

ラムダ式 (C#) の基礎文法

●ラムダ式の基本構文

左辺 => 右辺 // 「=>」は「goes to」(ゴーズ・トゥ)と読む。

●ラムダ式構文の左辺

() => 右辺 // パラメータがない場合
 param => 右辺 // パラメータが1つの場合
 (param1, param2) => 右辺 // パラメータが2つ以上の場合
 (int param) => 右辺 // パラメータの型指定

○左辺：パラメータの例

```
button1.Click += (sender, e) => <右辺>;
var query = dataSource.Where(i => <右辺>);
```

●ラムダ式構文の右辺

左辺 => 式
 左辺 => { 文; 文; 文; }

●右辺：「式」形式のラムダ

```
button1.Click += (sender, e) => MessageBox.Show("式!");
var query = dataSource.Where(i => i % 13 == 0);
```

●右辺：「文」形式のラムダ

```
button1.Click += (sender, e) => { MessageBox.Show("文!"); };
// 下記は、戻り値を返す例
var query = dataSource.Where(i => { return i % 13 == 0; });
// 下記は、複文の例
button1.Click += (sender, e) =>
{
    int count = 3;
    string msg = String.Format("{0}つの文です。", count);
    MessageBox.Show(msg);
}
```

詳細情報：「.NET開発『虎の巻』：C#ラムダ式 基礎文法最速マスター」

http://www.atmarkit.co.jp/fdotnet/rapidmaster/rapidmaster_01/rapidmaster_01.html

ラムダ式 (VB) の基礎文法

●ラムダ式の基本構文

Function(左辺) 右辺 ' VB 2010以降では「Sub」も利用可能

●ラムダ式構文の左辺

Function() 右辺 ' パラメータがない場合
 Function(param1, param2) 右辺 ' パラメータがある場合
 Function(param As Integer) 右辺 ' パラメータの型指定

○左辺：パラメータの例

```
AddHandler Button1.Click, Function(sender, e) <右辺>
```

●ラムダ式構文の右辺

Function(左辺) 式	Function(左辺)
Sub(左辺) 式	Return 文
	End Function

●右辺：「式」形式のラムダ

```
AddHandler Btn1.Click, Function(sndr, e) MessageBox.Show("式!");
Dim query = dataSource.Where(Function(i) i Mod 13 = 0)
```

●右辺：「文」形式のラムダ (VB 2010以降)

```
AddHandler Button1.Click,
Sub(sender, e)
    MessageBox.Show("文!")
End Sub
```

' 下記は、戻り値を返す例

```
Dim query = dataSource.Where( _
Function(i)
    Return i Mod 13 = 0
End Function)
```

' 下記は、複文の例

```
AddHandler Button1.Click, _
Sub(sender, e)
    Dim count As Integer = 3
    Dim msg = String.Format("{0}つの文です。", count)
    MessageBox.Show(msg)
End Sub
```

詳細情報：「.NET開発『虎の巻』：
VBラムダ式 基礎文法最速マスター」

[http://www.atmarkit.co.jp/fdotnet/rapidmaster/rapidmaster_02.html](http://www.atmarkit.co.jp/fdotnet/rapidmaster/rapidmaster_02/rapidmaster_02.html)