
    {
        // Swap characters
        char temp = strChars[startIndex];
        strChars[startIndex] = strChars[endIndex];
        strChars[endIndex] = temp;

        // Move towards middle
        startIndex++;
        endIndex--;
    }

    return new string(strChars);
}
```

Complexity

$O(n)$ time and $O(1)$ space.