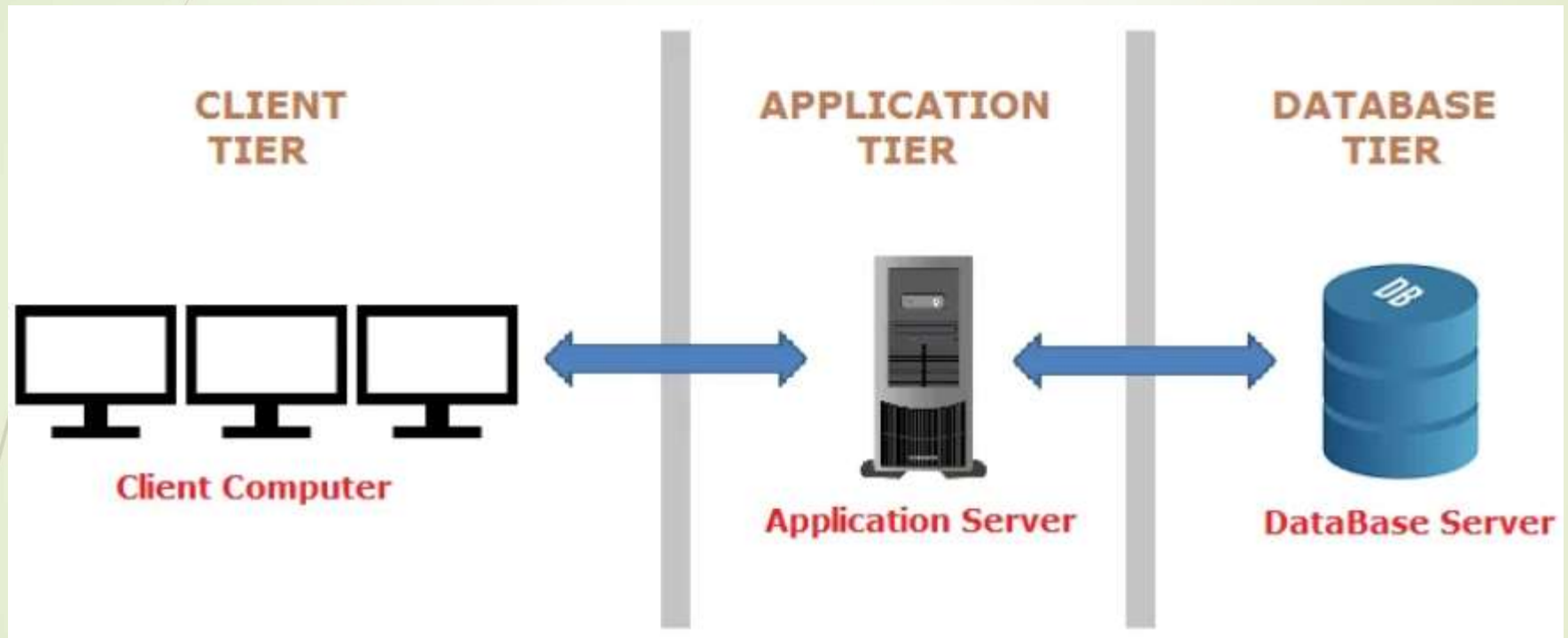


# Introduction to Web Design and Programming

# 3-Tier Application Architecture



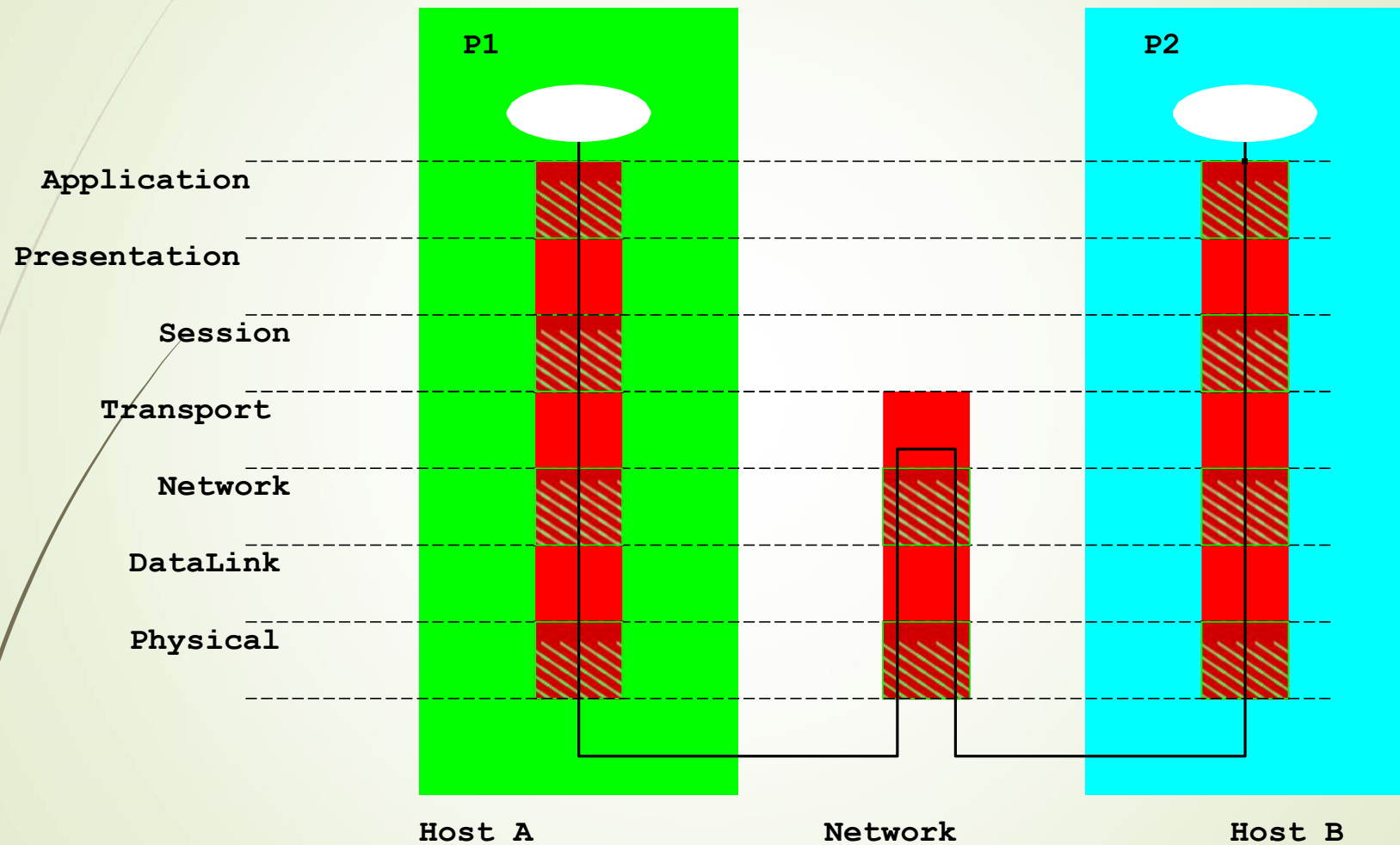


# Overview of the Web

## Web Is Part of the Internet

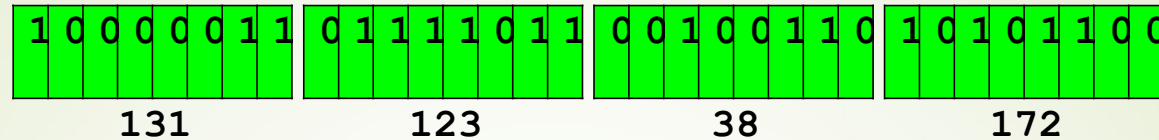
- The Web is one among many services on the Internet– email, file transfer, remote login, audio and video streaming, and many more.
- The Internet connects computer networks all over the globe including LANs and WANs. The Internet enables each connected computer, called a *host*, to communicate with any other hosts.
- In addition to host computers, the network infrastructure itself involves dedicated computers that perform network functions: hubs, switches, bridges, routers, and gateways.
- For programs and computers from different vendors, under different operating systems, to communicate on a network, a detailed set of rules and conventions must be established for all parties to follow. Such rules are known as *networking protocols*.

# Networking Layers



# IP Address

Web  
Overview-6



- On the Internet, each host has a unique *IP address* For example, tiger, a host at Kent State, has the IP address 131.123.38.172.
- This *dot notation* (or *quad notation*) gives the decimal value (0 to 255) of each byte.
- Similar to a telephone number, the leading digits of an IP address are like country codes and area codes while the trailing digits are like local numbers.
- A host may be configured with a fixed IP or may obtain an IP address assigned to it on-the-fly as it boots up and attempts to connect to the network initially.

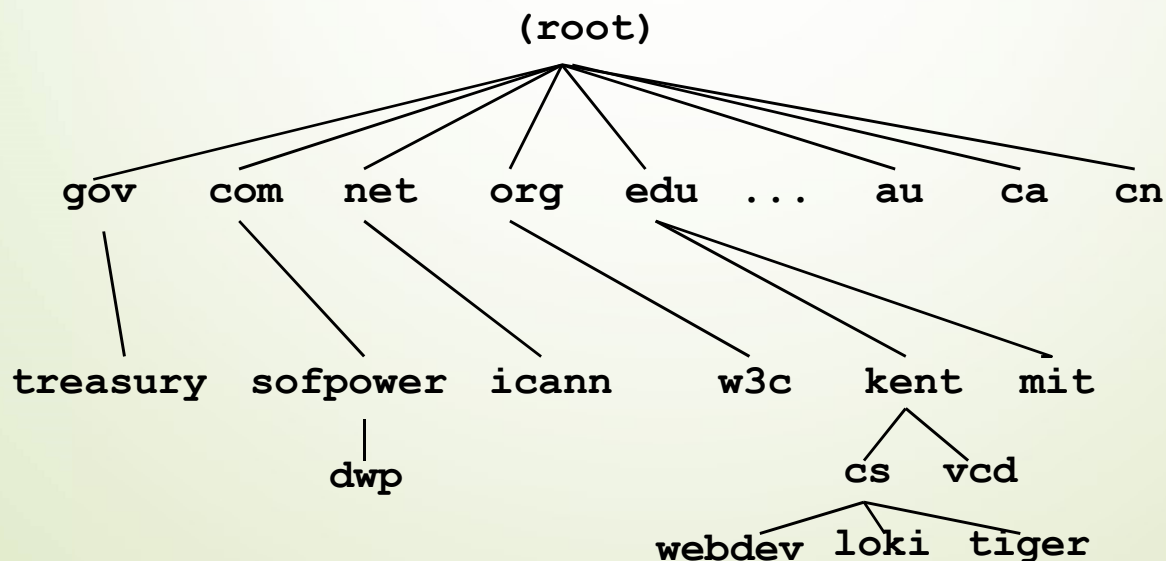
## Tracert command

# Domain Name System

Web Overview-7

- The IP address is easy on machines but hard on users. Therefore, each host also has a unique *domain-based name* composed of words, rather like a postal address. For example, the domain name for `tiger` is `tiger.cs.kent.edu` (at Department of Computer Science, Kent State University).
- With the domain names, the entire Internet host namespace is recursively divided into disjoint domains. The address for `tiger` puts it in the `kent` subdomain within `edu`, the *generic top-level domain* (generic TLD) for educational institutions.
- Other generic TLDs include `org` (nonprofit organizations), `gov` (government offices), `mil` (military installations), `com` (commercial outfits), `net` (network service providers), `info` (information sources), `name` (individuals), `uk`, (United Kingdom), `cn` (China), and so forth.

## The Domain Name Hierarchy



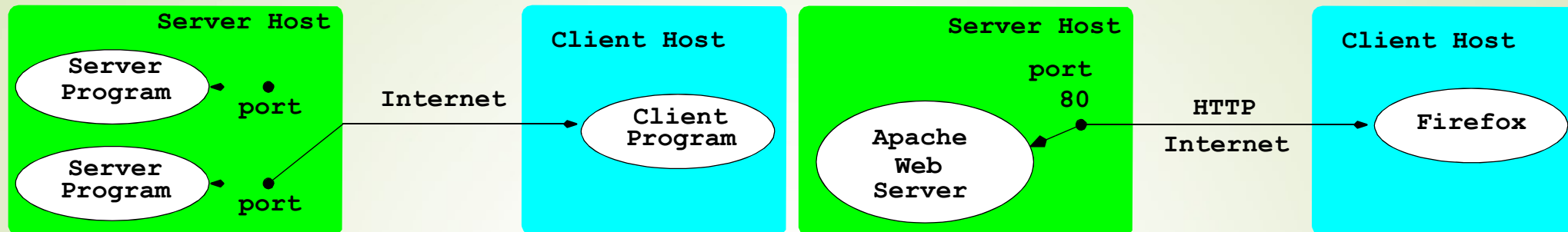
## Domain to IP



nslookup command



## Client and Server



- A *server* program provides a specific service on a host (the server host) that offers such a service. Example services are email (SMTP), secure remote login/file transfer (SSH/SFTP), and the World Wide Web (HTTP).
- Each *standard Internet service* has its own unique *port number* that is identical across all server hosts. The port number together with the IP address of a host identifies a particular server program. For example, SMTP uses port number 25, SSH/SFTP 22, and HTTP 80.
- A *client* program on a computer connects with a server on another host to obtain its service. Thus, a client program is the agent through which a particular network service can be obtained. Different agents are usually required for different services. Example clients are Mozilla Thunderbird® and Microsoft® Outlook® for email, OpenSSH for secure login/file transfer.

# URLs

- The Web uses *Uniform Resource Locators* (URLs) to identify (locate) many kinds of resources (files and services) available on the Internet.
- A URL may identify a host, a server port, and the target file stored on that host. URLs are used, for example, by browsers to retrieve information, and in HTML documents to link to other resources.

- A full URL usually has the form

*scheme://server:port/pathname?query\_string*

E.g. <https://www.google.com/search?q=Government+of+Korea>

- URLs are contained in webpages to create links to image, audio, video, and any other type of files anywhere on the Web

## URL Schemes

The *scheme* part indicates the information service type and therefore the protocol to use. Common schemes include

- http (Web service)
- ftp (file transfer service)
- mailto (email service)
- callto (Skype<sup>TM</sup> or similar service)
- file (local file system)
- https (secure Web service)
- sftp (secure file transfer service)

## Handling Different Content Types

- The Web uses the MIME defined content types.
- Many content types are in use. Popular types have standard file extensions.
- When a Web server returns a document to a browser, the content type is indicated via the HTTP message header `Content-Type`. For example, an HTML document is delivered with the content type `text/html`.
- The content type information allows browsers to decide how to process the incoming content. HTML, text, images, audio, and video can be handled by the browser directly.
- Other types such as PDF and Flash are usually handled by plug-ins or external helper programs.

## Content Types and File Suffixes

text/plain	txt sh c ...	text/html	html htm
application/pdf	pdf	application/msword	doc, docx
image/jpeg	jpeg jpg jpe	audio/basic	au snd
audio/mpeg	mpga mp2 mp3	application/x-gzip	gz tgz
application/zip	zip	audio/ogg	oga, ogg, spx
video/ogg	ogv	video/webm	webm

## HTML and HTML5

- HTML (the Hypertext Markup Language) is used to structure webpage contents.
- Since 1989, HTML has been constantly evolving and maturing. Today, the Web is moving toward HTML5, the next-generation HTML standard, which brings many new features and APIs (application programming interfaces).
- A document written in HTML contains ordinary text interspersed with *markup tags* and uses the `.html` filename extension.

## Some HTML Tags

Entire page	<code>&lt;html&gt;...&lt;/html&gt;</code>
Paragraph	<code>&lt;p&gt;...&lt;/p&gt;</code>
Meta data	<code>&lt;head&gt;...&lt;/head&gt;</code>
Unnumbered list	<code>&lt;ul&gt;...&lt;/ul&gt;</code>
Page title	<code>&lt;title&gt;...&lt;/title&gt;</code>
Numbered list	<code>&lt;ol&gt;...&lt;/ol&gt;</code>
Page content	<code>&lt;body&gt;...&lt;/body&gt;</code>
List item	<code>&lt;li&gt;...&lt;/li&gt;</code>
Level $n$ heading	<code>&lt;h<math>n</math>&gt;...&lt;/h<math>n</math>&gt;</code>
Comment	<code>&lt;!--...--&gt;</code>

## Ex: Sports

```
<!doctype html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<meta charset="UTF-8"/>
<head> <title>A Basic Webpage</title> </head>
<body><section>
  <h1>Big on Sports</h1>
  <p>Sports are fun and good for you ...</p>
  <p> What is your favorite sport? ... And here is a short list:
</p>
  <ol>
    <li> Baseball </li> <li> Basketball </li>
    <li> Tennis </li> <li> Soccer </li>
  </ol></section></body></html>
```



## A Sample Webpage



## Webpage Styling

- HTML provides page structure while *styling rules* controls page presentation.
- Styling is coded in *Cascading style sheets* (CSS) rules and attached to different parts of a webpage.
- Style rules are usually placed in separate files from the webpage. Isolating page styling from page structure makes it easy for Web designers to reuse styling rules in different pages and to enforce consistent visual styling over an entire website.
- For example, if we want to make all level-one headers dark blue, we can use this CSS rule:

```
h1 { color: darkblue }
```

## Web Hosting

- Publishing on the Web involves
  1. Designing and constructing the pages and writing the programs for a website
  2. Placing the completed site with a hosting service which runs web server programs to deliver your site to Web users.
- Each hosting account provides an amount of disk space, a monthly network traffic allowance, email accounts, Web-based site management and maintenance tools, and other access such as FTP and SSH/SFTP.
- To host a site under a given domain name, a hosting service associates that domain name to an IP number assigned to the hosted site.

## Dynamic Generation of Webpages

- Webpages are usually prepared and set in advance to supply some predetermined content. These fixed pages are *static*.
- A Web server can also deliver *dynamic pages* that are generated on-the-fly by programming on the server side.
- A dynamic page is produced when a request arrives often with input data.

## Use of Dynamic Webpages

- Managing user login and controlling interactive sessions
- Customizing a document depending on when, from where, by whom, and with what program it is retrieved
- Collecting user input (with HTML forms), processing such input data, and providing responses to the incoming information
- Retrieving and updating information in databases from the Web
- Directing incoming requests to appropriate pages; redirection to mobile sites is an example
- Enforcing certain policies for outgoing documents

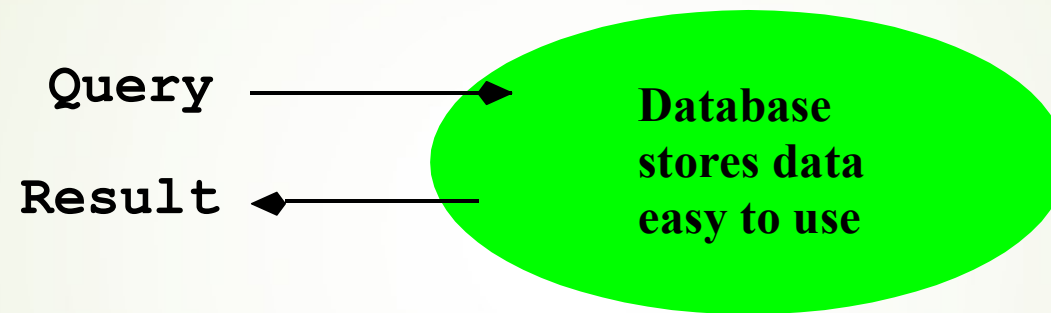
## Database Access

- A database is an efficiently organized collection of data for a specific purpose.
- A relational database is one that uses tables to organize and retrieve data.
- The Web can make databases accessible on the Internet.
- Databases can be used to manage and run a website.

## A Database Table

Last	First	Dept	Email
Wang	Paul	CS	<u><a href="mailto:pwang@kent.edu">pwang@kent.edu</a></u>

## Database Function





## Client-Side Scripting

- The actions of a Web browser can be defined and controlled by programming within a webpage. Such programming can supply customized user experiences and make webpages more responsive and useful for end users.
- The programs execute within the browser, which runs on the client host, the computer used to access the Web.
- For all major browsers, *JavaScript* is the standardized scripting language for client-side programming.
- Because the JavaScript language standard has been developed and maintained by the *ECMA* (European Computer Manufacturer Association), the language is also known as *ECMAScript* ([ecma-international.org](http://ecma-international.org)).

## Document Object Model

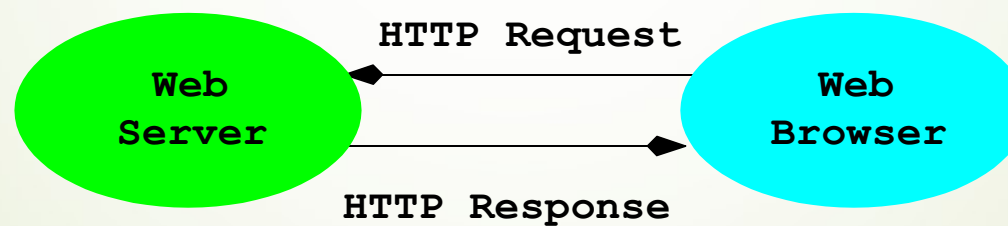
- The *Document Object Model* (DOM) is an application programming interface (API) definition specifying how a document processing program such as a Web browser represents a document as a tree-structured object (the DOM tree) and allows programmatic access and manipulation of the DOM tree.
- All major browsers provide DOM support, including HTML, XHTML, HTML5, and XML DOM interfaces. Because JavaScript is the standard, it is not surprising that major browsers all implement the required DOM interfaces in JavaScript.
- Combine JavaScript with DOM and CSS and you have a technique, known as *DHTML*, to make webpages responsive and dynamic.

# Hypertext Transfer Protocol

- HTTP is an application layer protocol that sits on top of TCP/IP, which provides reliable two-way connection between the Web client and Web server.
- HTTP/1.0 was standardized in the first part of 1996.
- HTTP/1.1 introduced important improvements and new features and is now the stable version.
- HTTPS (HTTP Secure) is a secure protocol that simply applies HTTP over a secure transport layer protocol *Transport Layer Security* (TLS).

## HTTP Transaction

1. *Connection*—A browser (client) opens a connection to a server.
2. *Query*—The client requests a resource controlled by the server.
3. *Processing*—The server receives and processes the request.
4. *Response*—The server sends the requested resource back to the client.
5. *Termination*—The transaction is finished, and the connection is closed unless it is kept open for another request immediately from the client on the other end of the connection.



## Web Development Principles

- Involve the principals of the website in every step of the planning and development process. Often, after learning what the Web and Internet can do, site owners will have important ideas on what the site can do for their businesses. It goes without saying that most of the site content must be supplied and composed by the principals.
- Make clear on the site entry page, what the website is and what audiences it is intended to serve.
- Know your audiences and anticipate what information they expect to find on your site. Design your site navigation accordingly and provide multiple links from different contexts to important parts of your site.
- Form follows function. Your site architecture, art and visual design, and page layout must enhance its functions and be *user œntric*. It is all about making it easy and attractive for your users/customers.

- Avoid extraneous audio, video, blinking text, and other technology show-off features, if they do not serve any good purposes.
- Avoid small fonts and dark backgrounds. Information on the site must be easy to read and understand.
- Make your site fast to load and responsive to user actions.
- Use fluid layout that responds well to varying screen sizes and resolutions.
- The site should be easy to maintain. Information that needs changing/updating must be easily editable by key business personnel within the organization that owns the site.
- Use open standards and test your site under all major browsers.

## Web Development Process

- Content: Create a content list or inventory; prepare content files ready to be included in webpages.
- Site map: Draw a relationship diagram of all pages to be created for the site, give each page appropriate titles, show page groupings and on-site and off-site links, distinguish static from dynamic pages, identify forms and server-side support. Major subsections of the site can have their own submaps.
- Skeletal site: Conceive an entry page, home page, typical subpages and sub-subpages, textual contents (can be in summary form), HTML forms with textual layout and descriptions of server-side support, structure of the file hierarchy for the site, and well-defined HTML coding standards for pages.

- Navigation: Follow the site architecture and site map to link the pages, use textual navigation links with rough placements (top, left, right, or bottom), avoid dead-end pages, and avoid confusing the user.
- Design concepts: features, characteristics, and look-and-feel of the site; the design must reflect client identity and site purpose.
- Storyboards: simple layout sketches based on content-only site for typical pages, HTML forms, and HTML form response pages; header, footer, margins, navigation bar, logo, and other graphical elements to support the delivery of content; client feedback and approval of storyboards.
- Page layout (for pages at all levels): content hierarchy and grouping; grids, alignments, constants, and variables on the page; placement and size of charts, graphs, illustrations, and photos; creative use of space and variations of font, grid,



and color; style options and variations.

- Home page/entry page: visuals to support the unique function and purpose of entry to the site and home page as required by site architecture.
- Page templates: Create templates for typical pages at all levels. Templates are skeleton files used to make finished pages by inserting text, graphics, and other content at marked places in the templates. In other words, a template is a page frame with the desired design, layout, and graphics ready to receive text, links, photos, and other content. A template page may provide HTML, style sheets, JavaScript, head, body, meta, link, and script tags, as well as marked places for page content. Templates enable everyone on the project team to complete pages for the site. Advanced templates may involve dynamic server-side features.

- Prototype pages: Use the templates to complete typical pages in prototype form, test and examine page prototypes, present prototype pages to the client, and obtain feedback and approval. Make sure that the layout system has been designed with enough versatility and flexibility to accommodate potential changes in content.
- Client-side programming: Write scripts for browsers and possibly other Web clients that will be delivered together with webpages to the client side. These scripts may include style sheets and JavaScripts. Client-side programs can make webpages more interactive and responsive.
- Server-side programming: Write programs for form processing, dynamic page generation, database access, e-business, and e-commerce features. Make sure these follow the site architecture, user orientation, and visual design.

# Questions?