

Web Technologies

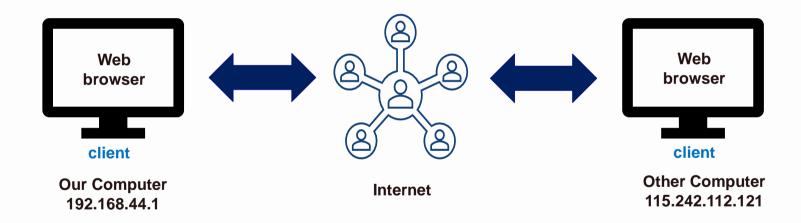




Introduction



### How the INTERNET works



Every resource over internet is uniquely identified by URI i.e. Uniform Resource Identifier It can be IP address or URL (Uniform Resource Locator)

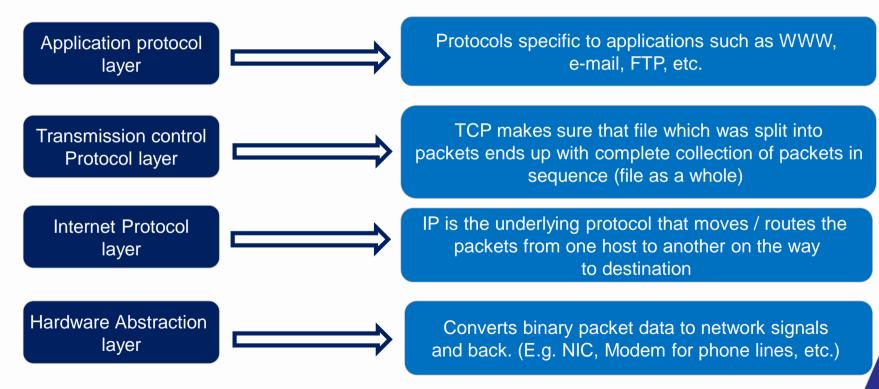


### The basis of communication over INTERNET

- Because the Internet is a global network of computers, each computer connected to the Internet must have a unique address.
- Internet addresses are in the form **nnn.nnn.nnn** where nnn must be a number from 0 255.
- This address is known as an IP address.
- The computers / computing devices / other resources communicate with each other via INTERNET



### How the INTERNET works





# DNS (Domain Name System)

- The DNS is a distributed database which keeps track of computer's names and their corresponding IP addresses on the Internet.
- The device which provides DN Services are known as DNS Servers
- No DNS server contains the entire database; they only contain a subset of it.
- If a DNS server does not contain the domain name requested by another computer, the DNS server redirects the requesting computer to another DNS server.



#### **Uniform Resource Locator**

- Every resource available on the Web HTML document, image, video clip, program, etc. has an address that may be accessed by a Uniform Resource Locator
- In Web Computing, a Uniform Resource Locator (URL) is a specific character string that constitutes a reference to an Internet resource.



#### More about URLS

Actual Resource Port Number (Optional): Protocol / Scheme: The name of the content server supports Tells the server being requested. If this ports. Actually which many part of URI is left out. application is server communication most web servers will identified by ports which protocol will be look for index.html is actually listening at used port http://www.IPLt20.com:80/allteams/select/mumbaiindians.html Server : Path: Unique name The path of the Server we're Resource being looking for. This will requested on server mapped be Query String (Not Shown): unique IP Address Some extra parameters We can also use would be appended starting directly IP address with a ? And with parameter in some cases (Name Value Pair) seperated by &

### Some more variants of URL

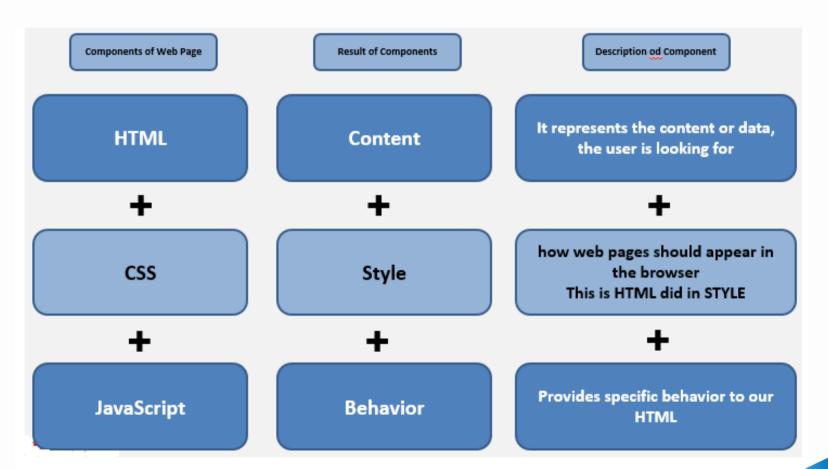
e.g. mailto:bob@example.com

- This will start an e-mail composer with the address bob@example.com in the "To" field
- Some schemes that require authentication allow a username, and perhaps a password too, to be embedded in the URI.

e.g. scheme://username:password@domain:port/path?query\_string#fragment\_id



# Anatomy of a web page





### Fundamentals of Web Technology

As far as the understanding of WEB is concerned, we have 2 sides of every web application:

- Client Side Make a request for specific resource
- Server Side Attends that particular request & provides information according to request submitted by client

The type of response generated by server can be of 2 types:

- Static Same for all the users all the time , pre-generated & Pre-defined Plain HTML
- Dynamic Keep changing every time , HTML generated programmatically Based on different input parameters



### Hypertext Text Transfer Protocol - HTTP

- HTTP is a protocol which has web specific features that works on Request Response Manner.
  - A Sequence of Browser Request -> Server Response
- It is a protocol used to deliver virtually all files and other data (collectively called resources) on the World Wide Web.
- A browser is an HTTP client because it sends requests to an HTTP server(Web server)
- The standard (and default) port for HTTP servers to listen on is 80.



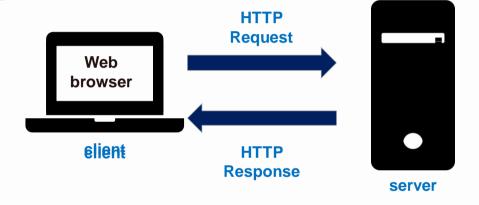
# HTTP – A Request – Response Protocol

#### Key Elements of Request Stream:

- 1. HTTP Method The action to be performed
- 2. The Page to Access a URL
- 3. Form Parameters Arguments to method

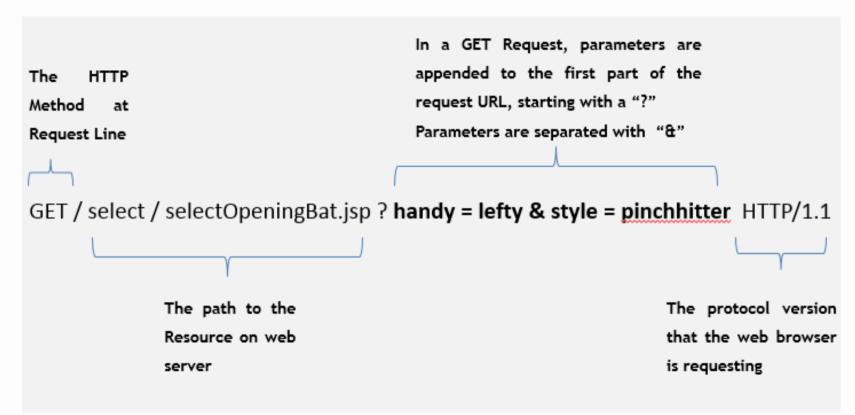
#### Key Elements of Response Stream:

- 1. A Status Code For whether request was successful
- 2. Content Type Text, Picture or HTML
- 3. The Content Actual HTML, Image





# Anatomy of HTTP **GET** REQUEST





# Anatomy of HTTP **GET** REQUEST

```
GET / select / selectOpeningBat.jsp ? handy = lefty & style = pinchhitter HTTP/1.1
```

HOST: www.IPLt20.com

User Agent: Mozilla/5.0, IE 5, CHROME 2.3, Netscape/7.1, Safari 44

Accept: text - HTML / XML XSD - DTD/ image - jpeg - gif - bmp / video - flv / mp4

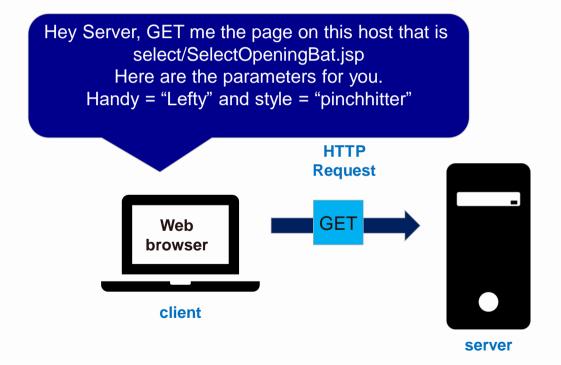
Accept Language : en – US / en - GEN

Accept CharSet: UTF - 8 / UTF - 16 / UTF - 32

**GET Request Header** 

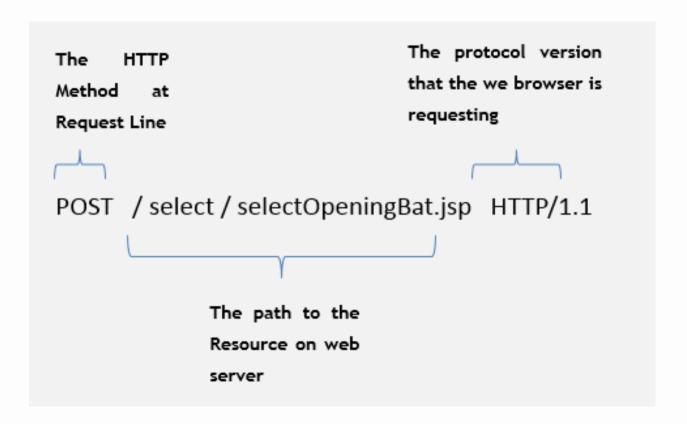


### HTTP GET REQUEST in Action





# Anatomy of HTTP **POST** REQUEST





# Anatomy of HTTP **POST** REQUEST

```
POST / select / selectOpeningBat.jsp HTTP/1.1
     HOST: www.IPLt20.com
     User Agent: Mozilla/5.0, IE 5, CHROME 2.3, Netscape/7.1, Safari 44
     Accept: text / HTML/ XML - XSD - DDTD / image - jpeg - gif - bmp / video - flv / mp-
     Accept Language : en – US / en - GEN
     Accept CharSet: UTF - 8 / UTF - 16 / UTF - 32
Request Header
                                                                    This time parameters
     handy = lefty & style = pinchhitter
```

preference = NatioanlPlayer & Alt = BigBashPlayer

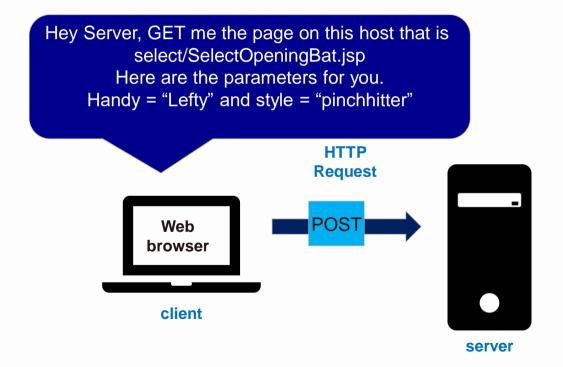
Request Body

This time parameters are down here in the body. So no limitation. They are not exposed as well



preference = WicketKeeper & Alt = Spinner

### HTTP POST REQUEST in Action





# Anatomy of HTTP Response with Response Body

#### HTTP / 1.1 200 OK

Set - Cookie NULL

Content – Type : Text/HTML

Date: Sat, 18 Jan, 2014

Server: Apache - Coyote/1.1

Connection State: Close

The Content Type is

related to the list

specified at Request's

"ACCEPT" Header

Response Header

< HTML > ------< /HTML>

Response Body

Response Body holds

the HTML which will

be rendered on

Browser

Here Connection status is close.

i.e. Once response is generated,

server shuts this connection.

This is STATELESS nature of HTTP

Protocol



#### Common status codes

- 200 OK
  - The request succeeded, and the resulting resource (e.g. file or script output) is returned in the message body
- 404 Not Found
  - The requested resource doesn't exist
- 301 Moved Permanently
  - The requested resource moved permanently
- 302 Moved Temporarily
  - The requested resource moved temporarily



### Common status codes

- 500 Server Error
  - An unexpected server error
  - server-side script that has bad syntax
  - server-side script fails for other reason





# Introduction

Quiz



"Complete the following statement: The data we send with method is appended to URL up in the browser bar, so whatever we send will be exposed."

POST
HEAD
GET
FIRST





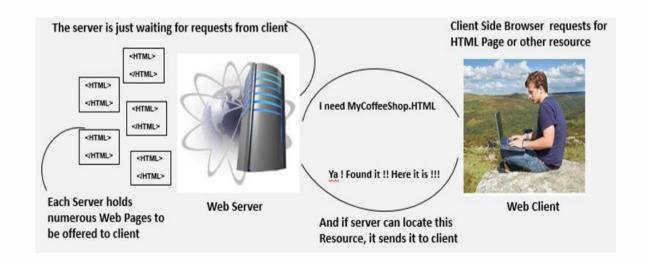
HTML



#### What does the Web server do?

In a REQUEST— RESPONSE Cycle ....

Web Servers have a full-time job on the Internet, seamlessly waiting for requests from Web Browsers. The server is just waiting for requests from client.





# HTML "Hello World" example

- The text between <html> and </html> describes the web page
- The markup text '<title>Hello
   HTML</title>' defines the browser page title.
- Typically, the text between <body> and </body> is the visible page content.
- The <a href="https://www.energive.com">httml> tag tells the browser your document is actually HTML. While some browsers will forgive you if you omit it.





HTML5



### Introduction

- HTML5 is the latest and most enhanced version of HTML.
- Technically, HTML is not a programming language, but rather a markup language.





- HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web.
- HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).
- The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash, Microsoft Silverlight, and Google Gears.



#### **Browser Support:**

- The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for *some* HTML5 functionality.
- The mobile web browsers that come pre-installed on iPhones, iPads, and Android phones all have excellent support for HTML5.



#### **New Features:**

- HTML5 introduces a number of new elements and attributes that helps in building a modern websites. Following are great features introduced in HTML5.
  - New Semantic Elements: These are like <header>, <footer>, and <section>.
  - Forms 2.0: Improvements to HTML web forms where new attributes have been introduced for <input> tag.
  - Persistent Local Storage: To achieve without resorting to third-party plugins.
  - WebSocket: A a next-generation bidirectional communication technology for web applications.
  - Server-Sent Events: HTML5 introduces events which flow from web server to the web browsers and they are called Server-Sent Events (SSE).



#### New Features (Continued...):

- Canvas: This supports a two-dimensional drawing surface that you can program with JavaScript.
- Audio & Video: You can embed audio or video on your web pages without resorting to third-party plugins.
- Geolocation: Now visitors can choose to share their physical location with your web application.
- Microdata: This lets you create your own vocabularies beyond HTML5 and extend your web pages with custom semantics.
- Drag and drop: Drag and drop the items from one location to another location on a the same webpage.



### HTML5 - Web Storage

HTML5 introduces two mechanisms, similar to HTTP session cookies, for storing structured data on the client side and to overcome following drawbacks.

- Cookies are included with every HTTP request, thereby slowing down your web application by transmitting the same data.
- Cookies are included with every HTTP request, thereby sending data unencrypted over the internet.
- Cookies are limited to about 4 KB of data. Not enough to store required data.

The two storages are **session storage** and **local storage** and they would be used to handle different situations.

The latest versions of pretty much every browser supports HTML5 Storage including Internet Explorer.



# HTML5 - Web Storage

#### Session Storage:

■ The Session Storage is designed for scenarios where the user is carrying out a single transaction, but could be carrying out multiple transactions in different windows at the same time.

#### Example:

If a user buying plane tickets in two different windows, using the same site. If the site used cookies to keep track of which ticket the user was buying, then as the user clicked from page to page in both windows, the ticket currently being purchased would "leak" from one window to the other, potentially causing the user to buy two tickets for the same flight without really noticing.

■ HTML5 introduces the *sessionStorage* attribute which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window i.e. session and as soon as you close the window, session would be lost.



### HTML5 - Web Storage

Following is the code which would set a session variable and access that variable:

```
<!DOCTYPE HTML>
<html>
<body>
  <script type="text/javascript">
   if( sessionStorage.hits ){
      sessionStorage.hits = Number(sessionStorage.hits) +1;
    }else{
      sessionStorage.hits = 1;
   document.write("Total Hits:" + sessionStorage.hits );
  </script>
  Refresh the page to increase number of hits.
  Close the window and open it again & check the result.
</body>
</html>
```



#### Local Storage:

- The *Local Storage* is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire user-authored documents or a user's mailbox, on the client side for performance reasons.
- Again, cookies do not handle this case well, because they are transmitted with every request.

#### Example:

■ HTML5 introduces the *localStorage* attribute which would be used to access a page's local storage area without no time limit and this local storage will be available whenever you would use that page.



■ Following is the code which would set a local storage variable and access that variable every time this page is accessed, even next time when you open the window:

```
<!DOCTYPE HTML>
<html>
<body>
 <script type="text/javascript">
   if( localStorage.hits ){
      localStorage.hits = Number(localStorage.hits) +1;
   }else{
      localStorage.hits = 1;
   document.write("Total Hits:" + localStorage.hits);
 </script>
 Refresh the page to increase number of hits.
 Close the window and open it again & check the result.
</body>
</html>
```



#### Delete Web Storage:

- Storing sensitive data on local machine could be dangerous and could leave a security hole.
- The Session Storage Data would be deleted by the browsers immediately after the session gets terminated.
- To clear a local storage setting you would need to call **localStorage.remove('key')**; where 'key' is the key of the value you want to remove. If you want to clear all settings, you need to call **localStorage.clear()** method.



Following is the code which would clear complete local storage:

```
<!DOCTYPE HTML>
<html>
<body>
 <script type="text/javascript">
   localStorage.clear();
   // Reset number of hits.
   if( localStorage.hits ){
      localStorage.hits = Number(localStorage.hits) +1;
   }else{
      localStorage.hits = 1;
   document.write("Total Hits :" + localStorage.hits );
 </script>
 Refreshing the page would not to increase hit counter.
 Close the window and open it again & check the result.
</body>
</html>
```



## **HTML5 Canvas**

- HTML5 element <canvas> gives you an easy and powerful way to draw graphics using JavaScript. It can be used to draw graphs, make photo compositions or do simple (and not so simple) animations.
- Here is a simple <canvas> element which has only two specific attributes width and height plus all the core HTML5 attributes like id, name and class etc.
  - <canvas id="mycanvas" width="100" height="100"></canvas>
- You can easily find that <canvas> element in the DOM using getElementById() method as follows:

var canvas = document.getElementById("mycanvas");



### **HTML5 Canvas**

■ Let us see a simple example on using <canvas> element in HTML5 document.

```
<!DOCTYPE HTML>
<html>
(head)
<style>
#mycanvas{
   border: 1px solid red;
</style>
</head>
<body>
   <canvas id="mycanvas" width="100" height="100"></canvas>
</body>
</html>
```



## HTML5 Canvas –The rendering context

- The <canvas> is initially blank, and to display something, a script first needs to access the rendering context and draw on it.
- The canvas element has a DOM method called **getContext**, used to obtain the rendering context and its drawing functions. This function takes one parameter, the type of context **2d**.
- Following is the code to get required context along with a check if your browser supports <anyas> element:

```
var canvas = document.getElementById("mycanvas");
if (canvas.getContext){
  var ctx = canvas.getContext('2d');
  // drawing code here
} else {
  // canvas-unsupported code here
}
```



#### HTML5 - Audio & Video

- HTML5 features, include native audio and video support without the need for Flash.
- The HTML5 <audio> and <video> tags make it simple to add media to a website. You need to set **src** attribute to identify the media source and include a controls attribute so the user can play and pause the media.

```
<video src="foo.mp4" width="300" height="200" controls>
    Your browser does not support the <video> element.
</video>

<audio src="foo.wav" controls autoplay>
    Your browser does not support the <audio> element.
</audio>
```



#### How the INTERNET works

- HTML5 Geolocation API lets you share your location with your favorite web sites. A Javascript can capture your latitude and longitude and can be sent to backend web server and do fancy locationaware things like finding local businesses or showing your location on a map.
- Today most of the browsers and mobile devices support Geolocation API. The geolocation APIs work with a new property of the global navigator object i.e. Geolocation object which can be created as follows:

var geolocation = navigator.geolocation;

 The geolocation object is a service object that allows widgets to retrieve information about the geographic location of the device.





# HTML and HTML5

Quiz



# Which of the following are new HTML5 tags?

table img video audio





CSS



# Generalized look and feel for components

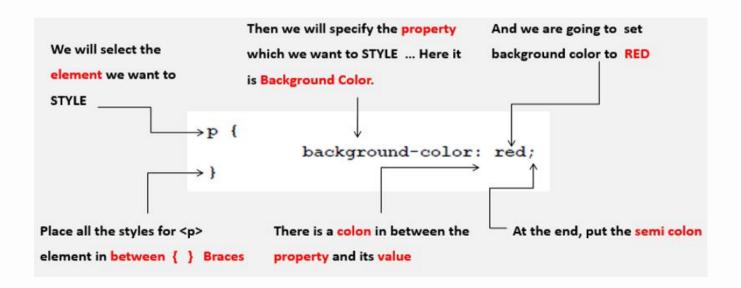
• In the Browser Context, HTML doesn't have ROOMS. HTML does have elements. These elements are the locations where we will put up STYLING. Here we are going to control the presentation of our elements.





# Using CSS with HTML

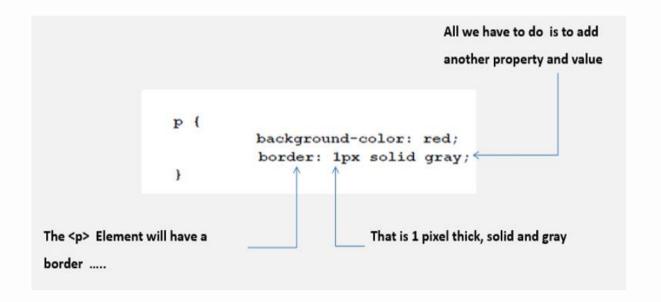
■ Following is an example to assign the background color as red to a paragraph.





# Adding multiple stylings

We can add multiple stylings to the same element





#### How the INTERNET works

- CSS can be added to HTML in the following ways
  - Inline using the style attribute in existing HTML elements
  - Internal using separate <style> element in the <head> section
  - External using an external CSS file
- The preferred way to add CSS to HTML, is to put CSS syntax in separate CSS files
- We can use a style sheet as a Template or Master Page so that multiple pages can use the same style sheet to implement same stylistic attributes.



# Using the style sheets

- Inline Styles .
  - An inline style can be used if a unique style is to be applied to one single element. .
  - To use inline styles, use the style attribute in the relevant tag.
  - The style attribute can contain any CSS property.
  - The example below shows how to change the text color and the left margin of a paragraph
     This is a paragraph.



# Font and background-based styling

```
<html>
<h2 Style = "font-family : verdana;" > This is a heading </h2>
 This is a paragraph 
</body>
<html>
<html>
<h2 Style = "background-color : red;"> This is a heading </h2>
 This is a paragraph 
</html>
```



# Text alignment example

```
<html>
<body>
<h1 style = "text aligned :center;"> Center aligned heading</h1>
 This is a paragraph
</body>
</html>
```



## Internal style sheet

 An internal style sheet can be used if one single document has a unique style. Internal styles are defined in the <head> section of an HTML page, by using the <style> tag, like this

```
<html>
<head>
<style type ="text/css">
body { background-color:yellow;}
p {color:blue}
</style>
</head>
</html>
```



#### How the INTERNET works

- An external style sheet is ideal when the style is applied to many pages.
- With an external style sheet, we can change the look of an entire website by changing one file.
- Each page must link to the style sheet using the <link> tag.
- The k> tag goes inside the <head> section





CSS3



## **CSS3 Introduction**

- CSS3 is completely backwards compatible, so you will not have to change existing designs. Browsers will always support CSS2.
- CSS3 is split up into "modules". The old specification has been split into smaller pieces, and new ones are also added.
- Some of the most important CSS3 modules are:
  - Selectors
  - Box Model
  - Backgrounds and Borders
  - Text Effects and Animations
  - 2D/3D Transformations
  - Multiple Column Layout
  - User Interface





CSS and CSS3

Quiz



# What is the correct HTML for referring to an external style sheet?

<style src="mystyle.css">

<link rel="stylesheet" type="text/css" href="mystyle.css">

<stylesheet>mystyle.css</stylesheet>

<link ref="stylesheet" type="text/css" href="mystyle.css">





Scripting language



## Dynamic content

- The Web Server Application only serves the pages.
- SO we can arrange
  - Another helper application that the Web Server can communicate with
  - Which can build Non Static, Just In Time Pages

```
CHTML>
CBODY>
The Current Time is 10: 10 AM On this Server

CHTML>
CBODY>
The Current Time is [InsertCurrentTime] On this Server

CHTML>
CBODY>
CHTML>
CHTML>
CHTML>
CHTML>
CBODY>
CHTML>
```

Static Non-Static



## Introduction to JavaScript

- Today's Websites need to go much beyond HTML
- Today's web site must be intelligent enough to accept users input and dynamically structure web page content, tailor made to a user's requirements.
- This requires a web development environment that will allow the creation of Interactive Web Pages. Also it should provide the facility for validating user input.
  - Conditional Checking Construct Case Checking Constructs
  - Well Controlled Loop Constructs.



# HTML and JavaScript

- HTML, itself, allows a very low level of Dynamicity and Interactivity. .
- Truly interactive pages can not be created using standard HTML Tags alone.
- Embedding JavaScript in HTML does this.
- For instance a Web Site can be created and hosted on a web server to take orders for product.
- At the same time, Form Data Validation must be done



# Strengths of JavaScript

- An Interpreted Language
  - Which requires no compilation steps.
  - The syntax is completely recognized by Browser directly.
- Obvious compatibility with HTML
  - HTML files containing JavaScript Instructions can easily be executed by Browser. No different platform is required (Editor or Compiler or IDE).
- Minimal Syntax Easy to Learn & Use Rapid Development
- Minimum Storage Requirement at Web Server and Downloadable Time.
- Procedural Capabilities
  - Conditional Checking & Branching Looping
- Supports Event Driven Programming efficiently
- <u>L</u>71

Ability to recognize the Button Press

# Writing JavaScript Into HTML

- JavaScript Syntax is embedded into HTML File.
- A Browser reads HTML files and interprets HTML Tags.
- Since entire JavaScript need to be included as an integral part of HTML document, when required, the browser needs to be informed that specific section of HTML Code is JavaScript.
- The browser will then use its built-in JavaScript Engine to interpret this code.
- For this purpose, <SCRIPT> </SCRIPT> Tag is used

```
<SCRIPT type="text/ javascript ">
```

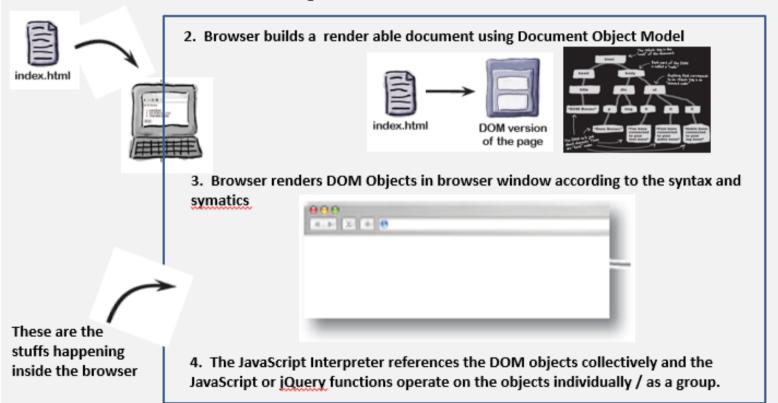
// JavaScript Code Snippet is written here

</SCRIPT>



#### Look into the browser

1. The Browser loads the HTML file, it gets from server

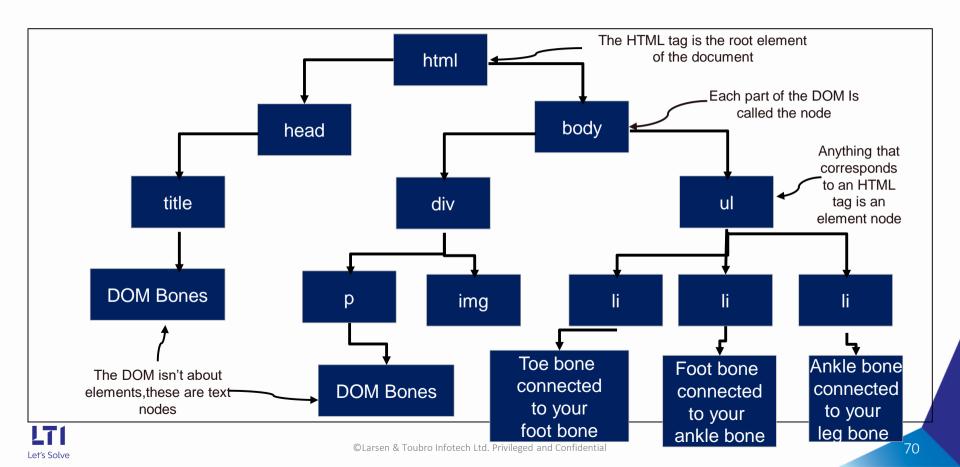


# According to Document Object Model

- The topmost object in DOM is the Browser itself.
- Next level in the model is browser's window.
- Then comes the document displayed in window.
- The DOM hierarchy continues downward to encompass Form and its individual elements like Text Boxes, Buttons & Boxes and so on.
- Browser
  - Anchor
  - Link
  - Form
    - Textbox Radio Button checkbox push button



# The hidden structure of a web page – with DOM



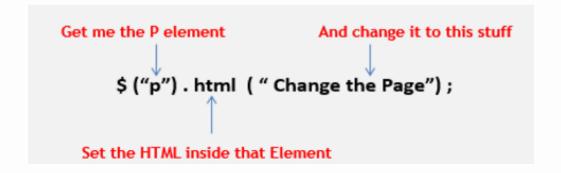
# jQuery

- jQuery is a fast, small, but feature-rich JavaScript library.
- It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simple.
- It is a way using JavaScript in more effective manner.
- jQuery is open source



# jQuery

- jQuery provides simplicity in writing codes for various tasks
- DOM manipulation
  - Using jQuery simplifies DOM manipulation and traversal for modifying the contents.
  - For this, jQuery uses an engine called as SIZZLE which is browser independent and open source





#### **jQuery**

- jQuery provides simplicity in writing codes for various tasks
- Event handling.
  - jQuery provides much simpler way to connect events with event handlers.
  - Which otherwise is very confusive through HTML Technology
  - You can easily bind and unbind existing event handlers
- AJAX Support
  - Enriched features of AJAX are as it is available for jQuery
  - We can call local script on server with query parameter which will modify the contents of HTML / XML elements



#### jQuery

- Query provides simplicity in writing codes for various tasks
- Animation Support .
  - Plenty of built in animation effects .
- Compatibility with languages
  - The jQuery script can be used with nearly all the web languages
  - PHP ASP JSP Servlet CGI s





### Scripting language

Quiz



**Group Company** 

## When a user views a page containing a JavaScript program, which machine actually executes the script?

The Web server

A central machine

The User's machine running a Web browser

**Application Server** 



#### How to create a function in JavaScript?

function myFunction() {...}

function:myFunction(){..}

function = myFunction()

None of the options





JSON



#### Introduction

- JavaScript Object Notation
  - Lightweight data-interchange format
  - Compared to XML
  - Simple format
  - Easy for humans to read and write
  - Easy for machines to parse and generate
  - JSON is a text format
  - Programming language independent
  - Uses conventions that are familiar to programmers of the C- family of languages, including C, C++,
     C#, Java, JavaScript, Perl, Python



#### Why JSON over XML?

- Lighter and faster than XML as on-the-wire data format
- JSON objects are typed while XML data is typeless
  - JSON types: string, number, array, boolean,
  - XML data are all string
- Native data form for JavaScript code
  - Data is readily accessible as JSON objects in your JavaScript code vs. XML data needed to be parsed and assigned to variables through tedious DOM APIs
  - Retrieving values is as easy as reading from an object property in your JavaScript code
  - A JSON notation begins with { (left brace) and ends with } (right brace)
  - Each name is followed by: (colon) and the name/value pairs are separated by, (comma)
  - Example : {"name": "Majrul", "company":"LTI"}





JSON

Quiz





# State True or False: JSON is lightweight.

TRUE

**FALSE** 





AngularJS and AngularX



#### What is Angular?

- **Angular** is a very powerful JavaScript Framework.
- It is used in Single Page Application (SPA) projects.
- It extends HTML DOM with additional attributes and makes it more responsive to user actions.

	Angular	jQuery
Animation support	Yes	Yes
2 way data binding	Yes	No
DOM manipulation	Yes	Yes
RESTful API	Yes	No
AJAX/JSONP	Yes	Yes
Deep Linking routing	Yes	No
Form validation	Yes	No



#### What is Angular?

#### Angular is a great tool that will:

- Enable you to create software quicker and with less effort
- Result in a more maintainable software
- Encourage good programming practices and design patterns like MVC
- Allow you to collaborate easier with other people
- Allow you to become proficient in a reasonable time
- Address problems that may arise in your software architecture such as Dependency Injection, DRY (Don't Repeat Yourself), etc

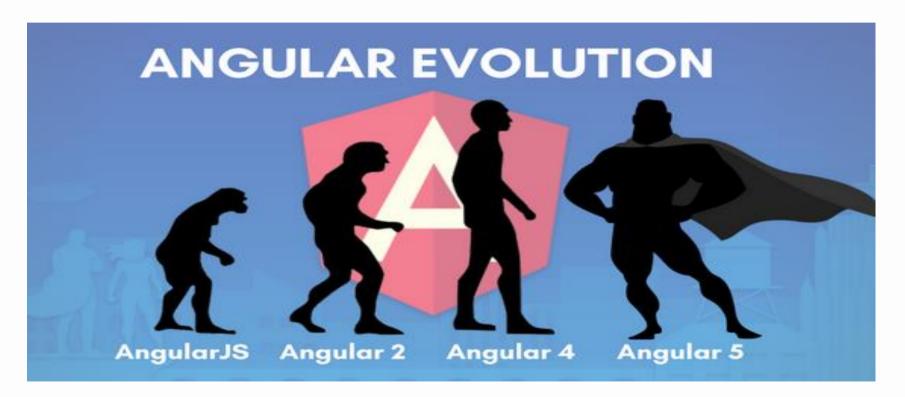


#### AngularJS vs Angular

- When it all started, this framework was called AngularJS, and alludes to what we now know as Angular 1.x. Then, Angular 2 arrived as a complete rewrite of the framework, improving from lessons learned and promising performance improvements, and a more scalable and more modern framework.
- The first version of Angular was named Angular 2. Later on, it was renamed to "Angular".
- Angular development is done using TypeScript which is a superset of JavaScript which offers many additional features than JavaScript
- Every time we use the term Angular we are referring to the latest version of the framework including Angular 2, Angular 4, Angular 5, Angular 6 and Angular 7.



#### **Angular Evolution**





#### New in Angular compared to AngularJS

- Angular is a complete rewrite of AngularJS
- An Angular application and its architecture are different from AngularJS. The main building elements for Angular are modules, components, templates, metadata, data binding, directives, services and dependency injection
- Angular does not have a "scope" concept or controllers, instead, it uses a component hierarchy as its main architecture
- Angular follows a modularity concept. Similar functionalities are kept together inside modules. This gives Angular an optimized lighter core
- The controller concept, which was present in AngularJS, was removed from Angular 2 and above which are component-based UI. This help developers divide applications in components with desired features. These helped improve the flexibility and reusability compared to AngularJS



#### AngularJS to Angular

Angular 2 recommends using the TypeScript language, which introduces these features:

- Static Typing
- Object Oriented Programming based on classes
- Support reactive programming using RxJS



#### AngularJS to Angular

On top of TypeScript features, Angular also includes the benefits taken from ES6:

- For/Of loops
- Improved dependency injection
- Iterators
- Reflection
- Dynamic loading
- Asynchronous template compilation
- Simpler Routing

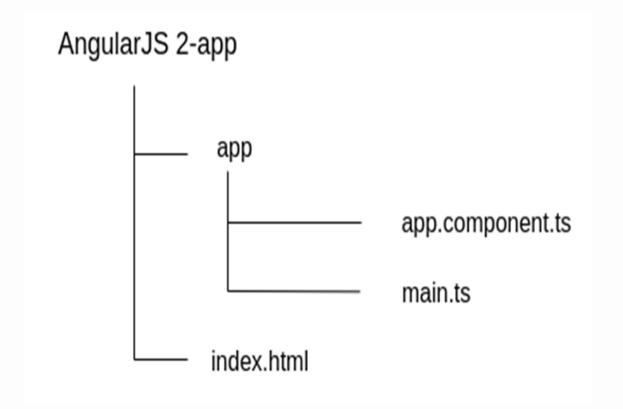


#### Angular2 File Structure

- $app/app.component.ts \rightarrow$  this is where we define our root component.
- app/app.module.ts → the entry Angular Module to be bootstrapped.
- *index.html* → this is the page the component will be rendered in.
- $app/main.ts \rightarrow$  is the glue that combines the component and page together.



### Angular2 File Structure (contd.)



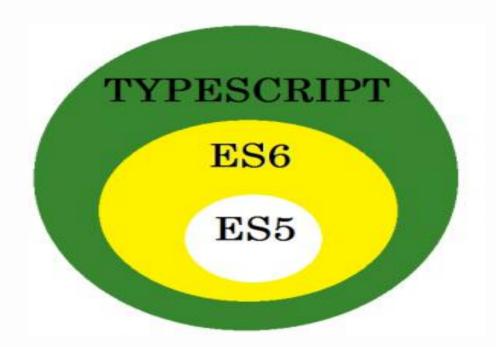


#### What is TypeScript?

- It is a superset of javascript.
- TypeScript is an open-source programming language.
- It is developed and maintained by Microsoft.
- TypeScript follows javascript syntactically but adds more features to it.



#### What is TypeScript?





#### Why do we need typescript?

- Strong Typing You can define variable with its data type if desired.
- OOPS features Classes, interfaces, constructors, access modifiers, fills, properties, generics
- Compile time errors We can catch errors at compile time before runtime.
- Great tooling We get access to great tools





### AngularJS and AngularX

Quiz



# Which language is used for writing component class in Angular?

TypeScript

JavaScript

Node.js

NPM



