

**IDENTIFICATION:** Regular Expression #1 / PELEOverall Purpose

The overall purpose for the group of 'Regular Expressions' assignments is to be able to use Regular Expressions for validation, finding, extracting and replacing text.

The whole group of assignments consist of 2 steps:

- 1. A basic understanding of Regular Expressions and the use for validation. (this assignment)**
2. Using Regular expressions for finding, extracting and replacing text.

Background Material:

- Overall view Wikipedia: [https://en.wikipedia.org/wiki/Regular\\_expression](https://en.wikipedia.org/wiki/Regular_expression)
- Nice presentation (though not C#) <https://cs.lmu.edu/~ray/notes/regex/>
- Microsoft C# - quick reference: <https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference>
- C# How to use: <https://www.c-sharpcorner.com/article/c-sharp-regex-examples/>
- Microsoft reference Regex-class: <https://docs.microsoft.com/en-us/dotnet/api/system.text.regularexpressions.regex?view=netcore-3.1>

## This Assignment: Regular Expression #1

### Purpose

The purpose of this assignment is to be able to understand simple patterns for regular expression and to validate input.

### Mission

You have to do two sides of regular expressions:

1. Investigate your knowledge of simple regular expressions
2. Create and test a simple module for validate user input
  - a. Create a validator using regular expressions in a library
  - b. Unittest your validator class

### Assignment 1: Try your knowledge of regular expressions

You should open the link and take the lessons building regular expressions:

<https://regexone.com/>

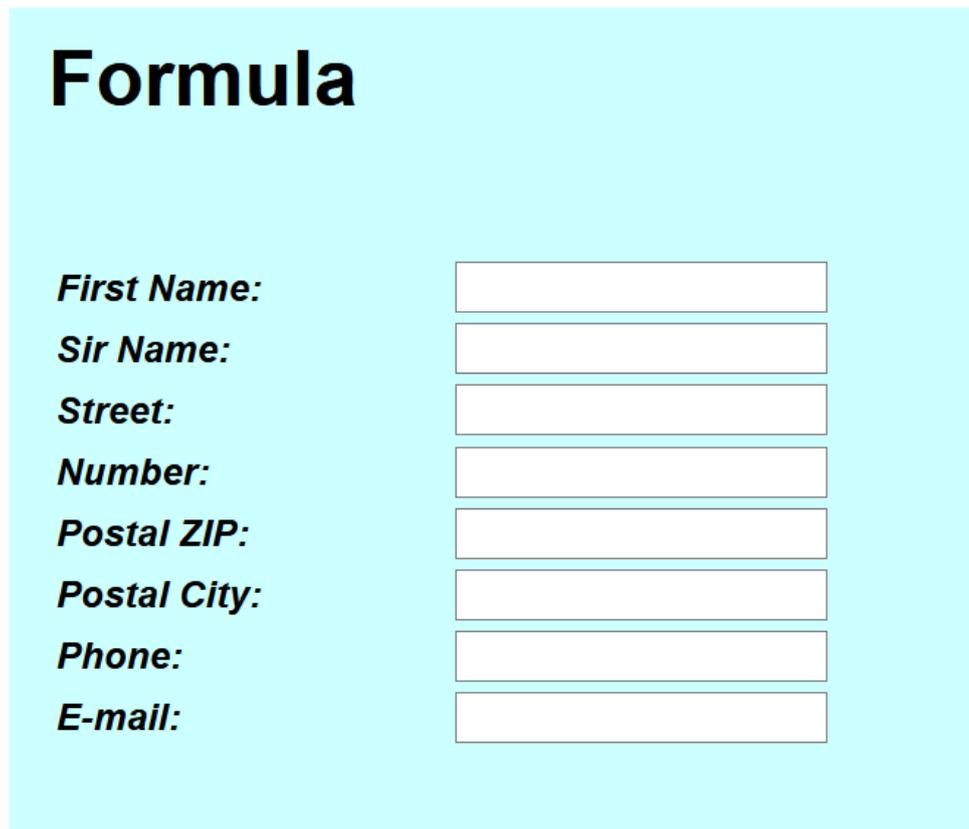
Make all 15 lessons

*Extra continue with additional lessons*

## Assignment 2.a: Try your knowledge of regular expressions

Create a .Net Core library, and create a class (remember to make it public) name it e.g. MyValidator.

You should make different methods to validate input from a virtual formula like the illustration below:



The image shows a form titled "Formula" on a light blue background. The form contains seven input fields, each preceded by a label in bold italic font:

- First Name:** [input field]
- Sir Name:** [input field]
- Street:** [input field]
- Number:** [input field]
- Postal ZIP:** [input field]
- Postal City:** [input field]
- Phone:** [input field]
- E-mail:** [input field]

### Constraints

- First Name, Sir Name, Street and Postal City are letters without any digits and at least one character long.
- Number could be 2, 99 but no longer than three digits but could be followed by several letters incl. digits e.g. 167 B, 1 i.e. number 167 B 1<sup>st</sup> floor
- Zip is always 4 digits
- Phone is always 8 digits long, but could be prefixed with e.g. +45 and perhaps a space
- E-mail is the normal letters, digits, hyphen, dot, underscore then '@' and the domain

Make five validator methods to fulfil these constraints, by using regular expressions.

## Assignment 2.b: Test your validator class

Create a Unittest to test your five methods in your Validator class.