



# **ASP.NET - Validation Controls**

# ASP.NET - Validation Controls

- ASP.Net validation controls validate the user input data to ensure that useless, unauthenticated or contradictory data dont get stored.
- ASP.Net provides the following validation controls:
  1. RequiredFieldValidator
  2. RangeValidator
  3. CompareValidator
  4. RegularExpressionValidator
  5. CustomValidator
  6. ValidationSummary

# ASP.NET - Validation Controls

- **The BaseValidator Class:**
- The validation control classes inherit from the BaseValidator class and inherit its properties and methods. Therefore, it would help to take a look at the properties and the methods of this base class, which are common for all the validation controls:

# ASP.NET - Validation Controls

Members	Description
ControlToValidate	Indicates the input control to validate.
Display	Indicates how the error message is shown.
EnableClientScript	Indicates whether client side validation will take.
Enabled	Enables or disables the validator.
ErrorMessage	Error string.
Text	Error text to be shown if validation fails.
IsValid	Indicates whether the value of the control is valid.

# ASP.NET - Validation Controls

Members	Description
SetFocusOnError	It indicates whether in case of an invalid control, the focus should switch to the related input control.
ValidationGroup	The logical group of multiple validators, where this control belongs.
Validate()	This method revalidates the control and updates the IsValid property.
IsValid	Indicates whether the value of the control is valid.
SetFocusOnError	It indicates whether in case of an invalid control, the focus should switch to the related input control.

# ASP.NET - Validation Controls

- **The RequiredFieldValidator:**
- The RequiredFieldValidator control ensures that the required field is not empty.
- It is generally tied to a text box to force input into the text box.

# ASP.NET - Validation Controls

- **The RangeValidator:**
- The RangeValidator control verifies that the input value falls within a predetermined range.

Properties	Description
Type	it defines the type of the data; the available values are: Currency, Date, Double, Integer and String
MinimumValue	it specifies the minimum value of the range
MaximumValue	it specifies the maximum value of the range

# ASP.NET - Validation Controls

- **The CompareValidator:**
- The CompareValidator control compares a value in one control with a fixed value, or, a value in another control.

Properties	Description
Type	it specifies the data type
ControlToCompare	it specifies the value of the input control to compare with
ValueToCompare	it specifies the constant value to compare with
Operator	it specifies the comparison operator, the available values are: Equal, NotEqual, GreaterThan, GreaterThanEqual, LessThan, LessThanEqual and DataTypeCheck



# ASP.NET - Validation Controls

- **The RegularExpressionValidator**
- The RegularExpressionValidator allows validating the input text by matching against a pattern against a regular expression.
- The regular expression is set in the ValidationExpression property.

# ASP.NET - Validation Controls

The following table summarizes the commonly used syntax constructs for regular expressions:

Character Escapes	Description
\b	Matches a backspace
\t	Matches a tab
\r	Matches a carriage return
\v	Matches a vertical tab
\f	Matches a form feed
\n	Matches a new line
\	Escape character

# ASP.NET - Validation Controls

Apart from single character match, a class of characters could be specified that can be matched, called the metacharacters.

Metacharacters	Description
.	Matches any character except \n
[abcd]	Matches any character in the set
[^abcd]	Excludes any character in the set
[2-7a-mA-M]	Matches any character specified in the range
\w	Matches any alphanumeric character and underscore
\W	Matches any non-word character

# ASP.NET - Validation Controls

Metacharacters	Description
\s	Matches whitespace characters like, space, tab, new line etc.
\S	Matches any non-whitespace character
\d	Matches any decimal character
\D	Matches any non-decimal character

# ASP.NET - Validation Controls

- Quantifiers could be added to specify number of times a character could appear

Quantifier	Description
*	Zero or more matches
+	One or more matches
?	Zero or one matches
{N}	N matches
{N,}	N or more matches
{N,M}	Between N and M matches

# ASP.NET - Validation Controls

- **The CustomValidator:**

- The CustomValidator control allows writing application specific custom validation routines for both the client side and the server side validation.
- The client side validation is accomplished through the ClientValidationFunction property.
- The client side validation routine should be written in a scripting language, like JavaScript or VBScript, which the browser can understand.
- The server side validation routine must be called from the control's ServerValidate event handler.
- The server side validation routine should be written in any .Net language, like C# or VB.Net.

# ASP.NET - Validation Controls

- **The CustomValidator:**

```
<asp:CustomValidator ID="CustomValidator1" runat="server"  
ClientValidationFunction=.cvf_func.  
ErrorMessage="CustomValidator"> </asp:CustomValidator>
```

# ASP.NET - Validation Controls

- **The ValidationSummary Control**
- The ValidationSummary control does not perform any validation but shows a summary of all errors in the page.
- The summary displays the values of the ErrorMessage property of all validation controls that failed validation.
- The following two mutually inclusive properties list out the error message:
  1. **ShowSummary:** shows the error messages in specified format.
  2. **ShowMessageBox:** shows the error messages in a separate window.



# ASP.NET - Validation Controls

## Validation Groups:

- Complex pages have different groups of information provided in different panels.
- In such a situation a need for performing validation separately for separate group, might arise. This kind of situation is handled using validation groups.
- To create a validation group, you should put the input controls and the validation controls into the same logical group by setting their *ValidationGroup* property.

# ASP.NET - Validation Controls

## • **Example:**

- The following example describes a form to be filled up by all the students of a school, divided into four houses, for electing the school president. We will be using the validation controls to validate the user input.
- The form in Design view:

# ASP.NET - Validation Controls

Start Page Default.aspx Default.aspx

## President Election 2010

President Election Form - Choose your president

Candidate:  Please choose a candidate

House: ☐ Red  
☐ Blue  
☐ Yellow  
☐ Green Enter your house name

Class:  Enter your class (6 - 12)

Email:  Enter your email

Errors:

- Error message 1.
- Error message 2.