Let's Solve

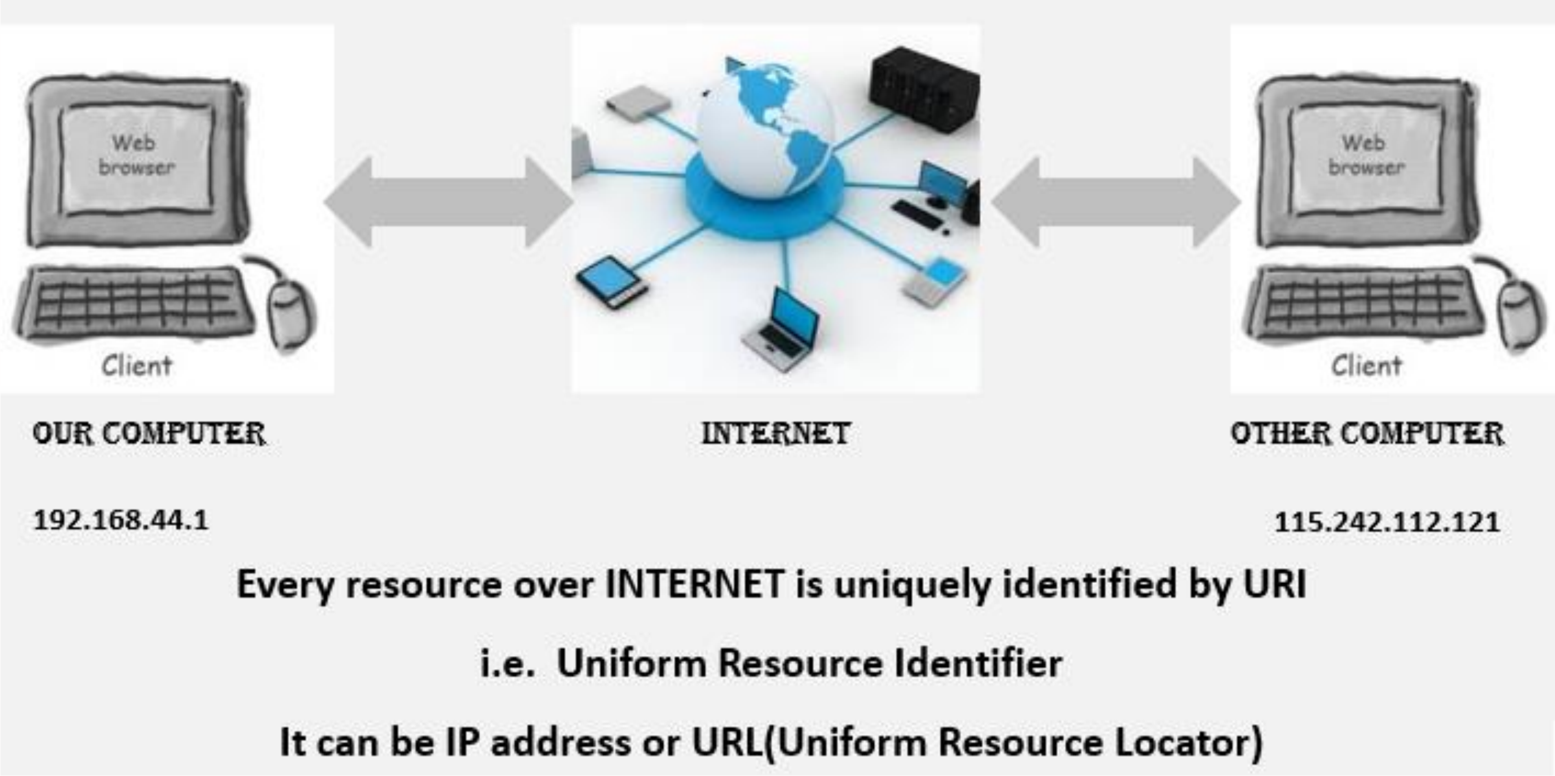
Celebrating 20 Years

WEB Technologies

A Larsen & Toubro
Group Company

Introduction

How does the INTERNET works



OUR COMPUTER

INTERNET

OTHER COMPUTER

192.168.44.1

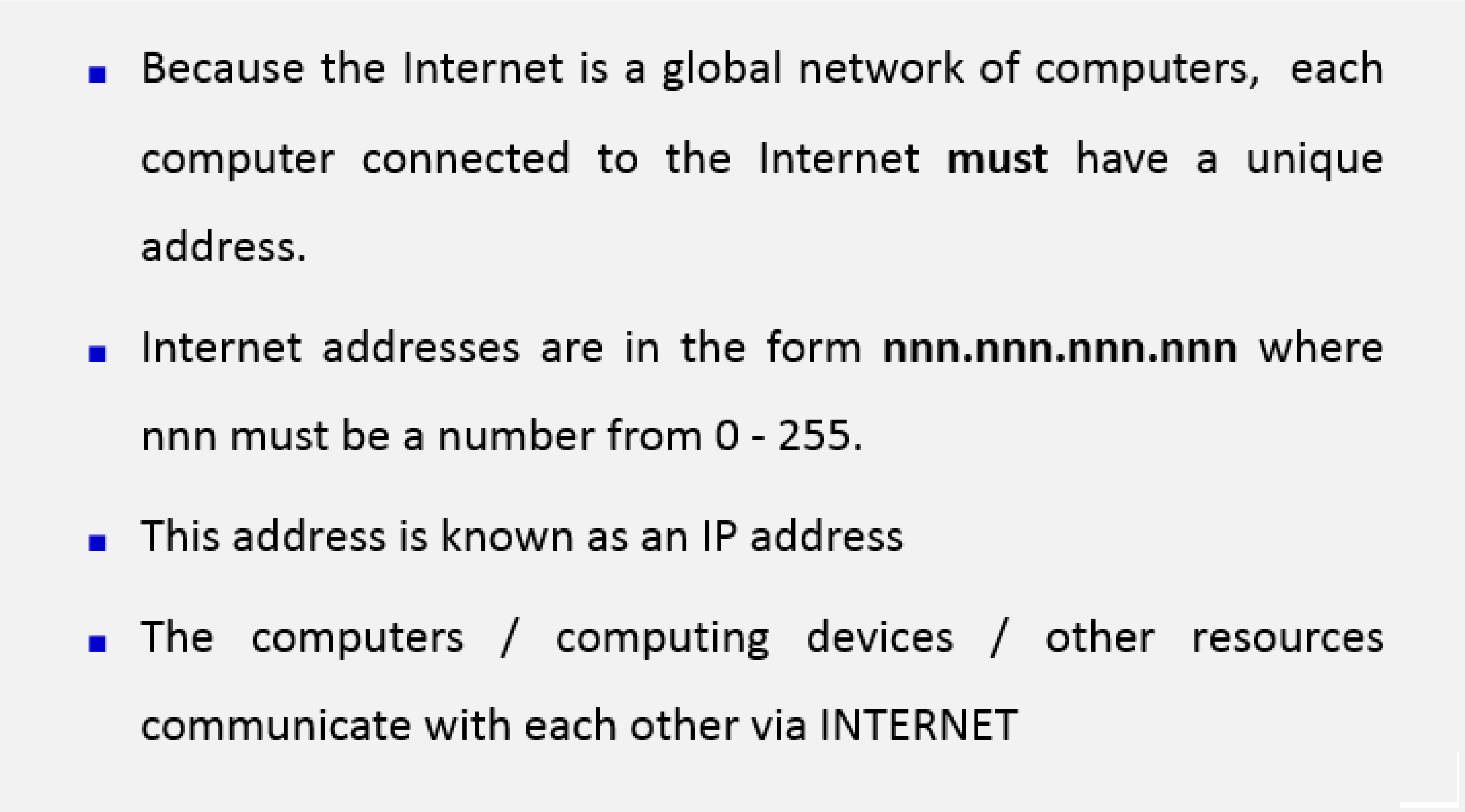
115.242.112.121

Every resource over INTERNET is uniquely identified by URI

i.e. Uniform Resource Identifier

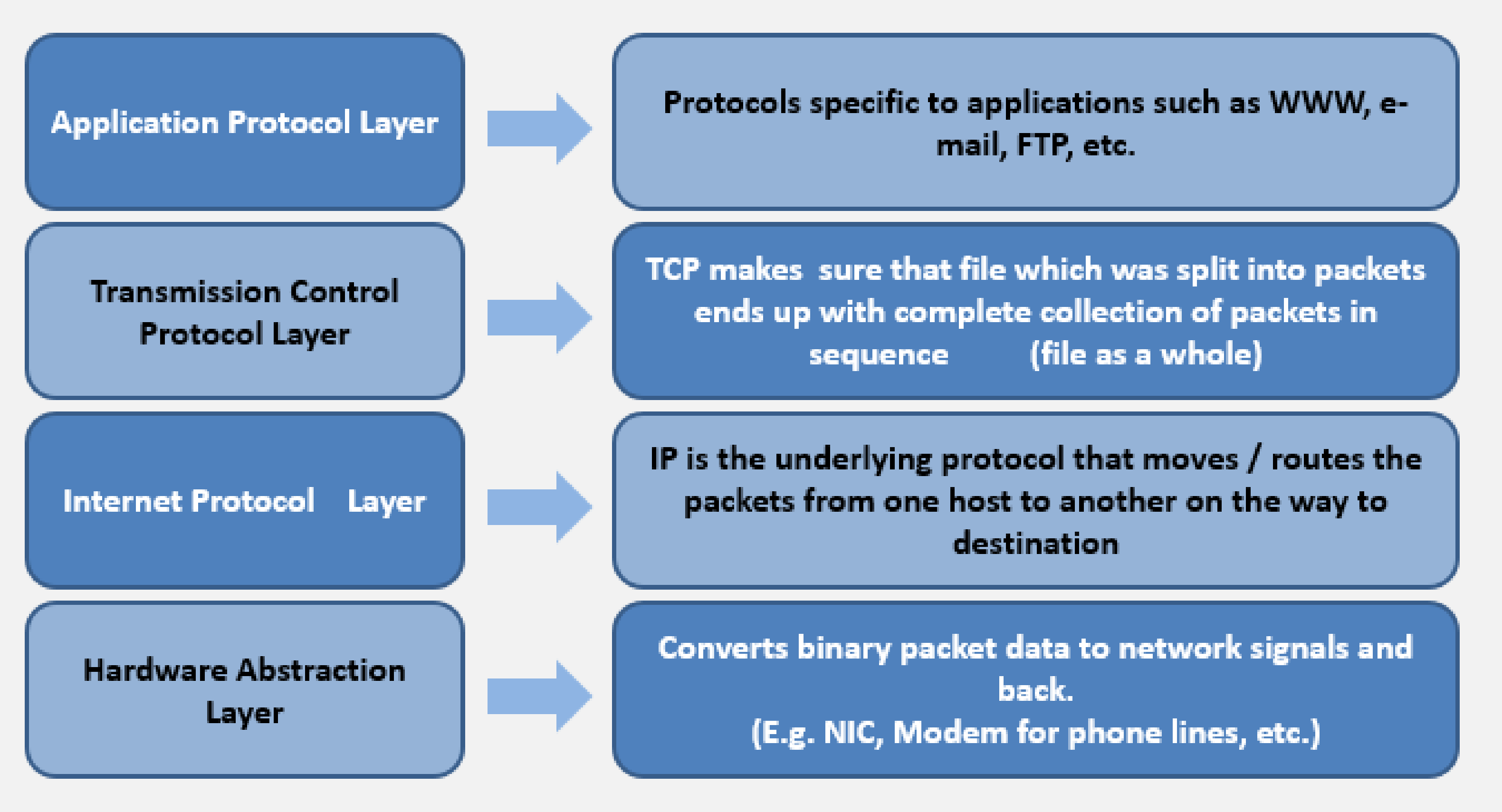
It can be IP address or URL(Uniform Resource Locator)

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.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

TCP / IP Protocol Stacks



Application Protocol Layer

Protocols specific to applications such as WWW, e-mail, FTP, etc.

Transmission Control Protocol Layer

TCP makes sure that file which was split into packets ends up with complete collection of packets in sequence   (file as a whole)

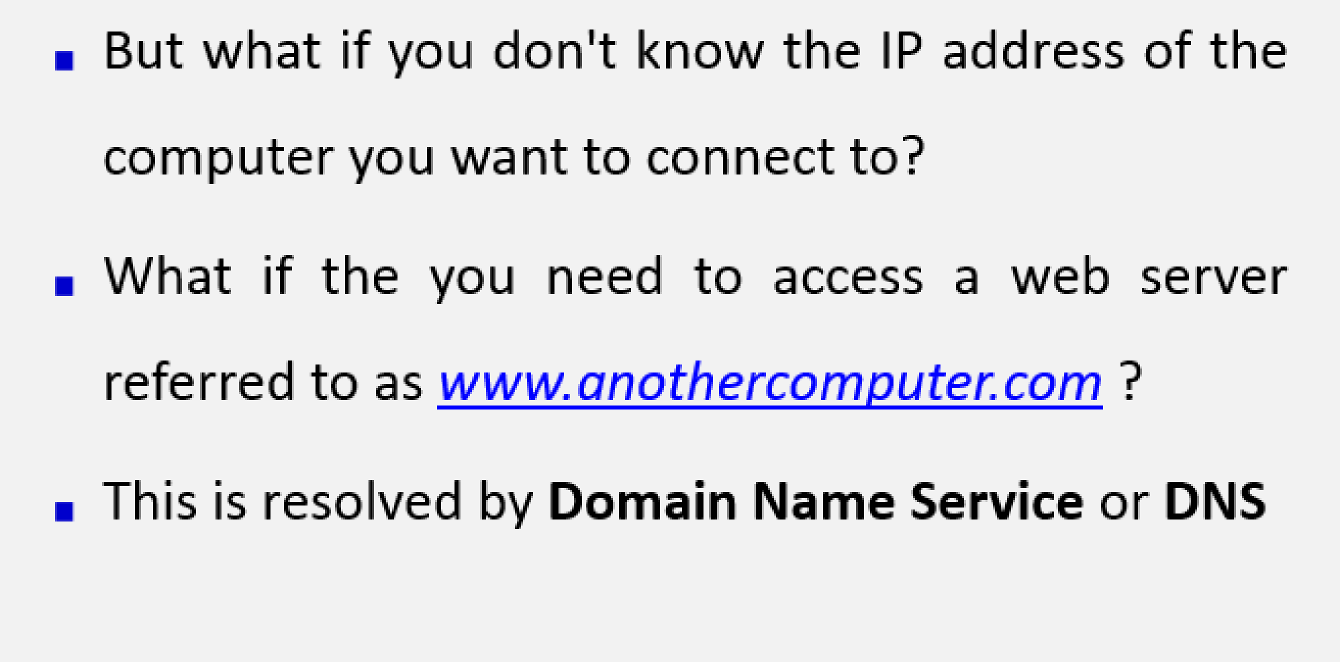
Internet Protocol   Layer

IP is the underlying protocol that moves / routes the packets from one host to another on the way to destination

Hardware Abstraction Layer

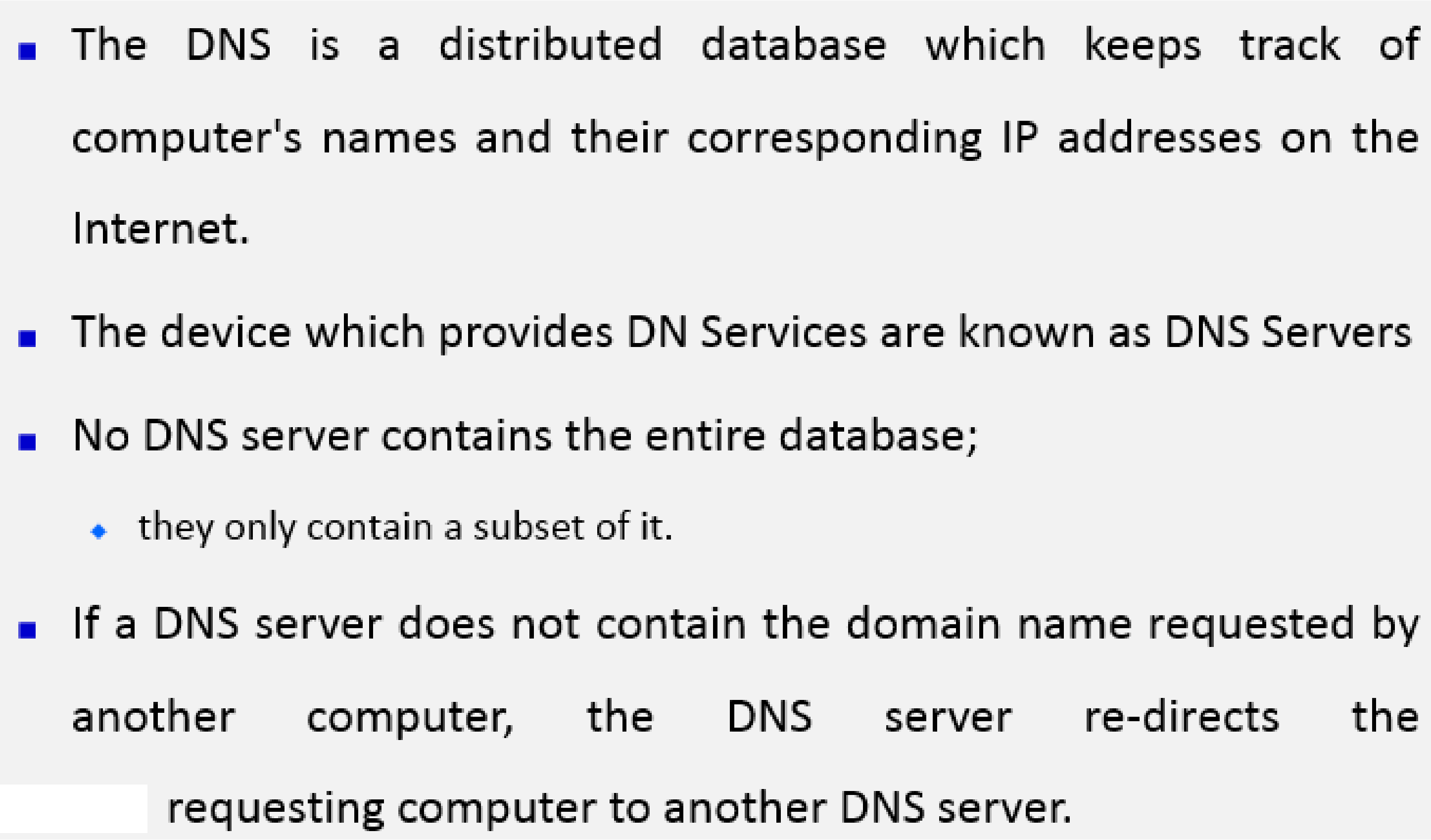
Converts binary packet data to network signals and back.
(E.g. NIC, Modem for phone lines, etc.)

Domain Names and Address Resolution



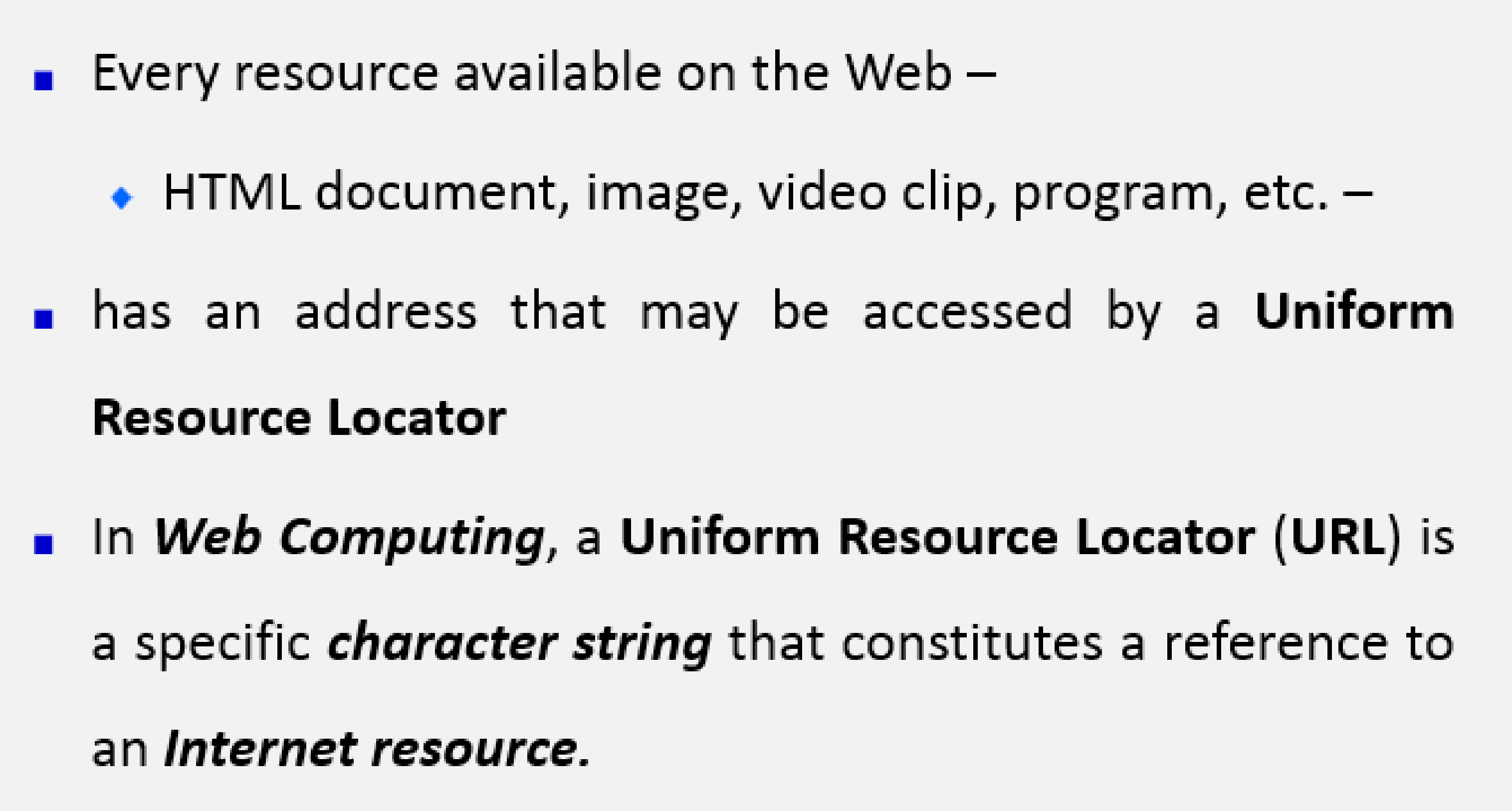
- But what if you don't know the IP address of the computer you want to connect to?
- What if the you need to access a web server referred to as *www.anothercomputer.com* ?
- This is resolved by **Domain Name Service** or **DNS**

Domain Names and Address Resolution



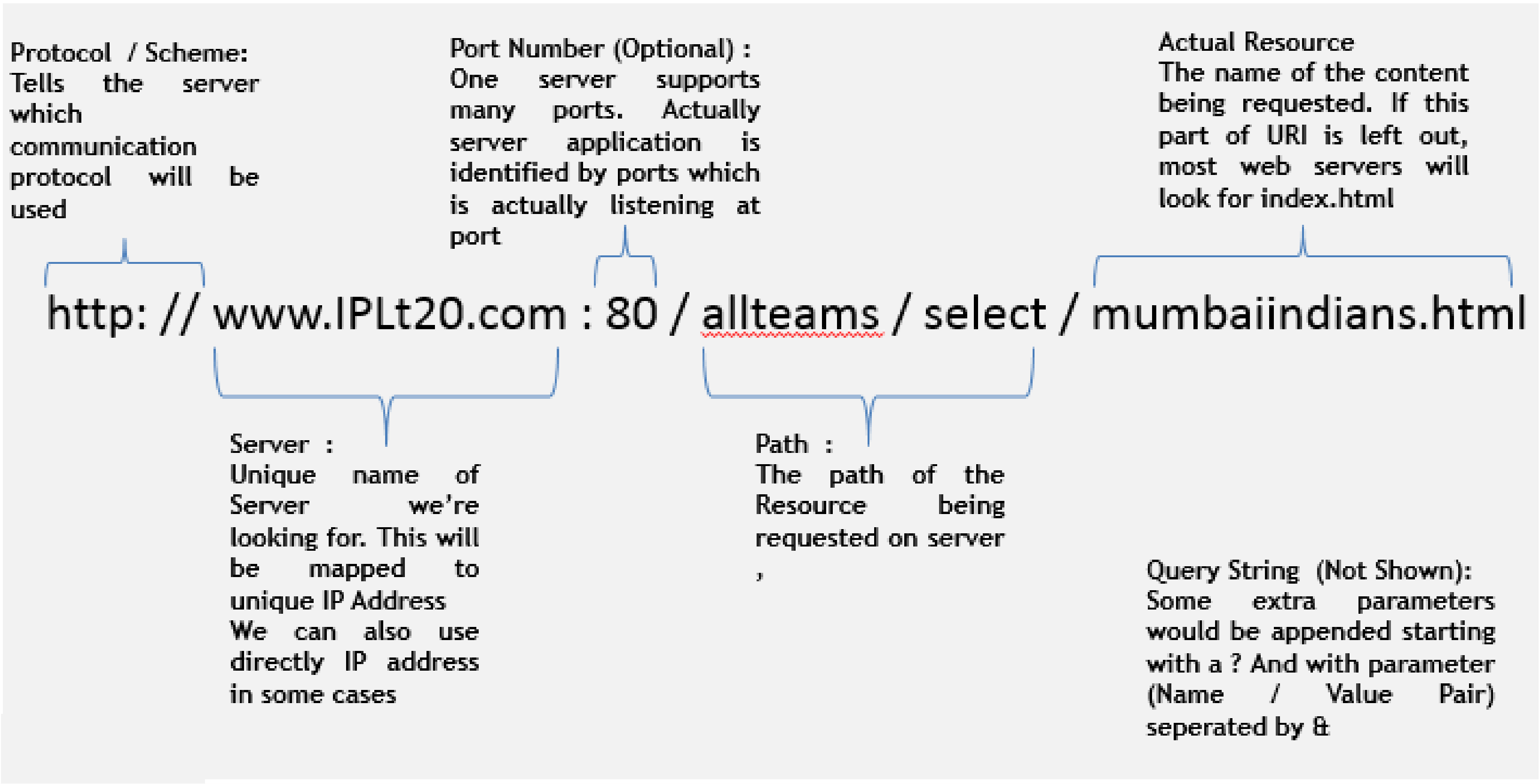
- The DNS is a distributed database which keeps track of computer's names and their corresponding IP addresses on the Internet.
- The device which provides DN Services are known as DNS Servers
- No DNS server contains the entire database;
  - they only contain a subset of it.
- If a DNS server does not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



http: // www.IPLt20.com : 80 / allteams / select / mumbaiindians.html

Protocol / Scheme: Tells the server which communication protocol will be used

Port Number (Optional) : One server supports many ports. Actually server application is identified by ports which is actually listening at port

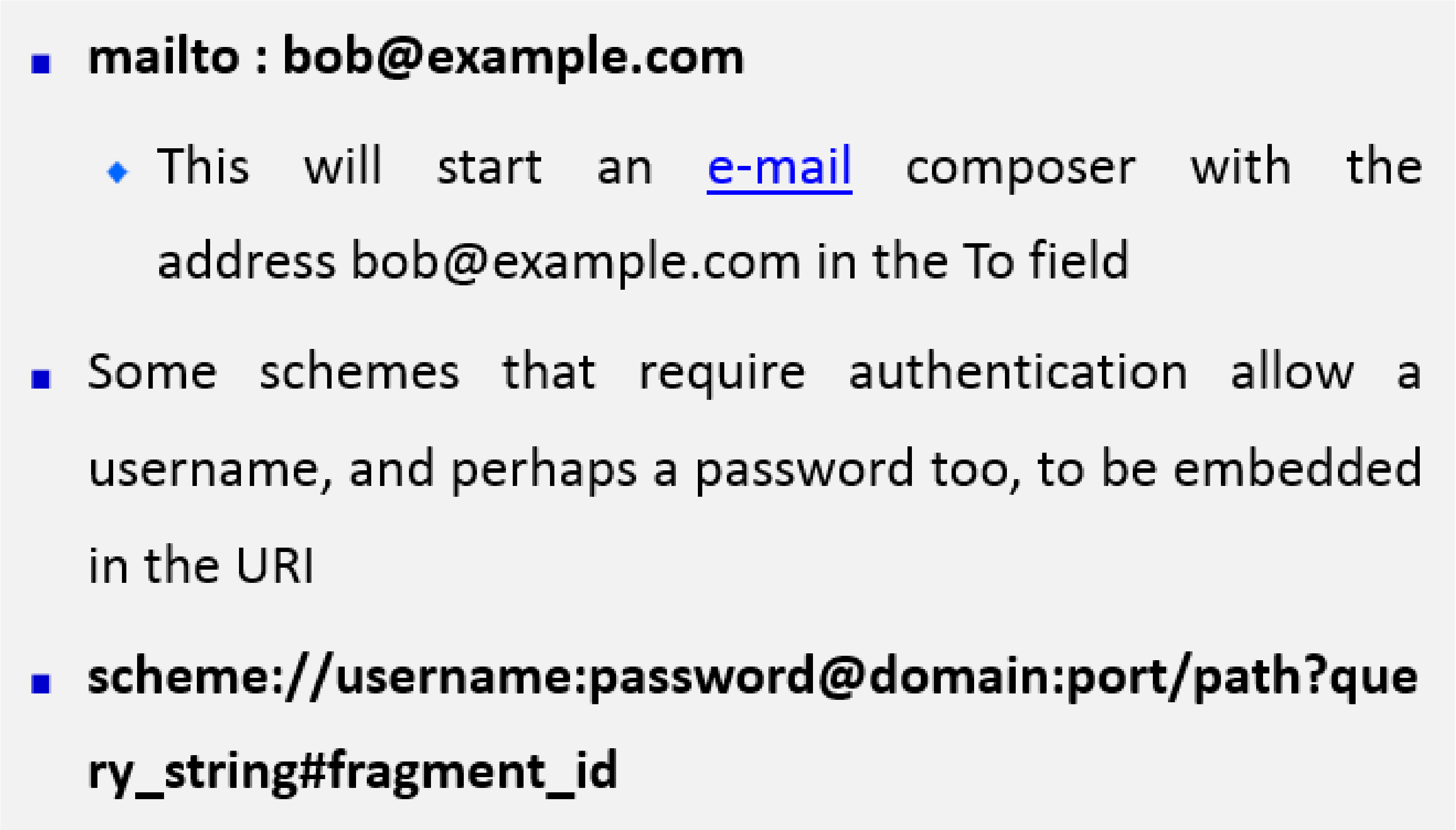
Actual Resource The name of the content being requested. If this part of URI is left out, most web servers will look for index.html

Server : Unique name of Server we're looking for. This will be mapped to unique IP Address We can also use directly IP address in some cases

Path : The path of the Resource being requested on server ,

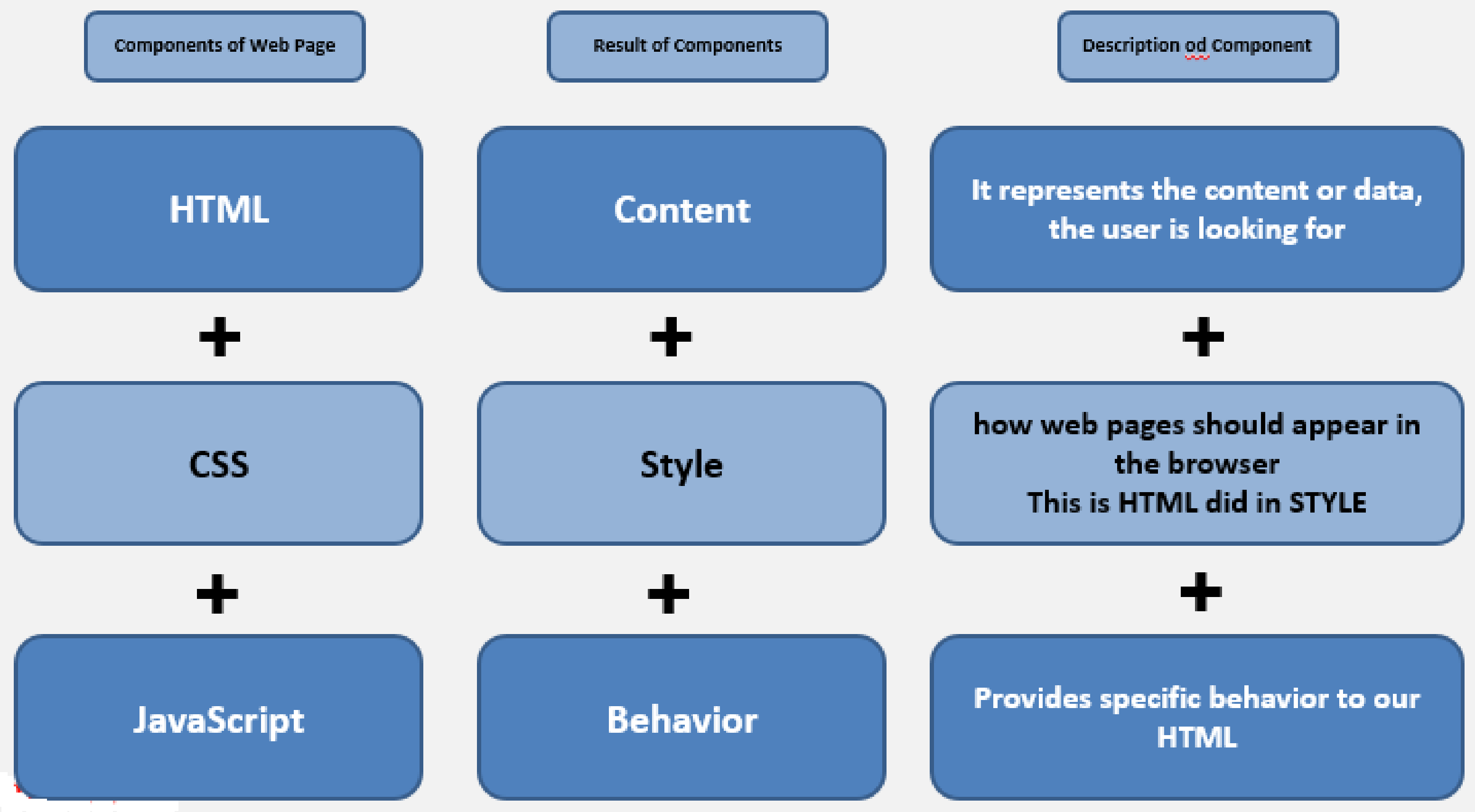
Query String (Not Shown): Some extra parameters would be appended starting with a ? And with parameter (Name / Value Pair) seperated by &

Some more variants of URL



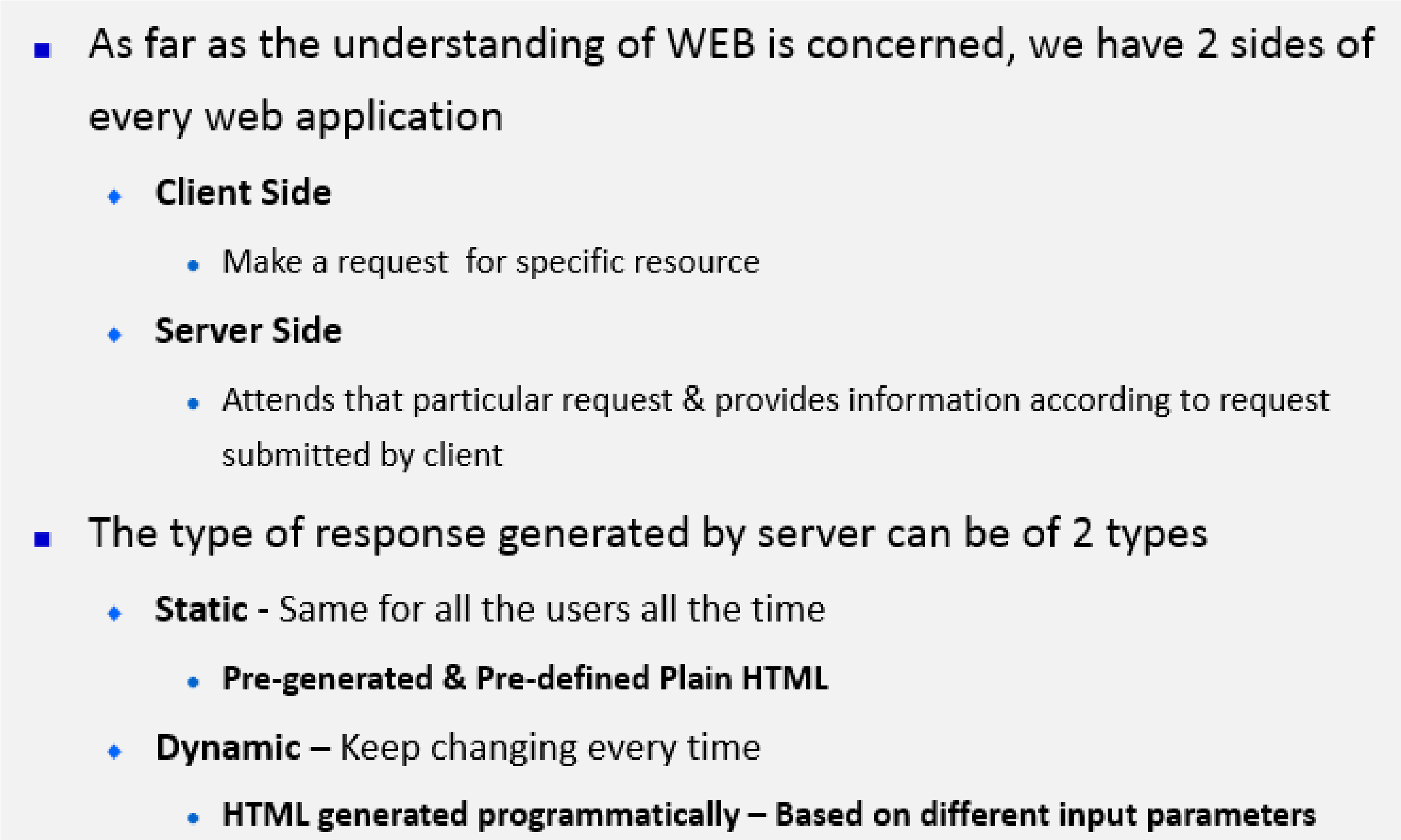
- **mailto : bob@example.com**
  - This will start an e-mail composer with the address bob@example.com in the To field
- Some schemes that require authentication allow a username, and perhaps a password too, to be embedded in the URI
- **scheme://username:password@domain:port/path?query_string#fragment_id**

Anatomy of web page



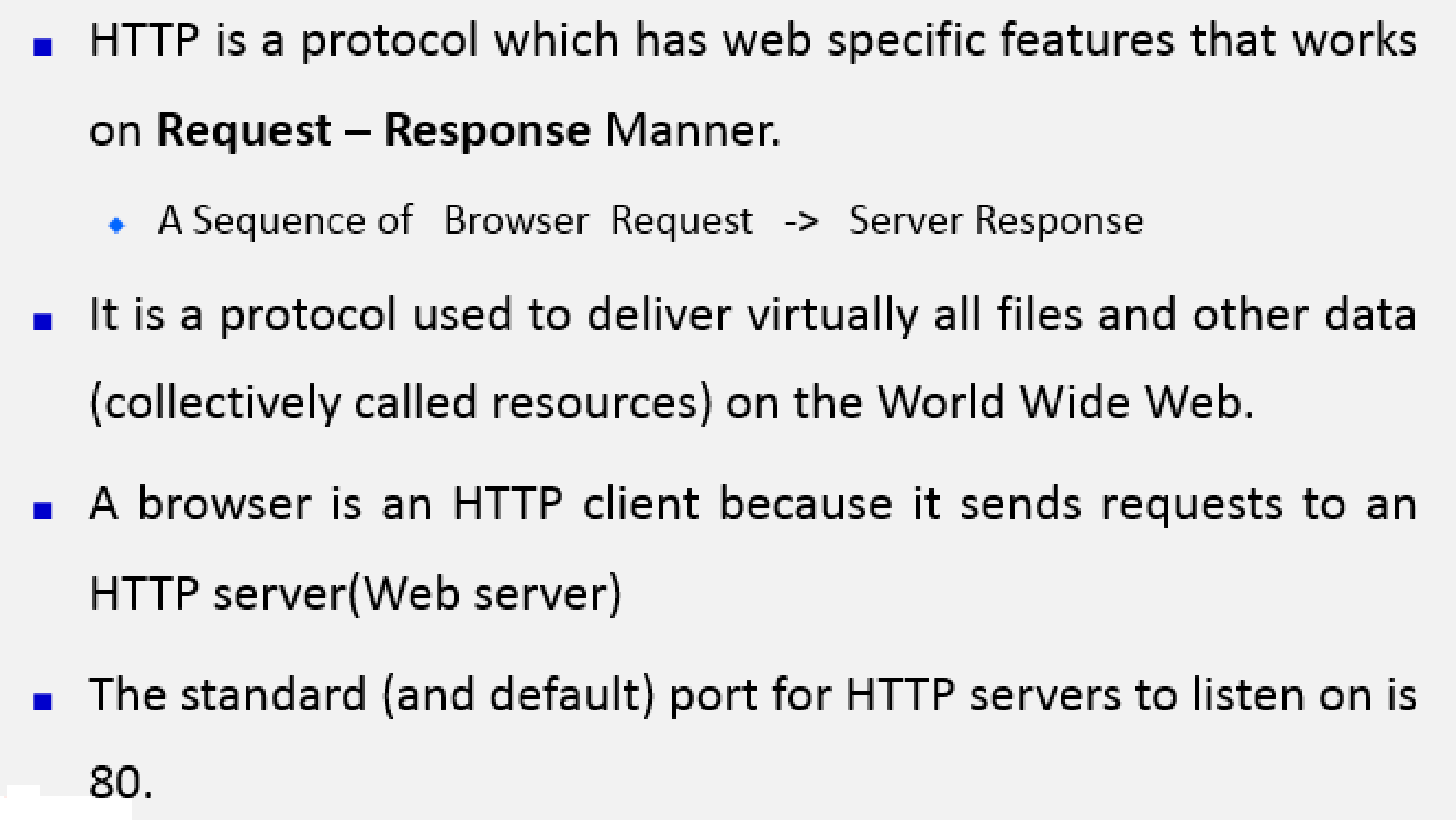
Components of Web Page	Result of Components	Description od Component
HTML	Content	It represents the content or data, the user is looking for
+	+	+
CSS	Style	how web pages should appear in the browser This is HTML did in STYLE
+	+	+
JavaScript	Behavior	Provides specific behavior to our HTML

Fundamentals of Web Technology



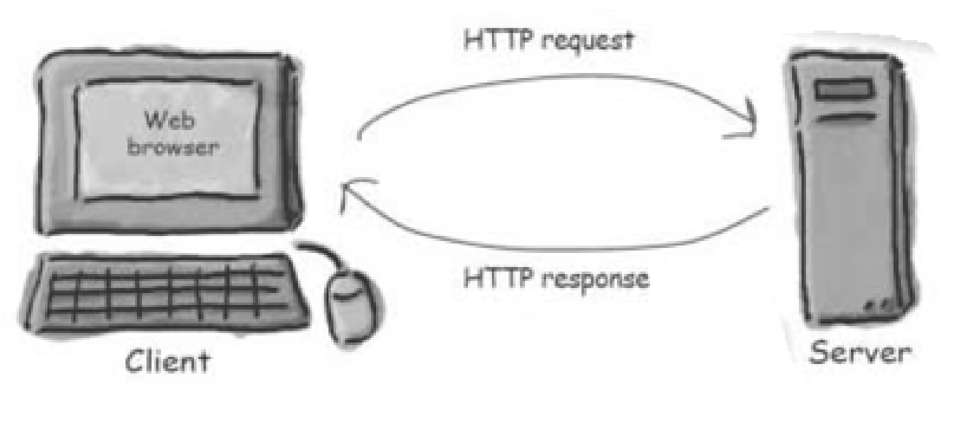
- As far as the understanding of WEB is concerned, we have 2 sides of every web application
  - **Client Side**
    - Make a request for specific resource
  - **Server Side**
    - Attends that particular request & provides information according to request submitted by client
- The type of response generated 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



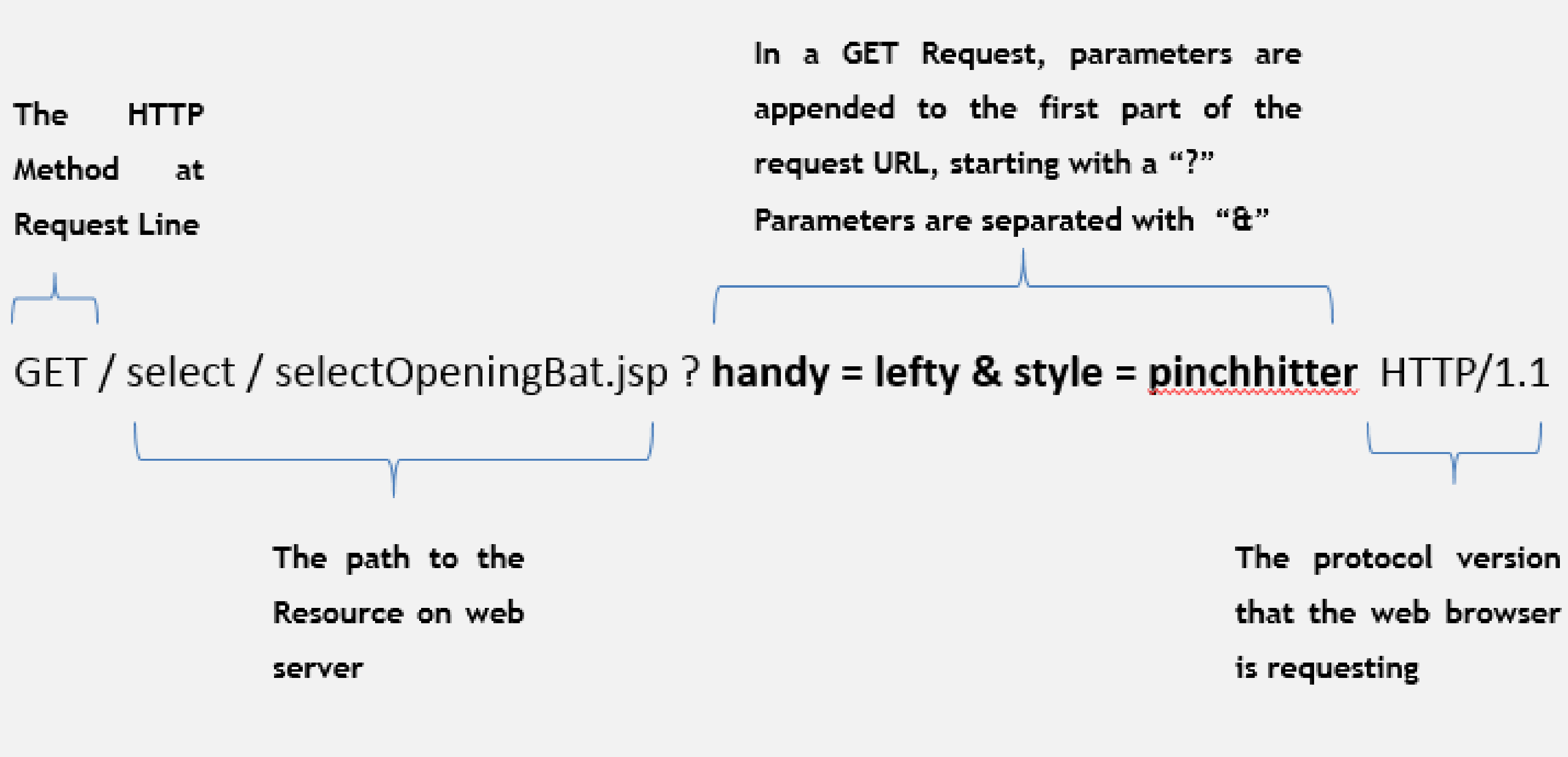
Key Elements of Request Stream :

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 etc

Anatomy of HTTP GET REQUEST



The HTTP Method at Request Line

In a GET Request, parameters are appended to the first part of the request URL, starting with a "?"

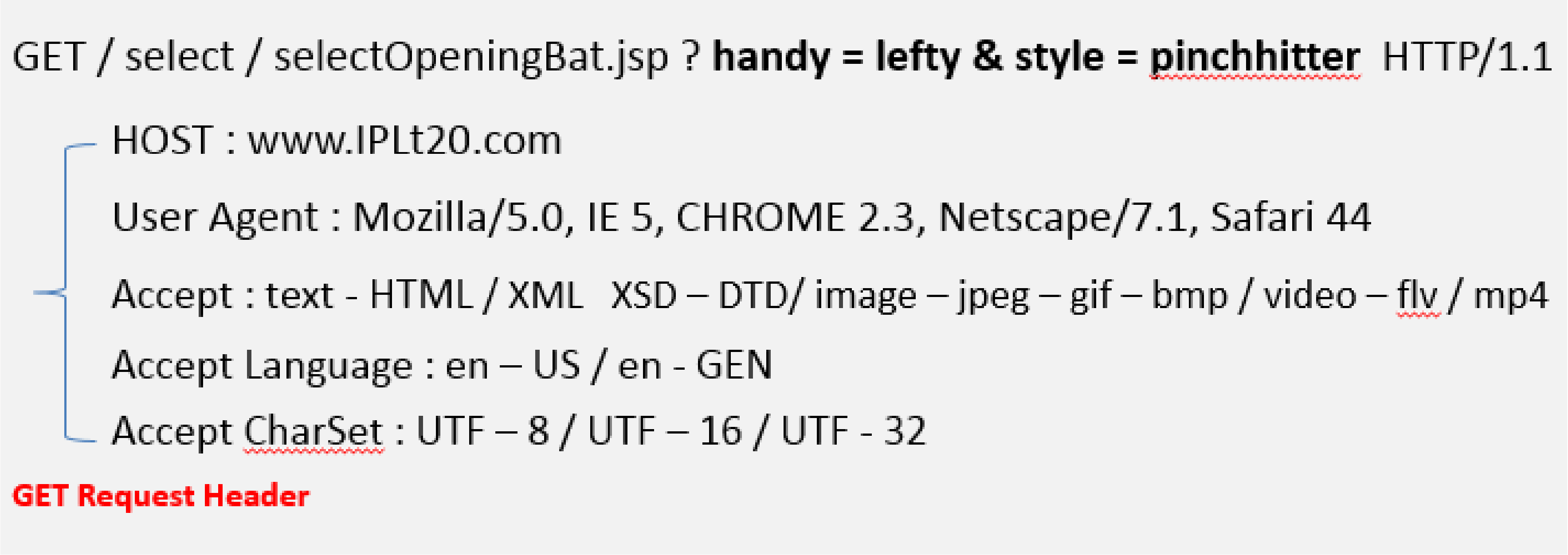
Parameters are separated with "&"

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

The path to the Resource on web server

The protocol version that the web browser is requesting

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

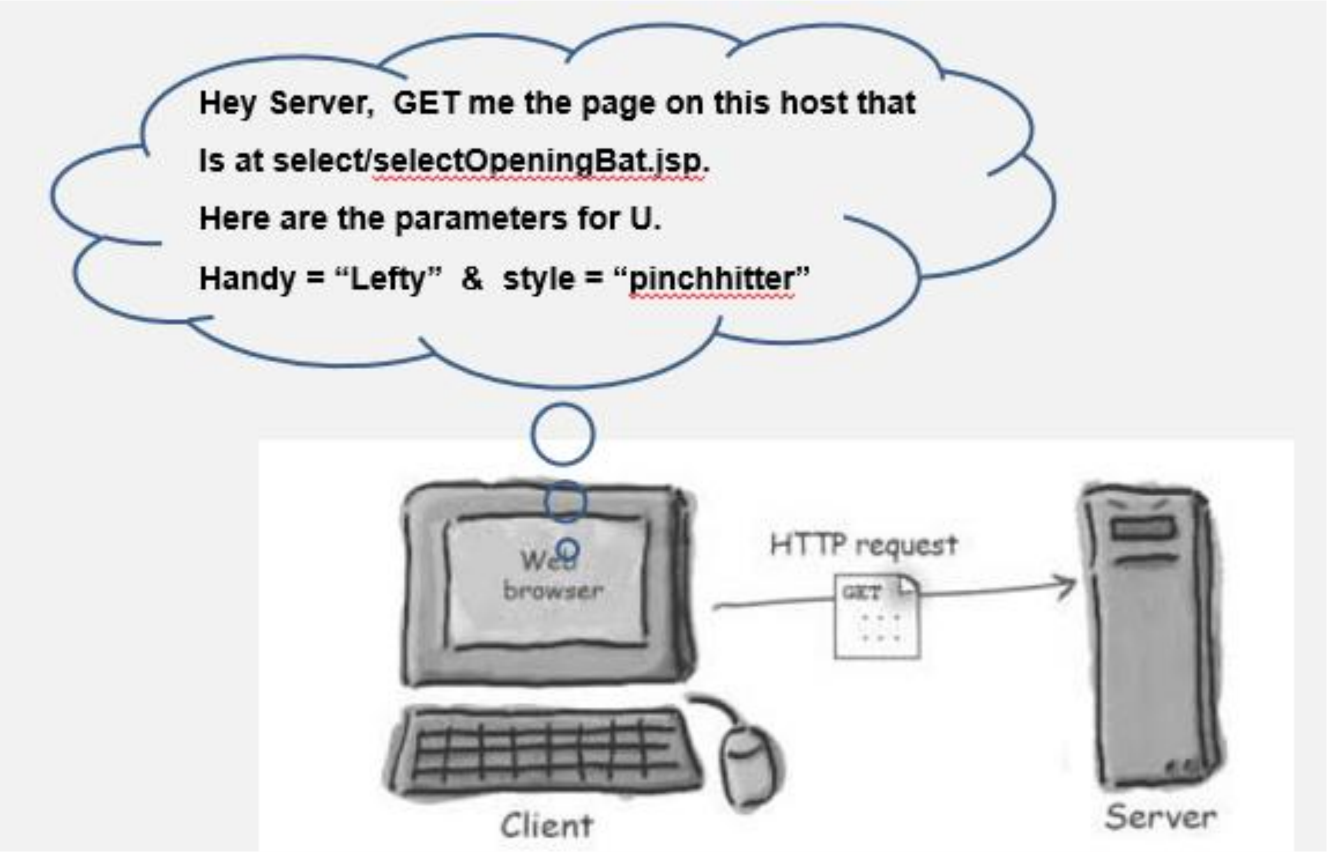
Accept : text - HTML / XML XSD – DTD/ image – jpeg – gif – bmp / video – flv / mp4

Accept Language : en – US / en - GEN

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

GET Request Header

HTTP GET REQUEST in Action

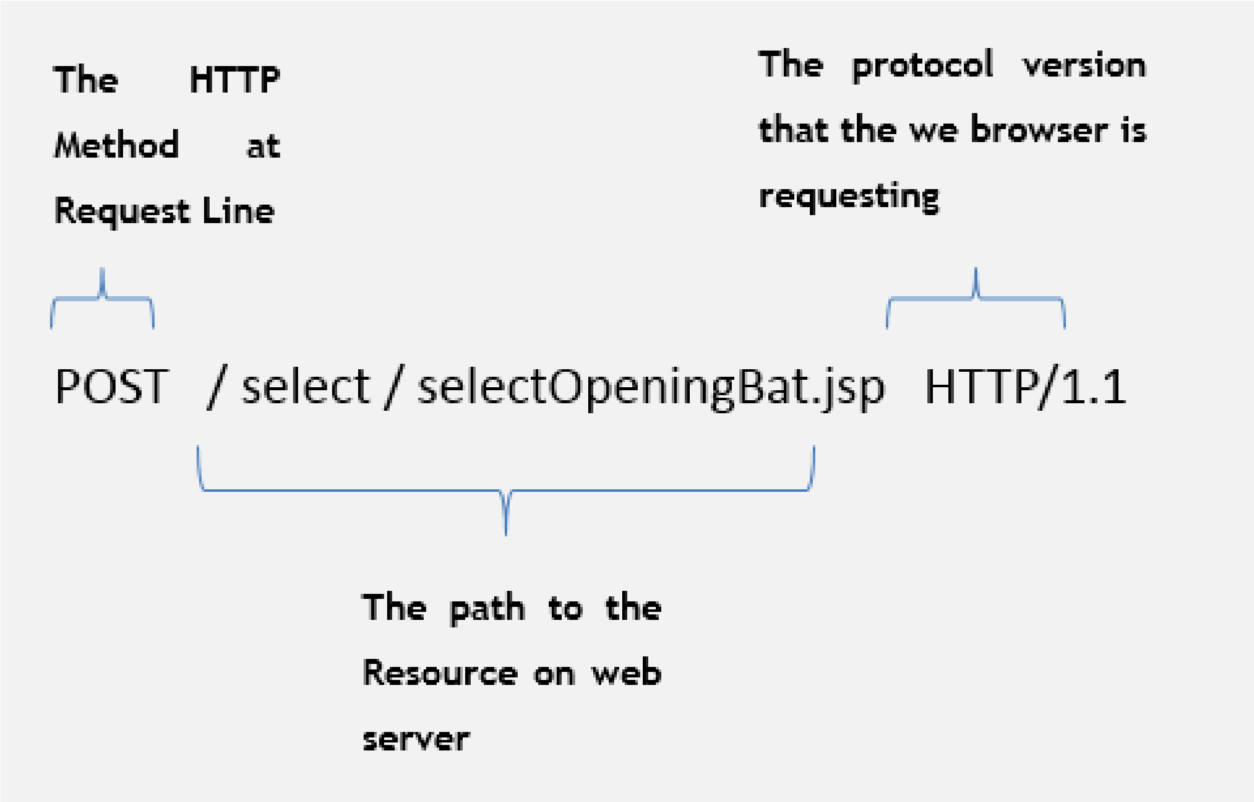


Hey Server, GET me the page on this host that Is at select/selectOpeningBat.jsp.

Here are the parameters for U.

Handy = “Lefty” & style = “pinchhitter”

Anatomy of HTTP POST REQUEST



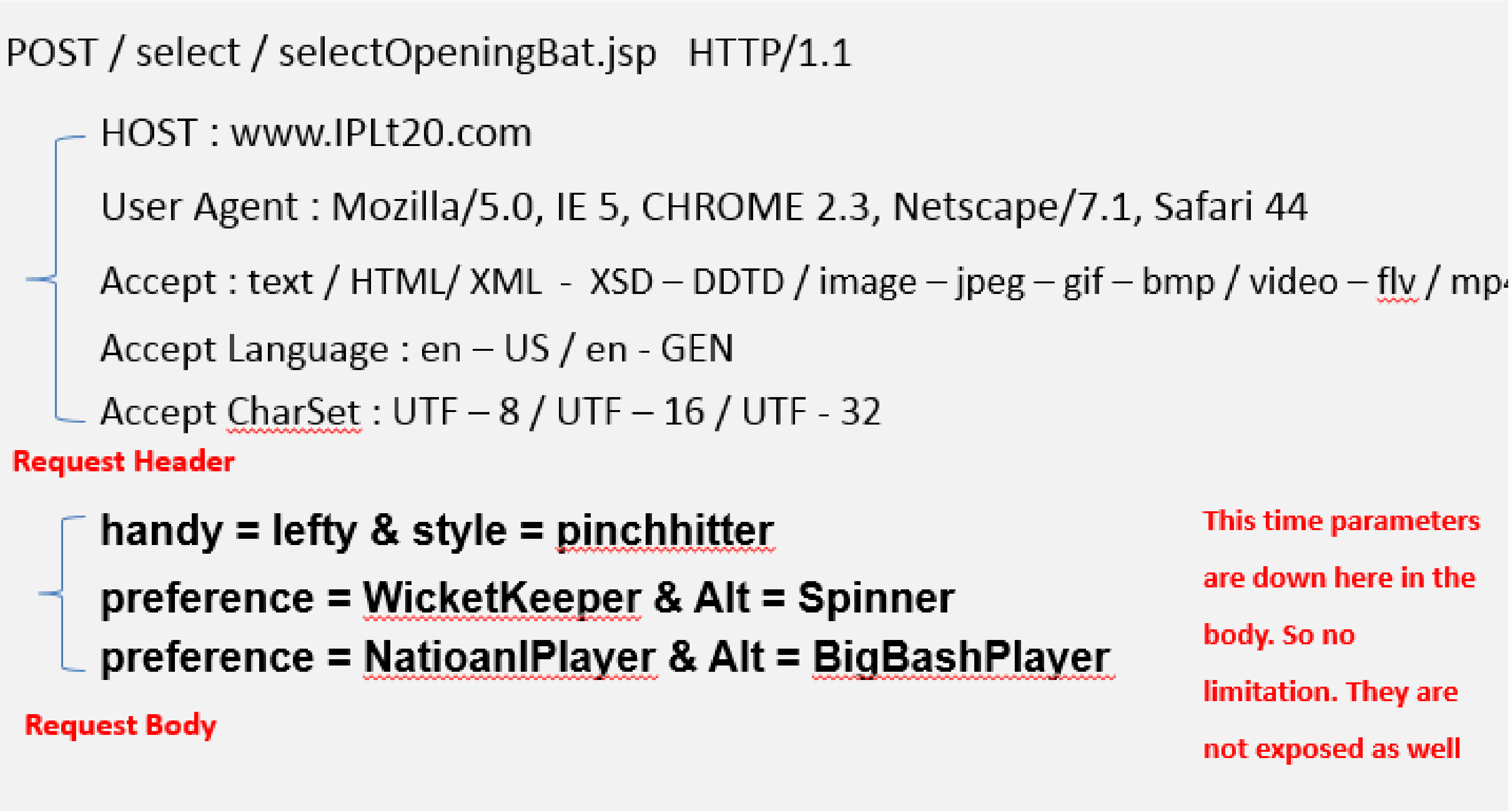
The HTTP Method at Request Line

The protocol version that the we browser is requesting

POST / select / selectOpeningBat.jsp HTTP/1.1

The path to the Resource on web server

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

handy = lefty & style = pinchhitter

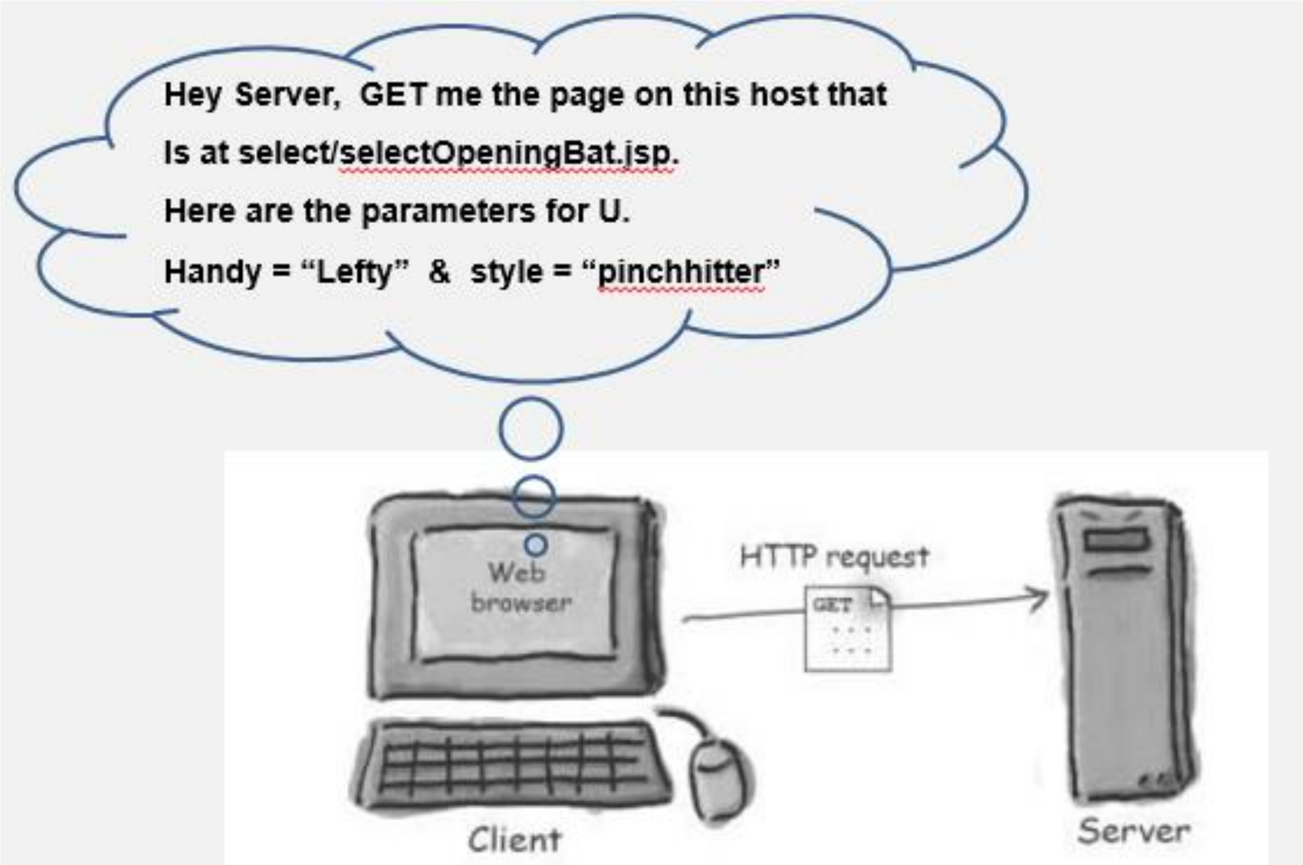
preference = WicketKeeper & Alt = Spinner

preference = NatioanlPlayer & Alt = BigBashPlayer

Request Body

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

HTTP POST REQUEST in Action



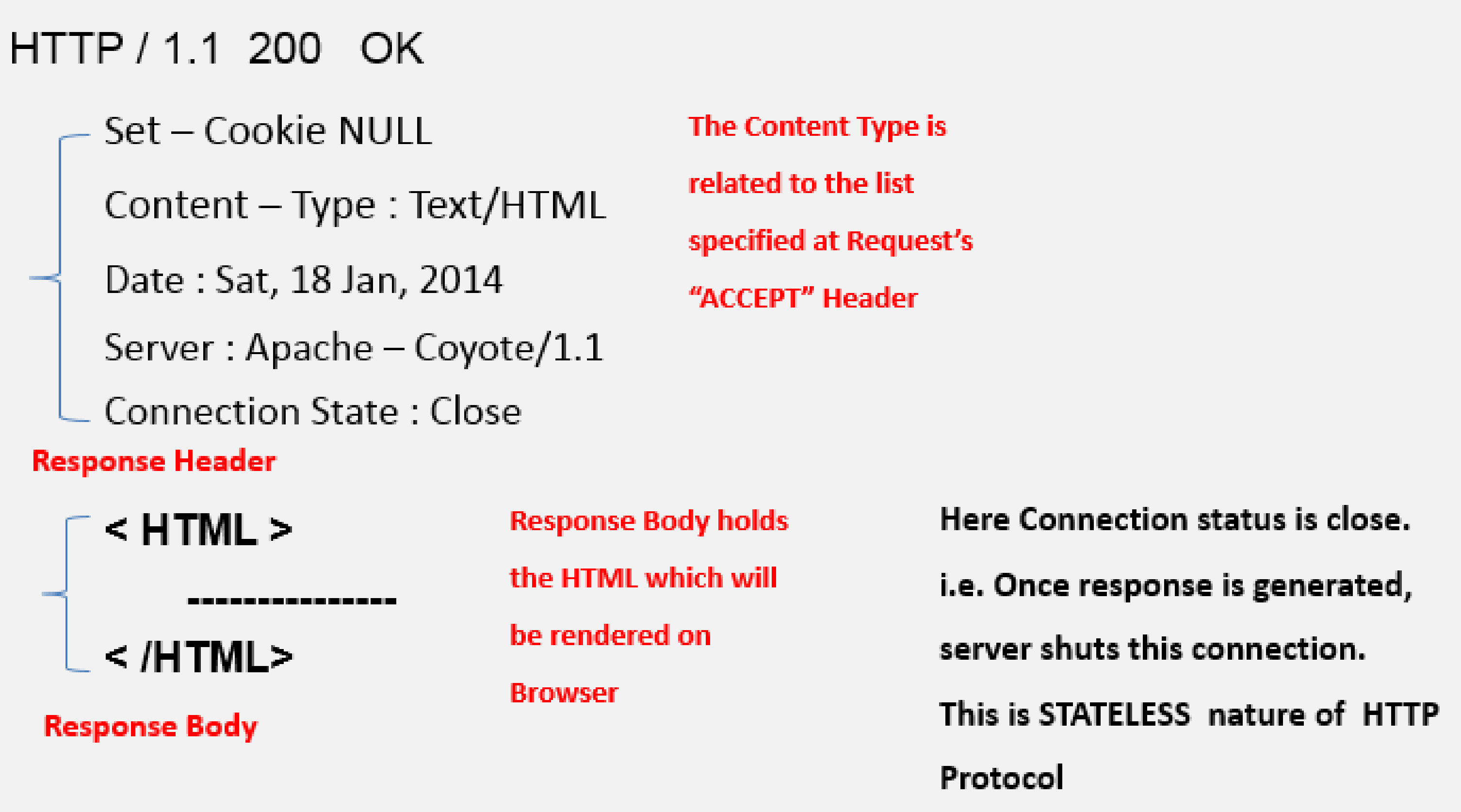
Hey Server, GET me the page on this host that

Is at select/selectOpeningBat.jsp.

Here are the parameters for U.

Handy = “Lefty” & style = “pinchhitter”

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

Response Header

The Content Type is related to the list specified at Request's "ACCEPT" Header

< HTML >

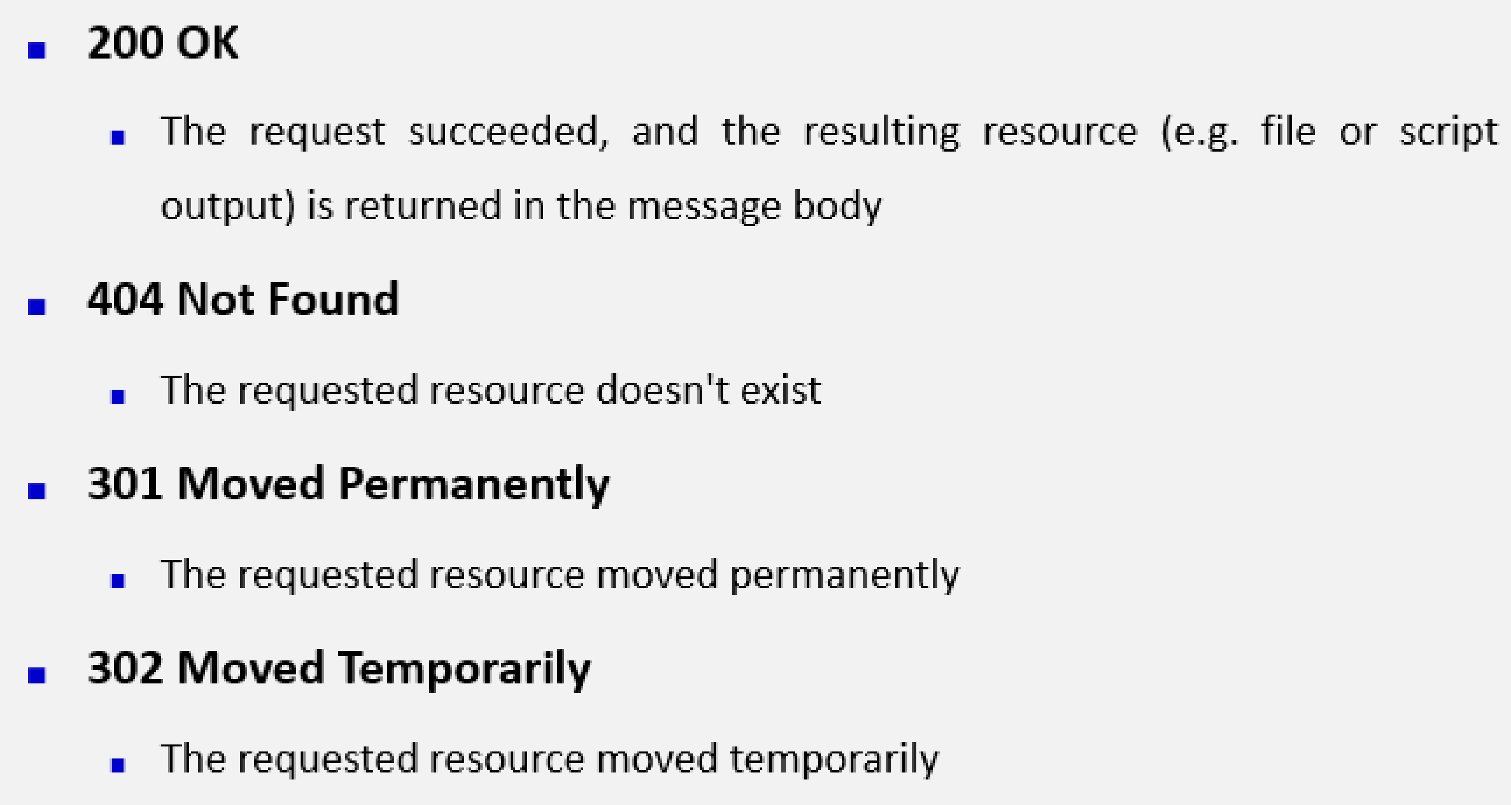
< /HTML>

Response Body

Response Body holds the HTML which will be rendered on Browser

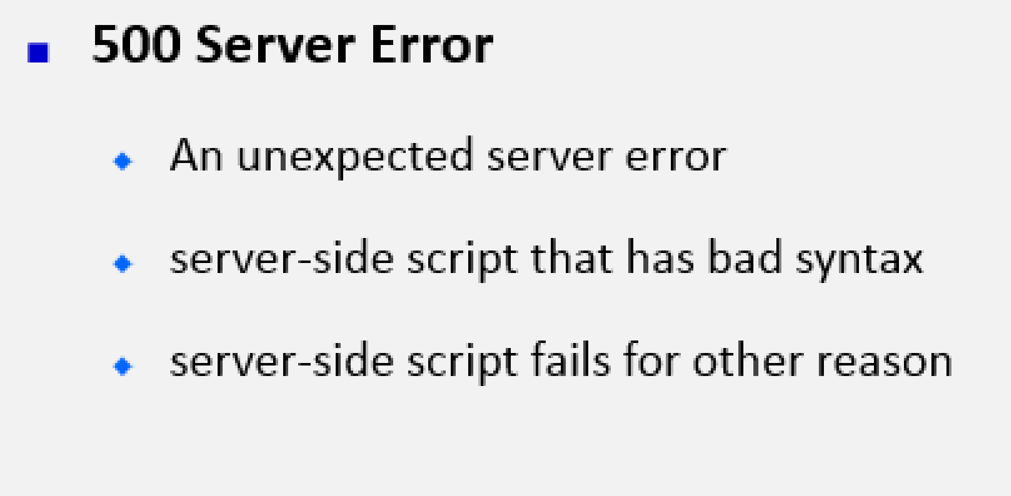
Here Connection status is close. i.e. Once response is generated, server shuts this connection. This is STATELESS nature of HTTP Protocol

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

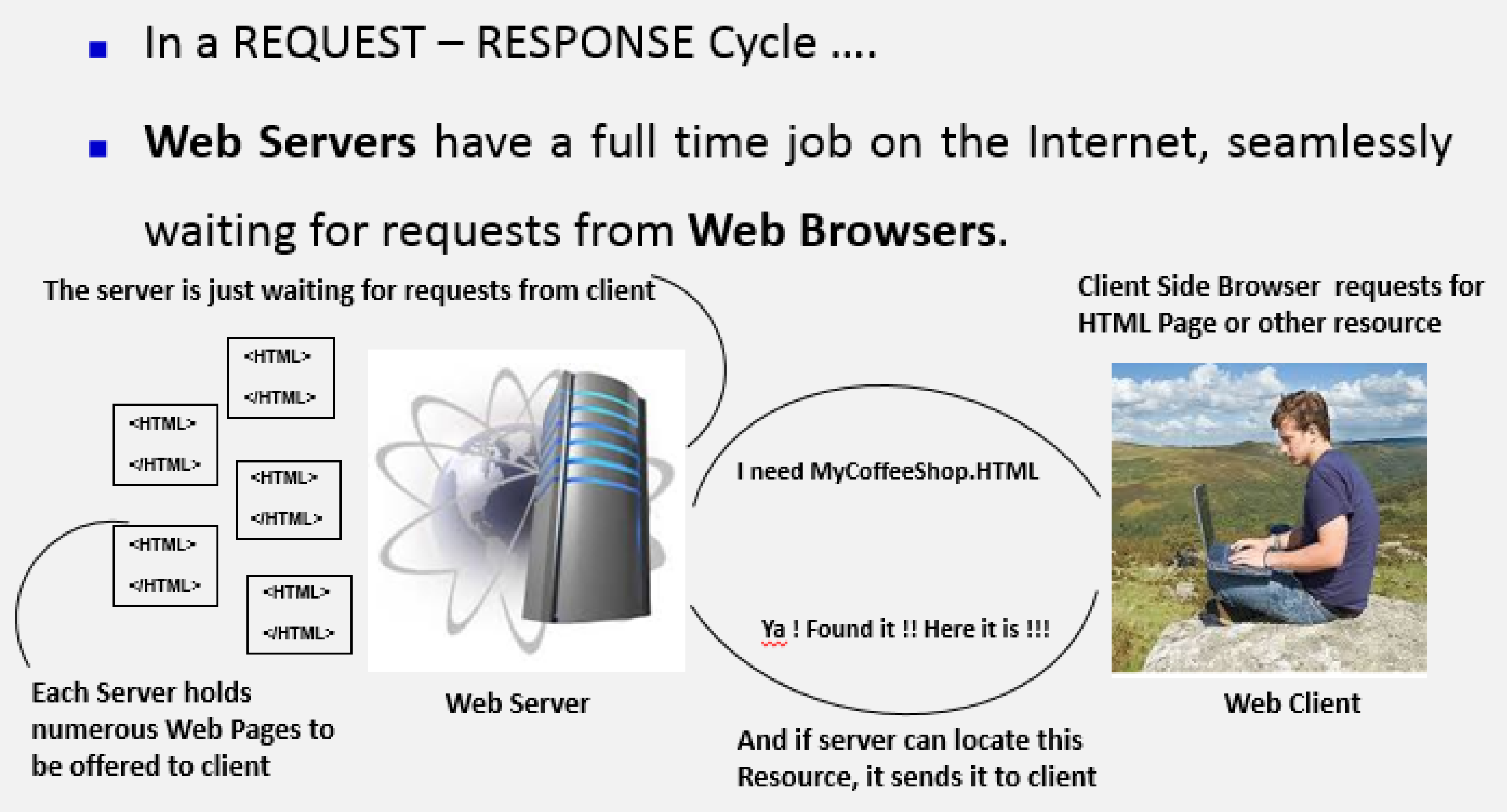
Let's Solve

Celebrating 20 Years

HTML

A Larsen & Toubro
Group Company

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

Client Side Browser requests for HTML Page or other resource

I need MyCoffeeShop.HTML

Ya ! Found it !! Here it is !!!

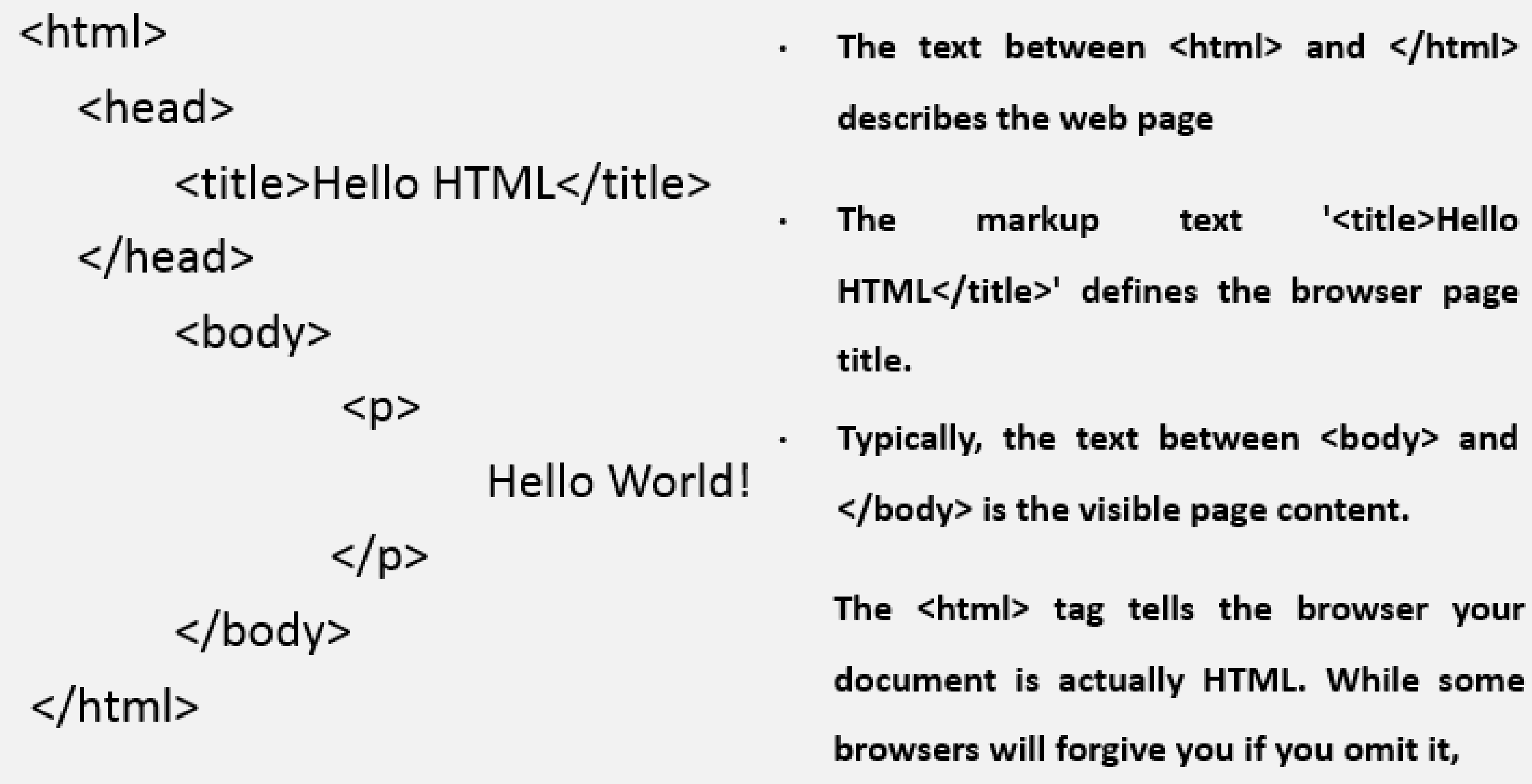
Each Server holds numerous Web Pages to be offered to client

Web Server

And if server can locate this Resource, it sends it to client

Web Client

Hello world – The HTML Code



```
<html>
     <head>
          <title>Hello HTML</title>
     </head>
          <body>
                    <p>
                              Hello World!
                    </p>
          </body>
 </html>
```

- 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 <html> tag tells the browser your document is actually HTML. While some browsers will forgive you if you omit it,

CSS

Generalized Look & 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,

This Mechanism is known as CSS – Cascading Style Sheet

Okay, let's get some design in this place...

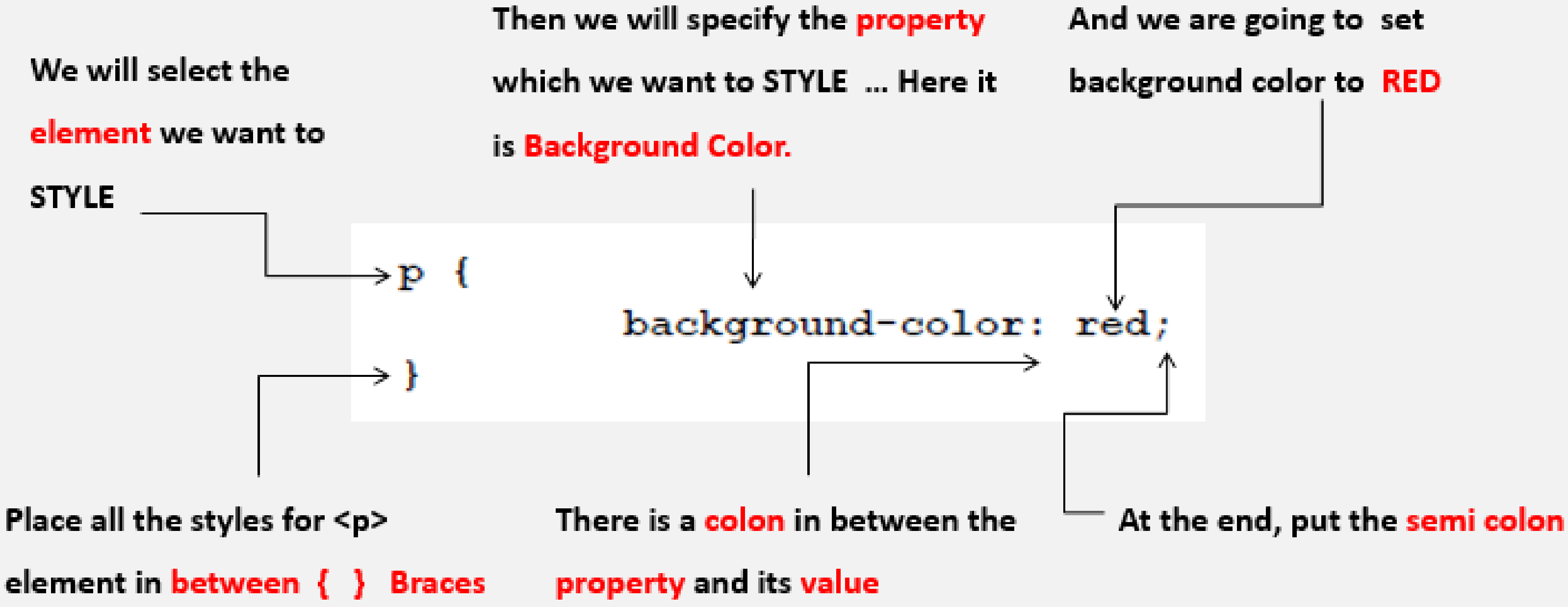
```
bedroom {
    drapes: blue;
    carpet: wool shag;
}
```

...and this bathroom needs some serious help!

```
bathroom {
        tile: 1in white;
        drapes: pink;
}
```

Using CSS with HTML

- Want to paint the walls of our <p> elements to red?
- No probs ………………. Here is the way……………………



We will select the element we want to STYLE

Then we will specify the property which we want to STYLE … Here it is Background Color.

And we are going to set background color to RED

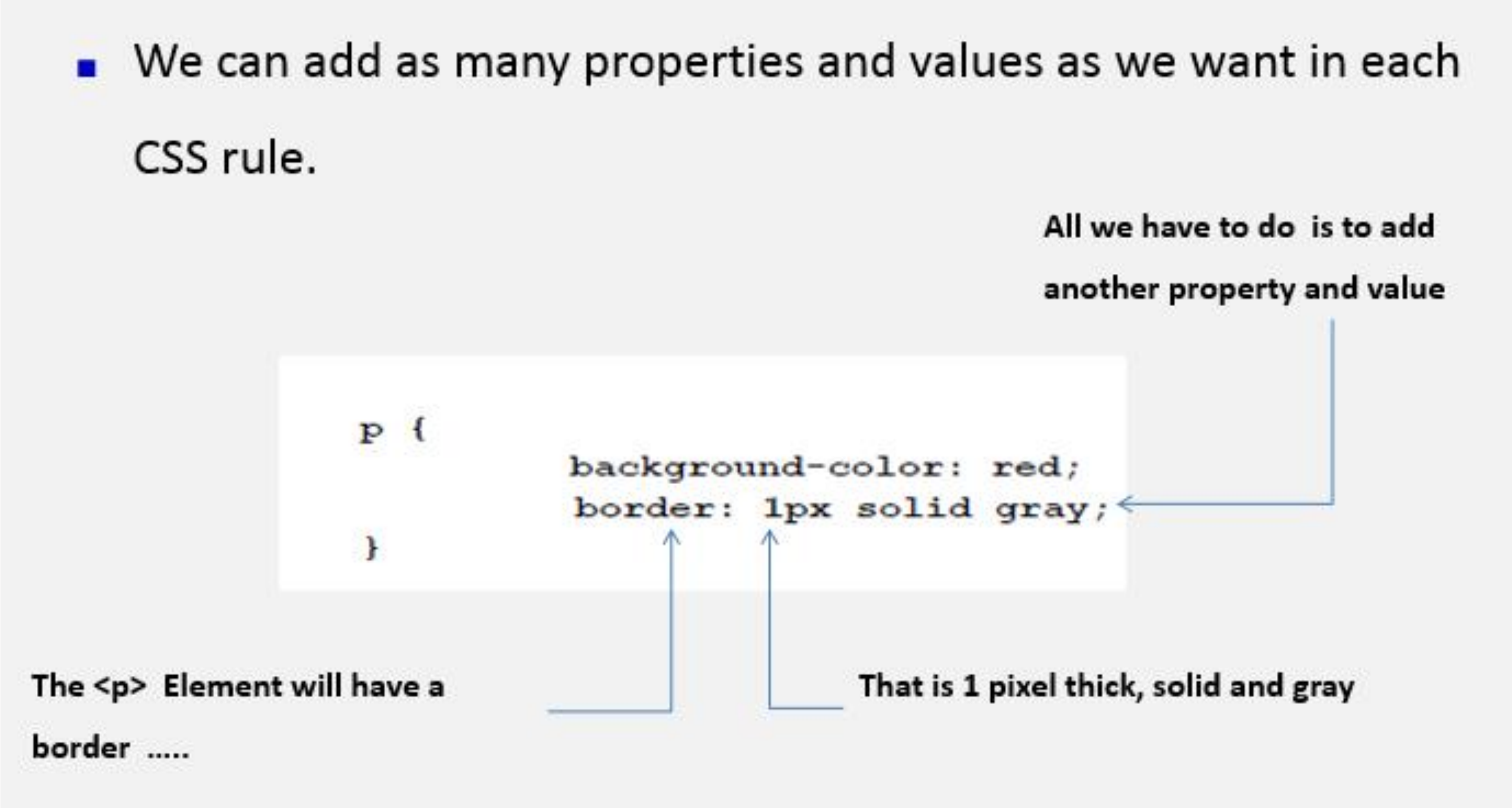
```
p {
        background-color: red;
}
```

Place all the styles for <p> element in between { } Braces

There is a colon in between the property and its value

At the end, put the semi colon

Wanna add more style?



- We can add as many properties and values as we want in each CSS rule.

All we have to do is to add another property and value

```
p {
        background-color: red;
        border: 1px solid gray;
}
```

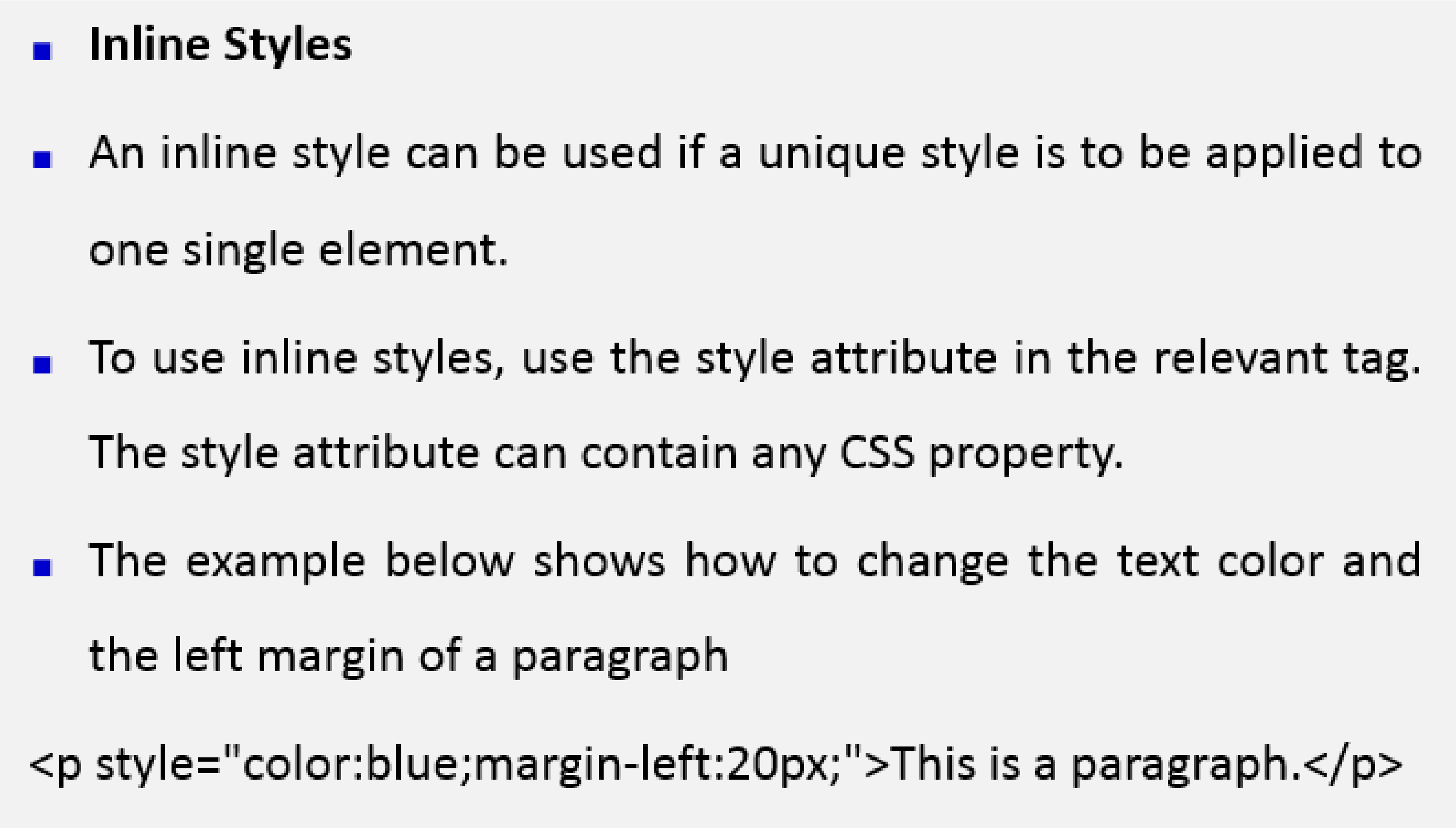
The <p> Element will have a border .....

That is 1 pixel thick, solid and gray

Styling HTML with CSS

- 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 obviously 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.

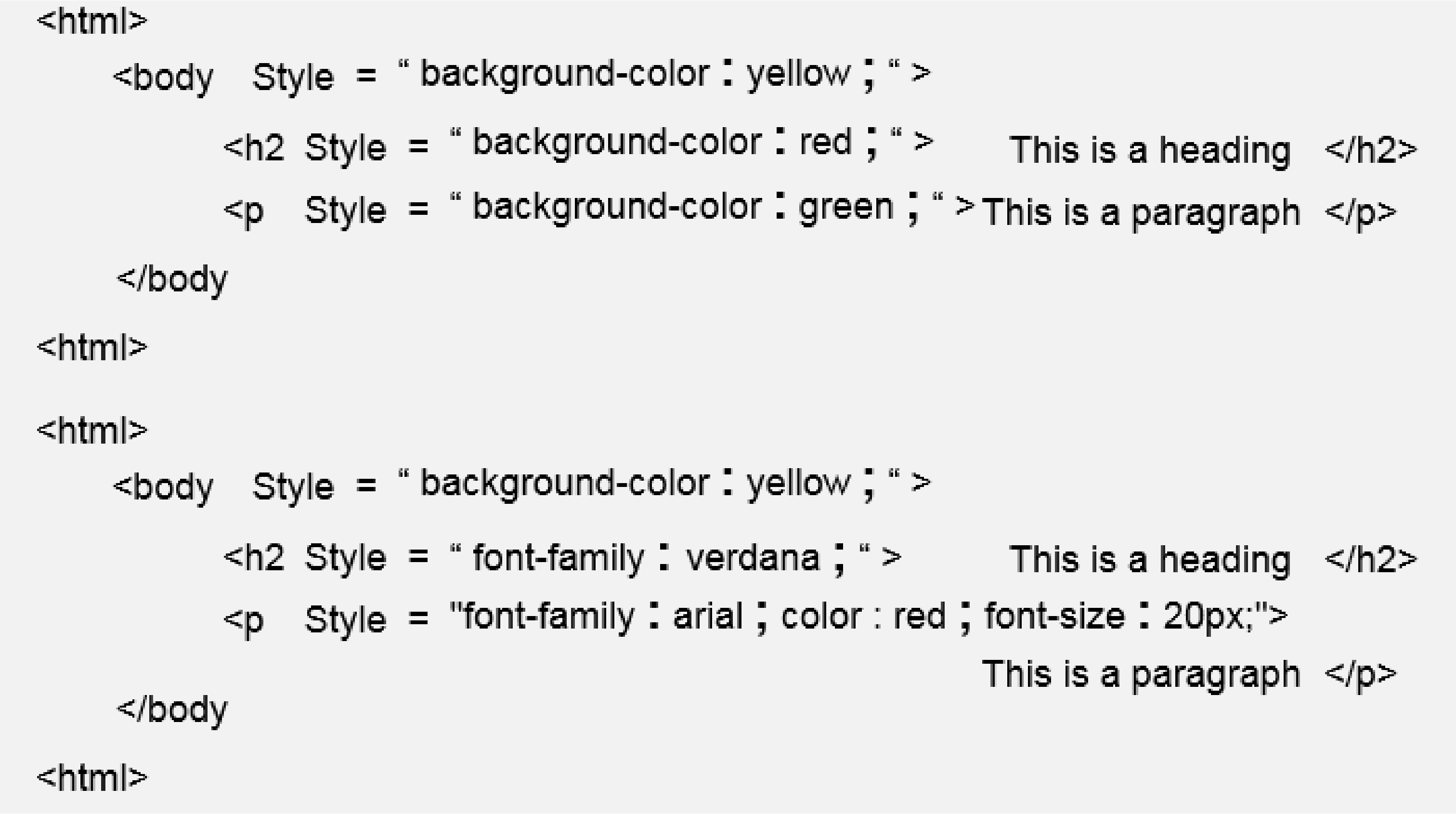
Using the Style Sheets



- **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

```
<p style="color:blue;margin-left:20px;">This is a paragraph.</p>
```

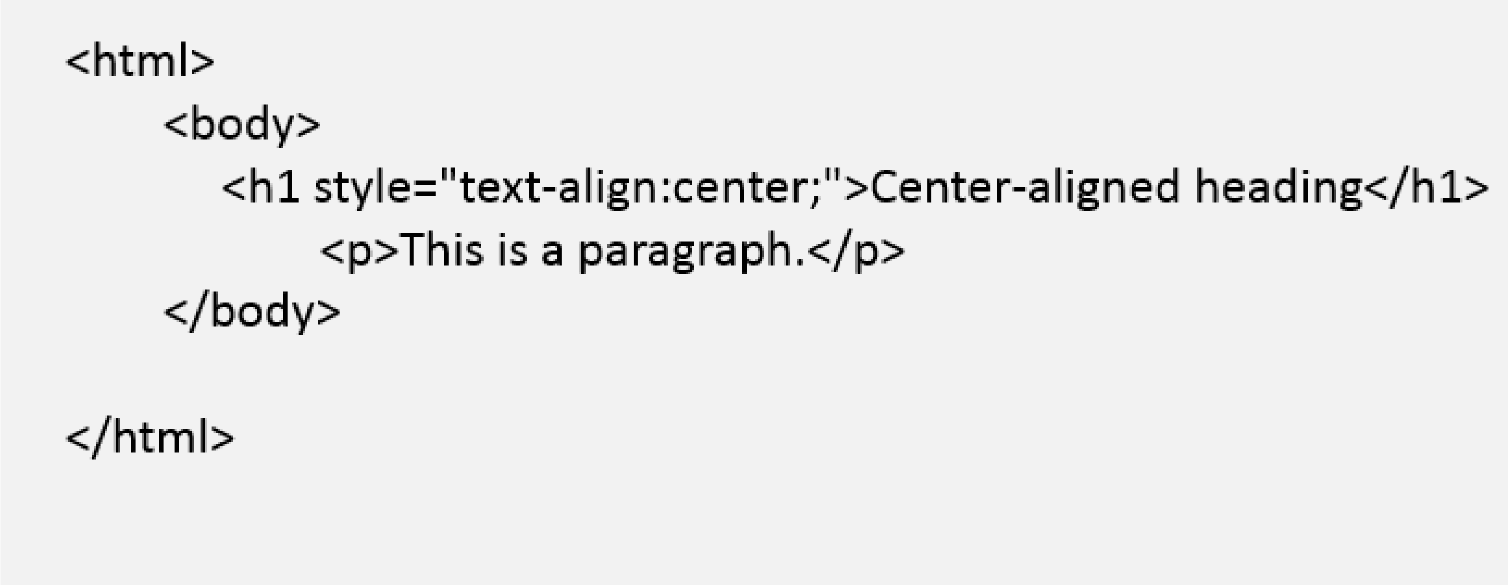
HTML Style - Background Color - Font, Color and Size



```
<html>
     <body   Style  =  “ background-color : yellow ; “ >
            <h2  Style  =  “ background-color : red ; “ >       This is a heading   </h2>
            <p    Style  =  “ background-color : green ; “ > This is a paragraph  </p>
     </body
<html>

<html>
     <body   Style  =  “ background-color : yellow ; “ >
            <h2  Style  =  “ font-family : verdana ; “ >          This is a heading   </h2>
            <p    Style  =  "font-family : arial ; color : red ; font-size : 20px;">
                                                                 This is a paragraph  </p>
     </body
<html>
```

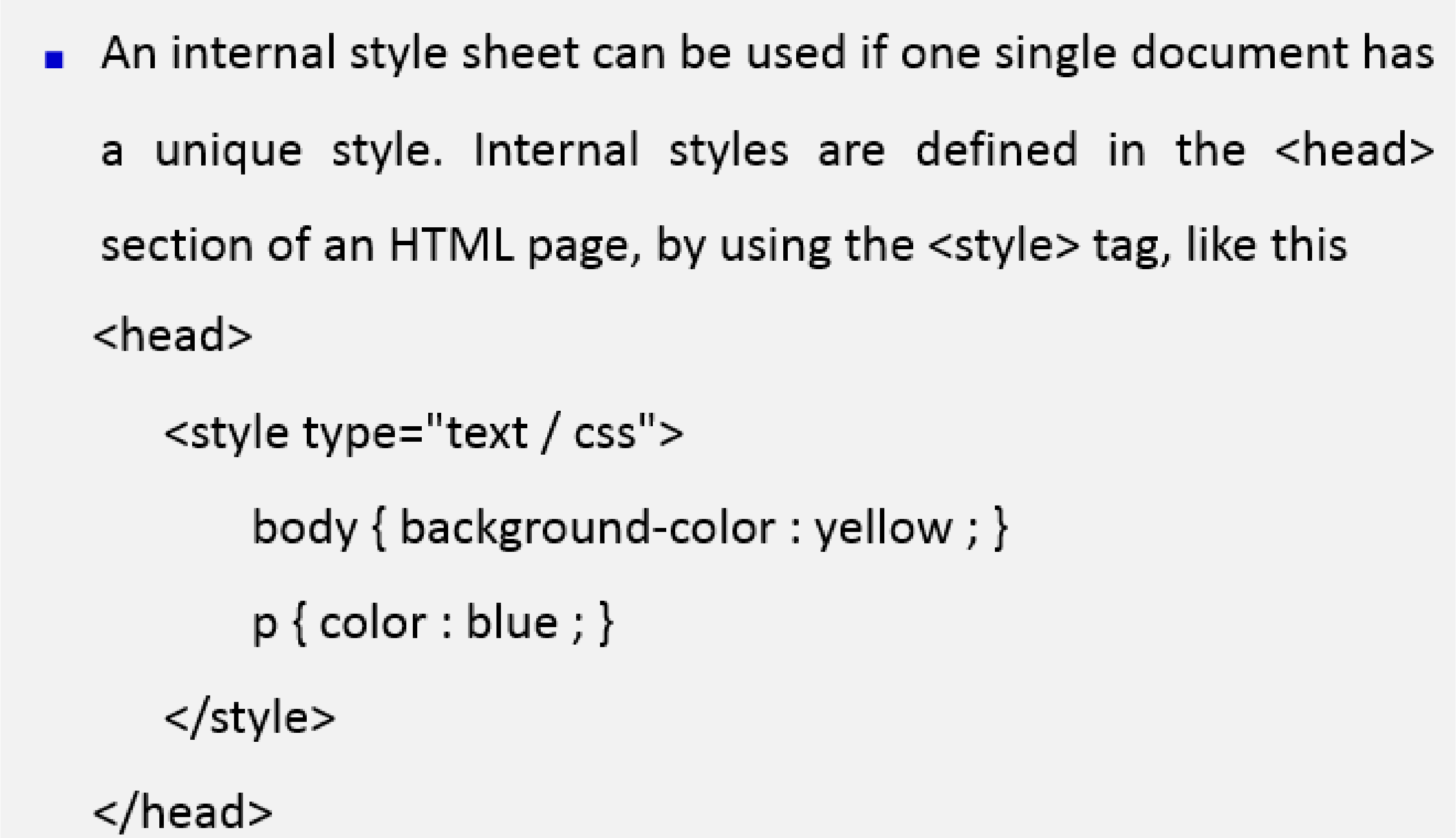
HTML Style Example - Text Alignment



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

</html>
```

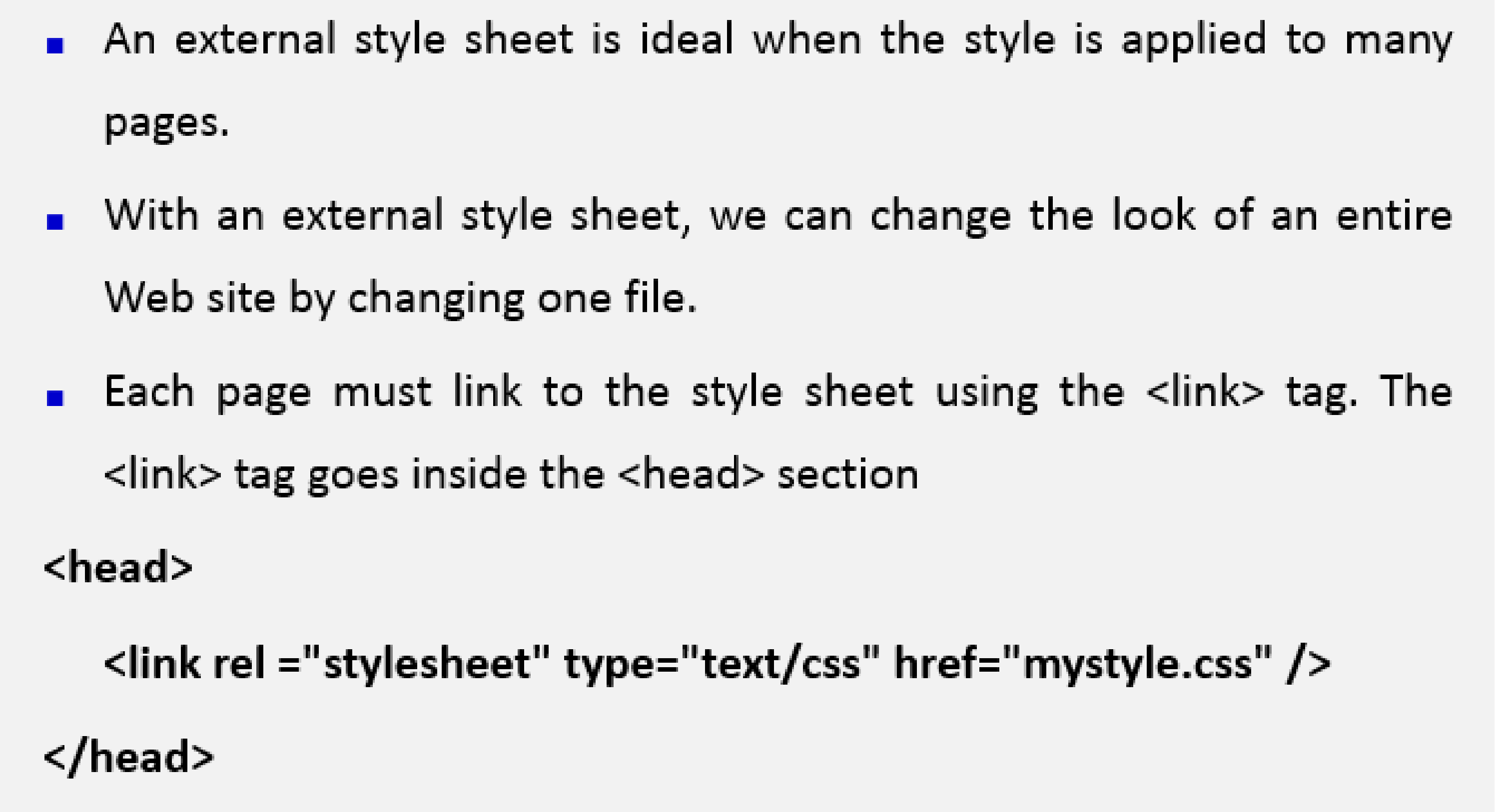
HTML Style Example - Text Alignment



- 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

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

External Style Sheet



- 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 Web site by changing one file.
- Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the <head> section

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

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 Overview

- 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.

HTML5 Overview

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.

HTML5 Overview

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).

HTML5 Overview

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:

- *For 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>
  <p>Refresh the page to increase number of hits.</p>
  <p>Close the window and open it again and check the result.</p>

</body>
</html>
```

HTML5 - Web Storage

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.

HTML5 - Web Storage

- 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>
  <p>Refresh the page to increase number of hits.</p>
  <p>Close the window and open it again and check the result.</p>

</body>
</html>
```

HTML5 - Web Storage

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.

HTML5 - Web Storage

- 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>
  <p>Refreshing the page would not to increase hit counter.</p>
  <p>Close the window and open it again and check the result.</p>

</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 <canvas> 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>
```

HTML5 - Geolocation

- 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 location-aware things like finding local businesses or showing your location on a map.

HTML5 - Geolocation

- 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.

CSS3

CSS3 Introduction

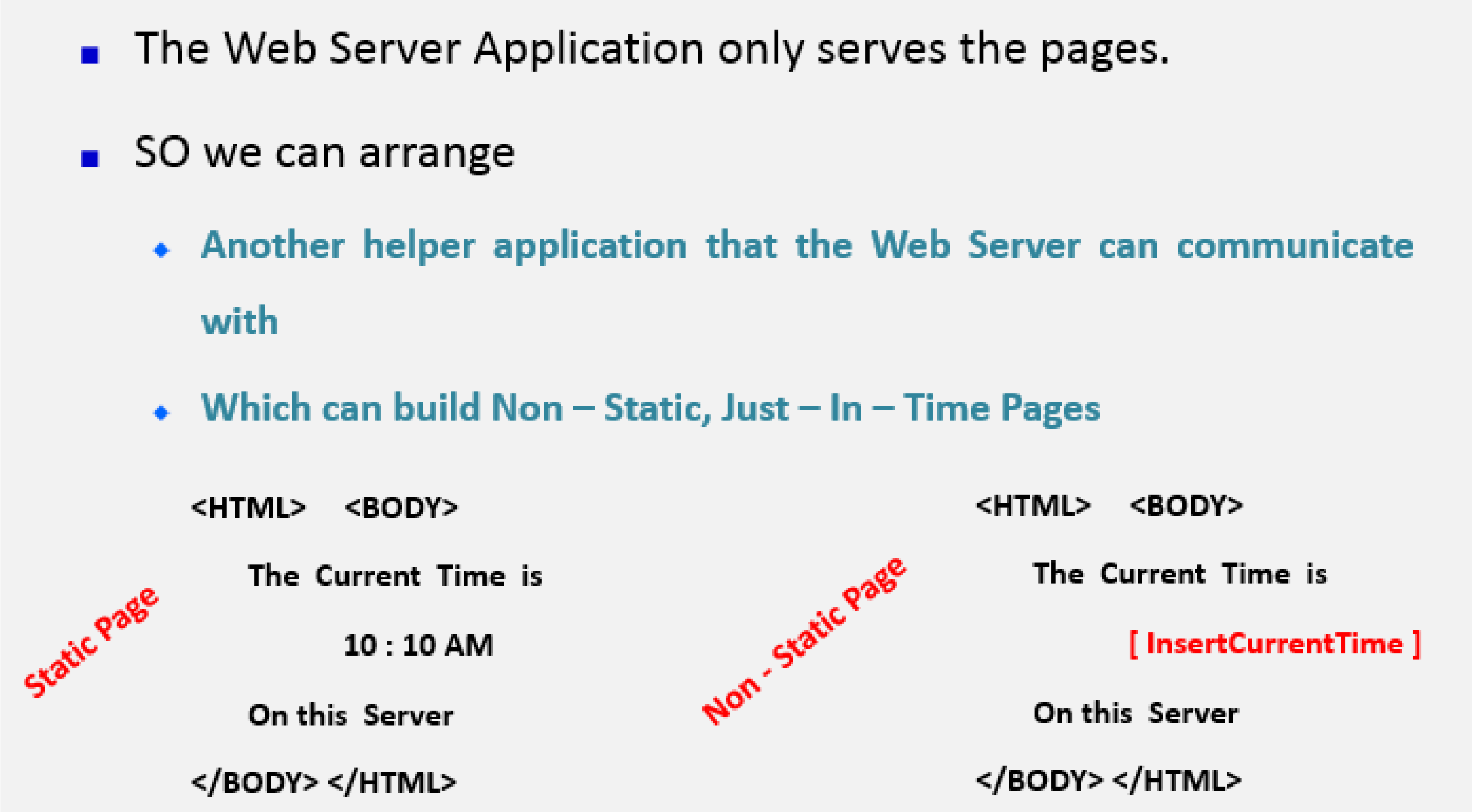
- CSS3 is completely backwards compatible, so you will not have to change existing designs. Browsers will always support CSS2.

- 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
  - 2D/3D Transformations
  - Animations
  - Multiple Column Layout
  - User Interface

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

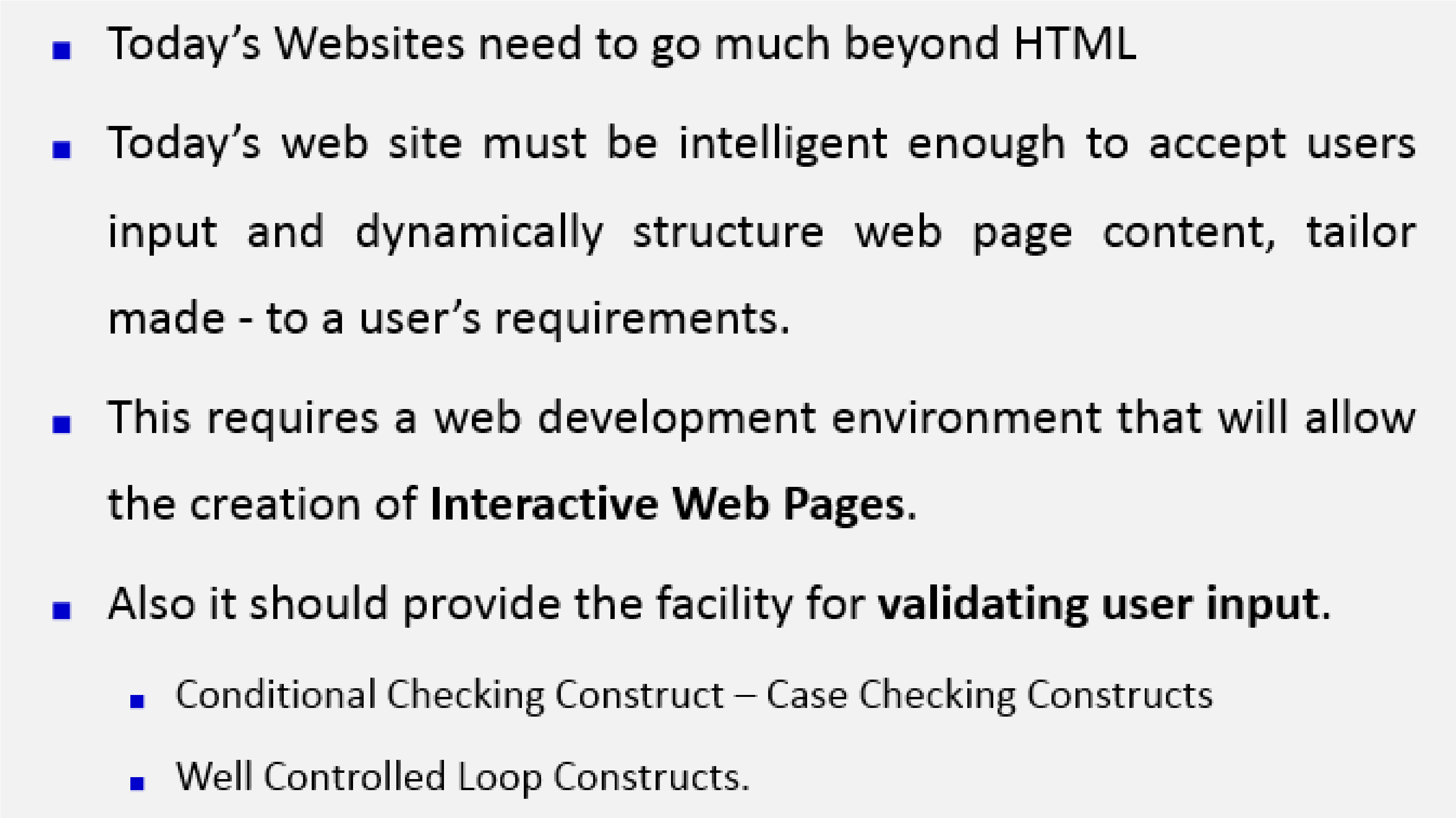
Static Page

```
<HTML>    <BODY>
     The  Current  Time  is
            10 : 10 AM
     On this  Server
</BODY> </HTML>
```

Non - Static Page

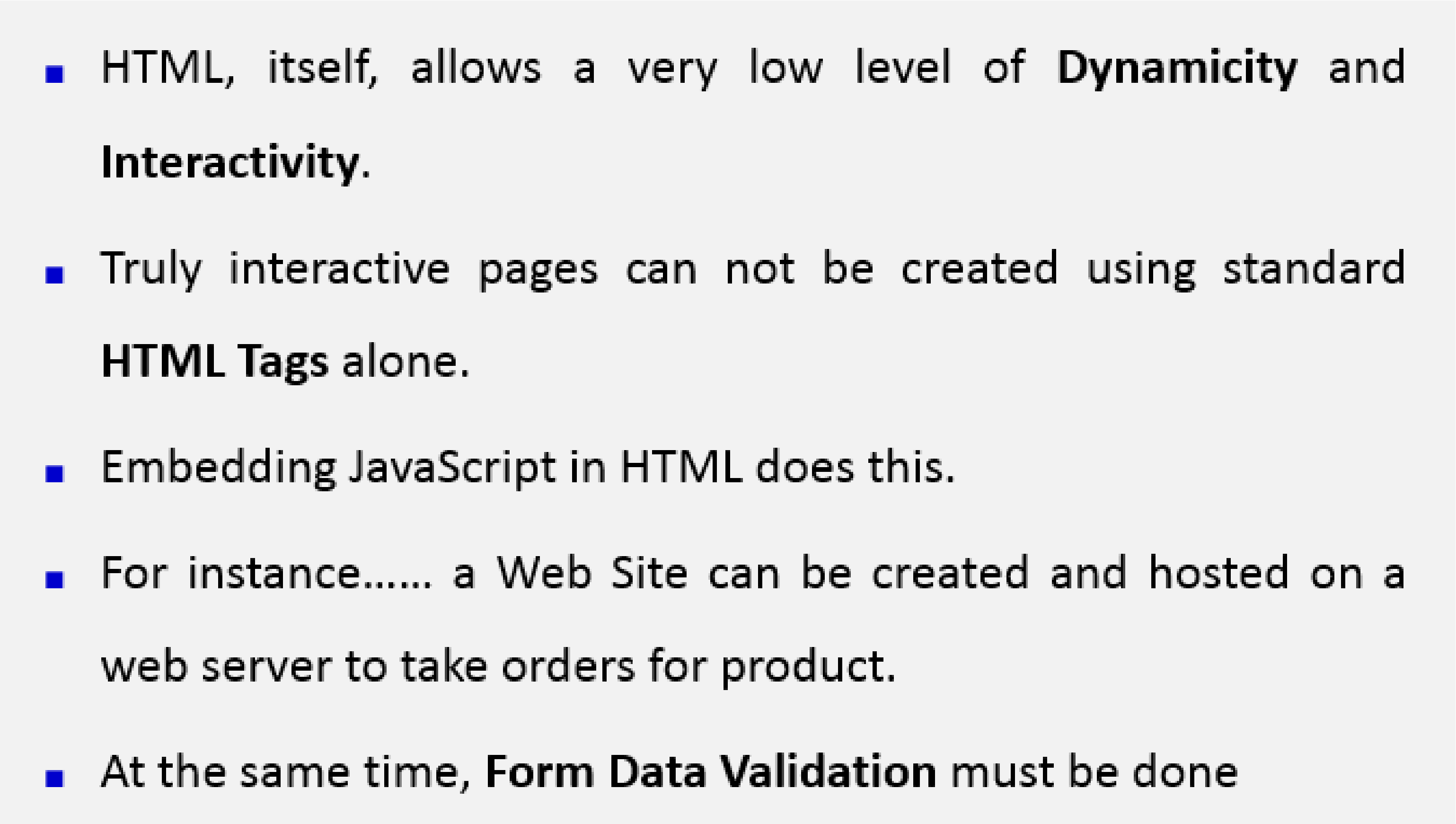
```
<HTML>    <BODY>
     The  Current  Time  is
               [ InsertCurrentTime ]
     On this  Server
</BODY> </HTML>
```

Introduction to JavaScript



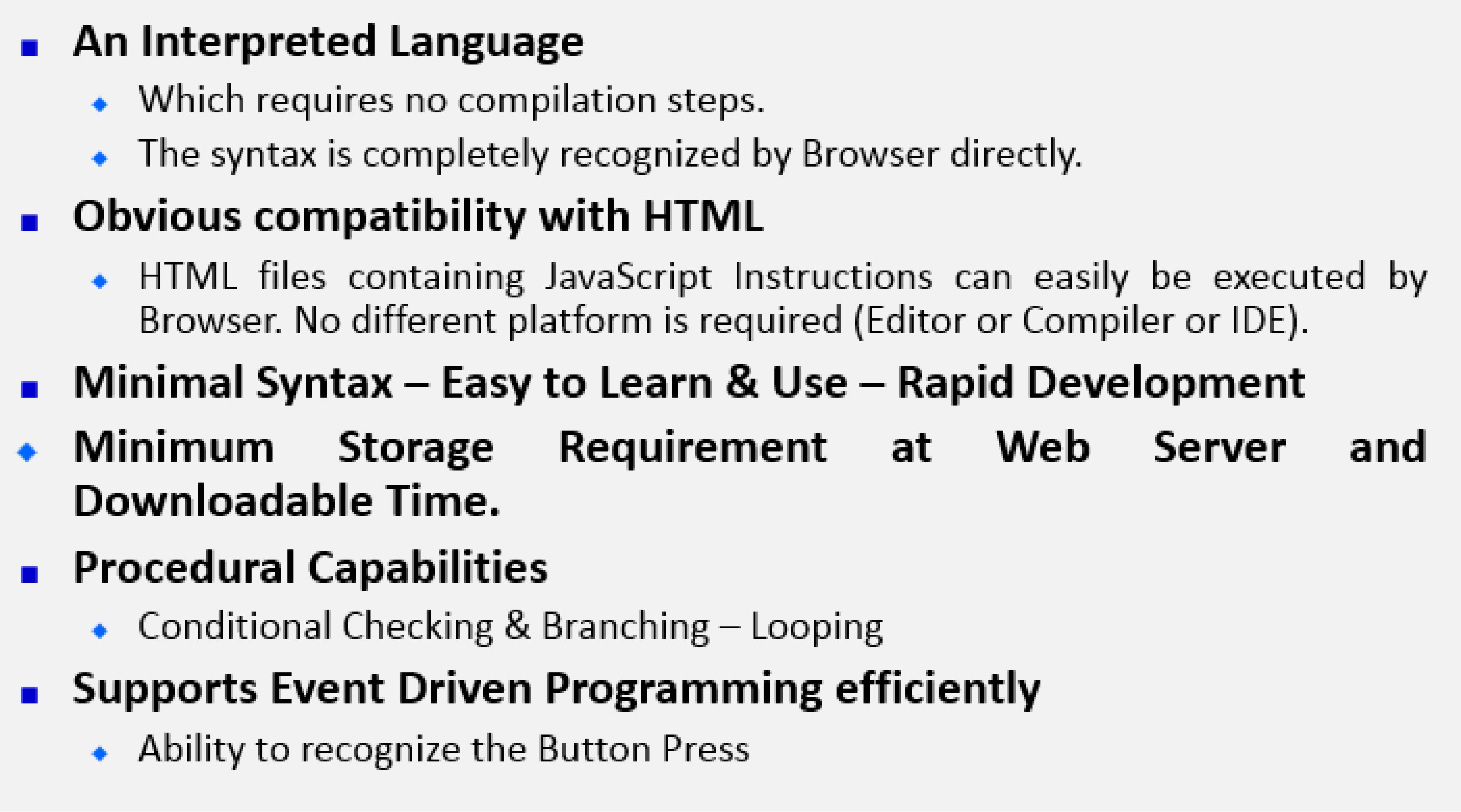
- Today’s Websites need to go much beyond HTML
- Today’s web site must be intelligent enough to accept users input and dynamically structure web page content, tailor made - to a user’s requirements.
- This requires a web development environment that will allow the creation of **Interactive Web Pages.**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

HTML & JavaScript



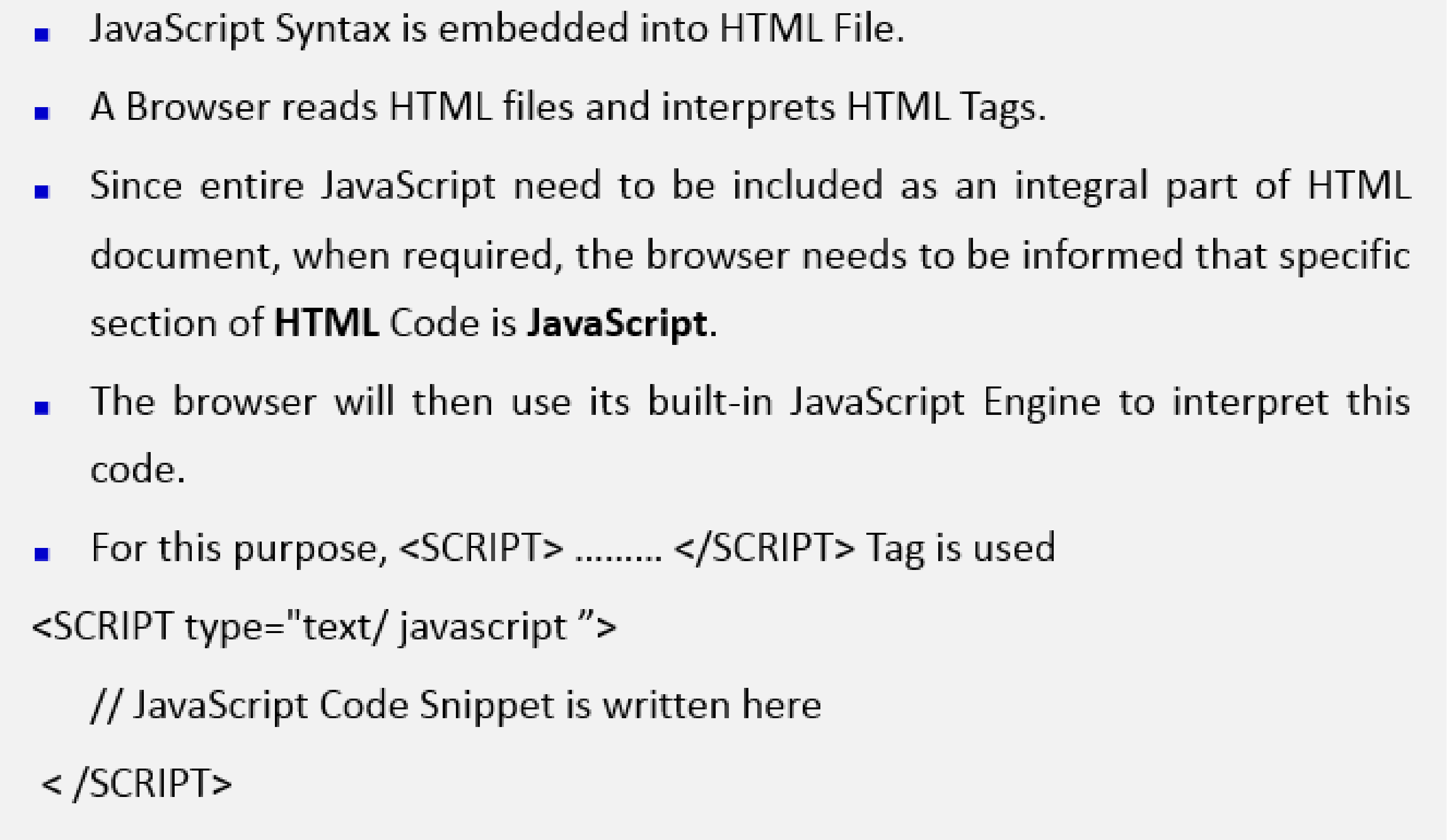
- HTML, itself, allows a very low level of **Dynamicity** and **Interactivity**.
- Truly interactive pages can not be created using standard **HTML Tags** alone.
- Embedding JavaScript in HTML does this.
- For instance…… a Web Site can be created and hosted on a web server to take orders for product.
- 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**
  - Ability to recognize the Button Press

Writing JavaScript Into HTML

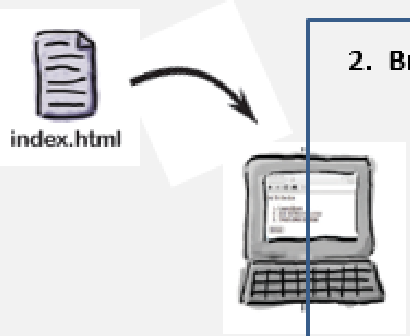


- JavaScript Syntax is embedded into HTML File.
- A Browser reads HTML files and interprets HTML Tags.
- Since entire JavaScript need to be included as an integral part of HTML document, when required, the browser needs to be informed that specific section of **HTML** Code is **JavaScript**.
- The browser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

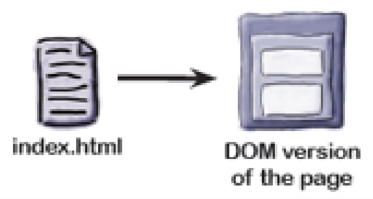
```
<SCRIPT type="text/ javascript ”>
    // JavaScript Code Snippet is written here
 < /SCRIPT>
```

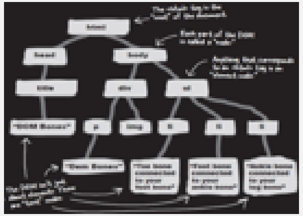
Look into the Browser

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



2. Browser builds a render able document using Document Object Model



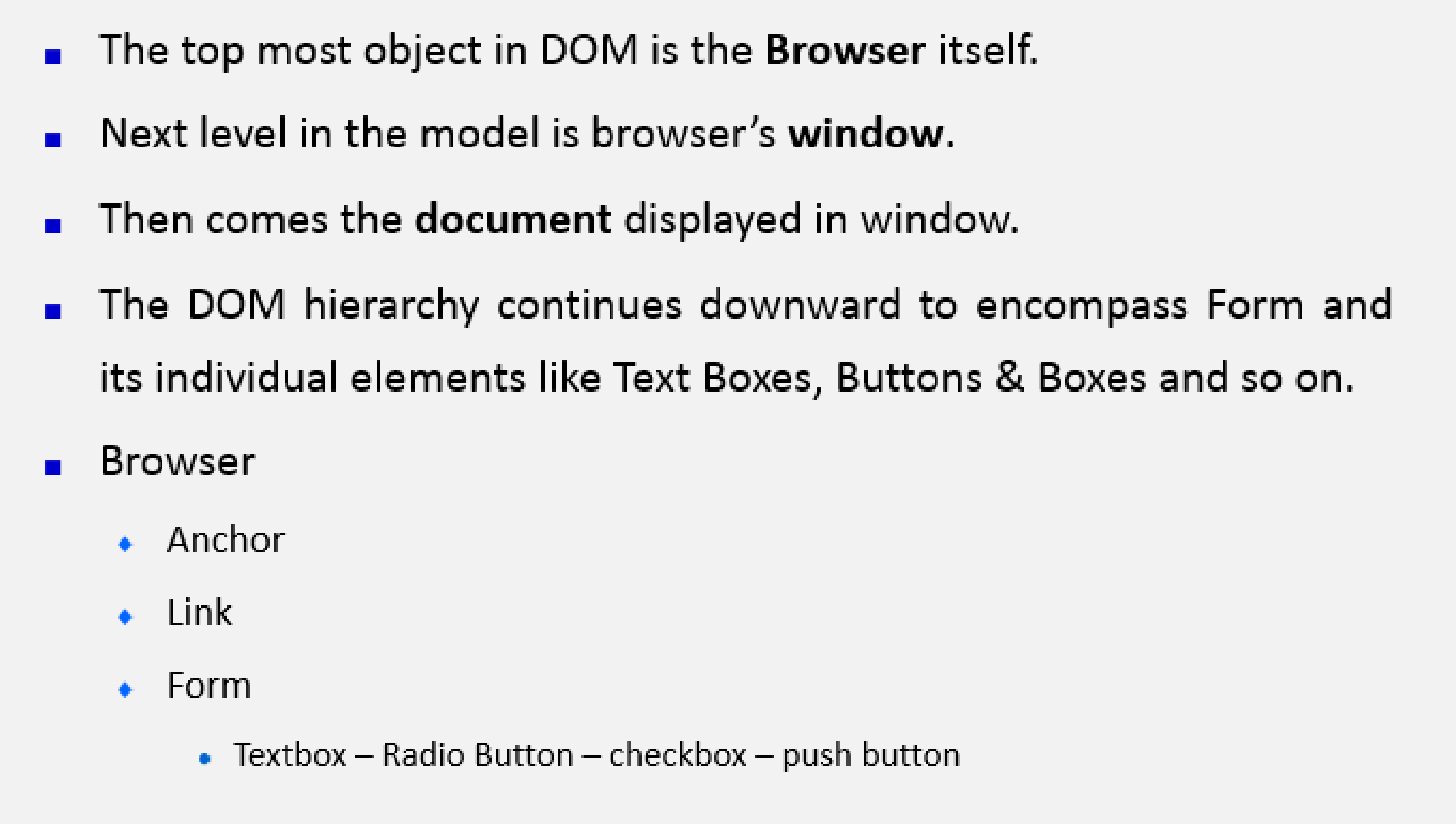


3. Browser renders DOM Objects in browser window according to the syntax and symatics

These are the stuffs happening inside the browser

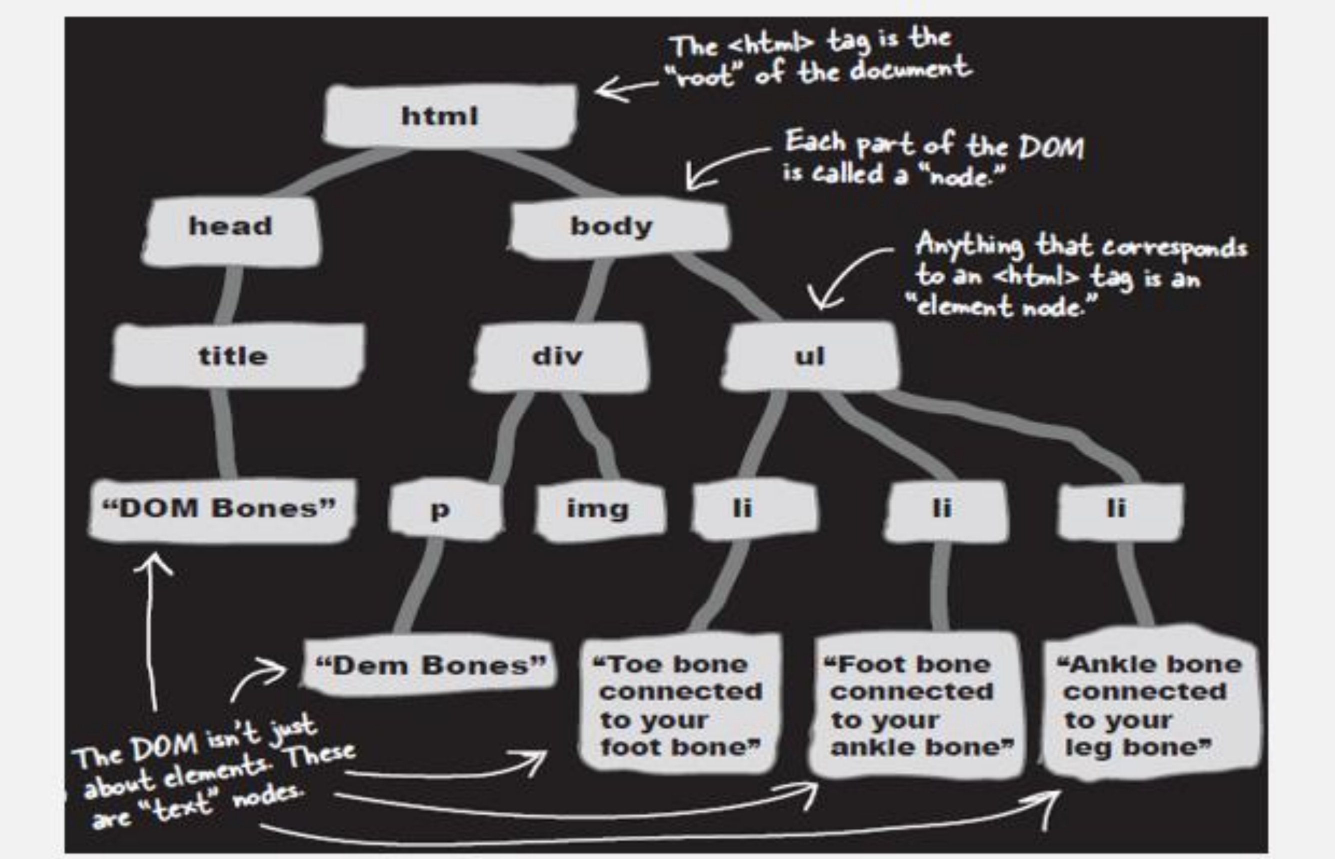
4. The JavaScript Interpreter references the DOM objects collectively and the JavaScript or jQuery functions operate on the objects individually / as a group.

According to Document Object Model

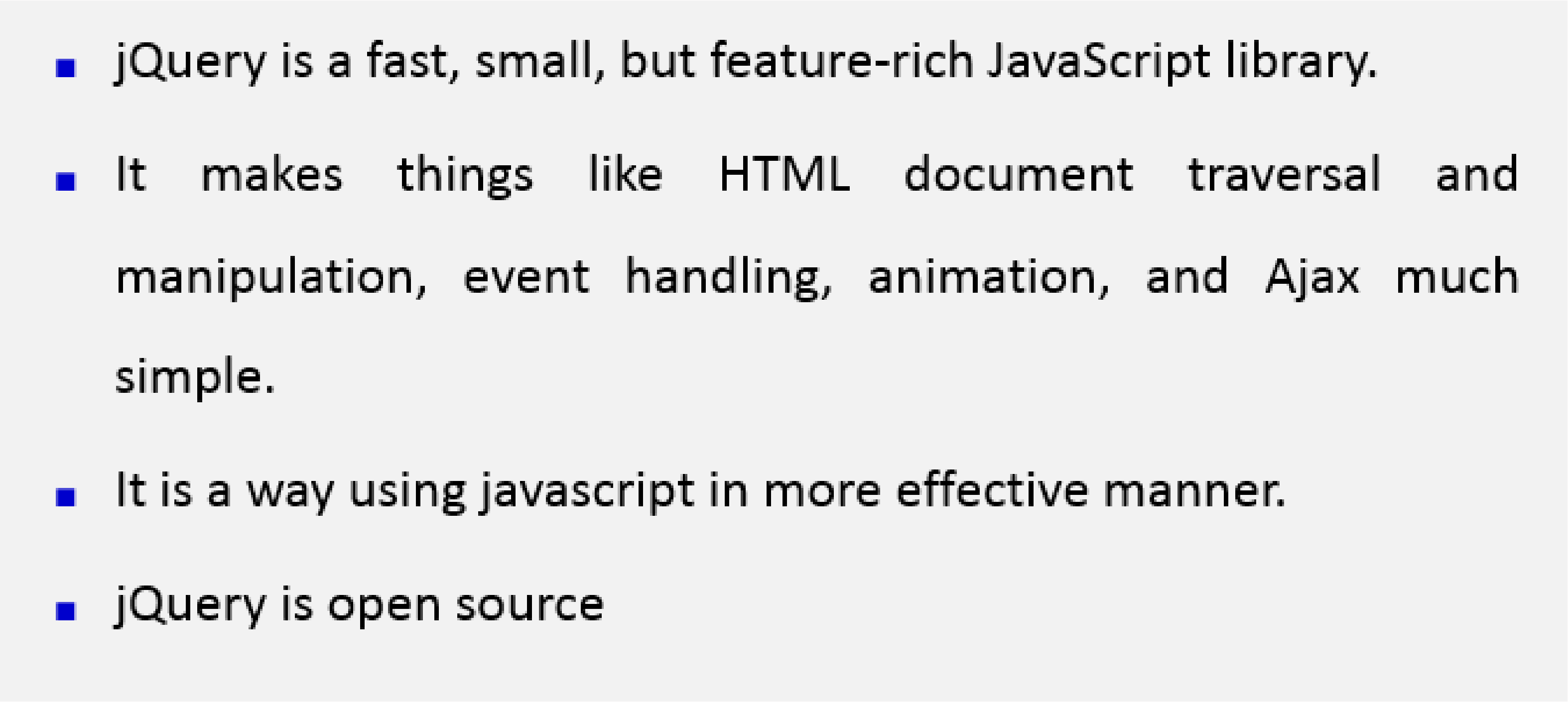


- The top most object in DOM is the **Browser** itself.
- Next level in the model is browser's **window**.
- Then comes the **document** displayed in window.
- The DOM hierarchy continues downward to encompass Form and its individual elements like Text Boxes, Buttons & Boxes and so on.
- Browser
  -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

The hidden structure of a web page – with DOM

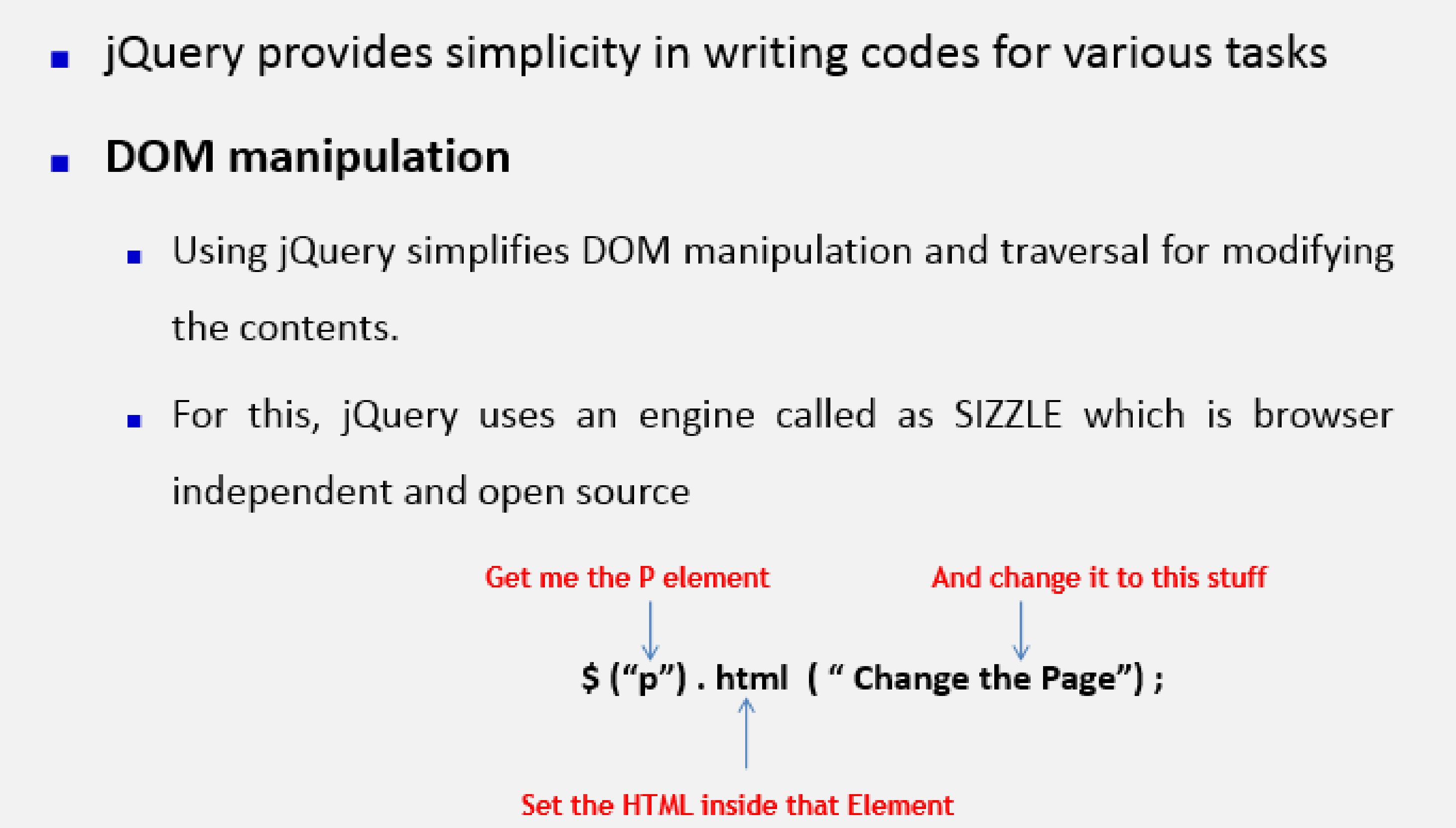

The <html> tag is the "root" of the document
html
Each part of the DOM is called a "node."
head
body
Anything that corresponds to an <html> tag is an "element node."
title
div
ul
"DOM Bones"
p
img
li
li
li
"Dem Bones"
"Toe bone connected to your foot bone"
"Foot bone connected to your ankle bone"
"Ankle bone connected to your leg bone"
The DOM isn't just about elements. These are "text" nodes.

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

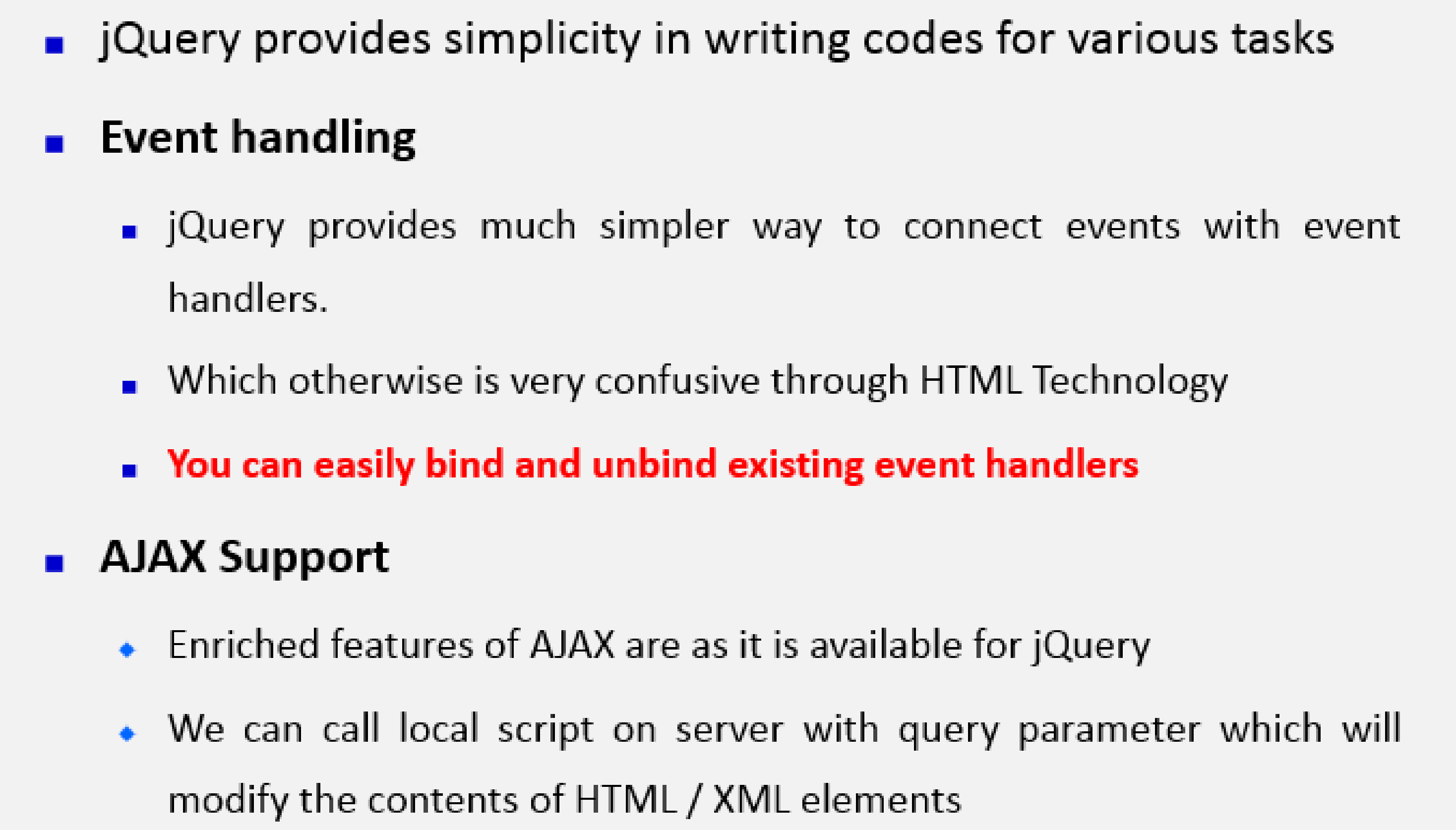
Get me the P element

And change it to this stuff

```
$ (“p”) . html  ( “ Change the Page”) ;
```

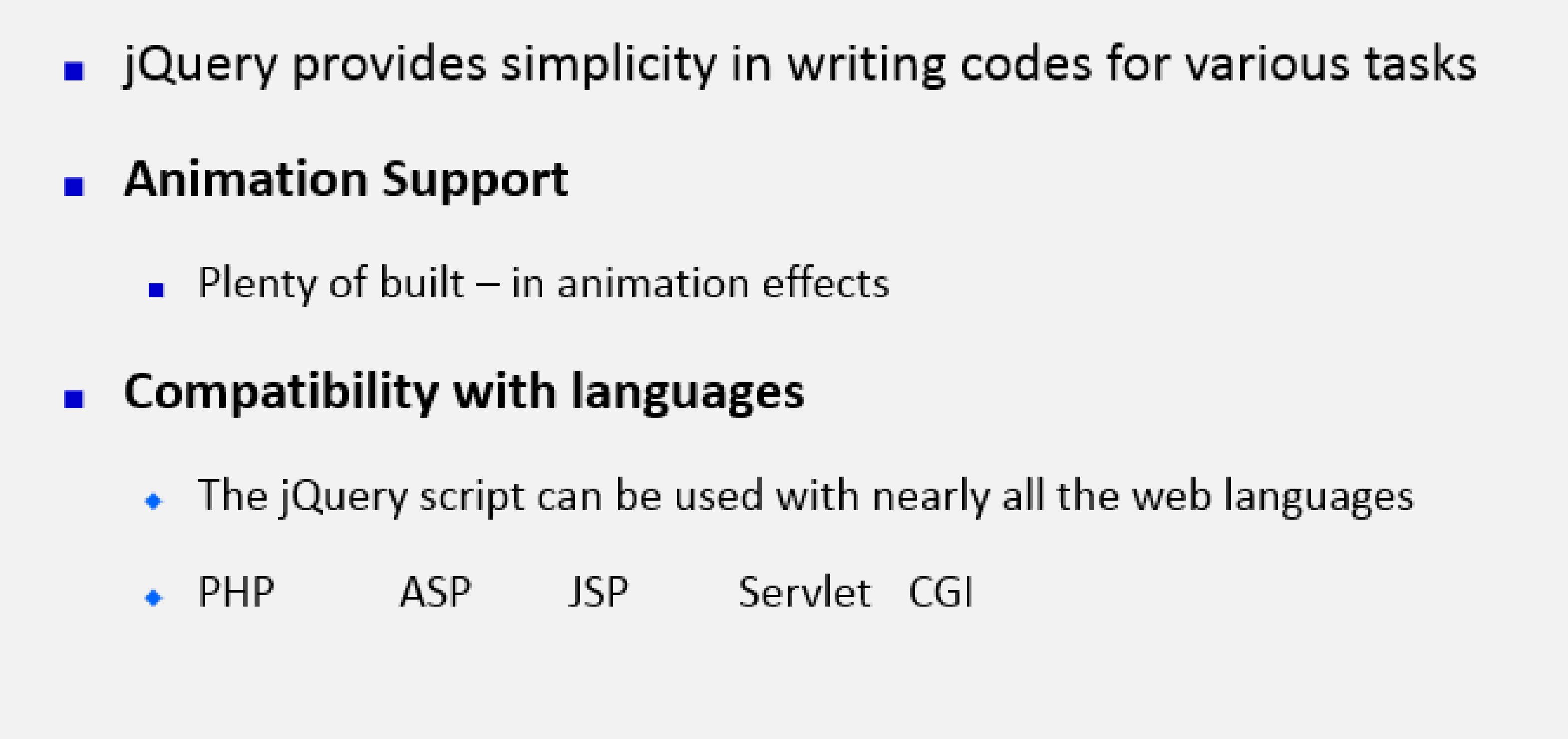
Set the HTML inside that Element

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



- jQuery 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

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

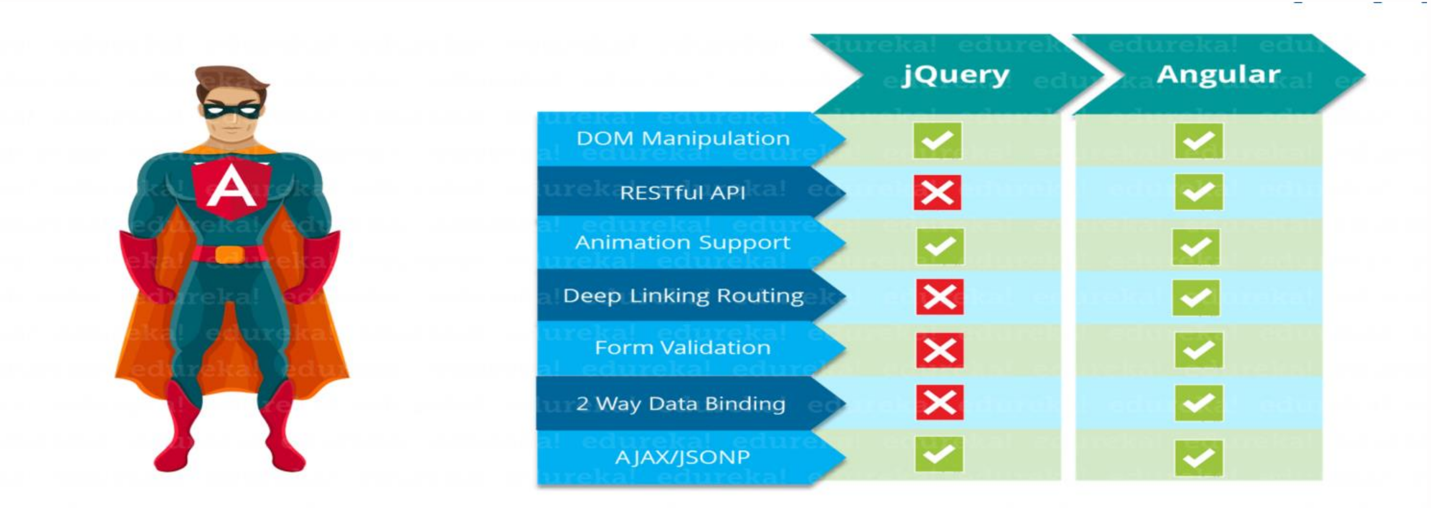
- JSON objects are typed while XML data is typeless
  - JSON types: string, number, array, boolean,
  - XML data are all string

- 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”}**

Angular JS & Angular X

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.



	jQuery	Angular
DOM Manipulation	✓	✓
RESTful API	✗	✓
Animation Support	✓	✓
Deep Linking Routing	✗	✓
Form Validation	✗	✓
2 Way Data Binding	✗	✓
AJAX/JSONP	✓	✓

What is Angular ?

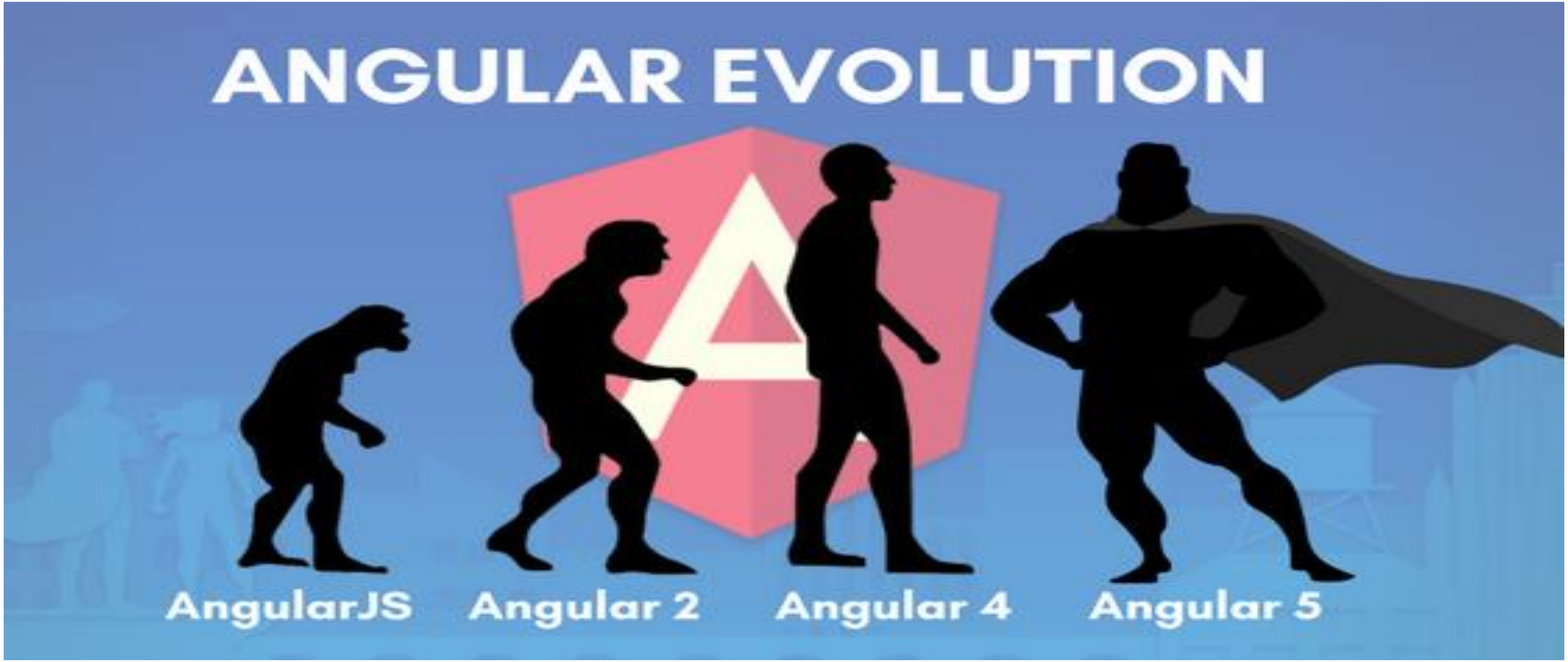
Angular is a great tool that will:

- 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

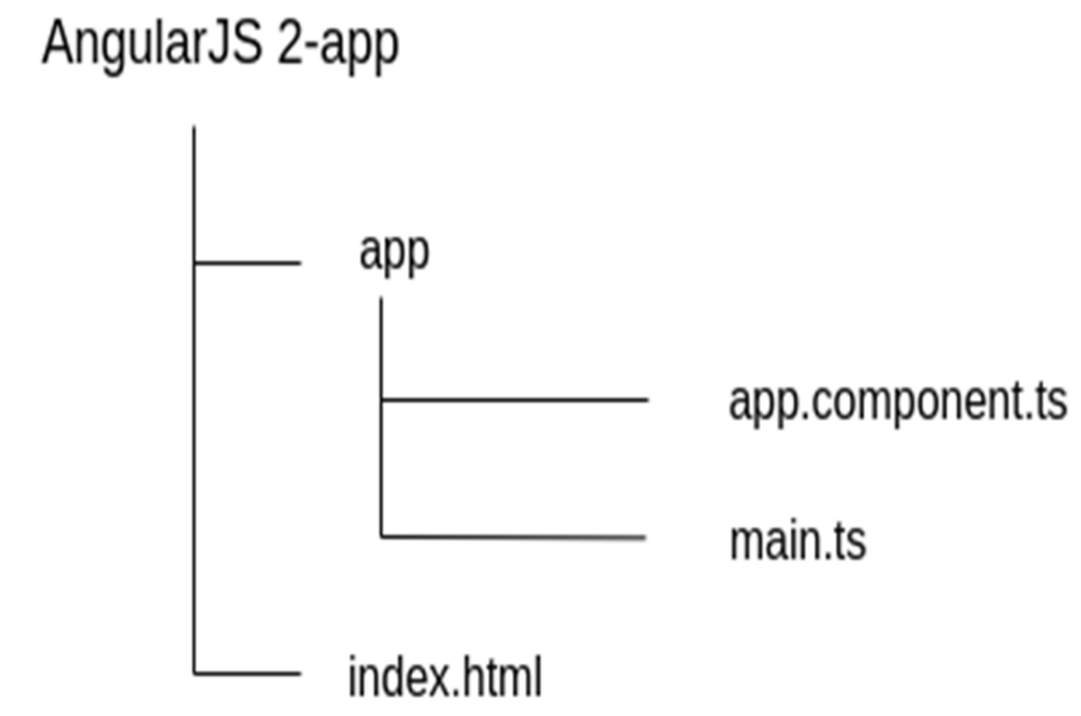
On top of TypeScript features, Angular also includes the benefits taken from ES6:

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

Angular2 File Structure

- *app/app.component.ts* → 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* → is the glue that combines the component and page together.

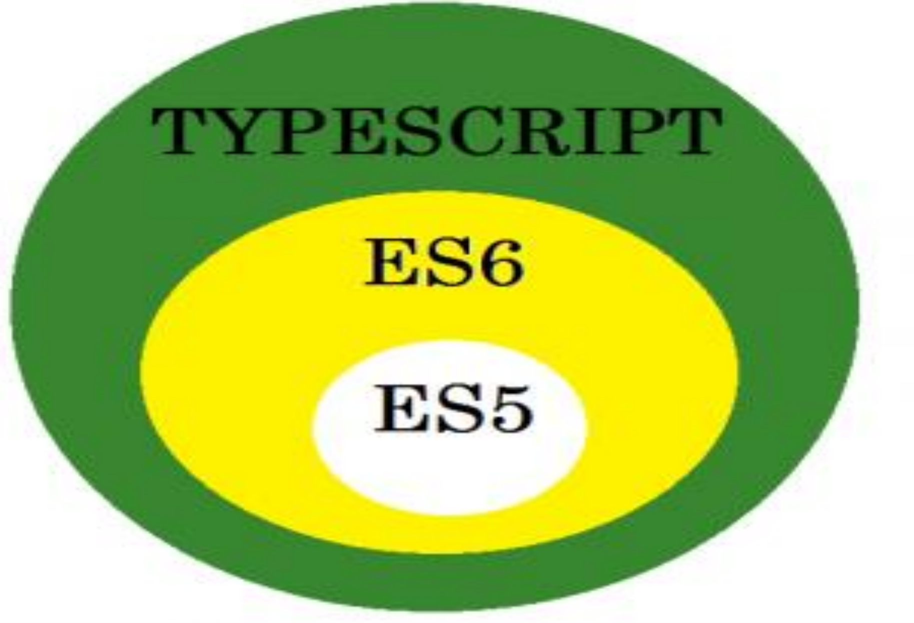
Angular2 File Structure contd..



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 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.

LTI

Let’s Solve