# **Regular Expression in C#**

**Регулярні вирази (Regular expressions)** - це потужний, гнучкий та ефективний метод обробки тексту. Широка нотація регулярних виразів, що відповідає шаблонам, дозволяє швидко аналізувати великі обсяги тексту:

* Знаходити певні шаблони символів.
* Перевіряти текст на відповідність заздалегідь визначеному шаблону (наприклад, адресі електронної пошти).
* Вилучення, редагування, заміна або видалення підрядків тексту.
* Додавання вилучених рядків до колекції для створення звіту.

Шаблони можуть складатися з будь-яких символьних літералів, операторів або конструкторів. У мові C# передбачено клас **Regex**, який можна знайти у просторі імен **System.Text.RegularExpression. Regex** - представляє незмінний регулярний вираз.

Цей клас буде виконувати наступні речі:

* Визначити, чи зустрічається шаблон регулярного виразу у вхідному тексті, викликавши метод **Regex.IsMatch.**
* Отримання одного або всіх входжень тексту, що відповідає шаблону регулярного виразу, шляхом виклику методу **Regex.Match** або **Regex.Matches**. Перший метод повертає об'єкт **System.Text.RegularExpression.Match**, який містить інформацію про відповідний текст. Другий повертає об'єкт **MatchCollection**, який містить по одному об'єкту **System.Text.RegularExpressions.Match** для кожного збігу, знайденого в тексті, що аналізується.
* Замініть текст, який відповідає шаблону регулярного виразу, викликавши метод **Regex.Replace.**

**Приклад 1**: Нижче наведено приклад використання regex для перевірки мобільного номера. Припустимо, ви створюєте форму, де вам потрібно перевірити введений користувачем мобільний номер, тоді ви можете використовувати **regex**.

class GFG {

// Main Method

static void Main(string[] args)

{

// Input strings to Match

// valid mobile number

string[] str = {"9925612824",

"8238783138", "02812451830"};

foreach(string s in str)

{

Console.WriteLine("{0} {1} a valid mobile number.", s,

isValidMobileNumber(s) ? "is" : "is not");

}

Console.ReadKey();

}

// method containing the regex

public static bool isValidMobileNumber(string inputMobileNumber)

{

string strRegex = @"(^[0-9]{10}$)|(^\+[0-9]{2}\s+[0-9]

{2}[0-9]{8}$)|(^[0-9]{3}-[0-9]{4}-[0-9]{4}$)";

// Class Regex Represents an

// immutable regular expression.

// Format Pattern

// xxxxxxxxxx ^[0 - 9]{ 10}$

// +xx xx xxxxxxxx ^\+[0 - 9]{ 2}\s +[0 - 9]{ 2}\s +[0 - 9]{ 8}$

// xxx - xxxx - xxxx ^[0 - 9]{ 3} -[0 - 9]{ 4}-[0 - 9]{ 4}$

Regex re = new Regex(strRegex);

// The IsMatch method is used to validate

// a string or to ensure that a string

// conforms to a particular pattern.

if (re.IsMatch(inputMobileNumber))

return (true);

else

return (false);

}

**Приклад 2:** Нижче наведено приклад використання **regex** для перевірки ідентифікатора електронної пошти. Припустимо, ви створюєте форму, де вам потрібно перевірити введений користувачем **email id**, тоді ви можете використовувати **regex**.

class GFG {

// Main Method

static void Main(string[] args)

{

// Input strings for Match

// valid E-mail address.

string[] str = {"parth@gmail.com",

"parthmaniyargmail.com",

"@gmail.com"};

foreach(string s in str)

{

Console.WriteLine("{0} {1} a valid E-mail address.", s,

isValidEmail(s) ? "is" : "is not");

}

}

// Method to check the Email ID

public static bool isValidEmail(string inputEmail)

{

// This Pattern is use to verify the email

string strRegex = @"\A(?:[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*@(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?)\Z";

Regex re = new Regex(strRegex, RegexOptions.IgnoreCase);

if (re.IsMatch(inputEmail))

return (true);

else

return (false);

}

### **Regex Syntax(Синтаксис регексу)**

Існує багато базових синтаксичних конструкцій, таких як квантори, спеціальні символи, класи символів, групування та альтернативи, що використовуються для регулярних виразів.

**Quantifiers**

**(Квантифікатори):**

| **Sub-expression(Greedy)**  **Підвираз(Жадібний)** | **Sub-expression(Lazy)**  **Під-вираз(Лінивий)** | **Matches**  **(Збіги)** |
| --- | --- | --- |
| **\*** | **\*?** | **Used to match the preceding character zero or more times.**  *Використовується для збігу з попереднім символом нуль або більше разів.* |
| **+** | **+?** | **Used to match the preceding character one or more times.**  *Використовується для збігу з попереднім символом один або кілька разів.* |
| **?** | **??** | **Used to match the preceding character zero or one time.**  *Використовується для збігу попереднього символу з нулем або один раз.* |
| **{n}** | **{n}?** | **Used to match the preceding character exactly n times.**  *Використовується для збігу з попереднім символом рівно n-разів.* |
| **{n, }** | **{n, }?** | **Used to match the preceding character at least n times.**  *Використовується для збігу з попереднім символом щонайменше n разів.* |
| **{n, m}** | **{n, m}?** | **Used to match the preceding character from n to m times.**  *Використовується для повторення попереднього символу від n до m разів.* |

**Special Characters**

**(Спеціальні символи)**

| **Sub-expression** | **Matches** |
| --- | --- |
| **^** | **Word after this element matches at the beginning of the string or line.**  *Слово після цього елемента збігається з початком рядка або строки.* |
| **$** | **Word before this element matches at the end of the line or string.**  *Слово перед цим елементом збігається з кінцем рядка або строки.* |
| **.(Dot)** | **Matches any character only once expect \n(new line).**  *Збігається з будь-яким символом лише один раз очікуючи \n(new line).* |
| **\d** | **It is use to match the digit character.**  *Використовується для зіставлення символу з цифрою.* |
| **\D** | **It is use to match the non-digit character.**  *Використовується для зіставлення з не цифровими символами.* |
| **\w** | **It is use to match any alphanumeric and underscore character.**  *Використовується для зіставлення з будь-яким алфавітно-цифровим символом та символом підкреслення.* |
| **\W** | **It is use to match the any non-word character.**  *Використовується для збігу з будь-якими не словесними символами.* |
| **\s** | **It is use to match the white-space characters.**  *Використовується для збігу з пробілами.* |
| **\S** | **It is use to match the non white-space characters.**  *Використовується для збігу з символами, що не є пробілами.* |
| **\n** | **It is use to match a newline character.**  *Використовується для збігу з символом нового рядка.* |

**Character Classes**

**(Класи символів)**

| **Sub-expression** | **Matches** |
| --- | --- |
| **[]** | **It is used to match the range of character**  *Використовується для зіставлення з діапазоном символів* |
| **[a-z]** | **It is used to match any character in the range of a-z.**  *Використовується для пошуку будь-якого символу в діапазоні a-z.* |
| **[^a-z]** | **It is used to match any character not in the range of a-z.**  *Використовується для порівняння з будь-яким символом, що не входить до діапазону a-z.* |
| **\** | **It is used to match Escaped special character.**  *Використовується для порівняння з екранованим спеціальним символом.* |

**Grouping and Alternatives**

**(Групування та альтернативи)**

| **Sub-expression** | **Matches** |
| --- | --- |
| **()** | **It is used for group expression**  *Використовується для групового виразу* |
| **(a|b)** | **| Operator is used for alternative either a or b.**  *| Оператор використовується для альтернативи a або b.* |
| **(?(exp) yes|no)** | **If expression is matched it gives yes otherwise it gives no.**  *Якщо вираз збігається, він повертає "так", інакше - "ні".* |