Questions:

1. What are ASP.NET Validation controls?
2. What is BaseValidator?
3. Where do the ASP.NET validation controls validate data, on the Client or on the Web Server?
4. What are the 6 different validation controls provided by ASP.NET?
5. What property of the validation control is used to specify which control to validate?
6. How to add a validation control to another control?
7. Hot can you specify a error message to a validation control?
8. How to display the error messages from the validation controls in one place?
9. Are the validation controls fired on the client side if JavaScript is disabled on the client browser?
10. What is the use of CausesValidation property of an ASP.NET button control?
11. Give an example of real time scenario where you might use CausesValidation property of an ASP.NET button control?
12. What is ASP.NET Custom Validator used for?
13. How do you programmatically check, if the client side validation is not bypassed by disabling the JavaScript on the client browser?
14. How do you programmatically invoke all validation controls on a page?
15. What is a validation group?
16. How do you create a validation group?
17. Explain how a validation group works when the Page is posted by clicking a button?
18. Can a DropDownList fire validation controls?
19. How do you programatically force all validation controls in a particular validation group to be fired?
20. What is SetFocusOnError property of a validation control used for?
21. What is InitialValue property of a RequiredFieldValidator?

Questions and Answers:

1. What are ASP.NET Validation controls?

Answer: ASP.NET provides validation controls to help you check Web form data entries before the data is accepted and saved in the Database. Validation controls can be used to address the following questions.

a) Did the user enter anything?
b) Is the entry the appropriate kind of data (For example, Date of Birth should be a valid Date, Name should be a string etc.)?

c) Is the data within a required range?(For example age cannot be greater than 100 years)

The validation controls check the validity of data entered in associated server controls on the client before the page is posted back to the server. Most validity problems can be caught and corrected by the user without a round-trip to the server.

2. What is BaseValidator?

Answer: It is the parent of all the validation controls. Validation Controls inherited from BaseValidator control are: RequiredFieldValidator, RangeValidator, CompareValidator, RegularExpressionValidator, CustomValidator.

3. Where do the ASP.NET validation controls validate data, on the Client or on the Web Server?

Answer: ASP.NET validation controls validate data first on the client and then on the web server. If a client disables JavaScript on the browser then, client side validations are bypassed and validations are performed on the web server. Client-side validation is provided by a JavaScript library named WebUIValidation.js, which is downloaded separately to the client. Validation controls also automatically provide server-side validation. Server-side validation is always performed, whether or not client-side validation has occurred. This double-checking ensures that custom validations are performed correctly and that client-side validation has not been circumvented.

4. What are the 6 different validation controls provided by ASP.NET?

Answer:

1) RequiredFieldValidator:Checks whether a control contains data
2) CompareValidator:Checks whether an entered item matches an entry in another control
3) RangeValidator:Checks whether an entered item is between two values
4) RegularExpressionValidator:Checks whether an entered item matches a specified format
5) CustomValidator:Checks the validity of an entered item using a client-side script or a server-side code, or both
6) ValidationSummary:Displays validation errors in a central location or display a general validation error description

5. What property of the validation control is used to specify which control to validate?

Answer: ControlToValidate property.

6. How to add a validation control to another control?

Answer: Draw a validation control on a Web form and set its ControlToValidate property to the control you want to validate. If you're using the CompareValidator control, you also need to specify the ControlToCompare property.

7. Hot can you specify a error message to a validation control?

Answer: Set the validation control's ErrorMessage property to the error message you want displayed if the control's data is not valid. Set the validation control's Text property if you want the validation control to display a message other than the message in the ErrorMessage property when an error occurs. Setting the Text property lets you briefly indicate where the error occurred on the form and display the longer ErrorMessage property in a ValidationSummary control.

8. How to display the error messages from the validation controls in one place?

Answer: Draw a ValidationSummary control on the Web form to display the error messages from the validation controls in one place. Provide a control that triggers a postback event. Although validation occurs on the client side, validation doesn't start until a postback is requested.

9. Are the validation controls fired on the client side if JavaScript is disabled on the client browser?

Answer: No, validation controls are not fired on the client side if JavaScript is disabled on the client browser.

10. What is the use of CausesValidation property of an ASP.NET button control?

Answer: CausesValidation property of an ASP.NET button control is used to determine if the validation controls should be fired when the button is clicked. If CausesValidation property is set to true, then validation is performed and if the CausesValidation property is set to false then validation is not done.

11. Give an example of real time scenario where you might use CausesValidation property of an ASP.NET button control?

Answer: Let us assume we have a Page that collects user information like name, age, date of birth, gender with a submit and reset buttons. When I click the submit button the information filled on the form should be validated and saved to the database. If I click the reset button then all the controls on the webform should default to their initial values without validation happening.So you have to set the CausesValidation property of the reset button to false for the validation to be bypassed. Otherwise you will not be able to post back the page to the server.

12. What is ASP.NET Custom Validator used for?

Answer: ASP.NET Custom Validator is used to perform complex types of validation not provided by the standard validation control, use a CustomValidator control and write code to perform the validation on the server side and optionally on the client side.

13. How do you programmatically check, if the client side validation is not bypassed by disabling the JavaScript on the client browser?

Answer: We use Page.IsValid property to determine if all the validations have succeeded. For this property to return true, all validation server controls in the current validation group must validate successfully.

14. How do you programmatically invoke all validation controls on a page?

Answer: Call Page.Validate() method. When this method is invoked, it iterates through the validation controls contained in the ValidatorCollection object associated with the Page.Validators property and invokes the validation logic for each validation control in the current validation group.

15. What is a validation group?

Answer: Validation groups allow you to group validation controls on a page as a set. Each validation group can perform validation independently from other validation groups on the page.

16. How do you create a validation group?

Answer: You create a validation group by setting the ValidationGroup property to the same name for all the controls you want to group. You can assign any name to a validation group, but you must use the same name for all members of the group.

17. Explain how a validation group works when the Page is posted by clicking a button?

Answer: During postback, the Page class's IsValid property is set based only on the validation controls in the current validation group. The current validation group is determined by the control that caused validation to occur. For example, if a button control with a validation group of LoginGroup is clicked, then the IsValid property will return true if all validation controls whose ValidationGroup property is set to LoginGroup are valid.

18. Can a DropDownList fire validation controls?

Answer: Yes, DropDownList control can also fire validation if the control's CausesValidation property is set to true and the AutoPostBack property is set to true.

19. How do you programatically force all validation controls in a particular validation group to be fired?

Answer: Call the Page.Validate(string GroupName) method and pass the name of the validation group. This will fire only the validation controls in that validation group.

20. What is SetFocusOnError property of a validation control used for?

Answer: Use the SetFocusOnError property to specify whether focus is automatically set to the control specified by the ControlToValidate property when this validation control fails. This allows the user to quickly update the appropriate control. If multiple validation controls fail and this property is set to true, the control specified in the ControlToValidate property for the first validation control receives focus.

21. What is InitialValue property of a RequiredFieldValidator?

Answer: Use this property to specify the initial value of the input control, Validation fails only if the value of the associated input control matches this InitialValue upon losing focus.