Basics of Web & Web Sites

HTTP

- HTTP → HyperText Transfer Protocol.
- HTTP is a text-based stateless protocol that defines how Web browsers and Web servers communicate.
- HTTP is a request/response protocol between clients and servers.
- An HTTP server listens on port 80. Port can be reconfigured.

HTTP Request

- When a browser make a request for a URL, it sends a request packet to the web server.

e.g. http://www.iconnectgroup.com/default.htm

```
GET /default.htm    HTTP/1.1
Host: http://www.iconnectgroup.com/
```

- The HTTP request first line has HTTP command to execute, the resource locator & the HTTP version.

HTTP Request

- An HTTP request has number of headers.
- An HTTP header is a line of text that provides additional information about the request.
- Some of the headers
  - Host: Identifies the server.
  - User-Agent : Identifies the type of requesting browser .
  - Connection: Closes a connection or keeps a connection alive.
  - If-Modified-Since: Provides client-side cache validation.

GET & POST

- GET and POST are the most commonly used HTTP methods.
- The GET verb means retrieve whatever information is identified by the request URL.
- The POST verb is used to request that the origin server accept the content enclosed in the request and process it .

HTTP Response

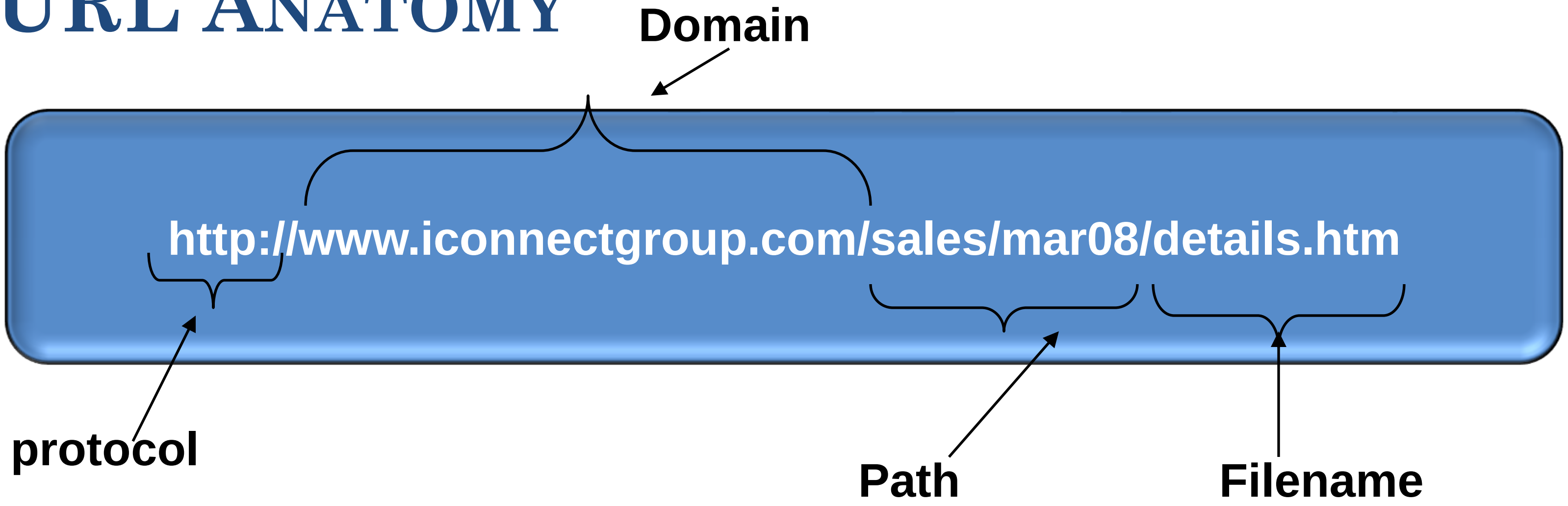
- The HTTP response starts with the message's protocol version and an exit code to indicate success or failure.
- HTTP response then contains the HTTP headers like content length, type, server etc.
- Message body follows headers with a newline skipped.

HTTP Response

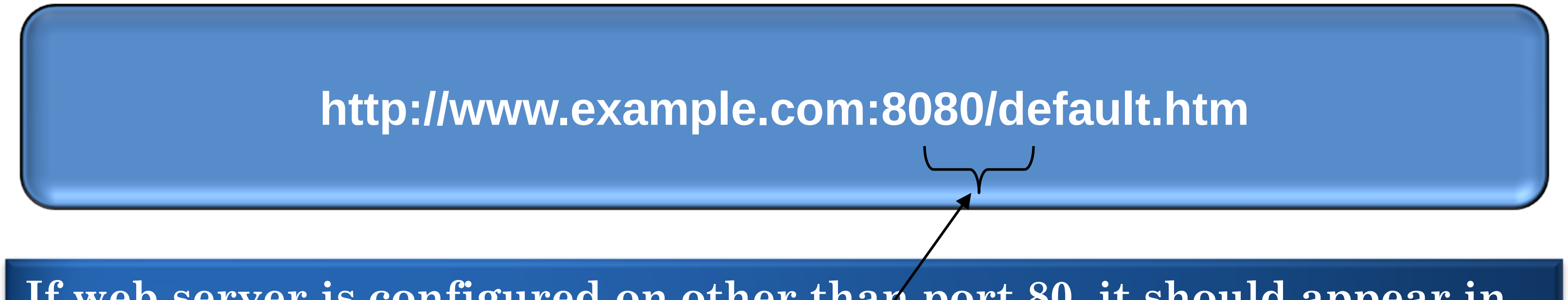
```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Type: text/html
Content-Length: 51

<html><body><h1>ASP.NET is cool!</h1></body></html>
```

- The code *200* means that all went OK with the request.
- Content-type is in Multipurpose Internet Mail Extensions (MIME) type.(e.g.text/xml)

URL ANATOMY



•URLs are the standard addressing system for resources on the Web.



If web server is configured on other than port 80, it should appear in URL

Web Page

- Web page can contain *text, graphics, forms, audio and video files*, and *interactive games*.
- Every Web page is created in HTML .
- HTML holds a Web page together; the graphics, content, and other information.
- HTML files that **produce** Web pages are text documents.
- Web pages aren't *merely* text documents.

HyperText Markup Language(HTML)

- HTML is a collection of instructions that one include along with pointers to content in a text file that specifies how the page should look and behave.
- Pointers are called *hyperlinks*.
- Web browsers read instructions written in HTML and renders page contents.
- Each Web browser interprets HTML in its own way & hence page appearance may differ.

XHTML

- XHTML is a reformulation of HTML 4.01 in XML.
- XHTML elements must be **properly nested**
- XHTML elements must always be **closed**
- XHTML elements must be in **lowercase**
- XHTML documents must have **one root element**.

HTML vs. XHTML

- The original formulation of HTML has some irregularities.
- XHTML uses an extremely regular and predictable syntax that's easier for software to handle.
- Most HTML & XHTML markups are identical.
- HTML and XHTML markup must be used differently in some cases.

Types Of (X)HTML

- The HTML and XHTML specifications use *Document Type Definitions* (DTDs) written in the Standard Generalized Markup Language (SGML).
- Previously HTML used elements for formatting as well as page's structure.
- Now CSS are used for formatting
- And HTML only describe a page's structure.
- The results are three types of (X)HTML
  - (X)HTML Transitional
  - (X)HTML Strict
  - (X)HTML Frameset

Types of (X)HTML

- (X)HTML Transitional
  - Formatting elements in XHTML Transitional are considered obsolete.
  - Allows use of formatting elements.
- (X)HTML Strict
  - It doesn't include any elements that describe formatting.
  - The type is designed to let CSS drive the page formatting.

Types of XHTML

- (X)HTML Frameset
  - Markup that allows you to display more than one Web page or resource at a time in the same browser window.
  - Includes frames.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

HTML & XHTML

- Both has three components
  - **Elements:** Identify different parts of an HTML page by using tags
  - **Attributes:** Information about an instance of an element
  - **Entities:** Non-ASCII text characters.

Elements

- Elements that describe content use a tag pair to mark the beginning and the end of the element.

  <tag>…</tag>

- Elements that insert something into the page are called empty elements and use single tag.

  <tag />

- In XHTML, all empty elements must end with a slash before the closing greater-than symbol.

```
<p>Welcome to iConnect</p>
<img src=“logo.jpg” width=“75” height=“100” />
```

Attributes

- Attributes allow variety in how an element describes content or works.
- e.g. <img /> uses *src* attribute to specify the location of image.
- One include attributes within the start tag of the element.
- XHTML syntax rules decree that attribute values must always appear in quotation marks.

Entities

- Non-ASCII characters like *trademark symbols, fractions,* and *accented characters* etc.

- Entities starts with & and end with ;.

  e.g. &eaccute; →é

  &uuml; →ü

  &copy; →©

  &lt;→< &gt; →> &amp; → &

Comments in HTML

- Comment starts with <!---

- It ends with --->

```
<!--------- Welcome to ASP.NET Class
                        iConnect Software Center
                        ---------------------------------->
```

Lists in HTML

- <ul>…</ul>
  - Creates an unordered list.
  - <li>…</li> constitutes item of list.
- <ol>…</ol>
  - Creates an ordered list.
  - <li>…</li> as above i.e. defines list item.

Tables in HTML

- <table> tag used to create table & it ends with </table>
- <tr>…</tr> defines the table row. Always appear in <table>…</table>.
- <td>…</td> defines the table cell. Appears in <tr>…</tr>
- *rowspan* attribute specifies the number of rows spanned by the current cell. For columns, use *colspan*.

Page Body Backgorund

- <body> tag has got an attribute *bgcolor* which sets background color of web page.
- *background* attribute is used to set the background image for a web page.

HTML Attribute	***Effect***
TEXT="#RRGGBB"	**Changes the color of the body text**
LINK="#RRGGBB"	**Changes the color of the link**
ALINK="#RRGGBB"	**Changes the color of the active link**
VLINK="#RRGGBB"	**Changes the color of the visited link**

Font Specification

- <font> tag has attributes *face* & *size*.
- *face* Sets the typeface NAME; a list of font names can be specified.
- *size* Changes the font size on a scale from 1 to 7.
- *color* Specifies the color for the text.

LINKS

- Anchor tag <a> used with attributes *href* and *name*.

- *href* Enables users to jump either to page within the same Web site or to other page out on the Internet.

- *name* Labels a spot within a document. That spot can then be part link URL so that readers can jump directly to it.

Image Manipulation

HTML Tag or Attribute	*Effect*	*Example*
<IMG SRC="...">	**Inserts an image**	**<IMG SRC="myPicture.gif">**
ALT="..."	**Specifies the text to display if the image isn't displayed**	**<IMG SRC="myPicture.gif" ALT="iConnectGroup">**
BORDER=n	**Controls the thickness of the border around an image in pixels**	**<IMG SRC="myPicture.gif" ALT="iConnectGroup" BORDER=5>**
HEIGHT=n	**Specifies the height of the image in pixels**	**<IMG SRC="picture.gif" HEIGHT="200">**
WIDTH=n	**Specifies the width of the image in pixels**	**<IMG SRC="picture.gif" WIDTH="150">**

Image Alignment

ALIGN="bottom"	Aligns the bottom of the image with the baseline of the current line
ALIGN="left"	Allows an image to float down and over to the left margin (into the next available space); subsequent text wraps to the right of that image
ALIGN="middle"	Aligns the baseline of the current line with the middle of the image
ALIGN="right"	Aligns the image with the right margin and wraps the text around the left
ALIGN="top"	Aligns the text with the top of the tallest item in the line
HSPACE=n	Controls the horizontal space (white space) around the image in pixels
VSPACE=n	Controls the vertical space (white space) around the image in pixels

Forms

- HTML forms are online versions of hard-copy forms that have check boxes and blanks to fill in.

- The HTML <*form*> tag is the only element authorized to transmit client-side data to the server.

HTML Tag or Attribute	*Effect*	*Example*
<FORM...> ... </FORM>	Encloses the entire form.	<FORM METHOD ="POST" ACTION="http://www.yourServer.yourScript" > form components</FORM>
ACTION="..."	Identifies what should happen to the data after the form is submitted.	<FORM METHOD ="POST" ACTION="http://www.yourServer.yourScript" > form components</FORM>
METHOD="..."	Identifies methods; valid options are GET or POST – one is required.	<FORM METHOD="POST" ACTION="http://www.yourServer.yourScript" > form components</FORM>

INPUT TAG

HTML Tag or Attribute	*Effect*	*Example*
<INPUT...>	Identifies some type of input field.	<INPUT TYPE="SUBMIT">
SIZE=n	Displays field *n* characters wide.	<INPUT TYPE="SELECT" SIZE=4>
TYPE=".."	Indicates the type of field. Valid types are TEXT, PASSWORD, CHECKBOX, RADIO, SUBMIT, RESET, FILE, IMAGE, BUTTON, and HIDDEN.	<INPUT TYPE="RADIO">
VALUE=".."	Indicates the value of the button (and the label for Submit and Reset).	<INPUT TYPE="BUTTON" VALUE="Click this button">

Input Tag

CHECKED	Shows which item is selected by default (used with check box and radio button).	<INPUT TYPE="CHECKBOX" CHECKED>
MAXLENGTH=n	Indicates the maximum number of characters in the field width.	<INPUT TYPE="TEXT" MAXLENGTH=25>
NAME=".."	Indicates the name of the field.	<INPUT TYPE="TEXT" NAME="HomeAddress">

Selection Lists

HTML Tag or Attribute	*Effect*	*Use in Pairs?*
<SELECT...> ... </SELECT>	Provides a list of items to select	Yes
MULTIPLE	Indicates that multiple selections are allowed	No
NAME="..."	Indicates the name of the field	No
SIZE=n	Determines the size of the scrollable list by showing *n* options	No
<OPTION...>	Precedes each item in an option list	Yes, optionally
SELECTED	Identifies which option is selected	No by default
VALUE="..."	Indicates the value of the field	No

TextArea

HTML Tag or Attribute	*Effect*	*Use in Pairs?*
<TEXTAREA ...> ...</TEXTAREA>	**Encloses a multiline text field. The enclosed text is the value displayed in the field.**	**Yes**
COLS=n	**Indicates the number of columns in the field.**	**No**
NAME="..."	**Indicates the name of the field.**	**No**
ROWS=n	**Indicates the number of rows in the field.**	**No**

JAVASCRIPT

- Javascript is a client-side scripting language & is case-sensitive.
- Client-side scripting is used to make pages interactive after they are sent to the browser.
- A common usage of client-side scripting is to check the data is valid or not.

Javascripting

- All JavaScript code must be put within the <script> tag.
- *language* attribute is set to “javascript”
- *type* attribute is set to “text/javascript”
- Inside script tag put HTML comment & within it put the Javascript code for backward compatibility.

JAVASCRIPTING

- One can save all the code without a </script> element in a special file (*.js) and then link it to the page using <script> element and its *src* attribute.

- <noscript>..</noscript> should be used. Its content are displayed if Javascript is disabled or unavailable with the browser.

```
<script src="myscript.js"> </script>
```

JAVASCRIPTING

```
<script type="text/javascript" language="javascript">
<!–
function say(text)
{
            alert(text);
}
//--->
 </script>
```

- The JavaScript code that you write outside of a function will be run when page loads.

- One can put a <script> element anywhere on the page and as many times as required.

Loops & Conditional Statements

- for loop
- do…while loop
- while loop
- if(condition){ statements}
- if(condition){ statements} else {statements}
- if(condition){ statements} else if(condition) {statements}else {statements}

JAVASCRIPT OBJECT

- JavaScript supports working with objects.

- String Object

  - The String object, used to represent text, is the most common object in JavaScript.

```
var myText1 = "This is my text.\n";
var myText2 = new String("This\t is tab.");
alert(myText1 + " " + myText2);
```

  - *toUpperCase* & *toLowerCase* functions of string object convert the text to upper or lower case.

String Object

- *indexOf* returns the index of specified character in the string.

```
pos1 = mailAddress.indexOf("@");
```

- *length* property returns the length of the string.

```
len = mailAddress.length;
```

- *substr* returns substring from specified character & of specified length.

```
firstThree = mailAddress.substr(0, 3);
```

Document Object

- The document object is used to represent the current web page.
- *bgColor* property is used to set background color. Define colors using hexadecimal code or standard names.
- *title* property access to the title of the page.

```
document.bgColor = color;//color="#ABCDEF" or "black"
```

```
document.title="iConnect Software";
```

DOCUMENT OBJECT

- *write* method is used to add text directly to web page.
- The text is added right to where the <script> element is positioned on the page.

```
<b>
<script type="text/javascript" language="javascript">
            document.write("Bold text.");
</script>
 </b>
```

Page Elements Access

- Every HTML element can have an ID attribute.
- Page elements can be referenced using the value of ID attribute or by using *document.getElementById()* function.
- *getElementById()* returns a reference to the element or *null*.

Page Elements Access

```
<html>
<body>
<div id="myDiv"> Hello </div>

<script language="javascript"type="text/javascript">

alert(myDiv.innerText);

var divtag=document.getElementById("myDiv");
if(!divtag) alert(divtag.innerText);
</script>

</body>
</html>
```

Accessing Attributes of Elements

- Element reference along with an attribute in lowercase separated by dot(.) allows to modify and access the value of the attribute.

- Style attributes can be accessed by adding ".style" between element reference & attribute.

- To access the value of the form field, one can use its *value* property.

```
table id="myTable" width="300" height="500" border="0"
              cellpadding="5" bgcolor="red">
<tr><td>Hello!</td></tr>
</table>
```

```
table = document.getElementById("myTable");

TableWidth = table.width;
TableHeight = table.height;
TableBorder = table.border;
TableCellPadding = table.cellPadding;
TableBgColor = table.bgColor;
```

```
<table id="myTable”
 style="width: 300px; height: 500px; border: solid 1px black;
              background-color: Red;">
 . . . </table>
```

```
TableWidth = table.style.width;
```

Events

- JavaScript is an event-driven scripting language.
- One can interact with all the events in the browser and react to them.
- Note that all events which affect the *document* element can also be applied to any element on the page.
- One can attach an event handler to an HTML element and define the action that will start when the event occurs.

```
<body onload="DoWelcome()">…</body>
```

Javascript function

EVENTS

Event	Elements affected	What starts this event?
onabort	Image	interruption of image loading (user has clicked on another link or STOP in the browser)
onblur	Button, Checkbox, FileUpload, Frame, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea, Window	when element loses focus (another element is selected)
onchange	FileUpload, Select, Text, Textarea	change of data inside an element
onclick	Button, Document, Checkbox, Link, Radio, Reset, Submit	click on an element
ondblclick	Area, Document, Link	double click on an element

Events

onfocus	Button, Checkbox, FileUpload, Frame, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea, Window	when an element gets focus (opposite of onblur)
onkeydown	Document, Image, Link, Textarea	pressing down a key on the keyboard
onkeypress	Document, Image, Link, Textarea	pressing (and releasing) a key on the keyboard
onkeyup	Document, Image, Link, Textarea	releasing a key on the keyboard
onload	Frame, Image, Layer, Window	end of loading
onmousedown	Button, Document, Link	pressing mouse button
onmouseout	Area, Layer, Link	mouse pointer exiting the element's area

Events

onmouseover	Area, Layer, Link	mouse pointer moving over an element
onmouseup	Button, Document, Link	releasing mouse button
onmove	Frame, Window	window or frame moving
onreset	Form	resetting the form
onresize	Frame, Window	changing size of a window or a frame
onsubmit	Form	submitting a form
onunload	Frame, Window	unloading a document (leaving the page or closing the window)

Events

ondragdrop	Frame, Window	drag and drop of a shortcut or a file to the browser window
onerror	Frame, Image, Window	error in the script or while loading an image (e.g. image not found)
onselect	Textarea	selecting text

WINDOW OBJECT

- JavaScript can be used to open new browser windows and to configure their properties.
- The *open* method is used to open new windows.
- Three parameters to *open*:
  - Address of the page that will be displayed in the newly opened window.
  - The name of the window.
  - The properties of the new window.
- The result of the *open* method is a reference to the newly opened window.

WINDOW OBJECT

```
myWin = window.open("myPage.html", "myPage", "height=300,width=400");
```

OR

```
<a href="mySecondPage.html" target="myWin">My link</a>
```

- Methods
  - close(): Closes the window.
  - moveTo(x,y): Moves the window to the location set by the two parameters.
  - resizeTo(width,height): Resizes the window.

window Object Properties

Property	Description
fullscreen	Opens the window in full-screen mode
height	Height of the new window in pixels. Minimum is 100.
location	Displays the address bar.
menubar	Displays the menu bar (File, Edit…).
resizable	Enables new window to be resizable.
scrollbars	Sets which scrollbars (vertical, horizontal) will be displayed, if necessary.
status	Display the status bar.
toolbar	Display the browser toolbar.
width	Width of the new window in pixels. Minimum is 100.

Web as a Development Platform

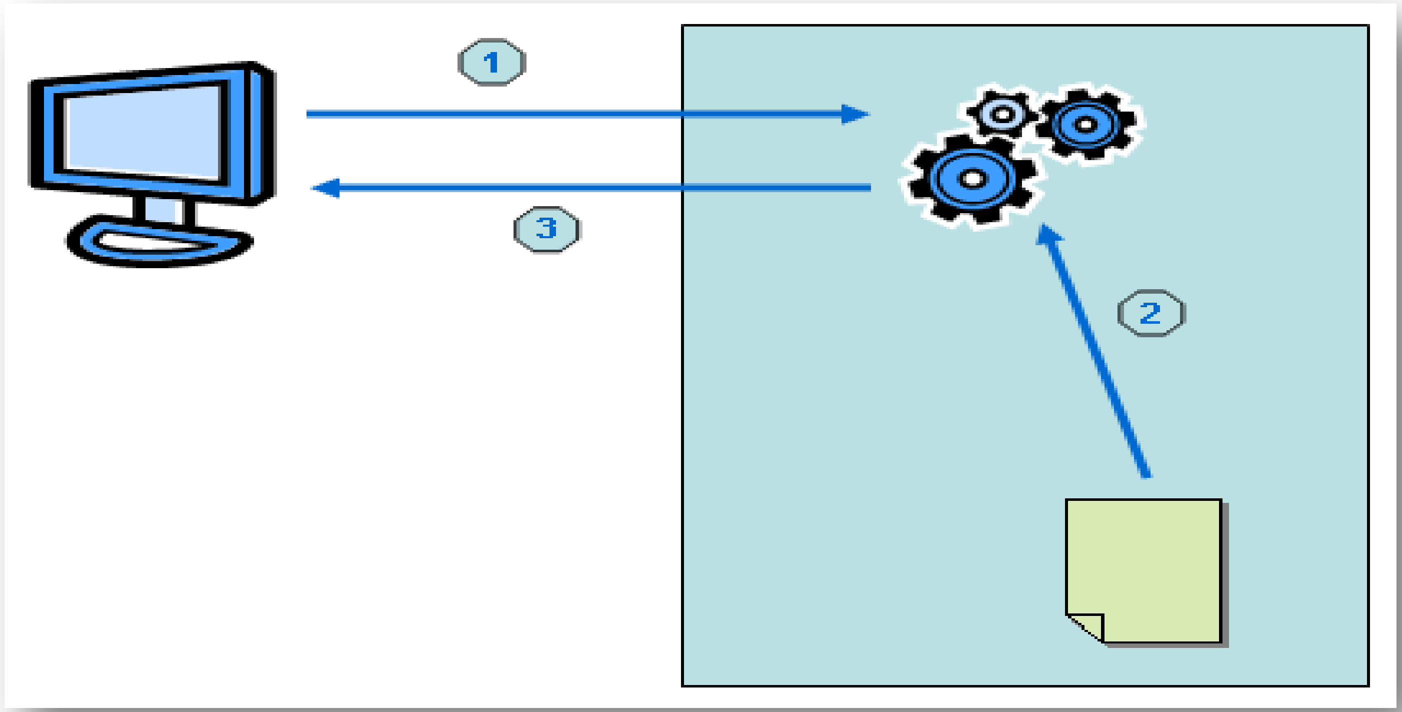
- The browser sends the GET request to the server.

```
GET /index.html HTTP/1.1
Host: www.example.com
Accept-Language: en
```

- When the Web server receives this request, it locates the root folder of the Web site on its hard drive, looks for a file called index.html.

- Web server sends the contents of the file back to the browser inside the HTTP response.

Web as a Development Platform



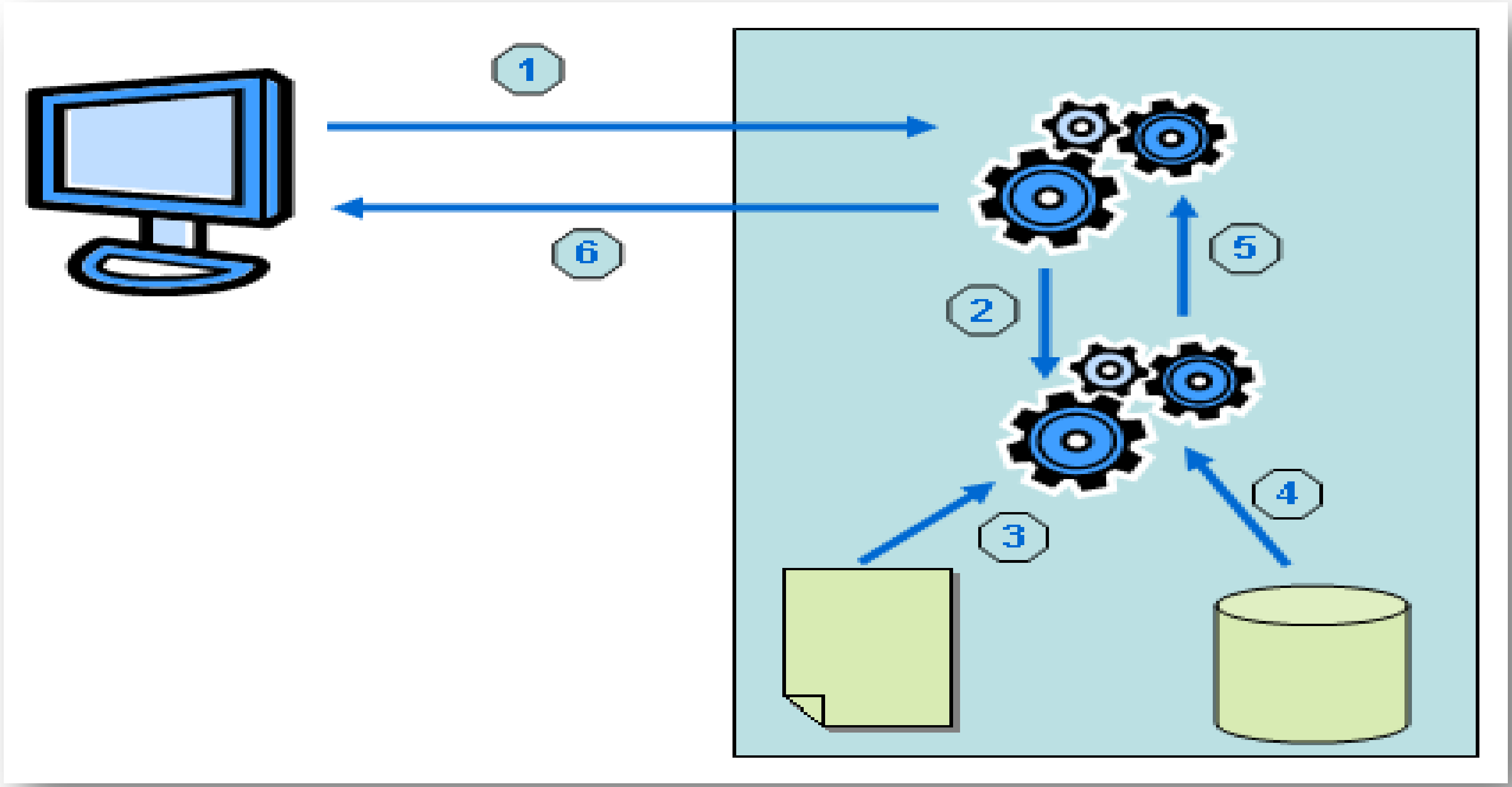
Every time the Web site owners want to change the content, they can upload a replacement for index.html

Web as a Development Platform

- A Web application that works with the Web server to create the page dynamically each time a browser requests it.

- A web server receives the request. The Web server recognizes that the file extension is something different than .html – for example, the file extension might be .aspx, which is an ASP.NET file – and it calls the appropriate Web application to process the request.

Web as a Development Platform



Web as a Development Platform

- The Web application reads the requested file from the file system.
- Web application processes the file and runs any instructions that it finds within.
- The Web application dynamically generates some HTML according to the instructions in the file & returns generated HTML to web server.
- The Web server forwards this HTML content to the browser.