C# String & Array Cheat Sheet for Coding Assessments

1. Character Methods (char)

Used to check or convert individual characters (usually inside a foreach loop).

- **char.IsLetter(c)**: Returns true if c is A-Z or a-z.
- **char.IsDigit(c)**: Returns true if c is 0-9.
- **char.IsLetterOrDigit(c)**: Returns true if c is a letter OR a number.
- **char.IsUpper(c) / char.IsLower(c)**: Checks the case of the letter.
- **char.ToUpper(c) / char.ToLower(c)**: Converts a single character's case.

2. String Information & Searching

Used to find things inside a string without changing it. Strings use .Length (a property, no parentheses).

- **str.Length**: Gets the total number of characters.
- **str.Contains("abc")**: Returns true if the substring exists inside the string.
- **str.StartsWith("A") / str.EndsWith("Z")**: Checks the beginning or end of the string.
- **str.IndexOf('x')**: Returns the int index of the *first* time 'x' appears (returns -1 if not found).
- **str.LastIndexOf('x')**: Returns the index of the *last* time 'x' appears.

3. Extracting & Splitting Strings

Used to pull pieces out of a string.

- **str.Substring(startIndex)**: Grabs everything from startIndex to the end.
  - *Example:* "Hello".Substring(1) -> "ello"
- **str.Substring(startIndex, length)**: Grabs a specific number of characters.
  - *Example:* "Hello".Substring(1, 2) -> "el"
- **str.Split(' ')**: Breaks a string into a string[] array based on a character (like a space or comma).

4. Modifying Strings

Important: Strings in C# are *immutable*. These methods do not change the original string; they return a *new* string. You must save the result: str = str.ToUpper();

- **str.ToUpper() / str.ToLower()**: Converts the whole string.
- **str.Replace("old", "new")**: Replaces all occurrences of a character or word.
- **str.Remove(startIndex, length)**: Deletes a chunk of the string.
- **str.Insert(index, "text")**: Shoves new text into the middle of the string.
- **str.Trim()**: Removes accidental spaces from the very beginning and very end.

5. Arrays & Reversing

Because strings cannot be changed character-by-character (e.g., str[0] = 'A' causes an error), you often have to convert them to arrays to swap things around.

- **str.ToCharArray()**: Converts "Cat" into ['C', 'a', 't'].
- **new string(charArray)**: Converts a char[] back into a normal string.
- **Array.Reverse(array)**: Flips an array backwards. (Modifies the array directly!).
- **Array.Sort(array)**: Sorts an array alphabetically or numerically.
- **string.Join("-", array)**: Glues an array of strings/chars together using a separator.
  - *Example:* string.Join("", new string[]{"A", "B"}) -> "AB"

💡 Pro-Tip: How to Swap Characters in C#

If a question asks you to swap the first and last letters of a string, do this:

```
string s = "Hello";

// 1. Convert to array so we can edit it
char[] arr = s.ToCharArray();

// 2. Do the swap using a temporary variable
char temp = arr[0];
arr[0] = arr[arr.Length - 1];
arr[arr.Length - 1] = temp;

// 3. Convert back to string
string swappedString = new string(arr); // Result: "oellH"
```