HTML5 & CSS3 Intro

Mohamed Mukthar Ahmed

HTML5 & CSS3 Intr

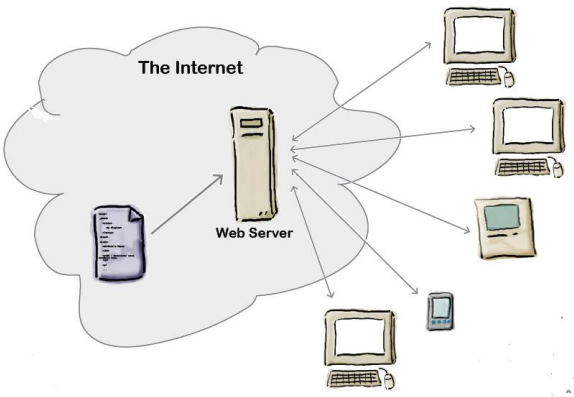
Outline

- The Web
- WebApp Components
- Web Server & Web Browser
- Static Web Pages
- Dynamic Web Pages
- JavaScript in Web Development
- HTML
  - Basic Structure
  - Some Basic Tags
  - Understanding Attributes
  - Nesting Elements
- CSS – Cascaded Style Sheets
  - Syntax & Selectors
- Getting Organized

Mohamed Mukthar Ahmed

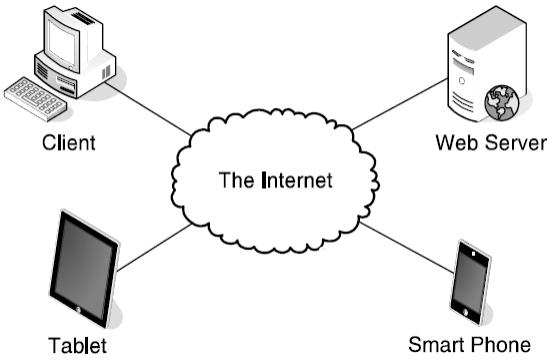
The Web

- The **Web** – Universal form of communication
- To use the Web effectively, we got to know about **HTML**



Web Application Components

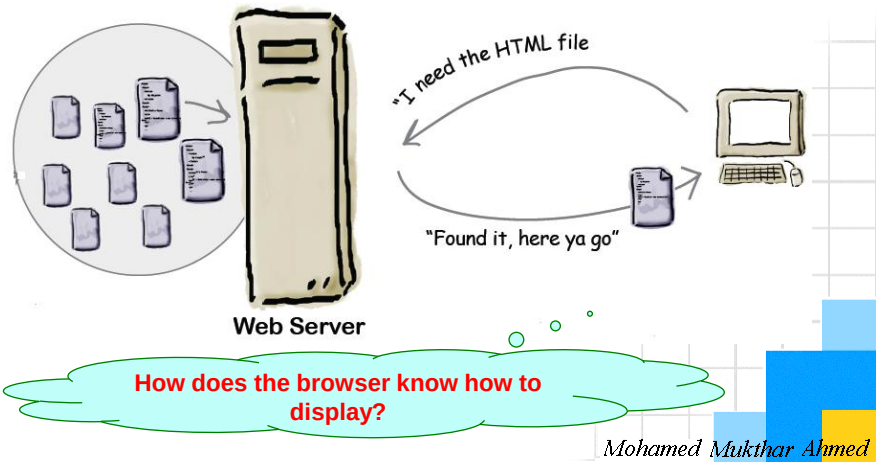
- Components of a Web Application



- A Web Application consists of clients, a web server and a network.

Web Server

- The **Web Server** – accepts **requests** from Web browsers and **responds**

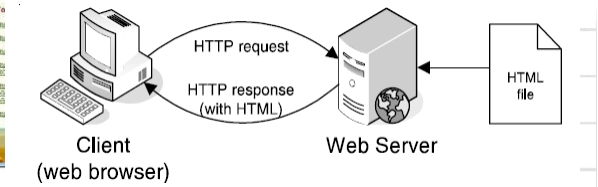


Web Browser

- Web servers store and serve web pages, which are created from HTML and CSS.
- Web Browsers retrieve pages and render their content based on the HTML and CSS

Static Web Pages

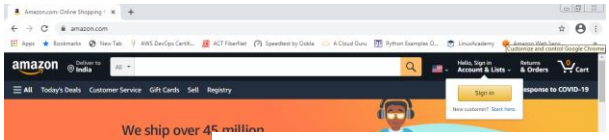
A static web page at http://www.valleytownhall.com/toobin.html

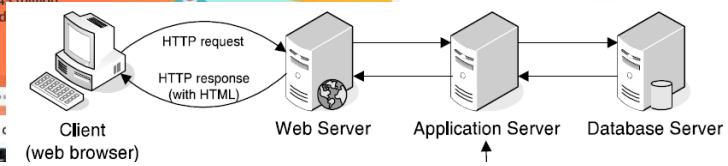


- A *static web page* is an HTML document that's stored on the web server and doesn't change. The filenames for static web pages have .htm or .html extensions.
- When the user requests a static web page, the browser sends an *HTTP request* to the web server that includes the name of the file that's being requested.
- When the web server receives the request, it retrieves the HTML for the web page and sends it back to the browser as part of an *HTTP response*.
- When the browser receives the HTTP response, it *renders* the HTML into a web page that is displayed in the browser.

Mohamed Mukthar Ahmed

Dynamic Web Pages



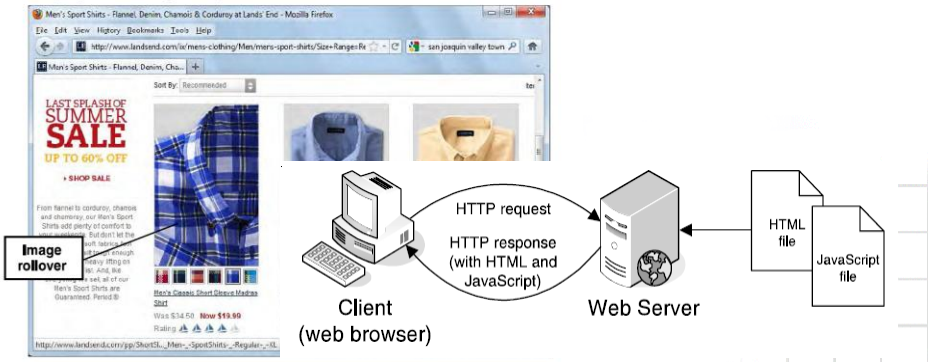


- A *dynamic web page* is a web page that's generated by a server-side program or script.
- When a web server receives a request for a dynamic web page, it looks up the extension of the requested file to find out which *application server* should process the request.
- When the application server receives a request, it runs the specified script. Often, this script uses the data that it gets from the web browser to get the appropriate data from a *database server*. This script can also store the data that it receives in the database.
- When the application server finishes processing the data, it generates the HTML for a web page and returns it to the web server. Then, the web server returns the HTML to the web browser as part of an HTTP response.

Mohamed Mukthar Ahmed

JavaScript in Web Development

A web page with image swaps and rollovers



- *JavaScript* is a *client-side scripting language* that is run by the *JavaScript engine* of a web browser and controls the operation of the browser.
- When the browser requests an HTML page that contains JavaScript or a link to a JavaScript file, both the HTML and the JavaScript are loaded into the browser.
- Because JavaScript runs on the client, not the server, it provides functions that don't require a trip back to the server. This can help an application run more efficiently.

Mohamed Mukthar Ahmed

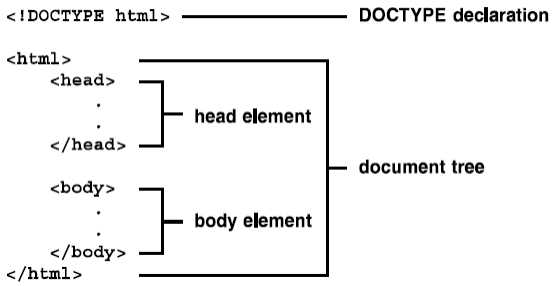
HTML

- To use the Web effectively, we got to know about **HTML**
- HTML is an abbreviation for HyperText Markup Language and is used to structure your web page
- Using HTML, we mark up content with tags to provide structure
- The basic **structure** of any HTML document will look as shown below:

Mohamed Mukthar Ahmed

Basic HTML Strutcure

The basic structure of an HTML5 document

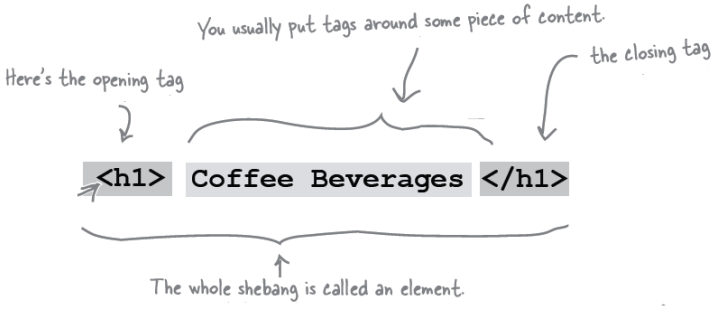



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>San Joaquin Valley Town Hall</title>
    </head>
    <body>
        <h1>San Joaquin Valley Town Hall</h1>
        <p>Welcome to San Joaquin Valley Town Hall.</p>
        <p>We have some amazing speakers in store for you this season!</p>
        <p><a href="speakers.html">Speaker information</a></p>
    </body>
</html>
```

Mohamed Mukthar Ahmed

Some Basic Tags

- **Heading** tags
  - <H1> <H2> <H3> … <H6>
- **Paragraph** tag
  - <p>
- Tags dissected



Mohamed Mukthar Ahmed

Tags Dissected

- Remember
  - Element = Opening Tag + Content + Closing Tag
- Opening tags can have attributes
- Closing tags have a "/" after the left angle bracket, in front of the tag name

The Line Break Element

- Whitespaces and line breaks are not displayed by the browser
- To insert carriage return use the **
** element
- **Recall**
  - Tags dissected
  - Element = Opening Tag + Content + Closing Tag
- When an element doesn't have any real content by design, we just use a shorthand to represent the element
- They are called **void elements**

The Anchor Element

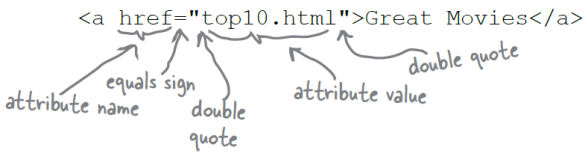
- Use the **<a>** – element to create a hypertext link to another web page
- The content of the **<a>** element becomes clickable
- The **href** attribute tell the browser the destination of the link

Understanding Attributes

- Attributes give you a way to specify additional information about an element

```
<style type="text/css">
<a href="irule.html">
<img src="sweetphoto.gif">
```

Do this (best practice)



Not this

```
<a href=top10.html>Great Movies</a>
```

No double quotes around the attribute value

Understanding Attributes

- Common attributes for identifying HTML elements.

An opening tag with an id attribute

```
<div id="page">
```

An opening tag with a class attribute

```
<a href="contact.html" title="Click to Contact Us" class="nav_link">
```

- *Attributes* can be coded within opening or empty tags to supply optional values.
- A *Boolean attribute* represents either an on or off value.
- The id attribute is used to identify a single HTML element so its value can be used for just one HTML element.
- A class attribute with the same value can be used for more than one HTML element.

Nesting Elements

- Putting one element inside another is called nesting

- While nesting make sure your tags match

How can we display a HTML tag?

CSS – Cascaded Style Sheet

- CSS – used for presentation
  - Describes how the content should be presented

- To add style use the <style> element
  - Generally placed inside the head of the HTML

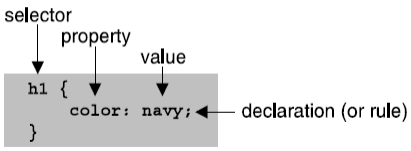
```
<html>
        <head>
                <title>                    </title>
                <style type="text/css">


                </style>
        </head>
```

Mohamed Mukthar Ahmed

CSS Syntax

- A CSS consists of rule sets.

The parts of a CSS rule set



- A CSS *rule set* consists of a selector and a declaration block.
- A CSS *selector* consists of the identifiers that are coded at the beginning of the rule set.
- A CSS *declaration block* consists of an opening brace, zero or more declarations, and a closing brace.
- A CSS *declaration* (or *rule*) consists of a *property*, a colon, a *value*, and a semicolon.
- To make your code easier to read, you can use spaces, indentation, and blank lines within a rule set.
- CSS *comments* begin with the characters /* and end with the characters */. A CSS comment can be coded on a single line, or it can span multiple lines.

Mohamed Mukthar Ahmed

CSS Selectors

Description

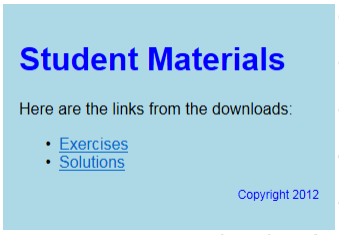
- To code a selector for an HTML element, you simply name the element. This is referred to as a *type selector*.
- If an element is coded with an id attribute, you can code a selector for that id by coding a pound sign (#) followed by the id value, as in #main.
- If an element is coded with a class attribute, you can code a selector for that class by coding a period followed by the class name, as in .base_color.

CSS Example

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Basic Selectors</title>
        <link rel="stylesheet" href="basic_selectors.css">
    </head>

    <body>
        <div id="main">
            <h1 class="base_color">Student Materials</h1>
            <p>Here are the links from the downloads:</p>
            <ul id="links">
                <li><a href="exercises.html">Exercises</a></li>
                <li><a href="solutions.html">Solutions</a></li>
            </ul>
            <p id="copyright" class="base_color">Copyright 2012</p>
        </div>
    </body>
</html>
```

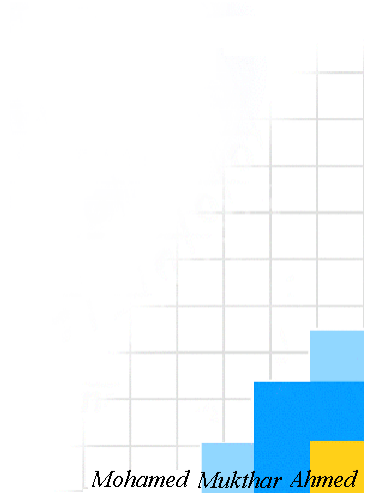
CSS Example

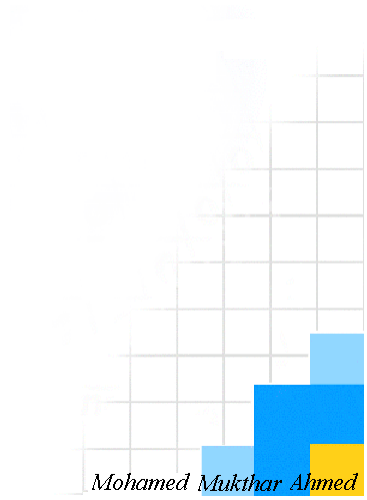
```
1  body {
2      font-family: Arial, sans-serif;
3  }
4  #main {
5      width: 300px;
6      padding: 1em;
7      background-color: lightblue;
8  }
9  #copyright {
10     font-size: 75%;
11     text-align: right;
12 }
13 .base_color {
14     color: blue;
15 }
```



Getting Organized

- Things are much more manageable if you organize your web pages, graphics, and other resources into a set of folders.
- Tell the browser the new location
- Plan the paths
  - Identify the source and the destination
  - Trace a path from the source to the destination


Mohamed Mukthar Ahmed


Mohamed Mukthar Ahmed