

# Scripting & Computer Environments

## *Web Programming: an Introduction*

IIIT-H

Sep {11,14}, 2013

# Today...

- The Internet
  - The Genesis
  - Architecture
  - Basic Networking Concepts
- The World Wide Web (WWW)
- Basic HTML

# Brainstorm

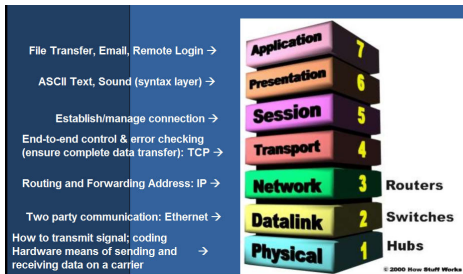
- Internet?
- The web?
- Network protocol? Any protocol you know?

## Internet (the Net)

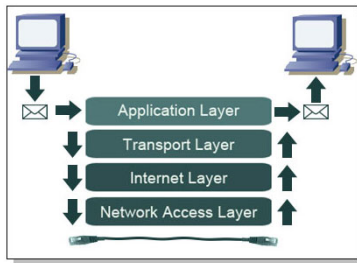
- A global system of interconnected computer networks using the standard Internet protocol suite called **TCP/IP**.
- A network of networks
- The information superhighway (infobahn)
- **I**nternet vs **i**nternet
- The Net vs the Web

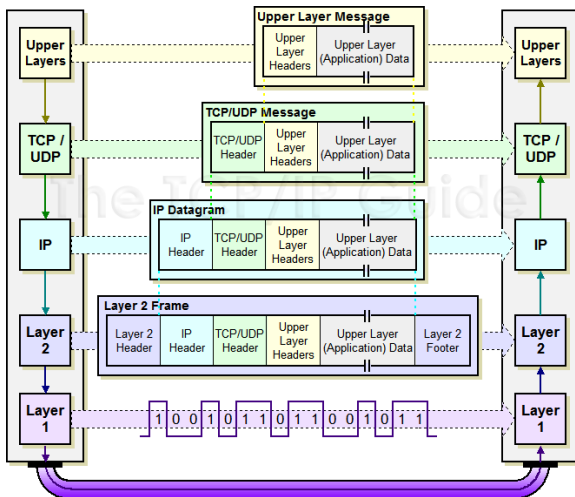
- Began as ARPANET @ US DoD (1960s-70s)
  - Initial killer apps: email, file transfer
  - Initial users: research institutions
- TCP/IP standardized (1982)
- Opened to commercial purposes (late 1980s)
- WWW by Tim Berners-Lee of CERN (network-based implementation of hypertext).
- The web browser revolution (Netscape 1994, IE 1995)
- Search engines (Lycos 1993, WebCrawler 1994, Yahoo! 1994, Altavista 1995, Google 1998 ...)
- The mobile revolution (present and beyond)

## 1. The OSI Model



## 2. The TCP/IP Model





## Protocol

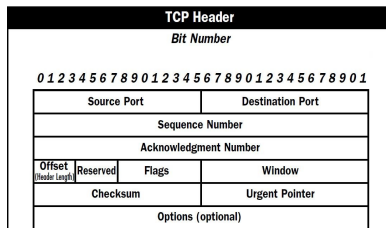
- System of digital message formats + rules for exchanging the messages.
- Describes the syntax, semantics, and synchronization of communication.
- **Internet Protocol (IP)** is the underlying communication protocol on the Internet.

Bits	0	3	4	7	9	15	16	31
Version	Header length		Type of service			Total length		
Identification						Flags	Fragment offset	
Time to live			Protocol			Header checksum		
32-bit source address								
32-bit destination address								
Options							Padding	

- IPv4 vs IPv6



- TCP=Transmission Control Protocol
- *Guaranteed* message delivery on top of IP.
- Provides *reliable, ordered* delivery of a stream of octets.
- TCP vs UDP protocols



- **Port** a number given to each program or service. (Check out the netstat command. )

SSH:22, WWW (HTTP):80, HTTPS:443, FTP:20/21, E-mail:25 ...

# The World Wide Web (WWW)

- The web is ...



- The most popular *service* on the Internet.
- A system of interlinked *hypertext* documents accessed via the Internet.
- **Hypertext** A text with references (hyperlinks) to other text.
- Documents: text, graphics, video, multimedia ...

# The World Wide Web (WWW)

- The web is ...



- The most popular *service* on the Internet.
- A system of interlinked *hypertext* documents accessed via the Internet.
- [Hypertext](#) A text with references (hyperlinks) to other text.
- Documents: text, graphics, video, multimedia ...

- The protocol to exchange or transfer hypertext.
- A request-response protocol in the client-server model.

HTTP request-HTTP reply .

- HTTP resources located by unique identifier: [Uniform Resource Locator \(URL\)](#).

Format:            `protocol host port path_to_document`

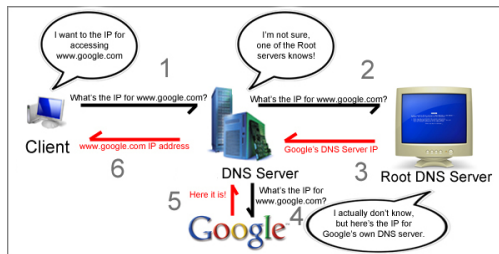
- The client internally sends commands:

```
GET filename
POST filename
PUT filename
HEAD filename      (only status info)
...
```

- Server may reply with HTTP error codes. Common ones:

- 100-101 → Intermediate Status
- 200-206 → Successful Response  
(e.g. 200 = OK )
- 300-307 → Redirects  
(e.g. 302 “Page moved temporarily”)
- 400-417 → Request Errors  
(e.g. 403 “Forbidden”, 404 “page not found” )
- 500-505 → Server Error (e.g. 505 “Internal server error”)

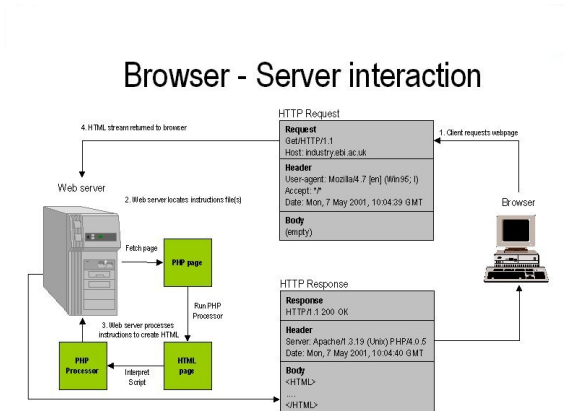
- Resolves names into IP and vice versa.
- Hierarchical distributed system
  - Top-level domains, subdomain, hostname ...



- The `nslookup` command and the `/etc/hosts` local cache file

# Typical Client-Server Interaction Model

## 1. Connection Establishment



## 3. Connection Teardown

- (Extensible) HyperText Markup Language (HTML/XHTML): for writing web pages (i.e. the structure/content).
- Cascading Style Sheets (CSS): defines how HTML elements are to be displayed (i.e. the style).
- Javascript: adds interactivity to HTML pages.
- PHP Hypertext Processor (PHP): dynamically creates web pages on a server.
- eXtensible Markup Language (XML): designed to structure, store and carry data, not display data (unlike HTML).
- Structured Query Language (SQL): for interaction with databases.

**Others:** Ruby, Python, JSP, ASP, AJAX, ColdFusion, XUL, JavaFX, Silverlight, etc



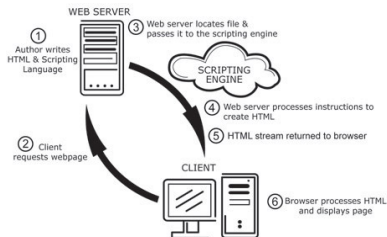
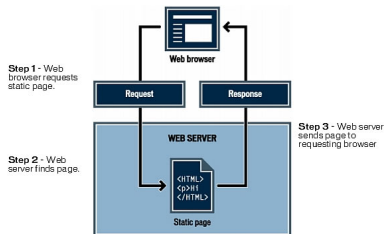
Quiz!

# HTML Basics

Sep 14, 2013

- Static vs Dynamic web pages
- Web 1.0 vs Web 2.0 vs Web 3.0

- Static vs Dynamic web pages



- Web 1.0 vs Web 2.0 vs Web 3.0

## Hyper Text Markup Language (HTML)

- Is a markup language i.e. a set of markup **tags**.
- Describes *content + structure* of web pages.  
(CSS - describes the appearance, Javascript - the behavior)
- Content surrounded with opening and closing tags.
- Each tag's name → **element**.
- Some tags can contain additional information → **attributes**.

```
<element> content </element>
```

```
<element attribute1="value1" attribute2="value2"> content </element>
```

Example:

```
<h1> This is heading </h1>
```

```
<p> This is paragraph </p>
```

```
<!-- This is comment -->
```

```
<!DOCTYPE html>           <!-- defines the document type -->
                           <!--version-dependent (HTML 4.01, HTML5...) -->
```

```
<html>
```

```
<head>
```

The header describes the page (tags: title, style, metadata ...)

```
</head>
```

```
<body>
```

The body contains the actual content of the page

```
</body>
```

```
</html>
```

HTML	1991	
HTML+	1993	
HTML 2.0	1995	
HTML 3.2	1997	
HTML 4.01	1999	
XHTML 1.0	2000	
HTML5	2012	(ongoing)

- HTML vs XHTML
  - Document structure
  - Element syntax
  - Attribute syntax

### ❶ *Block-level Elements:*

- Affect a large amount of content.
- Normally displayed with starting a new line (e.g. `<p>`, `<table>` ).

### ❷ *Inline Elements:*

- A small amount of content.
- Without starting a new line (e.g. `<b>`, `<img>`).

- Some *head* elements:

Page Title: `<title>`

```
<title> This is title of the web page </title>
```

Metadata: `<meta>`

```
<meta name="description" content="SACE Lecture Notes IIIT-H" />
<meta content="text/html; charset=utf-8" />
<meta http-equiv="refresh" content="20" />
```



### ❶ *Block-level Elements:*

- Affect a large amount of content.
- Normally displayed with starting a new line (e.g. `<p>`, `<table>` ).

### ❷ *Inline Elements:*

- A small amount of content.
- Without starting a new line (e.g. `<b>`, `<img>`).

- Some *head* elements:

Page Title: `<title>`

```
<title> This is title of the web page </title>
```

Metadata: `<meta>`

```
<meta name="description" content="SACE Lecture Notes IIIT-H" />
<meta content="text/html; charset=utf-8" />
<meta http-equiv="refresh" content="20" />
```

Headings :      `<h1>` to `<h6>`

`<h1>` This is heading 1 `</h1>`

`<h6>` This is heading 6 `</h6>`

Paragraph :      `<p>`

`<p>` This is paragraph 1 `</p>`

`<p>` This is paragraph 2 `</p>`

Line Break:      `<br/>`

Forces a new line.

Horizontal Rule:      `<hr/>`

A horizontal line to visually separate sections of a page.

Headings :      `<h1>` to `<h6>`

`<h1>` This is heading 1 `</h1>`

`<h6>` This is heading 6 `</h6>`

Paragraph :      `<p>`

`<p>` This is paragraph 1 `</p>`

`<p>` This is paragraph 2 `</p>`

Line Break:      `<br/>`

Forces a new line.

Horizontal Rule:      `<hr/>`

A horizontal line to visually separate sections of a page.

Headings :      `<h1>` to `<h6>`

```
<h1> This is heading 1 </h1>
```

```
<h6> This is heading 6 </h6>
```

Paragraph :      `<p>`

```
<p> This is paragraph 1 </p>
```

```
<p> This is paragraph 2 </p>
```

Line Break:      `<br/>`

Forces a new line.

Horizontal Rule:      `<hr/>`

A horizontal line to visually separate sections of a page.

Headings :      `<h1>` to `<h6>`

```
<h1> This is heading 1 </h1>
```

```
<h6> This is heading 6 </h6>
```

Paragraph :      `<p>`

```
<p> This is paragraph 1 </p>
```

```
<p> This is paragraph 2 </p>
```

Line Break:      `<br/>`

Forces a new line.

Horizontal Rule:      `<hr/>`

A horizontal line to visually separate sections of a page.

## Emphasis Elements : `<em>`, `<strong>`

- `<em>` for emphasis, `<strong>` for strong emphasis.
- Also `<b>` for bold, `<i>` for italics.

## Lists: `<ul>`, `<ol>`

- Unordered lists (`<ul>`) for a bulleted list of items.
- Ordered lists (`<ol>`) for numbered list of items.
- Each item as `<li>item </li>`.
- Can be nested.

## Emphasis Elements : `<em>`, `<strong>`

- `<em>` for emphasis, `<strong>` for strong emphasis.
- Also `<b>` for bold, `<i>` for italics.

## Lists: `<ul>`, `<ol>`

- Unordered lists (`<ul>`) for a bulleted list of items.
- Ordered lists (`<ol>`) for numbered list of items.
- Each item as `<li>item </li>`.
- Can be nested.

## Definition List : `<dl>`, `<dt>`, `<dd>`

- `<dl>` for list of definition of terms
- `<dt>` represents each term; `<dd>` for the definition.

## Quotations: `<blockquote>`, `<q>`

- `<blockquote>` for lengthy quotations.
- `<q>` for inline quotes.

## Code Environments: `<code>`, `<pre>`

- Use `<pre>` for verbatim text.



## Definition List : `<dl>`, `<dt>`, `<dd>`

- `<dl>` for list of definition of terms
- `<dt>` represents each term; `<dd>` for the definition.

## Quotations: `<blockquote>`, `<q>`

- `<blockquote>` for lengthy quotations.
- `<q>` for inline quotes.

## Code Environments: `<code>`, `<pre>`

- Use `<pre>` for verbatim text.

## Definition List : `<dl>`, `<dt>`, `<dd>`

- `<dl>` for list of definition of terms
- `<dt>` represents each term; `<dd>` for the definition.

## Quotations: `<blockquote>`, `<q>`

- `<blockquote>` for lengthy quotations.
- `<q>` for inline quotes.

## Code Environments: `<code>`, `<pre>`

- Use `<pre>` for verbatim text.

## Hyperlinks (links) : `<a>`

- Links (anchors) to other pages.
- The *href* attribute specifies the destination URL.

## Images : `<img>`

- Inserts a graphical object into the page.
- Attributes: `src`, `alt`, `height`, `width`
- Can be links too.

```

```

## Hyperlinks (links) : `<a>`

- Links (anchors) to other pages.
- The *href* attribute specifies the destination URL.

## Images : `<img>`

- Inserts a graphical object into the page.
- Attributes: `src`, `alt`, `height`, `width`
- Can be links too.

```

```

## Tables

- The `<table>` element creates tables.
- The `<tr>`, `<td>` and `<th>` elements define table rows, cells (data) and headings respectively.

- Some `<td>/<th>` attributes:

- |                        |              |                        |                  |
|------------------------|--------------|------------------------|------------------|
| • <code>align</code>   | (deprecated) | • <code>width</code>   | (in pixels or %) |
| • <code>bgcolor</code> | (deprecated) | • <code>rowspan</code> |                  |
| • <code>height</code>  | (deprecated) | • <code>colspan</code> |                  |

- Some `<table>` attributes (deprecated):

- `border` (in pixels)
- `align` values: {left, right, center, justify}
- `cellspacing="value"` (in pixels or %)
- `cellpadding="value"` (in pixels or %)
- `bgcolor="color_value"` (in hex codes #rrggbb or names)
- `summary="description"` for non-visual browsers

## Form

Group of UI controls that collect information from the user and send the information for processing.

The screenshot shows the Bharat Matrimony registration page. It features a header with navigation links like 'Register', 'Search', 'Upgrade', and 'Help'. Below the header, there are tabs for 'Online Matrimony', 'Private Matrimony', 'Elite Matrimony', 'Matrimony Details', and 'Mobile App'. The main content area is titled 'Matrimony Profile for' and includes a 'Free Membership' and 'Paid Membership' section. The profile form contains fields for Name, Age (with a range selector), Gender, Religion, Marital Status, Caste/Community, Country/Region, Mobile No., and E-mail. There is also a 'Login/Password' section. A 'Register' button is at the bottom right. A banner at the bottom says 'Find your life partner from over 20 million profiles'.

The screenshot shows the Facebook App Center sign-up page. It has a blue header with the Facebook logo and navigation links. The main content area is titled 'App Center' and features logos for Spotify and Google Play. The sign-up form includes fields for First Name, Last Name, Your Email, New Email, and New Password. There is a 'Select Date' dropdown for birth date and a 'Sign Up' button. A checkbox for 'I am' is also present. A banner at the bottom says 'Find great social music apps'.

The screenshot shows the Google India search page. It features the Google logo and a search bar. Below the search bar, there are links for 'Google Search' and 'The Floating Lady'.

The UI controls: inputbox, button, checkbox, radiobox, drop-down menu, file select box ...

- The `<form>` element creates forms.

```
<form action="URL">  
  form controls  
</form>
```

- The `action` attribute specifies the page that will process the submitted data.
- Other attributes:
  - `method` values: {get, post}
  - `id` (unique ID for the `<form>` element)
  - `name` (deprecated)
  - `onsubmit=event`
  - `onreset=event`
  - `accept`



# (X)HTML Form Controls

- The `<input>` element creates many of the UIs.
- Must be self-closed.

```
<input attribute="value" />
```

- Attributes:
  - `type` of input control
  - `name` name of the control passed to server.
  - `value` caption or initial value
  - `size` width of the control.
- Some `type` values: {text, button, radio, checkbox, submit, file, reset, password ...}

- 3 text input boxes:
  - 1 Single-line text input
  - 2 Password input
  - 3 Multi-line text input (textbox) → the `<textarea>` element

```
Username <input type="text" name="uname" value="" size="30" maxlength="30"/>
```

```
Password <input type="password" name="pw" value="" size="30" />
```

```
<textarea name="comment" rows="5" cols="10"> Your Comments </textarea>
```

- To submit or reset forms, to trigger events/scripts.
- Creating buttons:
  - a. Using the `<input>` element
  - b. Using the `<button>` element
  - c. Images as buttons.
- Some attributes: {`type`, `name`, `value`, `size`, `onclick`}
- `type` values: {`submit`, `reset`, `image`, `button`}

- **Checkbox** → created using the `checkbox` type attribute of `<input>` (`radio` for radiobox).
- When sent to server, any checked boxes will be sent with `value=on`
- **Radiobox** → “binary checkboxes”.
- Are grouped by `name` attribute.
- Value be specified for each.
- Related attributes: `{type, name, value, size, checked}`

- To select an item(s) from a list; & as an alternative to other controls.
- Created using the `<select>` and `<option>` elements.

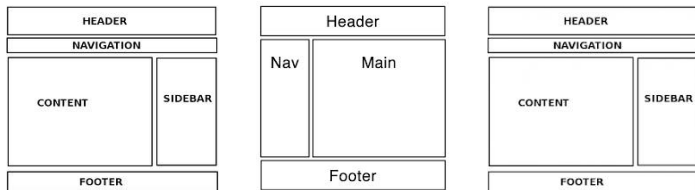
```
<select name="state">  
<option value="tl">Telangana</option>  
<option value="gj">Gujarat</option>  
<option value="kr">Kerala</option>  
<option value="tn">Tamil Nadu</option>  
</select>
```

- Attributes of `<select>`: {name, size, multiple}
- Attributes of `<option>`: {value, label, selected}
- The `<optgroup>` element is used to group list items.

- A.k.a *file upload box*
- Requirement → method attribute of `<form>` be `post`!
- Created using the `<file>` type attribute of the `<input>` element.

```
<input type="file" name="SomeName" />
```

- Some attributes: {name, value, accept}



- Some layout techniques:

- 1 Using Tables

- 2 Using `<div>`

- `<div>` is a block level element used for grouping (X)HTML elements.

- 3 Using Frames

- 4 Using CSS (later on).

- Frames divide our window into *panes*.
- Frameset: collection of frames.
- Advantages → each can be (re)loaded independently.
- Using the `<frameset>` and `<frame>` elements.
- The `<frame>` element specifies what goes in each frame.
- The `<frameset>` element:
  - Defines rows and columns (specified in pixels, percentage or the \* wildcard).
  - Replaces the `<body>` element.



- Multimedia = sound, music, videos, and animations ...
- Most browsers support various multimedia formats.
- Audio Formats:
  - MIDI (.mid, .midi)
  - Wave (.wav) - uncompressed
  - MP3 (.mp3, .mpga) - compressed
  - Real Audio (.rm, .ram) - audio-video format from Real Media.
- Video Formats:
  - AVI (Audio Video Interleave) - Microsoft
  - WMV (Windows Media)
  - Flash (.flv, .swf)
  - MPEG (.mpeg, .mpg)
  - MPEG-4 (.mp4)
  - Quicktime (.mov) - Apple

- Adding multimedia to your page:
  - Using Plug-ins
  - The `<embed>` and `<object>` elements
  - Using hyperlinks
  - Using external players

...