# Strings

# Functions for strings

There‘s a series of functions for working with strings.

You already know, how to define and assign strings:

- **string firstName = “John”;**
- **string lastName = “Kane”;**

# Information

We can get the length of a string returned:

- **string length = firstName.Length; // 4**

You can also get the index of the first occurrence of a character or string. Again, counting starts at 0:

- **int index = firstName.IndexOf("h"); // 2**

# Manipulation

We can make a string All-Uppercase or All-Lowercase:

- **string upper = firstName.ToUpper(); // JOHN**
- **string lower = firstName.ToLower(); // john**

You can get a sub-part of a string starting at a certain index (1) until the end of the string:

- **string sub = firstName.SubString(1); // "ohn"**

You can get a sub-part of a string starting at a certain index (1) with a certain length (2):

- **string sub = firstName.SubString(1, 2); // "oh"**

You can replace all occurances of a character or string with another one:
**string replace = fullName.Replace( 'n', 'd'); // "Johd Kade"**

# Concatenation

We can combine strings into a larger string using the +-Operator:

- **string fullName = firstName + " " + lastName; // John Kane**

We can combine strings using String-Interpolation. Put the $-operator before a string and variables between { and }:

- **string interp = $"{firstName} {lastName}"; // John Kane**

# Characters

You can get the character at any index of the string. The first index is 0:

- **char char1 = firstName[0]; // 'J'**

# Immutable

One thing to keep in mind:

Strings are immutable.

That means: A string can never be changed.

So, when you call a function on a variable, it never changes the variable itself.

Instead, it returns a new string that you can or need to assign to the same, or a new variable.

- **string fullName = "John Kane";**
- **fullName.Replace('n', 'd');**
- **Console.WriteLine(fullName);**

Output: John Kane

- **string fullName = "John Kane";**
- **fullName = fullName.Replace('n', 'd');**
- **Console.WriteLine(fullName);**

Output: Johd Kade

# Goal 1 - Length

Write a small program that prompts the user to enter a word and then displays the length of that word as an output.

Example:

- Input: Forsbergs
- Output: 9

# Goal 2 - Initials

Write a program that asks for the users full name and then displays their input name as initials and outputs it to them.

Example:

- Input: Gandalf Grey
- Output: G.G

# Goal 3 - Reverse

Write a program that takes a word as an input and then displays the word in a reversed order.

Example:

- Input: Chicken
- Output: nekcihC

# Goal 4 - Replace

Write a program that replaces a specific word in a sentence, prompt the user for the original string and the words to replace and then output the new string.

Example:

- Input: I hate playing games!
- Input: hate
- Input: love
- Output: I love playing games!

# Goal 5 - NoSpace

Write a program that takes a sentence as input and removes all the whitespaces.

Example:

- Input: Om nom nom nom!
- Output: Omnomnomnom!

# Goal 6 -Palindrome

Develop a program that takes a word as input and checks if the word is a palindrome, i.e. the word reads the same forwards and backwards.

Example:

- Input: Banana
- Output: Banana is not a palindrome
- Input: Level
- Output: Level is a palindrome