

# Regular Expression trong C#

Một **regular expression** là một pattern mà có thể được so khớp với một văn bản đầu vào. .Net Framework cung cấp một phương tiện regular expression mà cho phép so khớp với như thế. Trong C#, một pattern gồm một hoặc nhiều hằng ký tự, toán tử, hoặc construct.

## Construct cho định nghĩa Regular Expression trong C#

Có nhiều loại ký tự, toán tử và construct đa dạng mà giúp bạn định nghĩa Regular Expression trong C#. Bạn theo các link sau để có chi tiết về các construct này.

- [Character escape](#)
- [Lớp Character](#)
- [Anchor](#)
- [Grouping construct](#)
- [Quantifier](#)
- [Backreference construct](#)
- [Alternation construct](#)
- [Substitution](#)
- [Miscellaneous constructs](#)

## Lớp Regex trong C#

Lớp Regex trong C# được sử dụng để biểu diễn một Regular Expression. Nó có các phương thức được sử dụng phổ biến sau:

STT	Phương thức
1	<b>public bool IsMatch(string input)</b>  Chỉ rằng có hay không Regular Expression đã cho trong Regex constructor này tìm

	thấy một match trong chuỗi đầu vào đã xác định
2	<b>public bool IsMatch(string input, int startat)</b>  Chỉ rằng có hay không Regular Expression đã cho trong Regex constructor này tìm thấy một match trong chuỗi đầu vào đã xác định, bắt đầu tại vị trí startat đã cho trong chuỗi
3	<b>public static bool IsMatch(string input, string pattern)</b>  Chỉ rằng có hay không Regular Expression đã cho tìm thấy một match trong chuỗi đầu vào đã xác định
4	<b>public MatchCollection Matches(string input)</b>  Tìm kiếm chuỗi đầu vào đã xác định về tất cả sự xuất hiện của một Regular Expression
5	<b>public string Replace(string input, string replacement)</b>  Trong một chuỗi đầu vào đã xác định, thay thế tất cả chuỗi mà so khớp với một Regular Expression pattern với một chuỗi thay thế đã cho
6	<b>public string[] Split(string input)</b>  Chia một chuỗi đầu vào thành một mảng các chuỗi phụ tại vị trí được định nghĩa bởi một Regular Expression pattern đã xác định trong Regex constructor

Để có danh sách đầy đủ các phương thức và thuộc tính, bạn vui lòng đọc Microsoft Documentation về C#.

## Ví dụ 1

Ví dụ sau so khớp với các từ mà bắt đầu với 'S':

```
using System;  
using System.Text.RegularExpressions;
```

```
namespace RegExApplication
{
    class Program
    {
        private static void showMatch(string text, string expr)
        {
            Console.WriteLine("The Expression: " + expr);
            MatchCollection mc = Regex.Matches(text, expr);
            foreach (Match m in mc)
            {
                Console.WriteLine(m);
            }
        }

        static void Main(string[] args)
        {
            string str = "A Thousand Splendid Suns";

            Console.WriteLine("Matching words that start with 'S': ");
            showMatch(str, @"\bS\S*");
            Console.ReadKey();
        }
    }
}
```

Khi code trên được biên dịch và thực thi, nó sẽ cho kết quả:

```
Matching words that start with 'S':
The Expression: \bS\S*
Splendid
Suns
```

## Ví dụ 2

Ví dụ sau so khớp với các từ mà bắt đầu với 'm' và kết thúc với 'e':

```
using System;
using System.Text.RegularExpressions;

namespace RegExApplication
{
    class Program
    {
        private static void showMatch(string text, string expr)
        {
            Console.WriteLine("The Expression: " + expr);
            MatchCollection mc = Regex.Matches(text, expr);
            foreach (Match m in mc)
            {
                Console.WriteLine(m);
            }
        }
        static void Main(string[] args)
        {
            string str = "make maze and manage to measure it";

            Console.WriteLine("Matching words start with 'm' and ends with 'e':");
            showMatch(str, @"\bm\S*e\b");

            Console.ReadKey();
        }
    }
}
```

Khi code trên được biên dịch và thực thi, nó sẽ cho kết quả:

```
Matching words start with 'm' and ends with 'e':
The Expression: \bm\S*e\b
make
maze
manage
```

measure

## Ví dụ 3

Ví dụ sau thay thế các white space:

```
using System;
using System.Text.RegularExpressions;

namespace RegExApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            string input = "Hello World ";
            string pattern = "\\s+";
            string replacement = " ";
            Regex ngx = new Regex(pattern);
            string result = ngx.Replace(input, replacement);

            Console.WriteLine("Original String: {0}", input);
            Console.WriteLine("Replacement String: {0}", result);
            Console.ReadKey();
        }
    }
}
```

Khi code trên được biên dịch và thực thi, nó sẽ cho kết quả:

```
Original String: Hello World
Replacement String: Hello World
```