LESSON 1

# 98-363 Web Development Fundamentals

1.1 Customize the Layout and Appearance of a Web Page

1.2 Understand ASP.NET Intrinsic Objects

1.3 Understand State Information in Web Applications

1.4 Understand Events and Control Page Flow

1.5 Understand Controls

1.6 Understand Configuration Files

MTA Web Development Fundamentals 1 Test

LESSON 1.1

98-363 Web Development Fundamentals

# Customize the Layout and Appearance of a Web Page

LESSON 1.1

98-363 Web Development Fundamentals

## Lesson Overview

How do you create an attractive, usable, and accessible Web page?

In this lesson, you will practice:

- Adding external, internal, and inline styles
- Using tables, layers, or both to organize Web page elements
- Embedding images that were added to your page
- Designing your page layout for navigation

# Guiding Questions

1. What is the benefit of using an external cascading style sheet?
2. How can tables and layers be used to organize page elements?
3. How do you add images, audio, video, and other multimedia elements to a website?
4. How do you use classes, divs, and ids to organize elements on a page?
5. Most important, how will the site be used? Who is your target audience? What is the mission of the site?

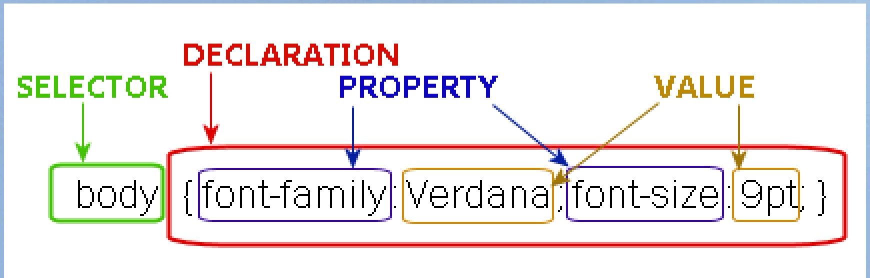
All these questions must guide your design.

# Cascading Style Sheets

- Decreases download time by loading the style when the first page is downloaded and made available to all subsequent Web pages
- The actual style applied to content follows these rules, with rule number one having the highest priority:
    1. Inline style [inside an Extensible Hypertext Markup Language (XHTML) element]
    2. Internal style sheet (inside the <head> tag)
    3. External style sheet
    4. Browser default
- Styles can be changed easily for all the pages by simply changing the style definitions in the one external style sheet document.

# Style Formatting

- The selector is the HTML tag.
- The declaration is in enclosed with curly braces {..}.
- The property is the HTML element.
- Finally, the value is an attribute to be used for each selector in your Web page(s).



```
body {font-family:Verdana; font-size:9pt; }
```

# Assignment

- Complete student activity 1.1

# Lesson Review

- What is the order of precedence for the three styles: internal, external, browser default, and inline?
- Explain what happens if you do not embed an external image or images.
- Describe the types of navigational structures common in websites. Why is important to plan for the correct type?

Microsoft

LESSON 1.2

98-363 Web Development Fundamentals

# Understand ASP.NET Intrinsic Objects

# Lesson Overview

What is the application life cycle of ASP.NET?

In this lesson, you will review these ASP.NET terms:

- Request object
- Server object
- Application state object
- Session state object
- Response object
- HttpContext object

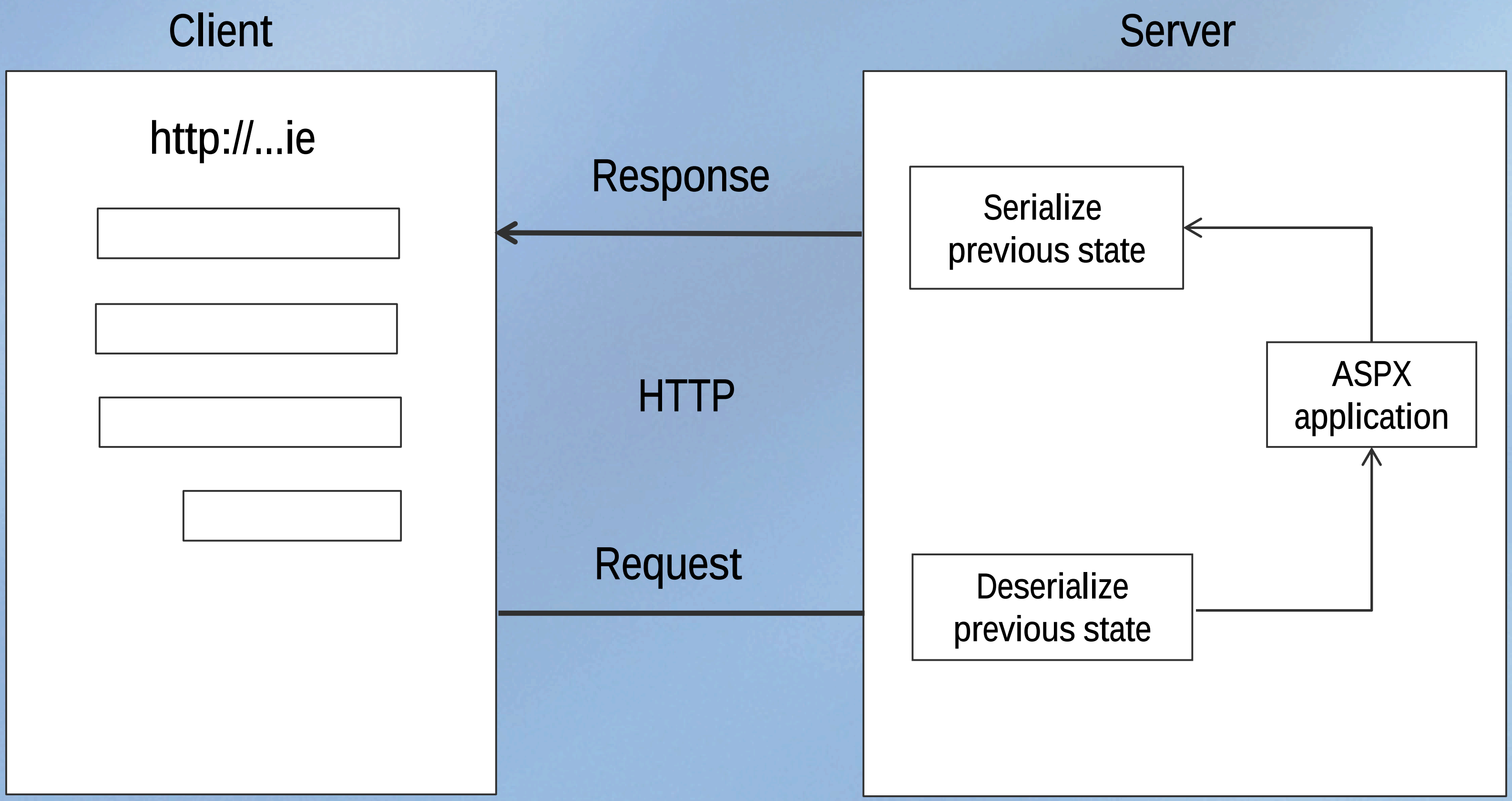
## Guiding Questions

1. How is data transmitted from one page to another (such as a shopping cart on an e-commerce site)?
2. What is the difference between a client and a server?
3. What is the Web form model in the Microsoft .NET Framework?

# General Information

- ASP.NET is an event-driven model of interaction to the Web.
- Data is collected by the client and forwarded to the server for *stateful* processing.
- The server processes output of client actions and triggers reactions in the form of responses to the client.
- The state of the application contains two types of information: state of client and state of session.
- Session state needs to manage data sent from one page to be used by another page later.
- The HttpContext class contains objects that are specific to the current application request.
- Reminder: the Hypertext Transfer Protocol (HTTP) *cannot* track session state—it is a “stateless” protocol.

# Web Form Model in the .NET Framework



# How Intrinsic Objects Work:

- Request—Represents the incoming request from the client to the Web server (for example, the data posted on a form).
- Application—Takes over the processing of an incoming request. It can handle one request at a time.
- Response—Represents the response sent back to the client from the Web server (for example, allowing writing text output).
- Server—Is used to access properties and methods on the server.
- Session state—Stores information about, or change settings for, a user session.
- HttpContext—Encapsulates all HTTP-specific information about an individual HTTP request.

## Assignment

- Complete student activity 1.2

# Lesson Review

- How do clients and servers communicate?
- What are some examples of applications that can be implemented in the ASP.NET environment?
- Can you list all the ASP.NET intrinsic objects that we discussed in this lesson?

LESSON 1.3

98-363 Web Development Fundamentals

# Understand State Information in Web Applications

# Lesson Overview

How does ASP.NET state management work?

In this lesson, you will review these ASP.NET terms:

- View state
- Control state
- Application state
- Session state

# Guiding Questions

1. How does the ASP.NET Framework help with state management control?
2. What data types can be stored in view state?
3. What are hidden fields, and how are they used?
4. What are cookies?

# Questions to help you determine what state to choose:

- How much information do you need to store?
- Does the client accept persistent or in-memory cookies?
- Do you want to store the information on the client or on the server?
- Is the information sensitive?
- What performance and bandwidth criteria do you have for your application?
- What are the capabilities of the browsers and devices that you are targeting?
- Do you need to store information per user?
- How long do you need to store the information?
- Do you have a Web farm (multiple servers), a Web garden (multiple processes on one machine), or a single process that serves the application?

## How does the ASP.NET Framework help with state management control?

- ASP.NET provides state management facility at four levels: application, session, page, and request.

- ASP.NET helps with state management control by maintaining state and page information over multiple requests for the same or different pages.

- *Application state* is a data repository available to all classes in an ASP.NET application. Application state is stored in memory on the server and is faster than storing and retrieving information in a database.

# How the ASP.NET Framework helps with state management control (continued)

- *Control state* persists control information that must be retained between postbacks, even if view state is disabled for the page or for a control. Control state is stored in one or more hidden fields. This can include information such as items added to a shopping cart.

- *Session state* enables you to store and retrieve values for a user as the user navigates ASP.NET pages in a Web application.

- *View state* is used to preserve page and control values between client and server. It provides state information. *Note:* this only provides state information for a specific ASP.NET page.

# What data types can be stored in view state?

- Data is stored in hidden fields and can contain:
  - — Strings
  - — Integers
  - — Boolean values
  - — Array objects
  - — ArrayList objects
  - — Hash tables
  - — Custom type converters

  - (**Note:** you can store other types of data as well, but the class must be compiled with the Serializable attribute so that its values can be serialized for view state.)

# What are cookies?

- A *cookie* is a small bit of text that accompanies requests and pages as they go between the Web server and browser. The cookie contains information the Web application can read whenever the user visits the site.
  - Cookies will persist beyond the current session, which allows the application to save information for future sessions.
  - Cookies are convenient because they prevent the user from having to retype information on subsequent visits. However,  there are privacy issues surrounding their use.

## Lesson Review

- Discuss situations where you would use each state and why.
- One of the most used forms of state is session state. Why?
- When are cookies used? Why are they used?

LESSON 1.4

98-363 Web Development Fundamentals

# Understand Events and Control Page Flow

LESSON 1.4

98-363 Web Development Fundamentals

# Lesson Overview

How do events play a role in creating dynamic Web pages?

What events play a key role in the life cycle of a Web application or Web page?

In this lesson, you will review:

- Events and their role in Web applications
- Various concepts related to controlling page flow

# Application Life Cycle

Figure 1

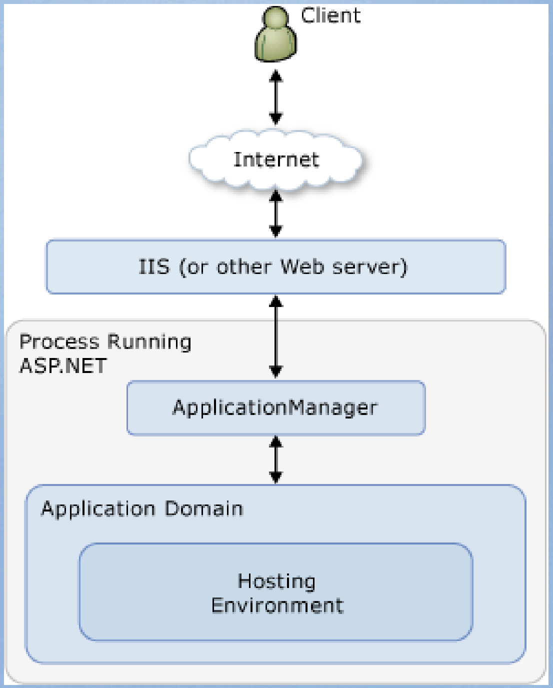
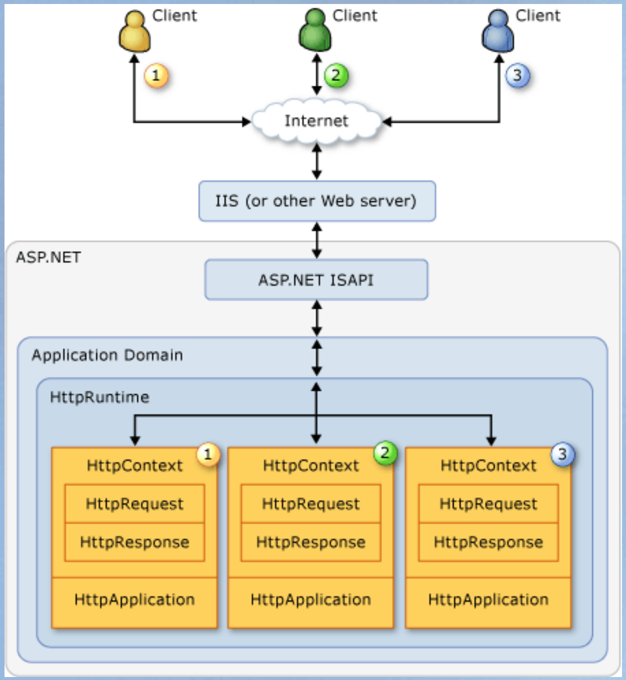


Figure 2



# Page Life Cycle

When an ASP.NET page is requested, a series of page events occurs.

These events always occur in the same order, which is referred to as the *page event life cycle*. An easy way to remember the sequence of events is SILVER:

S – Start
I – Initialize
L – Load
V – Validate
E – Event Handling
R – Render

# Lesson Review

- Life cycle
- Page event
- Application event
- Key players related to page control:

  *Cross-page posting, Response.Redirect, Server.Transfer, IsPostBack, AutoEventWireup*

## Assignment

- Complete student activity 1.4

LESSON 1.5

98-363 Web Development Fundamentals

# Understand Controls

# Lesson Overview

How do controls play a role in creating dynamic Web pages?

How do you decide which control is appropriate for a particular scenario?

In this lesson, you will review the appropriate use of:

- User controls
- Web controls
- Server controls
- Validation controls

# User Controls

User controls enable you to reuse code and common user interface (UI) components.

User controls have the following features:

- A user control resides in a Web Form page.
- A user control participates in the event life cycle for the Web Form.
- A user control has its own page logic.

# User Controls (continued)

- User controls are self-contained.
- None of the methods and properties of the user control conflict with the methods or properties of the hosting page.
- User controls can be used more than once in a hosting page without causing property and method conflicts.
- User controls can be written in a different language from the main hosting page**.**

# Web Controls

- Web controls serve as the base class that defines the methods, properties, and events common to all controls in the *System.Web.UI.WebControls* namespace.
- Web controls deliver content that renders in all commonly used browsers.
- You can control the appearance and behavior of a Web server control by setting properties defined in this class. For example, the background color and font color of a control are set by using the `BackColor` and `FontColor` properties, respectively.

# Server Controls

- ASP.NET Web server controls are objects on ASP.NET Web pages that run when the page is requested and that render markup to the browser.
- Can be similar to familiar Hypertext Markup (HTML) elements, such as buttons and text boxes.
- Other controls encompass complex behavior. Example: calendar controls and controls that you use to connect to data sources and display data.

# Validation Controls

- You enable validation of user input by adding validation controls to your page (for example, testing for valid dates or values within a range).
- Validation controls perform input checking in server code.
- Validation controls can be used with any controls that you put on an ASP.NET Web page, including both HTML and Web server controls.

LESSON 1.6

98-363 Web Development Fundamentals

# Understand Configuration Files

# Lesson Overview

Before you can deploy your ASP.NET Web application, you must organize the Web application settings in the Machine.config and Web.config files.

In this lesson, you will review:

- Machine.config files
  - Machine-level settings
- Web.config files
  - Application- and directory-level settings

## Guiding Question

How many Web.config files does your application use?

## Overview of Configuration Files

What are Machine.config and Web.config files?

- Extensible Markup Language (XML) files
- CamelCase  (first letter cap—cap for each new word)
- Extendable

# Machine.config

- The Machine.config file is located in the following directory:

C:\Windows\Microsoft .NET \Framework\version\CONFIG\Machine.config.

- Advantage: Storing settings in the Machine.config file can make your system easier to maintain because you have only one configuration file to search, edit, and maintain.

- Disadvantage: When you deploy a Web application to a new server, the Web application settings that are in the Machine.config file are not copied to the new Web server.

# Web.config

- Use the Web.config file to share information and settings between Web pages.
- A single Web.config file is typically located in the root folder of the Web application.
- Place additional Web.config files in the folder of the virtual directory to which they belong.

## Assignment

Complete student activity 1.6

# Lesson Review

- The settings in the Machine.config file apply to all ASP.NET directories and subdirectories.
- Additional configuration information is contained in the Web.config files, which are located in the same directories as the application files.
- Child directories inherit the settings of the parent directories unless the settings are overridden by a Web.config file in the child directory.

LESSON 1

98-363 Web Development Fundamentals

# Complete Quia Test:

MTA WebFund1 Test