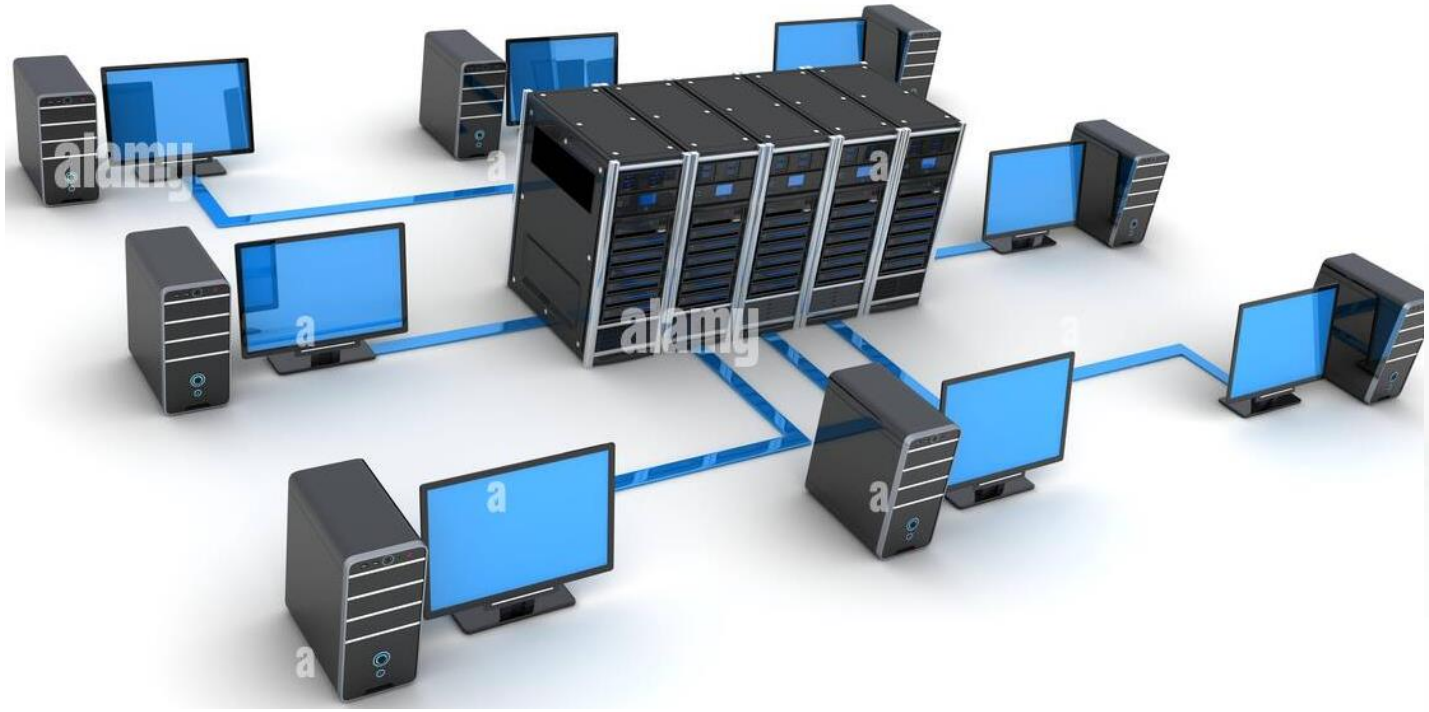


Server machine Vs Desktop machine

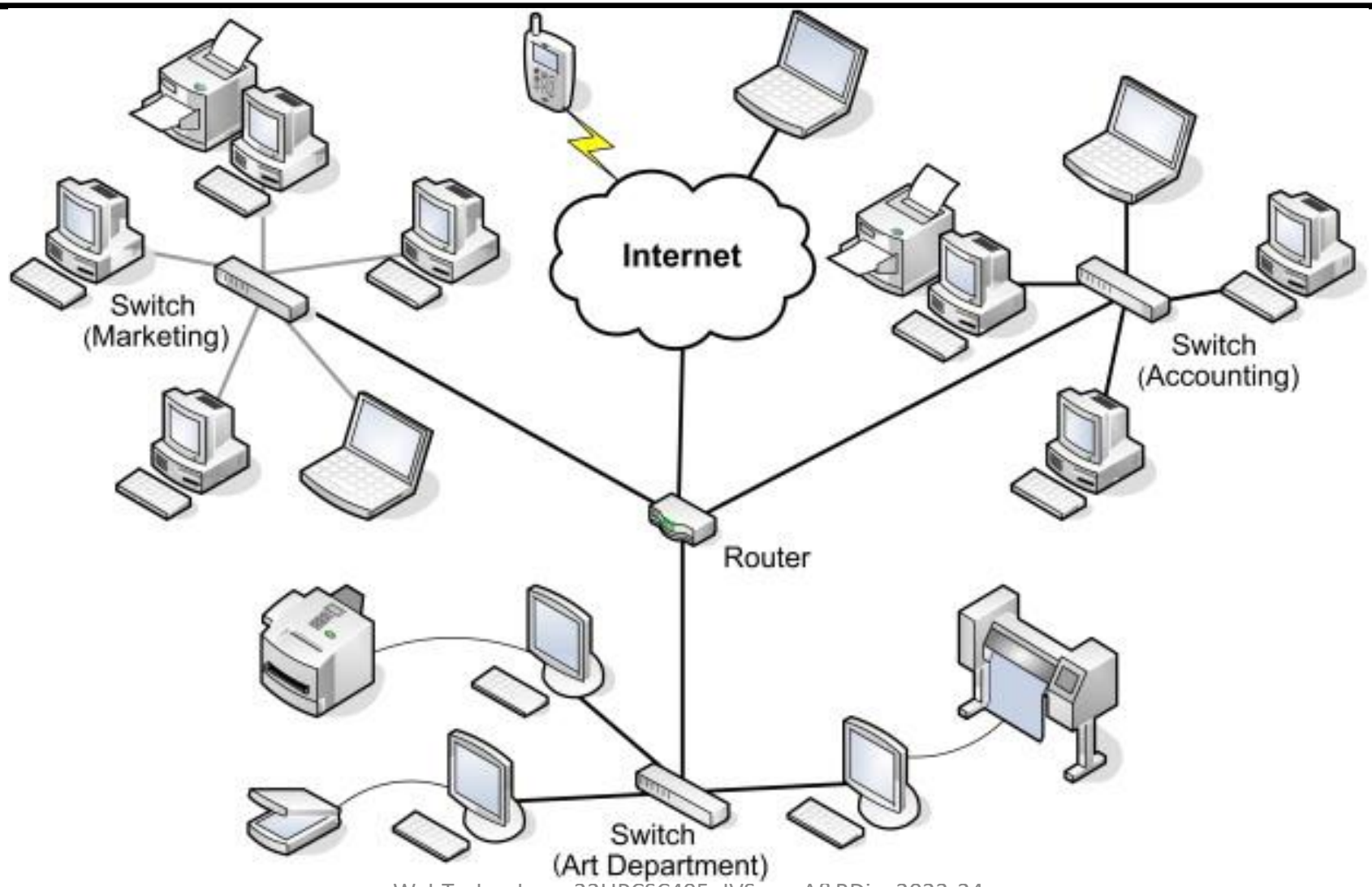
Machine Types - Servers & Desktop

Feature	Server	Desktop
Purpose	Provide services and resources to other devices on a network	Run individual user applications and tasks
Number of Users	Multiple users simultaneously	Single user
Hardware	More powerful processors, larger memory capacity, redundant components for reliability	Less powerful processors, moderate memory capacity, fewer redundant components
Software	Server operating systems (e.g., Windows Server, Linux)	Desktop operating systems (e.g., Windows, macOS)
Uptime	Designed for continuous operation (24/7)	Designed for occasional or regular use, not continuous operation
Security	High focus on security features and data protection	Lower focus on security compared to servers
Cost	Generally more expensive due to powerful hardware and features	Generally less expensive than servers
User Interface	Often accessed remotely through command line or management tools	Graphical user interface (GUI) for user interaction
Examples	File servers, web servers, email servers, database servers	Personal computers, workstations, gaming PCs

Application Types - Standalone Vs Web applications

Aspect	Standalone Applications	Web Applications
Deployment	Installed locally on a device.	Accessed via a web browser, hosted on a remote server.
User Interface	Often has a native look and feel.	Rely on web technologies, accessed through a browser.
Updates and Maintenance	Manual updates by users or administrators.	Centralized updates managed on the server.
Platform Independence	Platform-specific (Windows, macOS, Linux).	Platform-independent, accessible from any browser.
Resource Utilization	Direct access to local resources for optimal performance.	Relies on device capabilities, subject to network latency.
Security Model	Operates within the device's security boundaries.	Rely on server and browser security measures.
Example	Microsoft Word, Adobe Photoshop, AutoCAD	Gmail, Google Docs, or any web application served over the internet.

Network and Internet



Unit-1

Introduction to Web

Content

- ✓WWW1.0
- ✓HTML
- ✓HTML5
- ✓XHTML
- ✓XML
- ✓XSD
- ✓DTD
- ✓DOM-XML

How Internet Started?

Introduction to Web

Key Milestones in Evolution-Internet

→1950's:

ARPA (Advanced Research Project Agency)

→1970

ARPANET creates a precursor to Transmission Control Protocol (TCP)

→1971

Universities added to the net
Telnet and FTP are available

→1972

First Electronic mail sent

→1973

ARPANET connected to England and Norway

→1974

TCP Starts being used for communicating across a system of networks

→1982

US DoD starts building a defense data network based on ARPANET technology

→1983

ARPAnet splits into ARPANET and MILNET

Key Milestones in Evolution-Internet

→1983

-- The Internet was in place
-- TCP/IP standardized

→1986

NSF (National Science and Foundation) implemented NSFnet
A system of a regional network of routers connected over a backbone network.

→1991

Archie & Gopher were released

→1992

Internet links more than 17,000 networks in 33 countries ; 3 million hosts.

→1993

World Wide web was launched

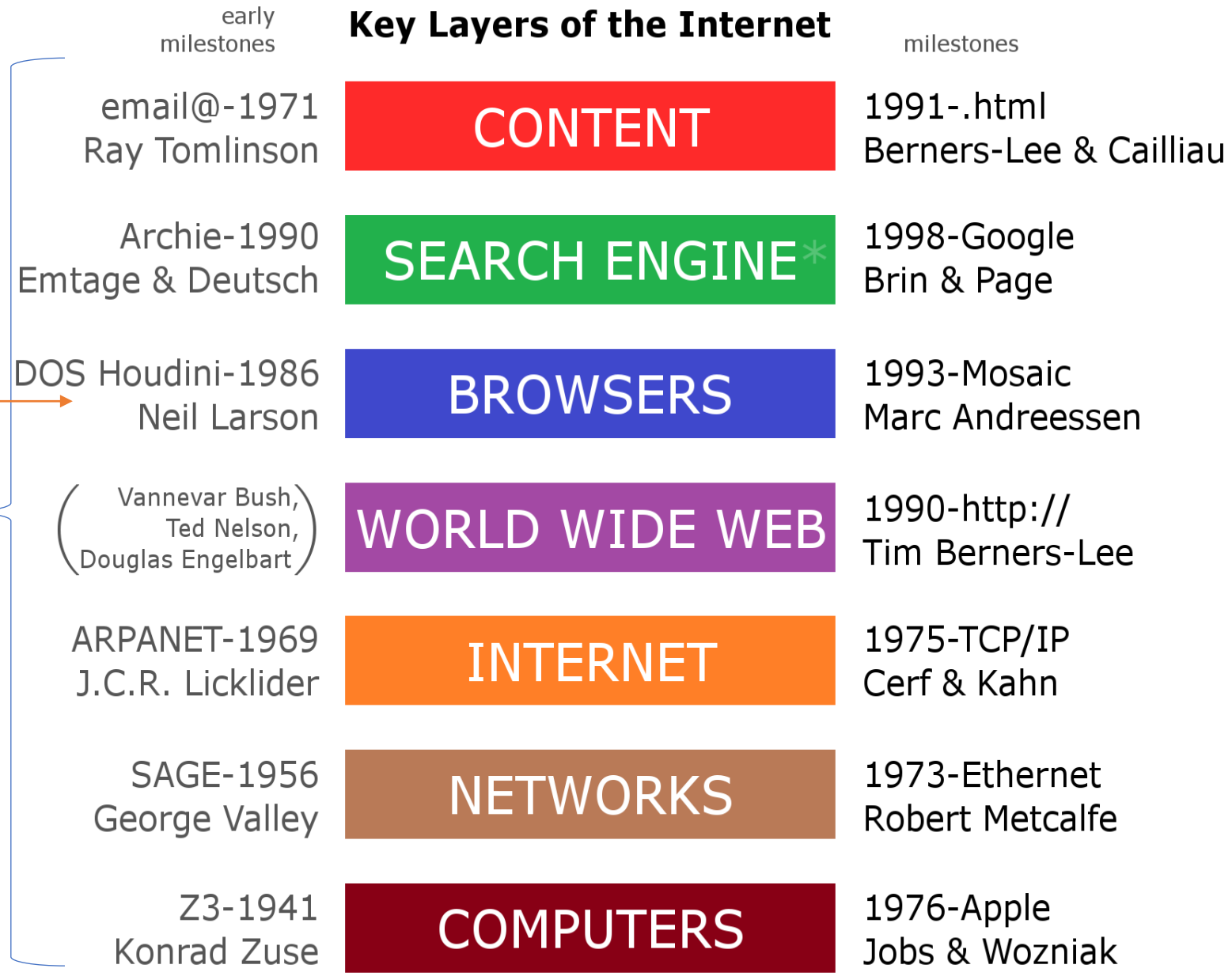
→ 1995

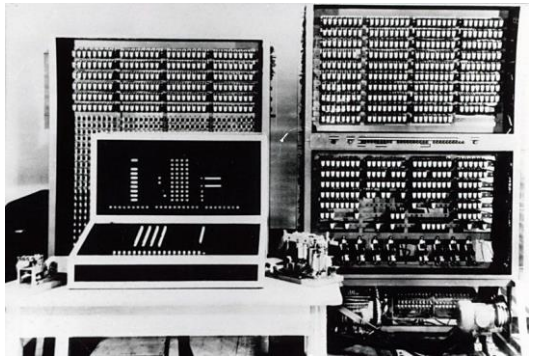
Interconnected network providers start offering services
About 30 million users were added to the network

Showing how users are connected to content (ie, web pages) served by content providers

DOS Houdini knowledge network program

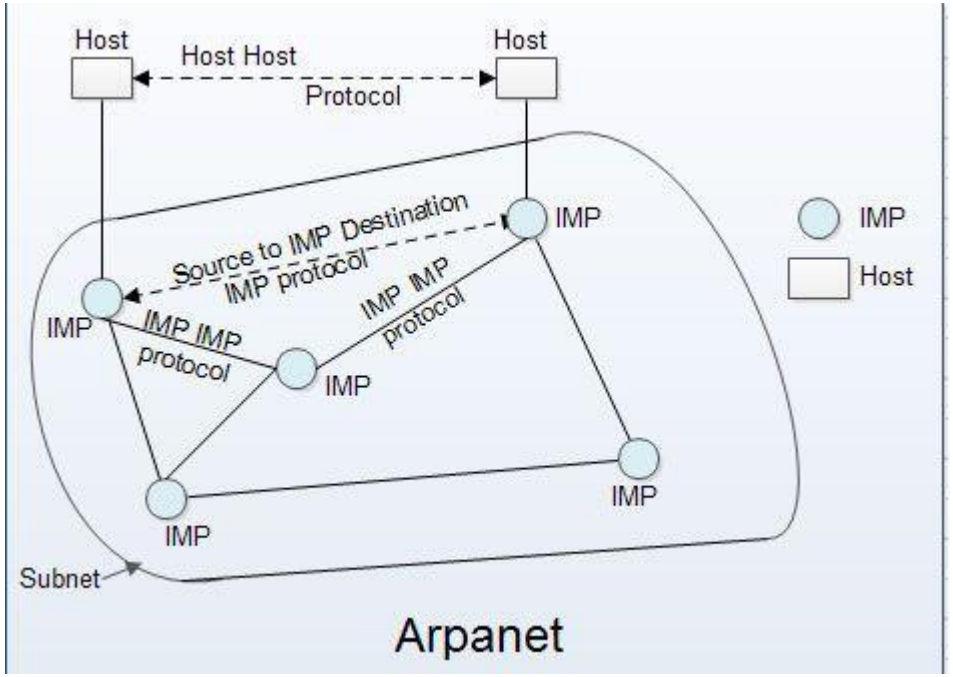
Key contributions were made by some of the instrumental people involved in inventing and developing the technology.

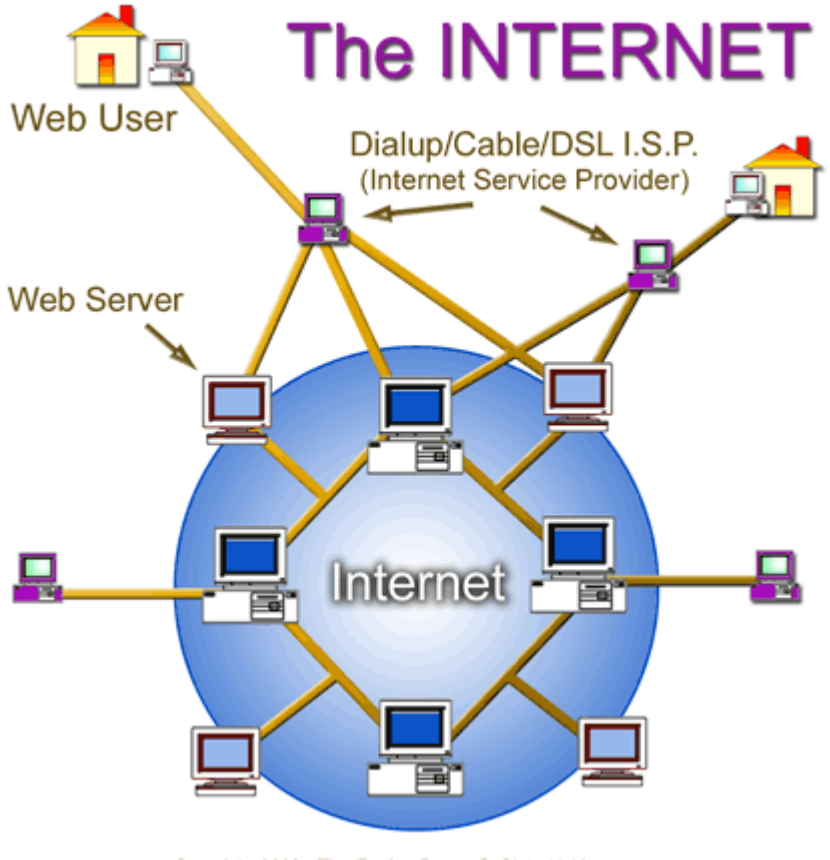




World's First programmable computer - built by **German engineer Konrad Zuse Z3**

The first air defense system -Large computers and associated networking equipment - SAGE





What is Internet?

Definition

- ✓ The Internet is a collection of computers and other devices connected by equipment that allows them to communicate with each other.

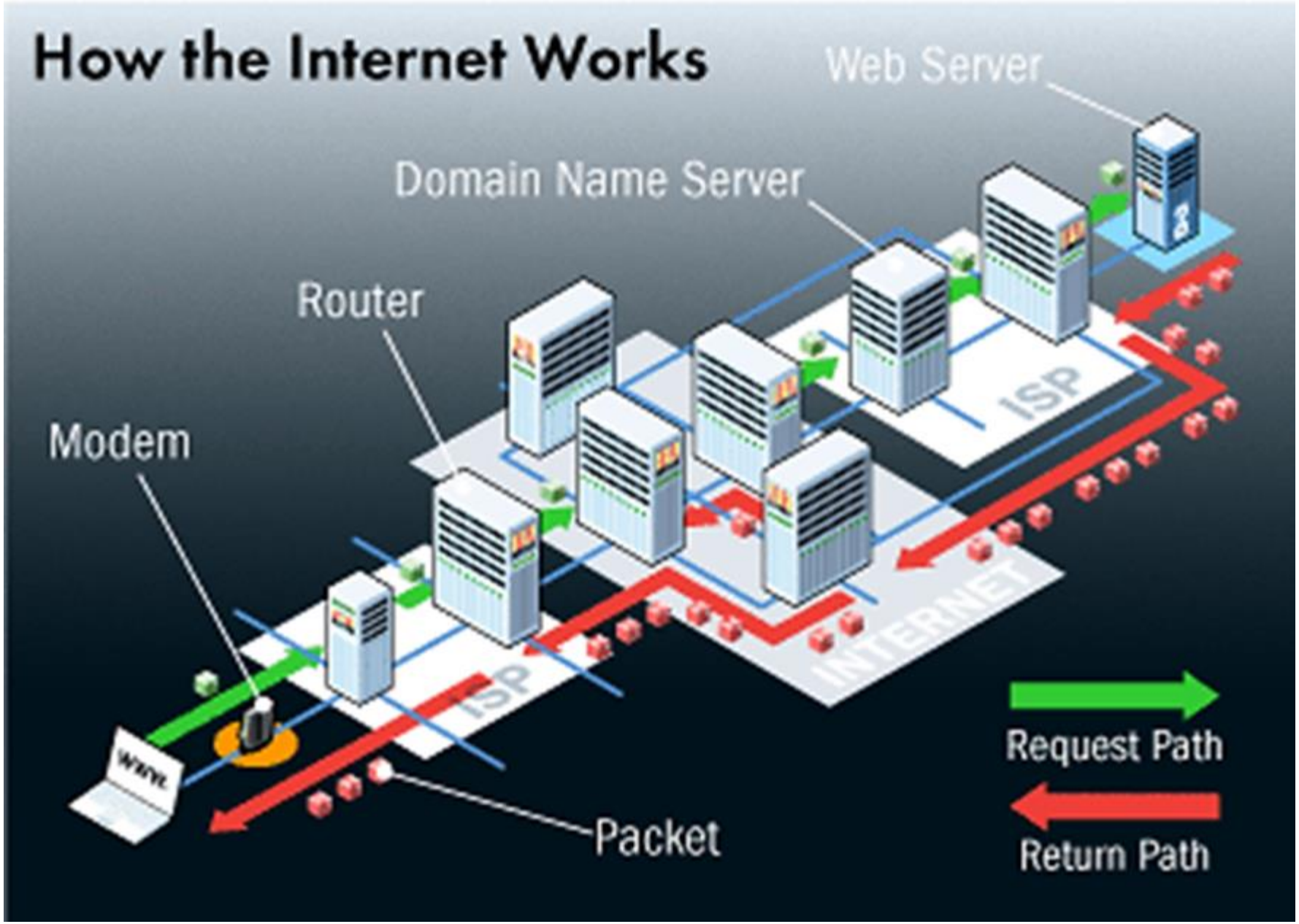
or

- ✓ The Internet is a huge collection of computers connected to communications networks.

or

- ✓ Internet is actually a network of networks rather than a network of computers.

How Internet Works


How the Internet Works
Web Server
Domain Name Server
Router
Modem
ISP
ISP
INTERNET
Request Path
Return Path
Packet

Network Terminologies

→Modem:

A modem is a connecting device that converts analog and digital signals in real-time for two-way communication.

→Router:

- Router is a connecting device that forwards data packets between computer networks.
- It performs traffic directing functions that is data packet is forwarded to its destination code

Network Terminologies

- Packet: A packet normally represents the smallest amount of data that can traverse over a network at a single time.

- How it works:
  - →Whenever a node on a network sends some data over the network
  - →It passes the data frame to the switch
  - →Later to the router.
  - →The router, after looking at the destination IP addresses, encapsulates the data and routes it toward the recipient.
  - →This encapsulated data is the packet that is forwarded over the network.

Network Terminologies

→Protocols:

Formal standards and policies are comprised of rules, procedures, and formats that define communication between two or more devices over a network.

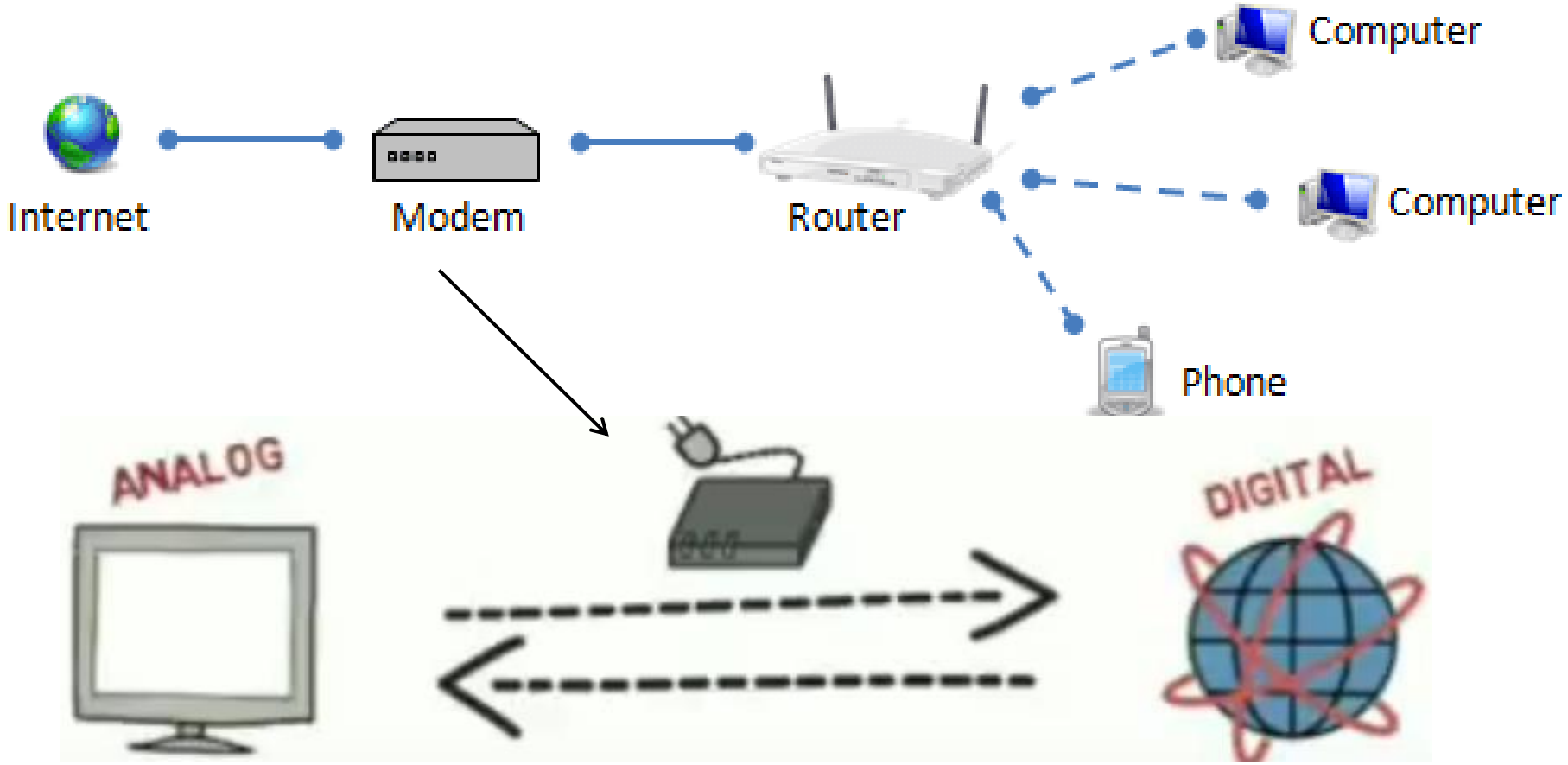
Network protocols govern the end-to-end processes of timely, secure, and managed data or network communication.

→There are several broad types of networking protocols:

1. Network communication protocols:
   Basic data communication protocols, such as TCP/IP, and HTTP.
2. Network security protocols:
   Implement security over network communications and include HTTPS, SSL, and SFTP.
3. Network management protocols:
   Provide network governance and maintenance and include SNMP and ICMP.

Network Terminologies

→Modem & Router:



Internet Applications

- Telnet
- File Transfer Protocol (FTP)
- Electronic Mail (Email)
- Gopher
- Internet Relay Chat (IRC)
- Usenet News
- World Wide Web (WWW)

Class Quiz -I

Answer the following

1. Name any two editors used to edit the HTML or any code.

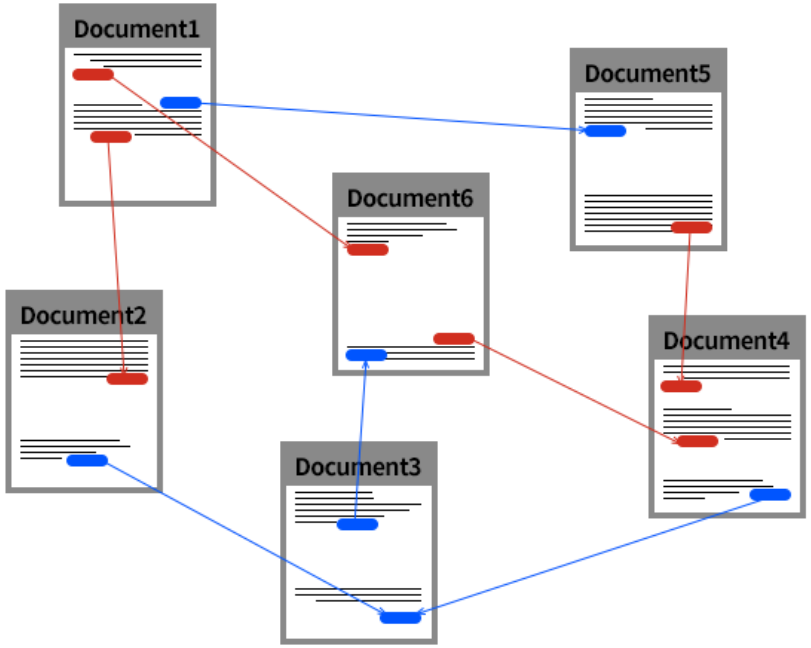
2. State True or False: Desktop operating system can be used in server machine to host the web applications.

3. Give any one example for a standalone application.

4. Documents or web pages are stored in _____ machine so that the client machine can locate and get access to it.

5. _____ is a collection of networks of networks.

6. Name any two internet applications.



World Wide Web

What is Web?

- An interconnected system of information resources accessed through the Internet and consisting of interlinked hypertext documents and other resources. Users access the Web through web browsers and navigate between resources using hyperlinks.

Or

- Web is a set of related web pages typically served from a single web domain.

World wide web (WWW)

- Basically a huge collection of inter-linked documents.
  - Billions of documents
  - Inter-linked in any possible way
- It allows multimedia documents to be shared between machines.
  - Ex: Documents containing: text, image, audio , video & animation

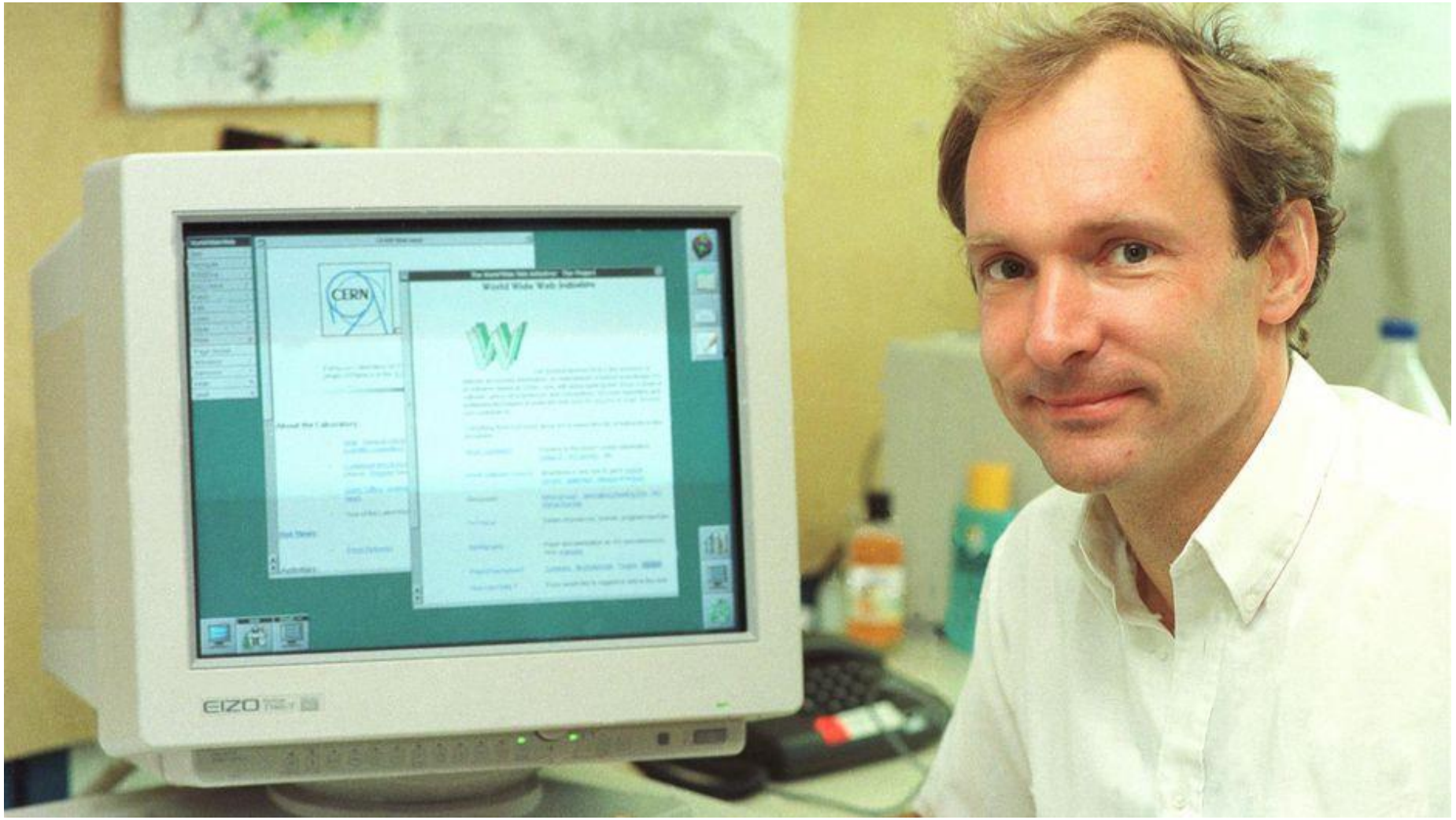
Contd…

- Where do the documents reside?
  - On web servers.
  - Also called “Hyper Text Transfer Protocol (HTTP)” servers.
- They are typically written in
    - Hyper Text Markup Language (HTML)
- Documents get formatted/displayed using
    - Web browsers
        - Internet Explorer
        - Netscape
        - Mosaic

Contd…

- A possible solution to the proliferation of different protocols being used on the Internet
- Origins
  - Tim Berners-Lee at CERN proposed the Web in 1989
    - Purpose: To allow scientists to have access to many databases of scientific work through their computers
  - Document form: hypertext
  - Hypermedia – more than just text – images, sound, etc.
- The Web uses one of the protocols, HTTP, that runs on the Internet--there are several others (telnet, mailto, etc.)

Tim Berners-Lee at CERN proposed the Web



Web browsers

- Browsers are clients - always initiate, servers react (although sometimes servers require responses)
- Mosaic - NCSA (Univ. of Illinois), in early 1993
  - First to use a GUI, which led to an explosion of Web use
  - Initially for X-Windows, under UNIX, but was ported to other platforms by late 1993
- Most requests are for existing documents, using HyperText Transfer Protocol (HTTP)
  - But some requests are for program execution, with the output being returned as a document

Web servers

- Provide responses to browser requests, either existing documents or dynamically built documents
- Browser-server connection is now maintained through more than one request-response cycle
- All communications between browsers and servers use Hypertext Transfer Protocol (HTTP)

Web server operations

- Web servers run as background processes in the operating system
  - Monitor a communications port on the host, accepting HTTP messages when they appear
- All current Web servers came from either
  - The original from CERN
  - The second one, from NCSA

What is the difference between the Internet and the Web?

Difference between Internet and WWW

Internet	WWW
Internet is a global network of networks.	WWW stands for World Wide Web, it is a collection of resources i.e., web pages, multimedia files, etc.
Internet is infrastructure.	WWW is service on top of that infrastructure.
Internet is primarily hardware-based.	WWW is more software-oriented as compared to the Internet.
Internet is superset of WWW.	WWW is a subset of the Internet.
The first version of the Internet was known as ARPANET.	In the beginning WWW was known as NSFNET.
Internet uses IP address.	WWW uses HTTP.

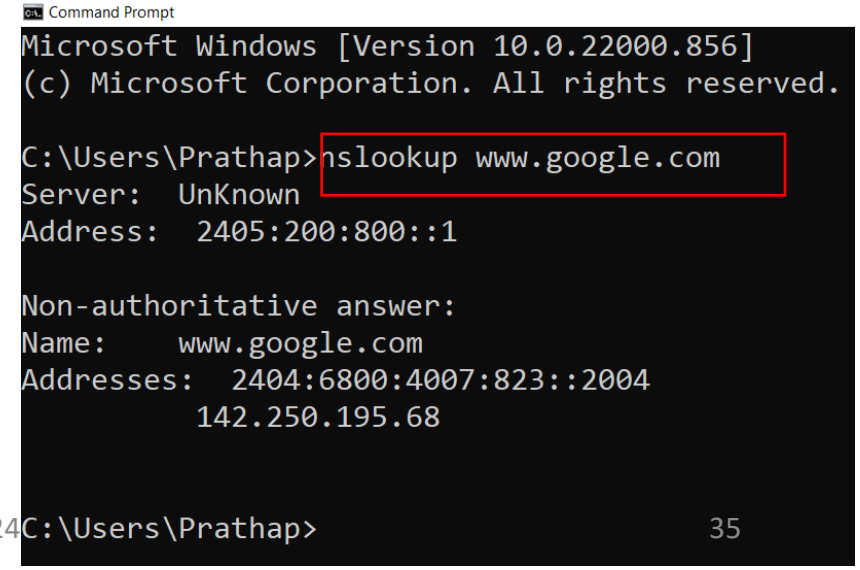
DNS

Domain Name System

Domain Name System - DNS

- The global database system for Internet addressing, mail and other information.
  - Much easier to use and memorize.
- Concept of domains and sub-domains.
  - Domain management is distributed.
  - DNS servers translate domain names to IP addresses.
- Used to convert an IP address to a name
- nslookup is used to get the address of any domain name

  nslookup is a network administration command-line tool for querying the Domain Name System to obtain the mapping between a domain name and IP address



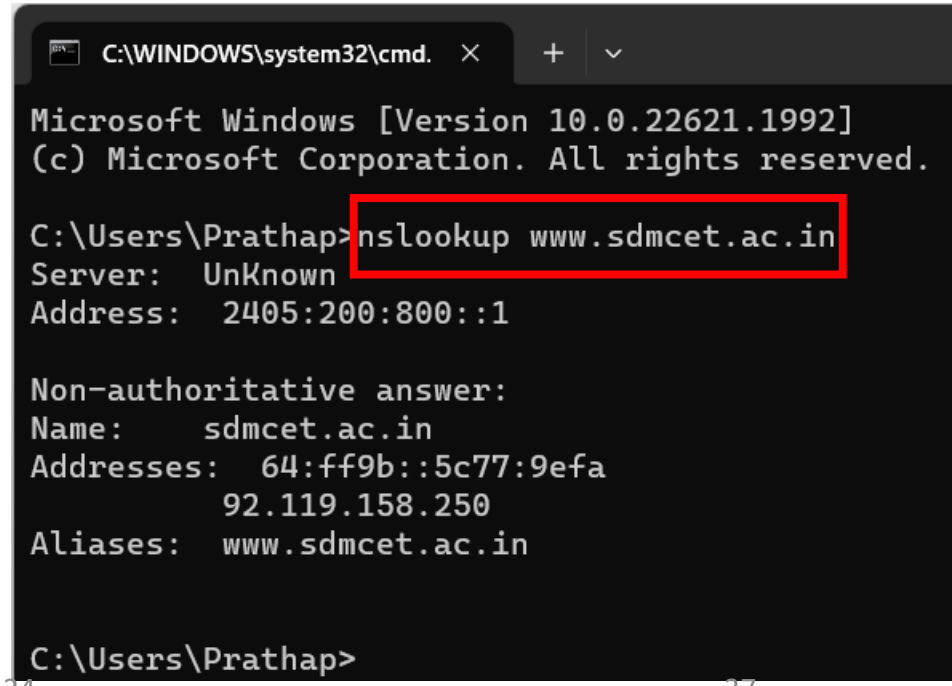
Contd…

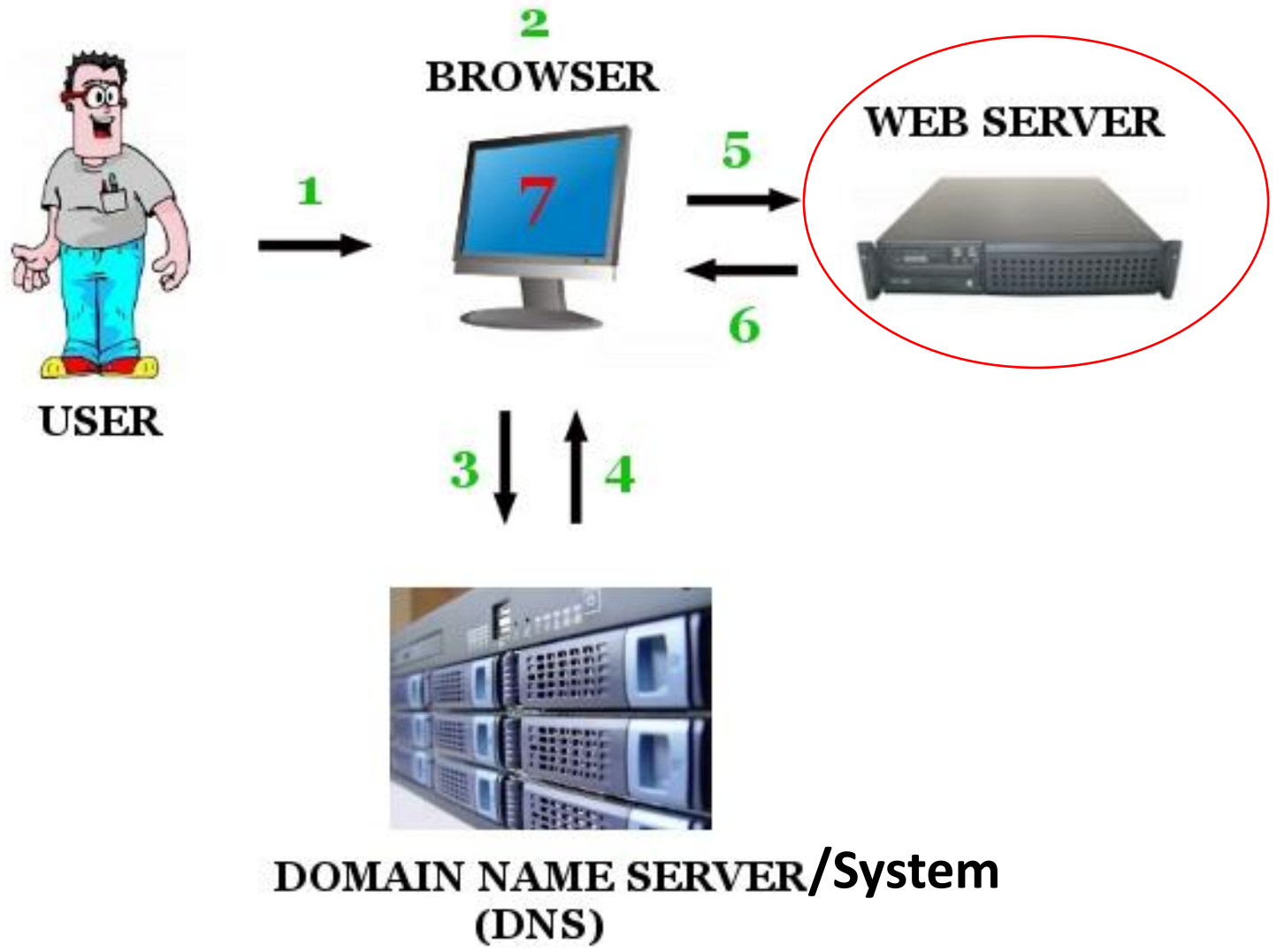
- Domain Name System (DNS) is the Internet's equivalent of a phone book
- DNS maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.
- This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.
- Information from all the domain name servers across the Internet is gathered together and housed at the Central Registry.

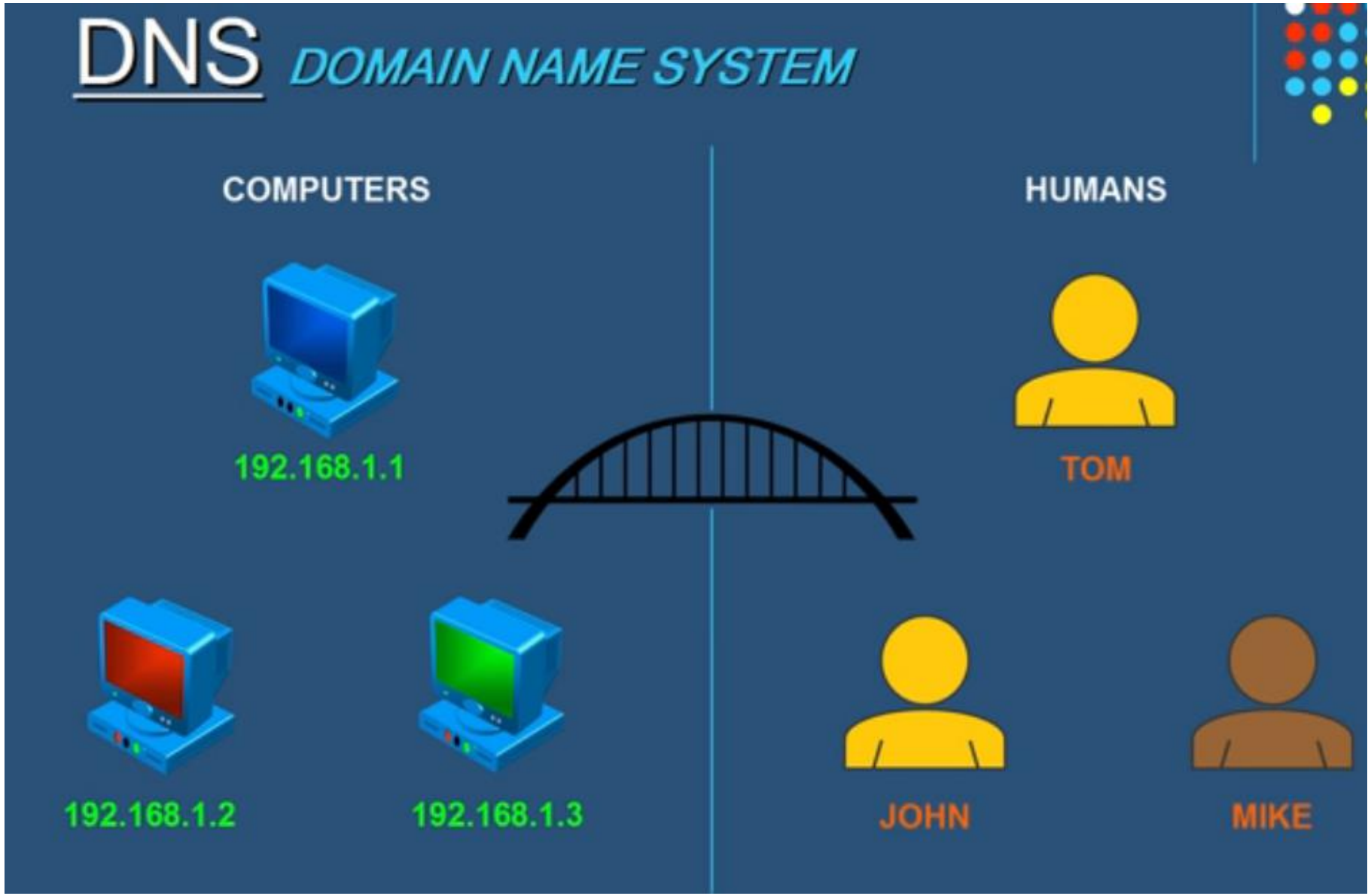
Domain Name System - DNS

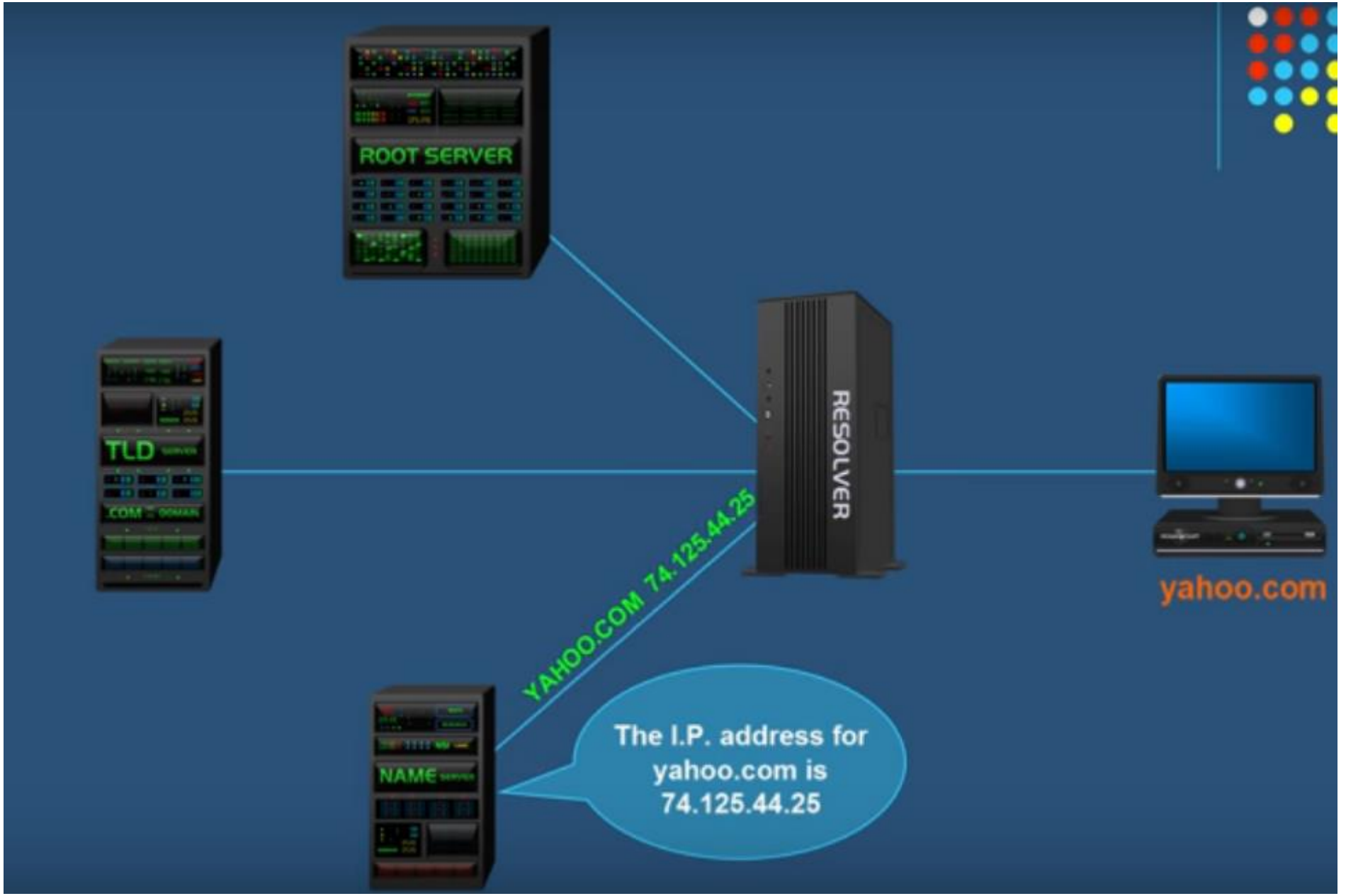
- The global database system for Internet addressing, mail and other information.
  - Much easier to use and memorize.
- Concept of domains and sub-domains.
  - Domain management is distributed.
  - DNS servers translate domain names to IP addresses.
- Used to convert an IP address to a name
- nslookup is used to get the address of any domain name

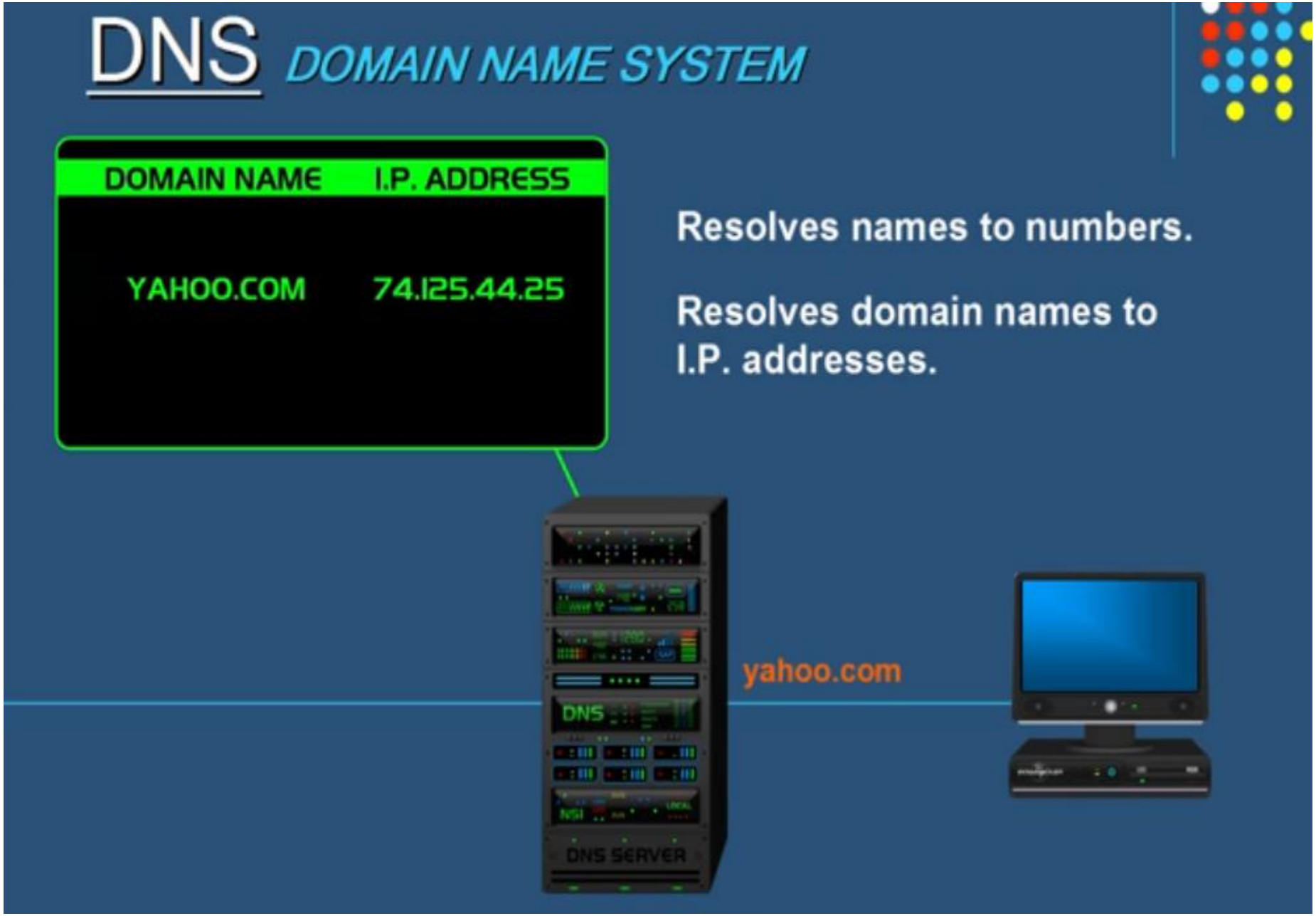
  nslookup is a network administration command-line tool for querying the Domain Name System to obtain the mapping between a domain name and IP address




2
BROWSER
WEB SERVER
1
5
6
USER
3
4
DOMAIN NAME SERVER/System
(DNS)


DNS
DOMAIN NAME SYSTEM
COMPUTERS
HUMANS
192.168.1.1
TOM
192.168.1.2
192.168.1.3
JOHN
MIKE


ROOT SERVER
TLD
COM
RESOLVER
YAHOO.COM 74.125.44.25
NAME
The I.P. address for yahoo.com is 74.125.44.25
yahoo.com


DNS
DOMAIN NAME SYSTEM
DOMAIN NAME
I.P. ADDRESS
YAHOO.COM
74.125.44.25
Resolves names to numbers.
Resolves domain names to I.P. addresses.
DNS
DNS SERVER
yahoo.com

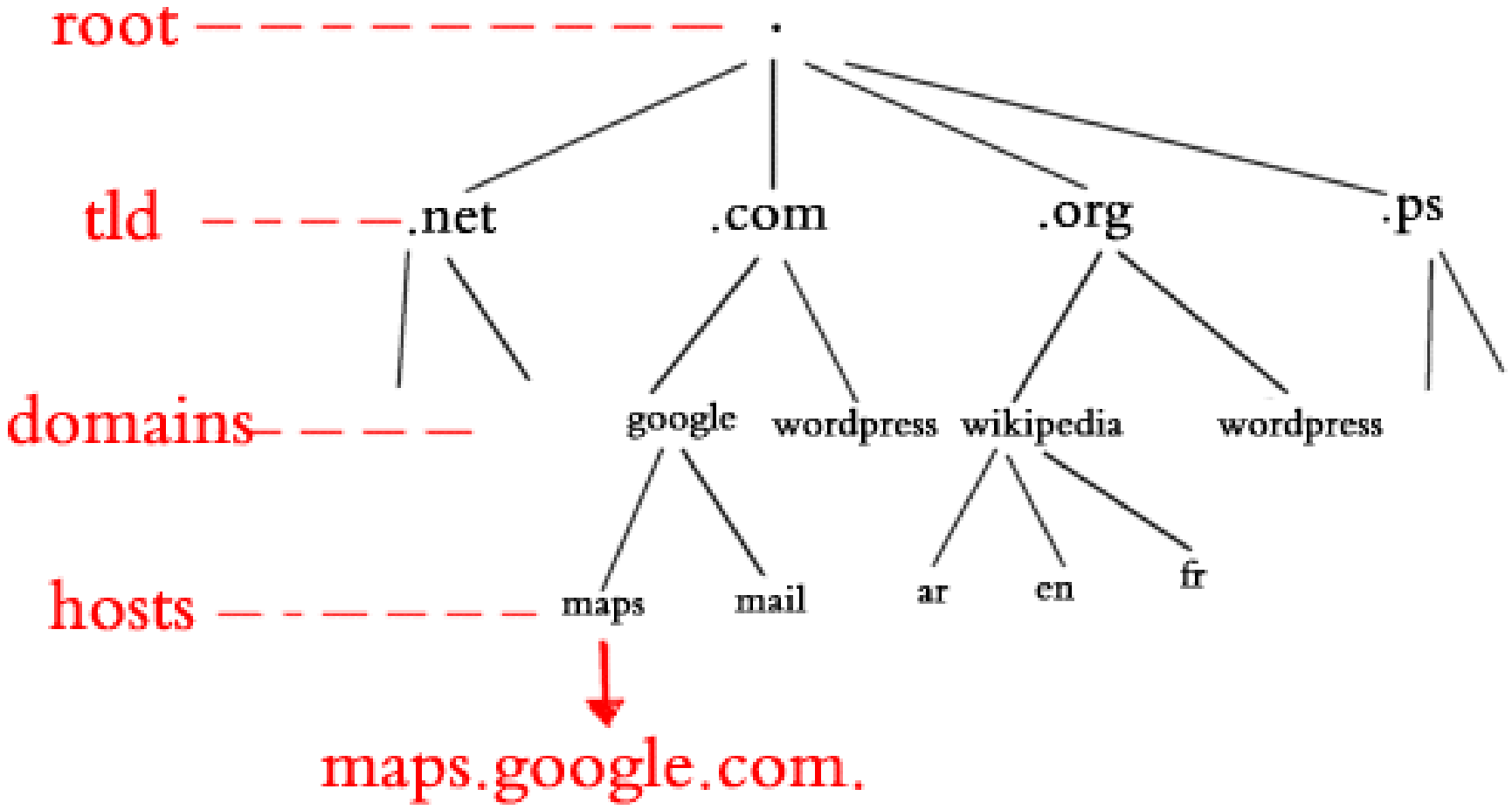
Top Level Domains

- com- Commercial
- org- Non-profit
- net – Network service provider
- gov – US govt.
- mil – millitary
- edu – Education
- au – Australian
- in - Indian
- us – U.S.
- co.in – commercial (Second top-level domain)

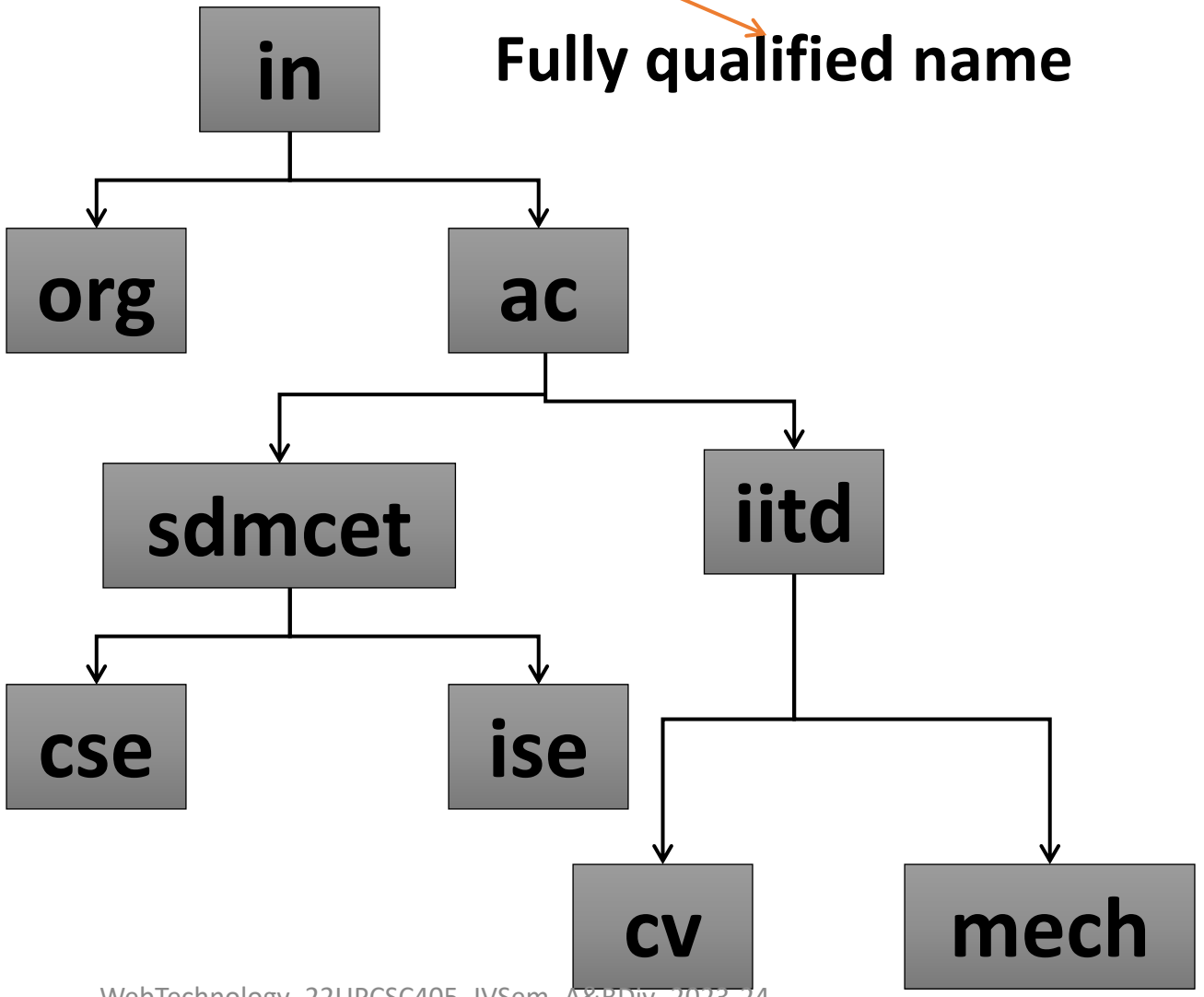
- https://root-servers.org/ - who are all maintaining root servers across the country
- iana.org/domains/root/db – To view TLD

- IANA contracts with root server operators globally to maintain the physical servers.
- NIXI is responsible for setting policies and maintaining the database of registered ".in" domain names.
- Third-party domain name registrars like GoDaddy or BigRock

Domain Names and Address Resolution



Example:www.sdmcet.ac.in



Class Quiz -2

Answer the following

1. State True or False: The web is a collection of multimedia documents.

2. _____protocol is used in web servers.

3. _____ are client applications used to display the server responses in a system.

4. State True or False: The Internet is a repository of online resources.

5. Identify the TLD for the following URL: “iana.org/domains/root/db”

6. The communication between browsers and the server machine takes place through _____ protocol.

Answer the following

7. State True or False: DNS is used to identify the online resources.

8. Example for second top-level domain.

9. Resources are served from the _____ machine.

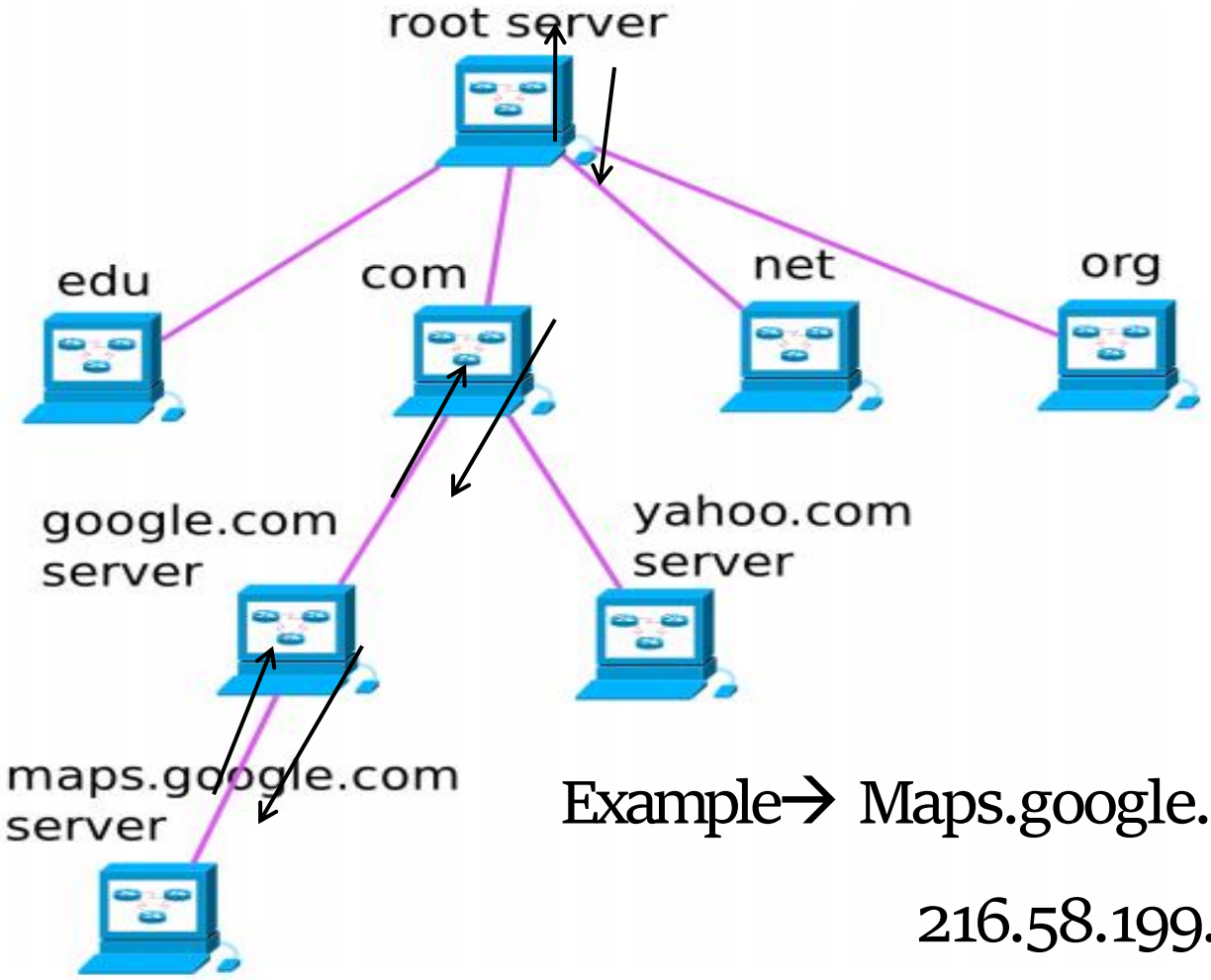
10. Construct a domain address resolution tree for the following URL which contains the root DNS server: “www.outlook.com”

11. Identify the domain name for the following URL: localhost/Myproject/index.html

12. State True or False: nslookup helps to query the online resources.

Example: Hierarchy of Name Server

C:\Windows\System32\drivers\etc



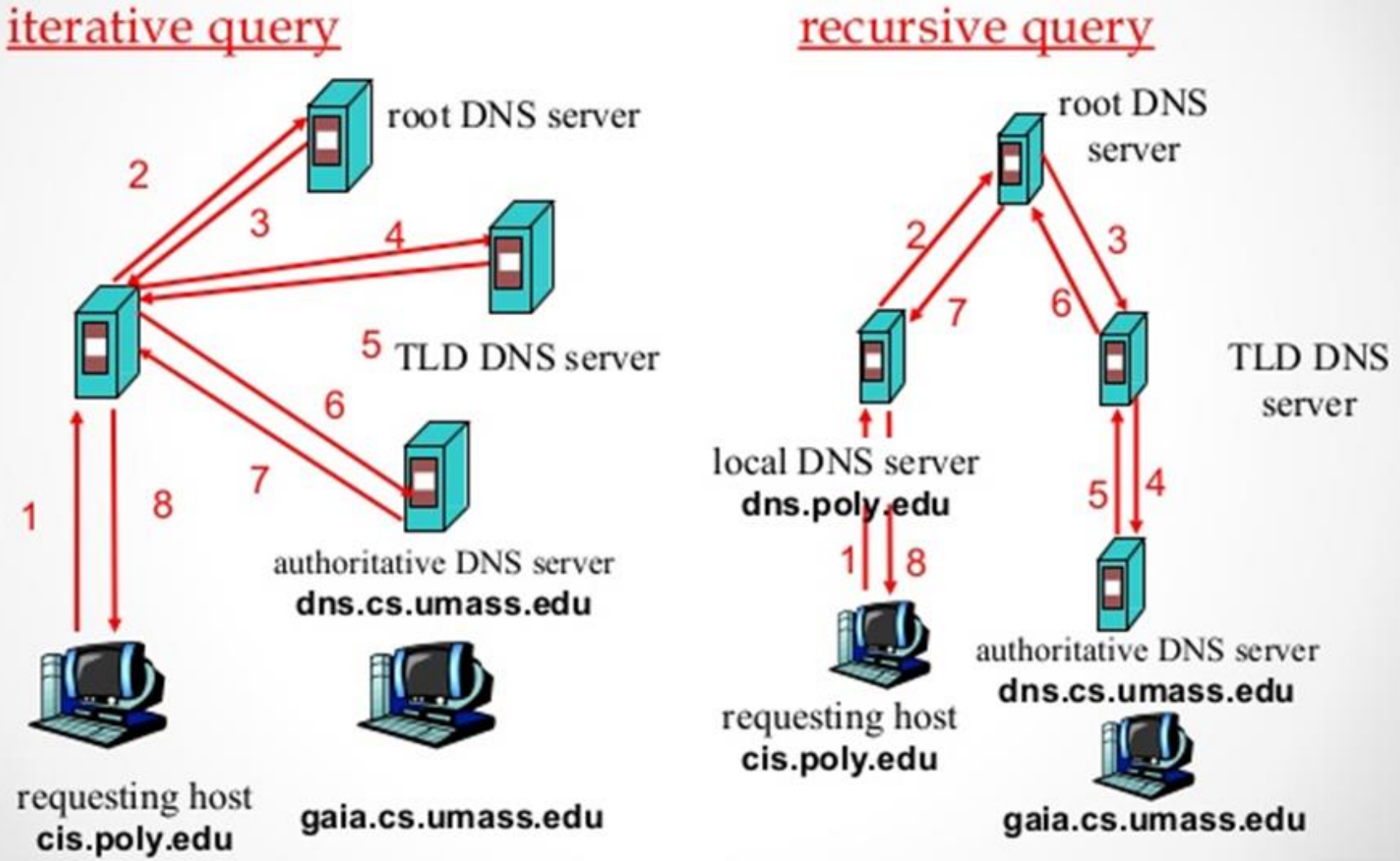
Example→ Maps.google.com

216.58.199.132

DNS name resolution example

Sequentially sends requests

One request & one response

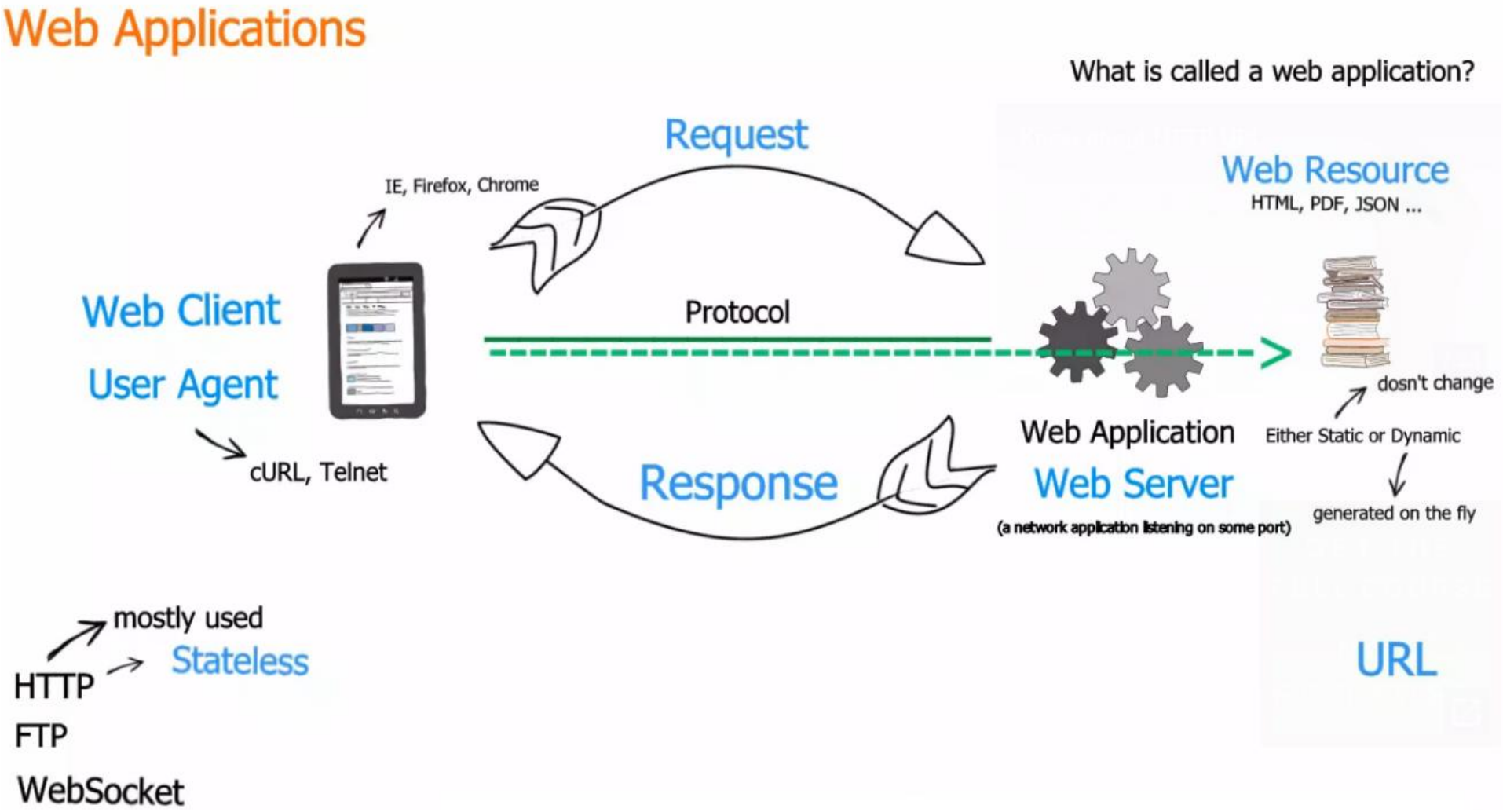


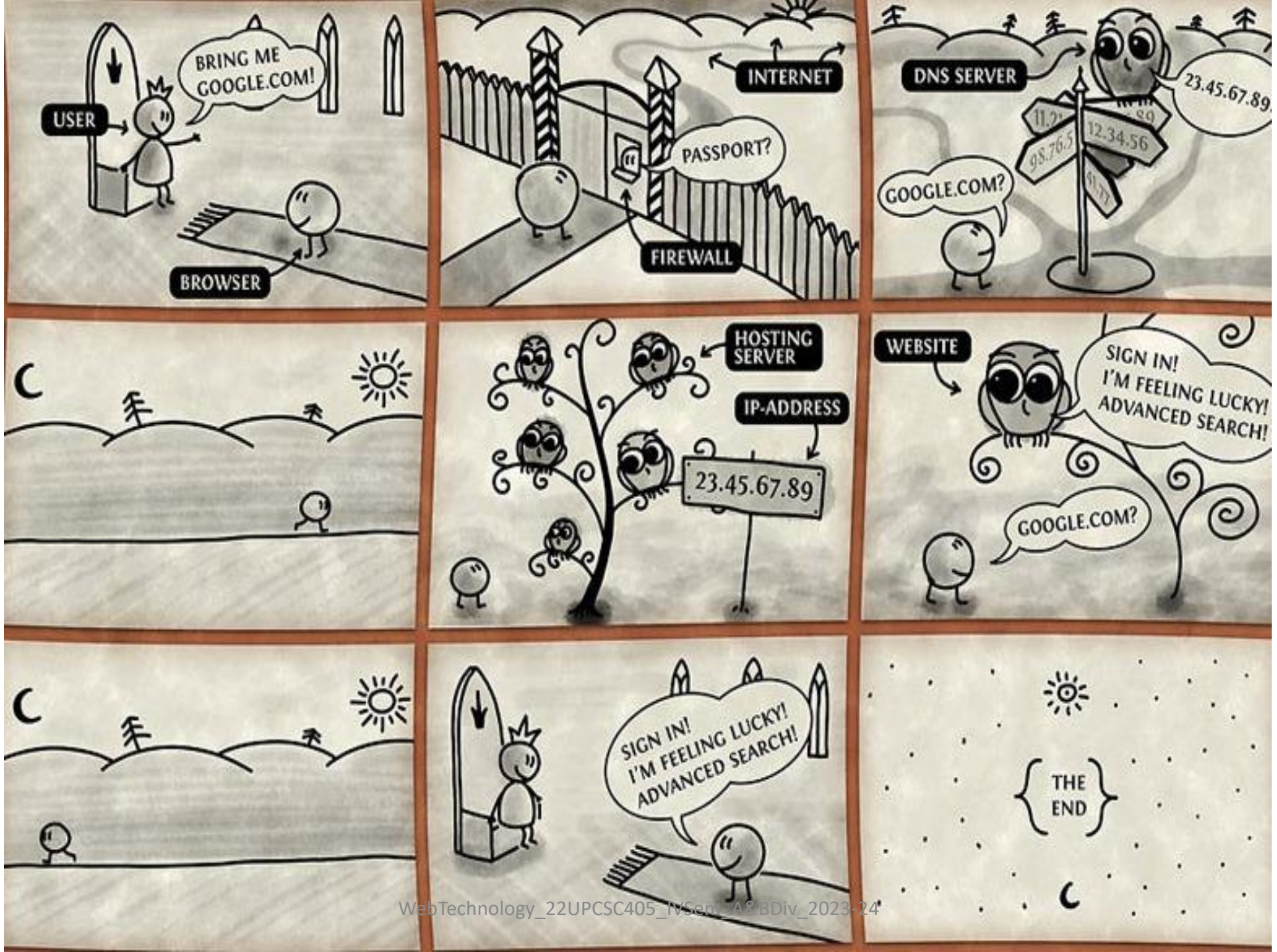
Servers:

- Apache began as the NCSA server, httpd, with some added basic features.
- Apache is open-source software.
- Apache is capable of providing a long list of services beyond the basic process of serving documents to clients.
- IIS under windows XP, the IIS is accessed by going to the control panel, administrator tools, and IIS Admin.
- IIS behavior is managed through a Windows-based management program.
- Apache behavior is controlled by a configuration file that is edited by a manager to change Apache's behavior.

How does the web work?

How Web Applications Works




USER
BRING ME GOOGLE.COM!
BROWSER
INTERNET
PASSPORT?
FIREWALL
DNS SERVER
23.45.67.89
12.34.56
GOOGLE.COM?
HOSTING SERVER
IP-ADDRESS
23.45.67.89
WEBSITE
SIGN IN!
I'M FEELING LUCKY!
ADVANCED SEARCH!
GOOGLE.COM?
SIGN IN!
I'M FEELING LUCKY!
ADVANCED SEARCH!
THE END

Contd…

- Host companies and Internet Service Providers(ISP) interact with the Central Registry on a regular schedule to get updated DNS information.
- Example- www.TechSolutions.ac.in

  S1. ISP views the DNS associated with the domain name.

  S2. DNS translate it into equivalent IP address.

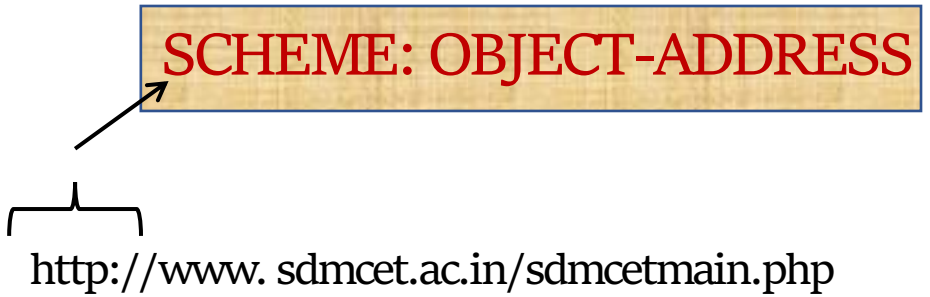
  S3. Then it redirects request to proper host server.

  S4. Page/document is sent as a response to the client.
- To host any web page user must register domain name with the ISP to make it publically available.
- It takes about 12-36 hrs, Domain name servers world-wide to be updated and able to access the information.

URLs

- Used to identify documents(resources) on the internet.
- There are many different kinds of resources identified by different forms of URLs.
- URL Formats:

Example:



- For the HTTP protocol, the object address is: a fully qualified domain name/doc path.
- For the file protocol, only the doc path is needed.
- www - Technically, it's a subdomain traditionally used to indicate that a site is part of the web

Example - www

- Excluding subdomain:
  - maps.Google.com
  - mail.Google.com
  - bard.Google.com

- Including subdomain:
  - www.sdmcet.ac.in
  - www.Google.com
  - www.outlook.com

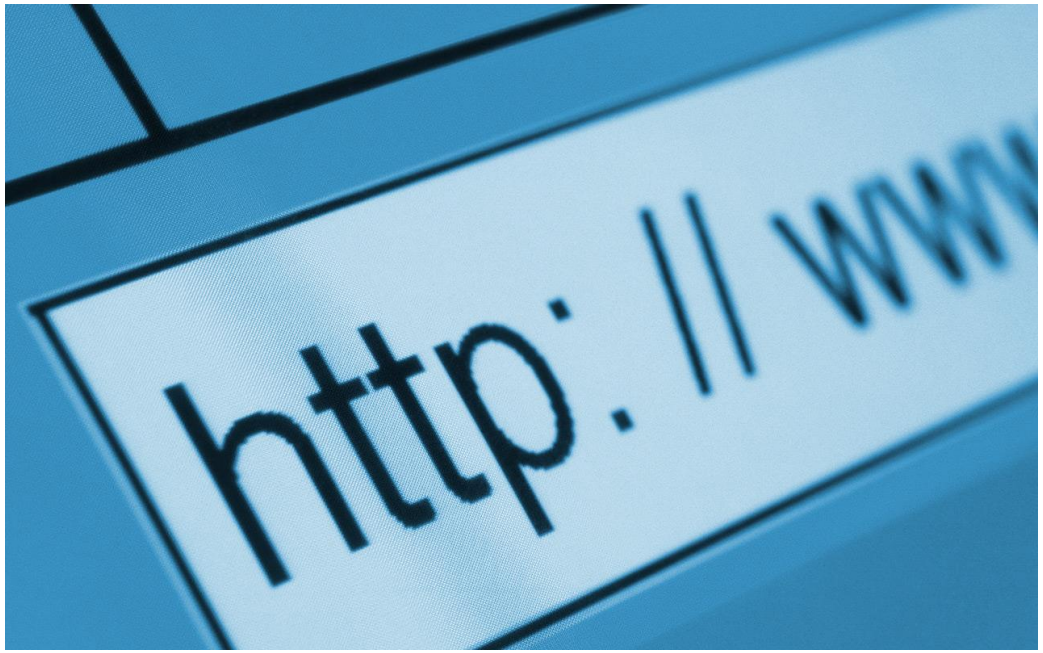
Contd…

- **Scheme**: It is a communication protocol.

  Ex:- http, ftp, gopher, telnet, file, mailto and news

- Messages to a host machine must be directed to the appropriate handling process running on the host.
- Such processes are identified by their associated port number.
- The default port number of web server processes is 80.
- If the default port is not used, a port number is appended to the host.

  Example:- http://localhost:8090/homepage.html



HTTP

Hypertext Transfer Protocol(HTTP)

- The protocol used by ALL Web communications
- Request Phase
  - Form:

    HTTP method domain part of URL HTTP ver.

    Header fields

    blank line

    Message body
- An example of the first line of a request:

  GET /degrees.html HTTP/1.1

HTTP methods

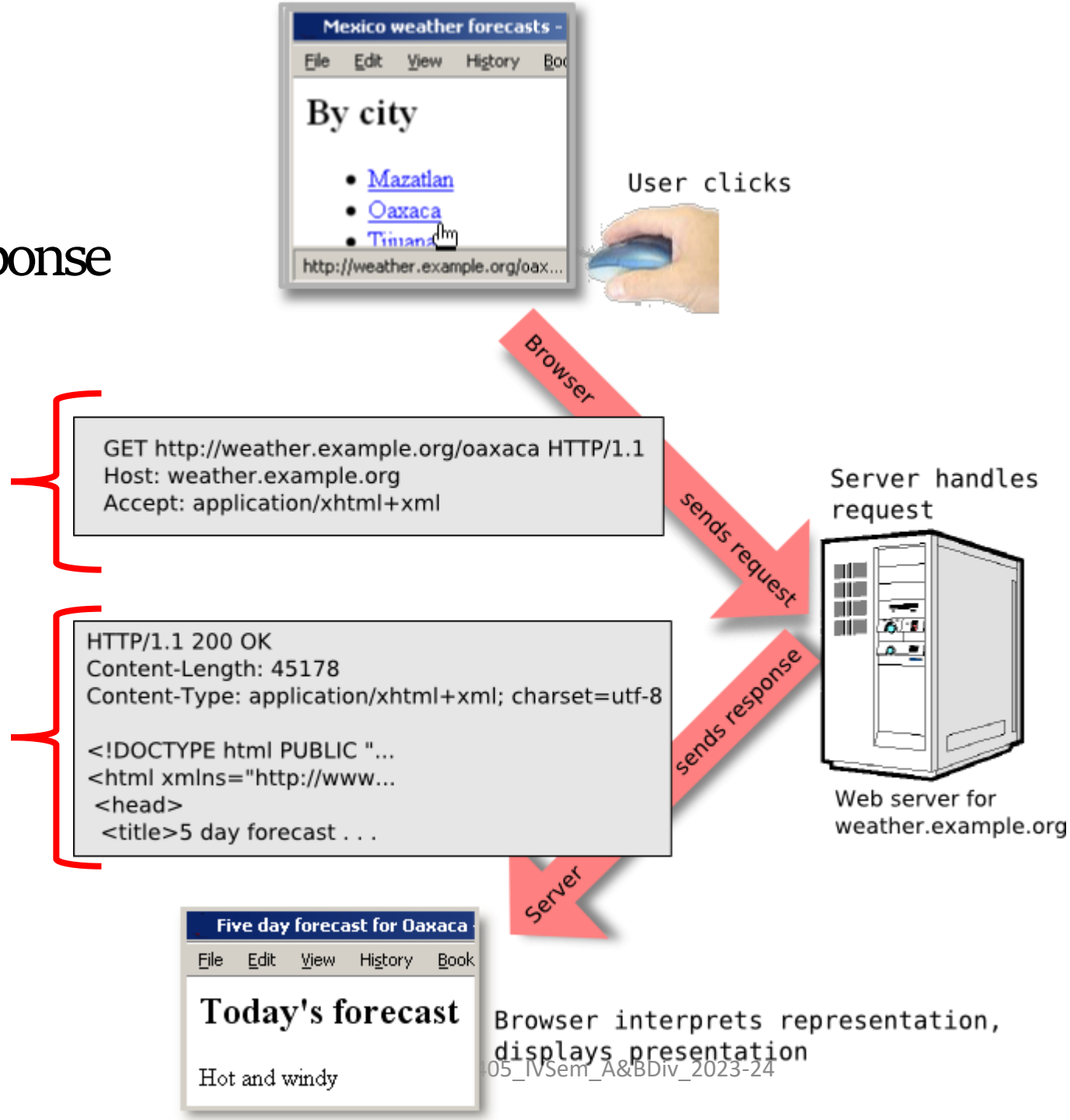
Method Name	Description
GET	Retrieves data from a web server by specifying parameters in the URL portion of the request.
HEAD	Same as GET, but transfers the status line and header section only.
POST	A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
PUT	Replaces all current representations of the target resource with the uploaded content.
DELETE	Removes all current representations of the target resource given by a URI.

Example

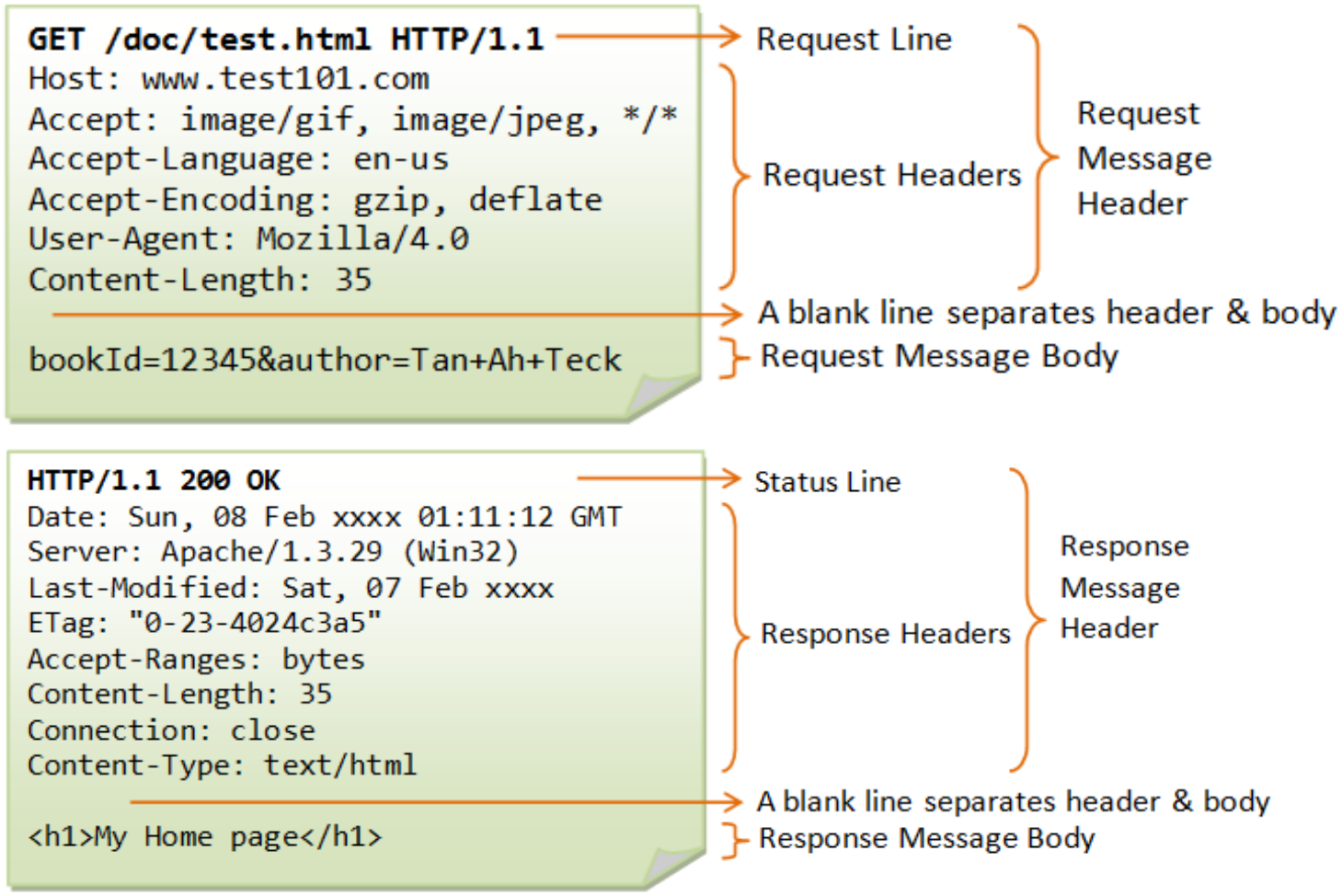
HTTP Request-Response Message Format

Request Message Format

Response Message Format



HTTP Request-Response Message Format



HTTP Request message-General format

- Contains 3 parts
  - Request line: command, web page requested, HTTP version number
  - Request header: includes browser in use, date, and some other info
  - Request body: contains information that was sent to the server (optional)

Sample request

GET salisbury/home.html HTTP/1.1

HOST: www.su.edu

Request Line

DATE: Tues 08 Sept 2006 13:45:34 EST

User-Agent: Mozilla/7.0

Referrer: http:salisbury.edu/~welcome

Request Header

HTTP Response message-General format

- Contains 3 parts:
  - Response status: http version #, status code, and reason phrase (description of status code)
  - Response header: optional info including server being used, date, URL of web page
  - Response body: the website (in HTML)

Sample Response

HTTP/1.1 200 OK — Response Status

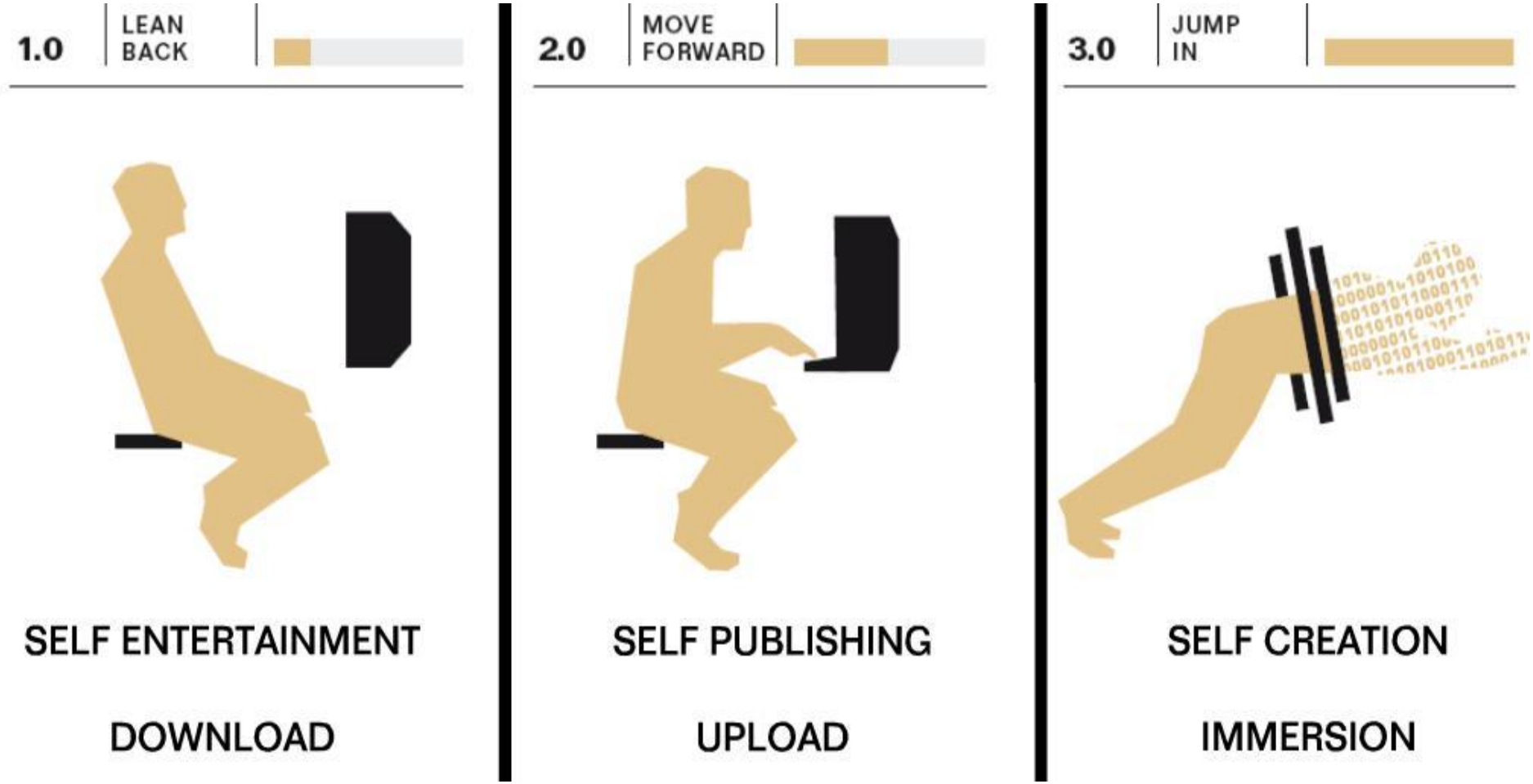
Date: Tues 08 Sept 2006 13:45:34 EST
Server: Apache
Location: http:salisbury.edu/~welcome — Response Header
Content-Type: text/html

```
<html>
<head>
<title> SALISBURY</title>
</head>
<body>
<p> welcome to salisbury</p>
.....
</body>
</html>
```

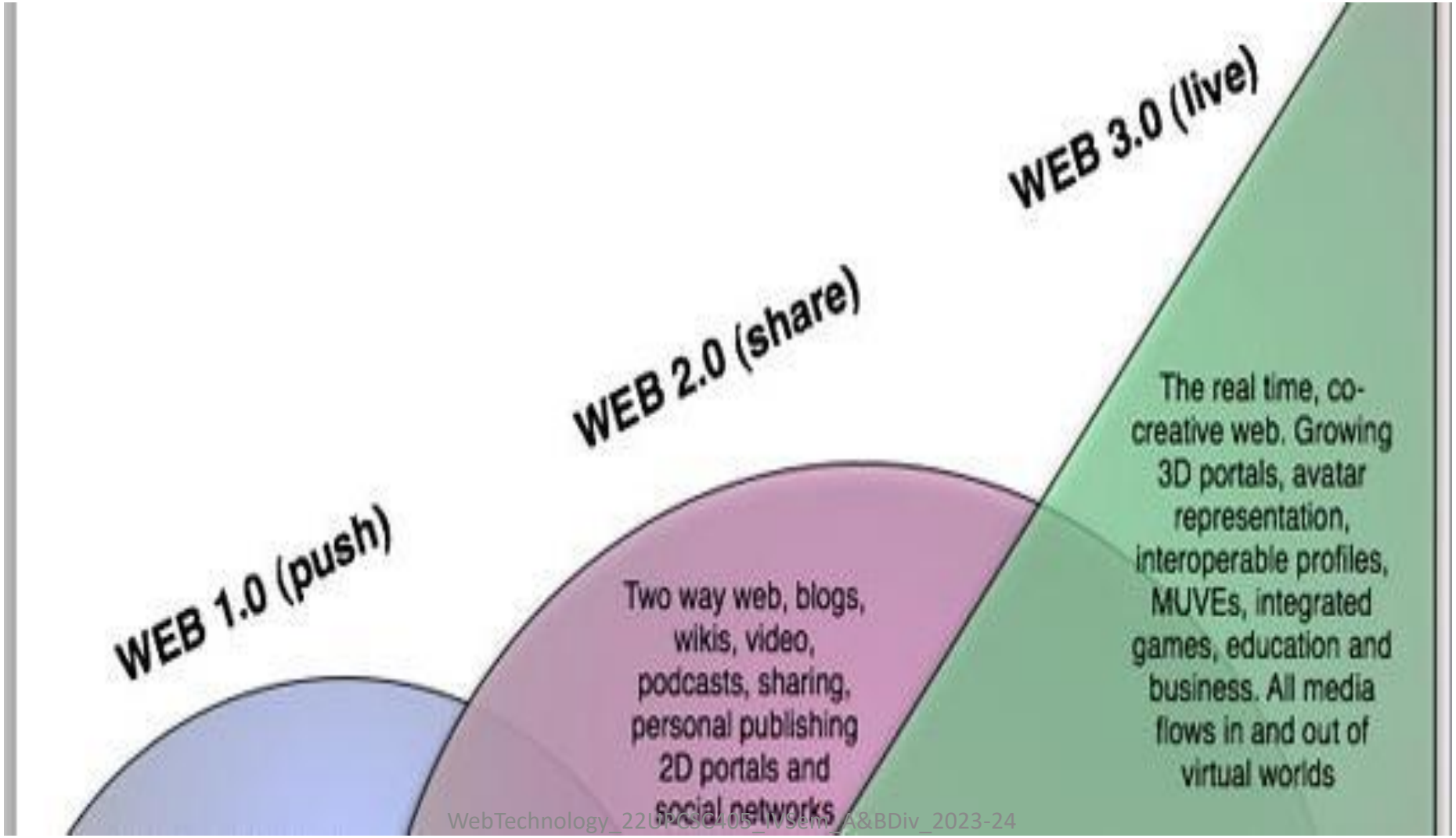
Response Body in HTML

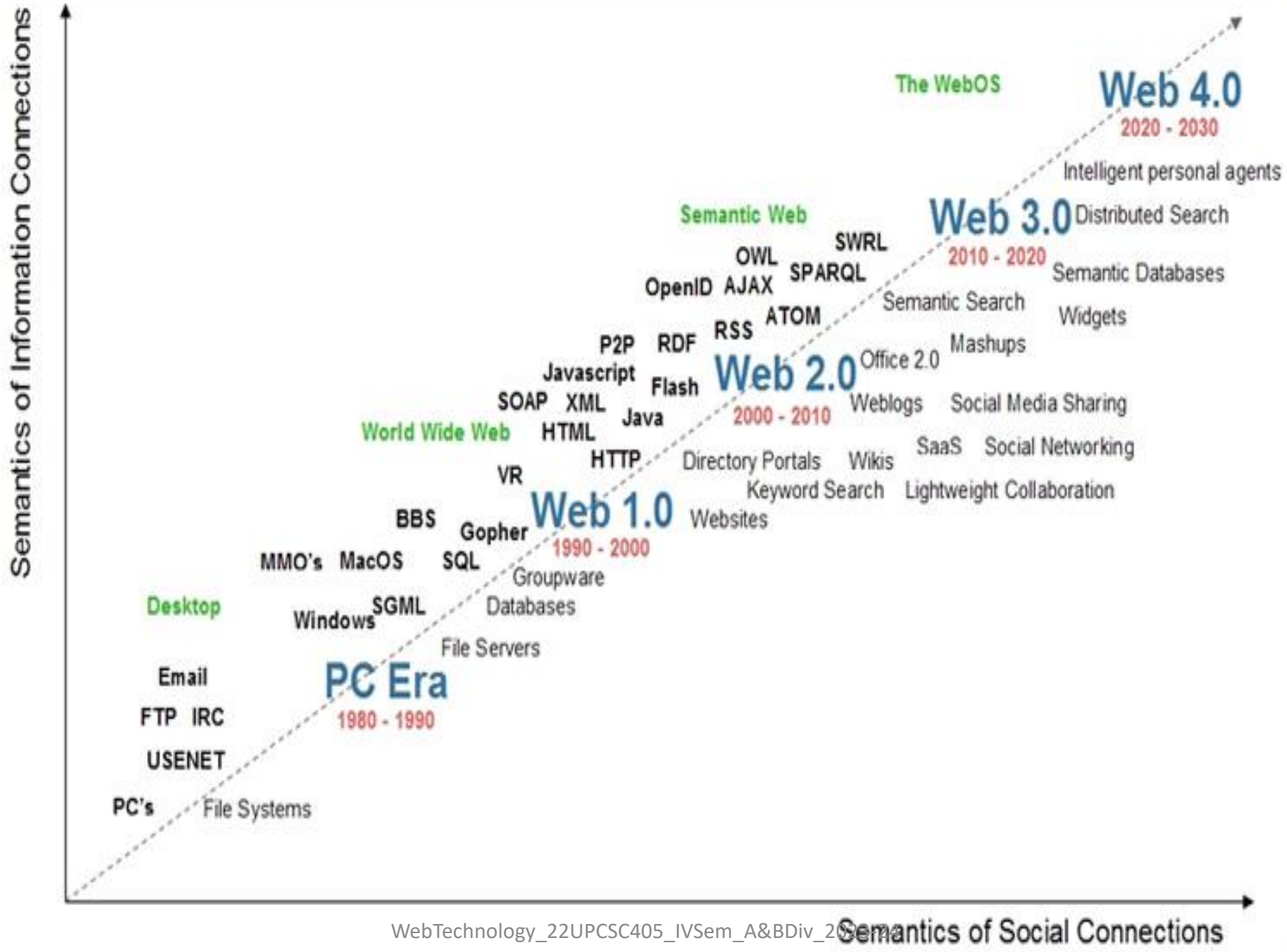
Difference between Web, Web1.0 & Web 2.0

Web, Web1.0, Web2.0 and Web3.0



Web1.0, Web 2.0 & Web 3.0




Semantics of Information Connections
Semantics of Social Connections
The WebOS
Web 4.0
2020 - 2030
Intelligent personal agents
Semantic Web
Web 3.0
Distributed Search
SWRL
OWL
SPARQL
2010 - 2020
Semantic Databases
OpenID
AJAX
Semantic Search
Widgets
ATOM
RSS
Mashups
P2P
RDF
Office 2.0
Javascript
Flash
Web 2.0
SOAP
XML
Java
Weblogs
Social Media Sharing
2000 - 2010
World Wide Web
HTML
SaaS
Social Networking
HTTP
Directory Portals
Wikis
VR
Keyword Search
Lightweight Collaboration
BBS
Gopher
Web 1.0
Websites
1990 - 2000
MMO's
MacOS
SQL
Groupware
Desktop
SGML
Databases
Windows
File Servers
Email
PC Era
FTP
IRC
1980 - 1990
USENET
PC's
File Systems

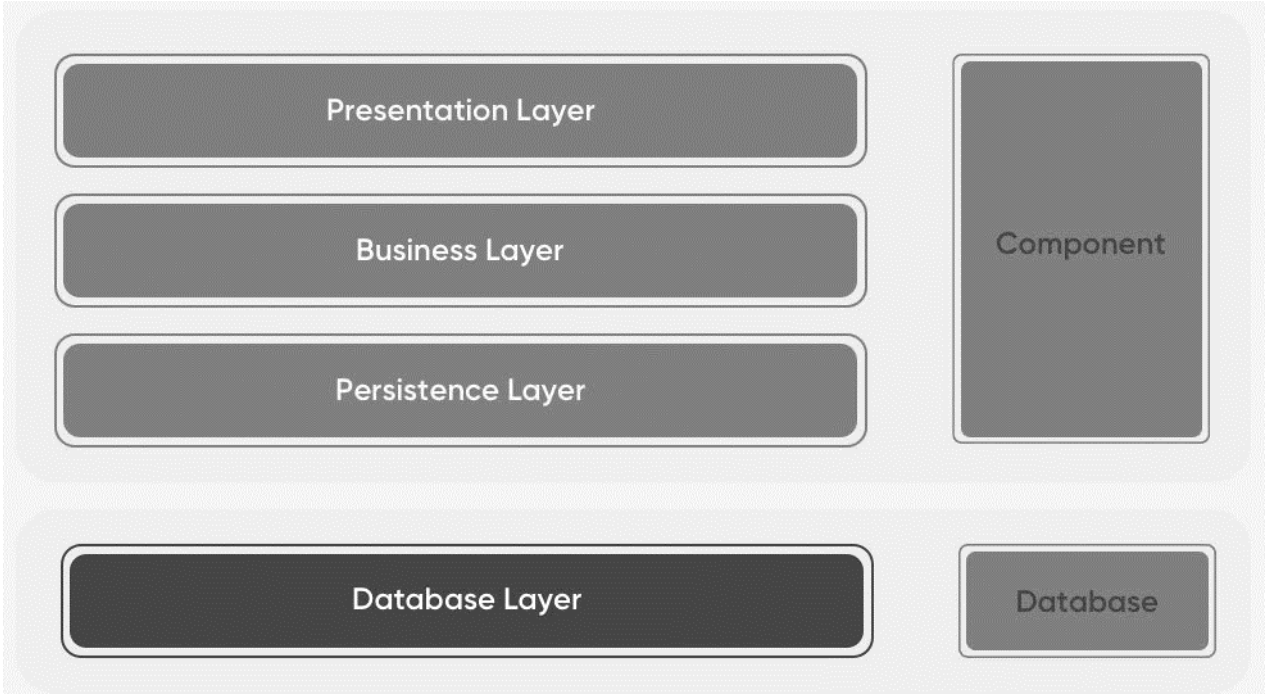
Web 1.0 / 2.0 / 3.0 Summary

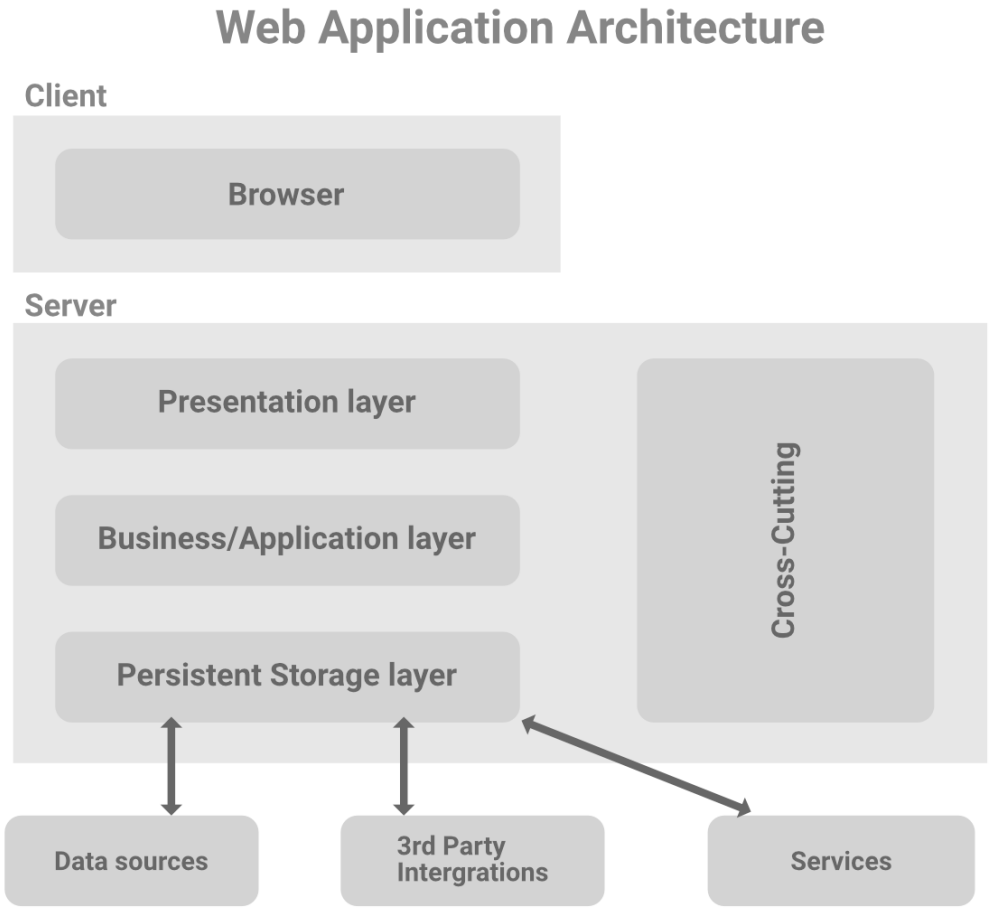
Crawl	Walk	Run
Web 1.0	Web 2.0	Web 3.0
Mostly Read-Only	Wildly Read-Write	Portable & Personal
Company Focus	Community Focus	Individual Focus
Home Pages	Blogs / Wikis	Lifestreams / Waves
Owning Content	Sharing Content	Consolidating Content
Web Forms	Web Applications	Smart Applications
Directories	Tagging	User Behavior
Page Views	Cost Per Click	User Engagement
Banner Advertising	Interactive Advertising	Behavioral Advertising
Britannica Online	Wikipedia	The Semantic Web
HTML/ Portals	XML / RSS	RDF / RDFS / OWL

Web Architecture

WWW Architecture

- It is a client-server architectural model.

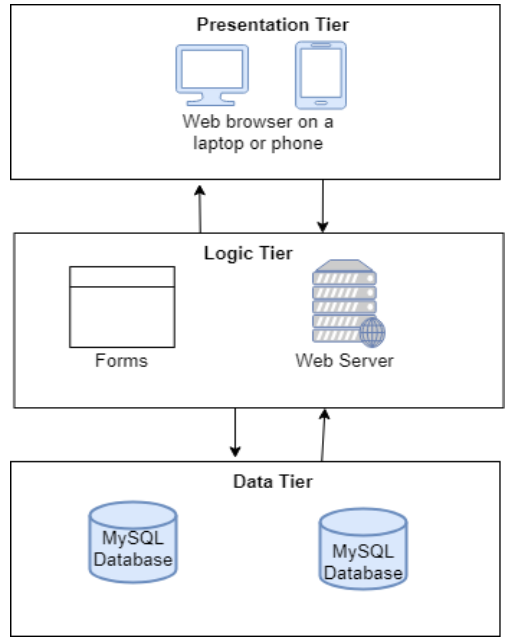


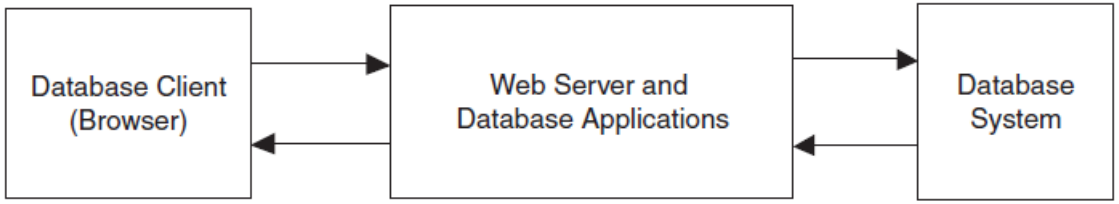


Types of web architecture

- Mainly web architecture is divided into two types:
  1. Two-tier architecture
  2. N-tier architecture.
- Two-tier architecture
  - It includes any two web components:
    1. Presentation layer
    2. Database layer or business logic layer
- N-tier architecture
  - A multilayered client-server architecture - Presentation, Processing, and Data functions are divided into logically and physically separate tiers.

3-tier architecture





Class Quiz -3

Answer the following

1. _____ is the web server used for handling web-based request and response messages.
2. Identify different types of URL name resolvers used in DNS servers
3. State True or False: FTP is a stateless protocol
4. Identify any one key difference between web1.0 & web3.0
5. State True or False: DNS is a target server used to process the request messages.
6. Three tier architecture includes_____ components
7. www is a part of a Scheme in the general format of a URL

Answer the following

8. List and name types of web architecture
9. _____number is used to identify a specific process to which an internet message is to be forwarded when it arrives at a server.
10. "Accept" in a request message specifies _____type
11. List components of web architecture
12. State True or False: The presentation layer is used to collect the data
13. Name three parts of the response message format
14. Name three parts of the request message format
15. State True or False: Sharing content and interactive advertising are examples of web2.0

Introduction to HTML

Web 1.0, web 2.0

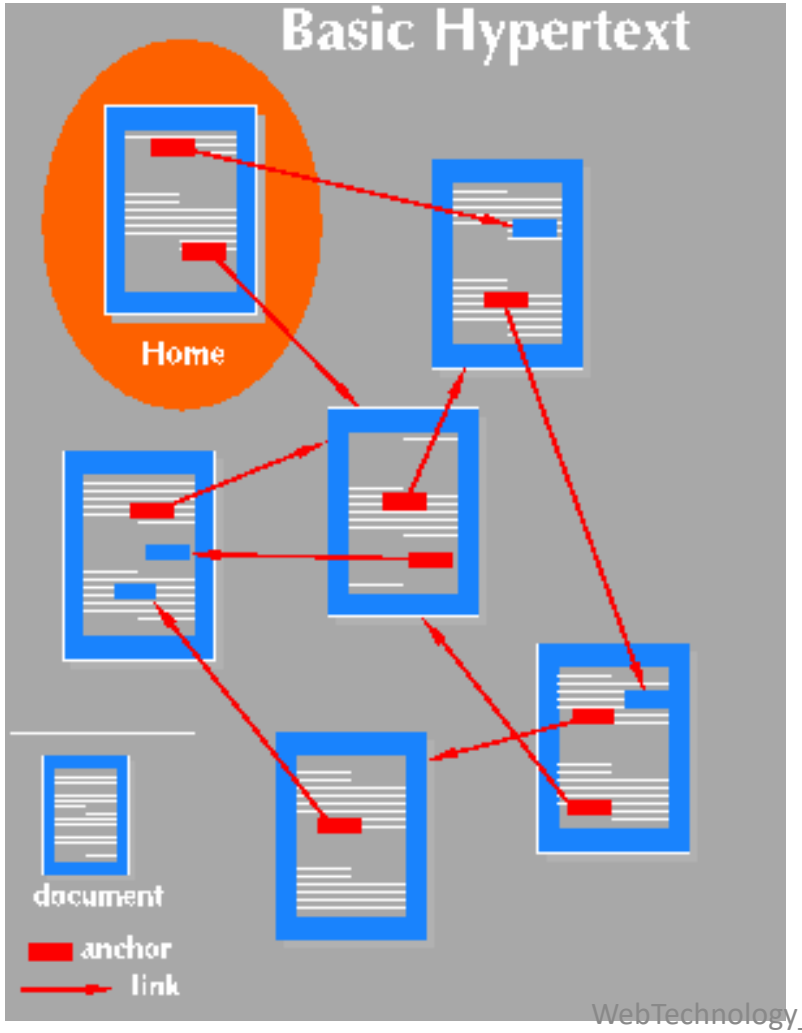
HTML

- Hyper Text Markup Language.
  - Constitutes a collection of *platform-independent* styles that define the various components of a web document.
  - Styles indicated by markup tags.
- Editors:
  - Plain-text documents can be created using any text editor.
  - WYSIWYG editors are also available.

Contd…

- What is a markup language?
  - One where we can embed special tags or formatting commands in the text.
  - To describe how the text should be displayed/printed.
- HTML is a markup language
  - Special formatting codes (tags) to adjust fonts, create bulleted lists, create forms, display images, create tables, etc.

Hypertext



- Hypertext is the text which contains links to other texts.

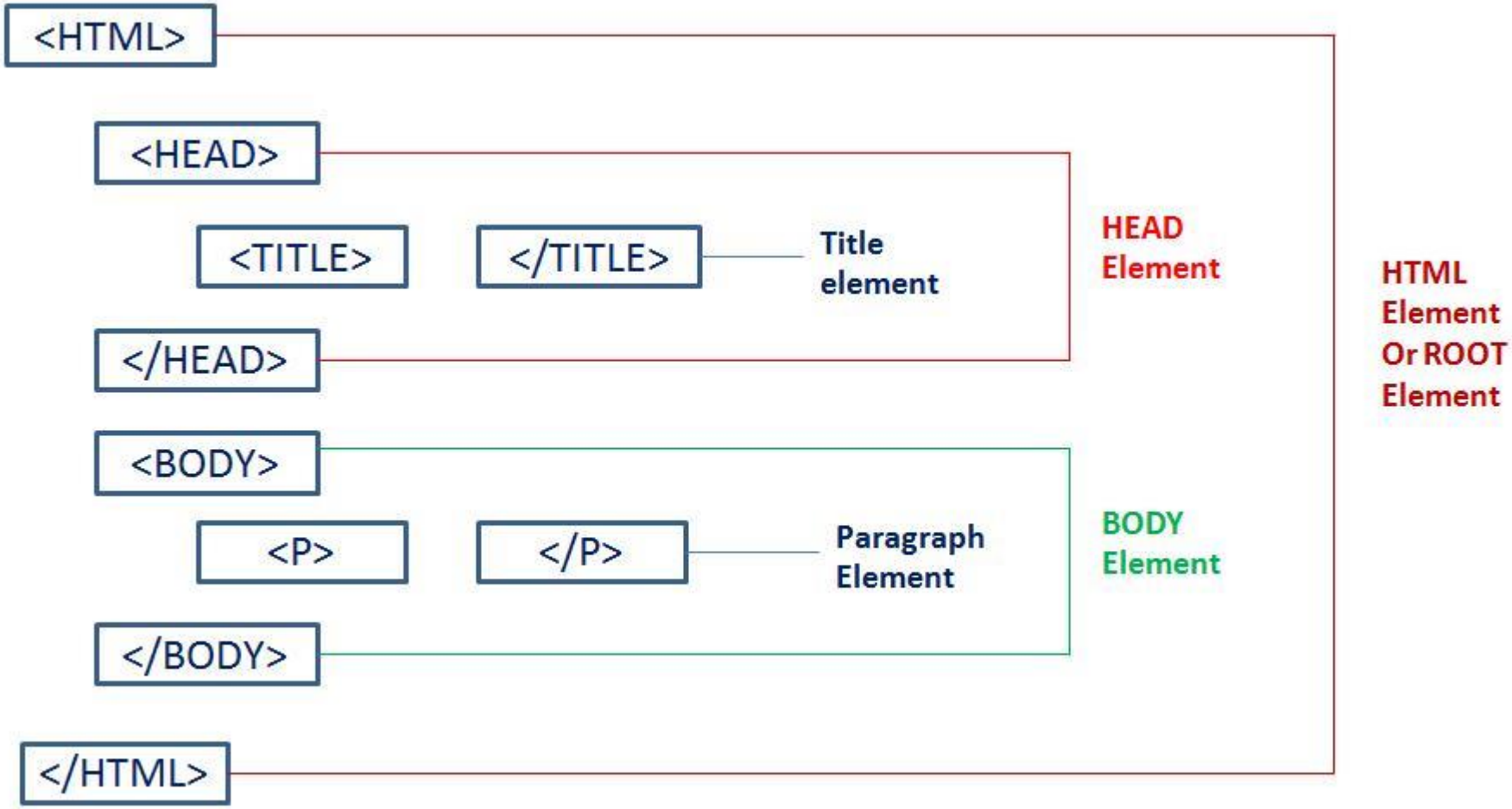
HTML Tags

- The left and right angle brackets are used to enclose all special instructions, called tags.
- Two classes of tags:
  - Those which appear in pairs:

     Italic text
  - Those which appear individually.

    <img src= “imagefile.jpg”>

Structure of HTML



Important Points

- Most of the tags belong to the first category.

  <tag-name>……..</tag-name>

- Tags are not case sensitive

  <HEAD>, <Head> and <head> are all equivalent.

- Tags may be nested

  <html><head>…</head><body>…</body></html>

- Most browsers have a VIEW SOURCE menu option.
- The HTML version of the page can be displayed.

Browsers features

- Browsers ignore all extra spaces and carriage returns within an HTML document.
- Browsers have the following feature:
  - A reformat of the document to fit in the current display area. (resizing of the window)
- It is good practice to use white spaces in an HTML document.
  - Improves readability.

Attributes of HTML elements

- Some tags can have one or more (named) attributes to define some additional characteristics of the tag.
- Defines the behavior of that element.
- Attributes should always be applied with a start tag.
- The Attribute should always be applied with its name and value pair.

```
<img   src="image.jpg">

<body text="#FFFFFF"  bgcolor="#0000FF">

<body text="white" bgcolor="blue">
```

Contd…

- Unrecognized tags
  - Browsers normally ignore tags it does not recognize(spelling mistakes).
- Comment lines
  - Comments are included between

    <!--- and --->
  - Comments cannot be nested

    <!--- Comment here --->

    <!--- Multiline comments

    two line comments--->

HTML document structure

- An HTML document consists of two major portions:
- Head
  - Contains information about the document, like the title metadata describing the contents.
- Body
  - Contains the actual matter of the document.
- Gets displayed within the browser window.

Simple HTML document

```
<html>          } optional
   <head>
      <title> Title of the documents</title>
   </head>
    <body text="blue" bgcolor="black"> Content of the document.
     This is an <i> italic word</i> style
      </body>
</html>
```

Comparison between head, body tags

1	Header tag information is rendered not displayed on the web page.	Body tag information is displayed on the web page.
2	Javascript in header tag is processed before loading or rendering of page.	Javascript in the body tag is executed as and when it loads the body tag contents

Grouping Content

Grouping Content

- The and elements allow you to group together several elements to create sections or subsections of a page.

in-line element	Block level element
It does not cause a line break	By default, creates a line break
can be used within tags	It is used to make separate containers or boxes within a page
  This is crazy 	

Structural HTML Tags

- <html>…..</html>
  - Used to bracket an entire HTML document.
  - Optional for most browsers.
  - Attributes:
    - lang=language code
    - Language code indicates the primary language of the document.
    - example: hi-Hindi, en-English
- Example:

```
<html lang="en">
...
</html>
```

Contd…

- <head>….</head>
  - Used to provide information about a web page.
  - Nests within itself other tags like
  - <base>, <meta> , <title>.
- <base>
  - Attribute: href=url
  - Specifies the base pathname for all relative URLs in the document
  - No end tag.

Contd.

- <meta> tag describe metadata within an HTML document
- Metadata will not be displayed on the page but is machine parable.

```
<head>
   <meta name="description" content="Free Web tutorials">
   <meta name="keywords" content="HTML,CSS,XML,JavaScript">
   <meta name="author" content="Name of the author">
</head>
```

- <title>….</title>

Specifies the title of a HTML document.

Usually appears on the title bar of the browser window.

Contd…

- <body>….</body>
  - Actual content is to be displayed on the page is placed between body tag
- Attributes:
  - background=url: specifies an image file as background.
  - bgcolor=color: Sets the background color of the document.
  - text=color: Sets the default color for the normal text in the document.

Contd…

- link=color: Sets the default color for all the links in a document.
- alink=color: Sets the color of active links.
- vlink=color: Sets the color for the visited links.

Example:

```
<body alink="green">
<p><a href="http://www.sdmcet.ac.in">sdmcet</a></p>
<p><a href="http://www.google.co.in">google</a></p>
</body>
```

Specifying colors

- Two ways:
  - By specifying the red, green, blue or RGB components.
- Example:
  - <body text="#FFFFFF" bgcolor="#0000FF">

Color Name	HEX
Blue	0000FF
BlueViolet	8A2BE2
Chocolate	D2691E
DarkKhaki	BDB76B

HTML Color Coding Methods

- Three different methods to set colors on your web page:
  1. Color names – You can specify color names directly like green, blue or red.
  2. Hex codes – A six-digit code representing the amount of red, green, and blue that makes up the color.
  3. Color decimal or percentage values – This value is specified using the rgb( ) property.

HTML color coding

- HTML Colors - Color Names
  - W3C has listed 16 basic color names that will
- validate with an HTML validator
  - Major browsers supports 200 different colors
  - W3C Standard 16 Colors names

	Black		Gray		Silver		White
	Yellow		Lime		Aqua		Fuchsia
	Red		Green		Blue		Purple
	Maroon		Olive		Navy		Teal

Contd.

1. By specifying the color name.
   - Some of the colors that are supported by browsers are:

     aqua black blue gray green lime

     navy olive purple red ….etc.

- Many other colors are possible.
- Example:

```
<body text=white>
 <body bgcolor="yellow">
```

HTML color coding

2. HTML Colors - Hex Codes

- A hexadecimal is a 6 digit representation of a color
- Format:

1	2	3	4	5	6
Red(RR)		Green(GG)		Blue(BB)	

hexadecimal code will be preceded by a pound or hash sign #

Color	Color HEX
	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

Hex code 000000 to FFFFFF

000000	000033	000066	000099	0000CC	0000FF
003300	003333	003366	003399	0033CC	0033FF
006600	006633	006666	006699	0066CC	0066FF
009900	009933	009966	009999	0099CC	0099FF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF
330000	330033	330066	330099	3300CC	3300FF
333300	333333	333366	333399	3333CC	3333FF
336600	336633	336666	336699	3366CC	3366FF
339900	339933	339966	339999	3399CC	3399FF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF
660000	660033	660066	660099	6600CC	6600FF
663300	663333	663366	663399	6633CC	6633FF
666600	666633	666666	666699	6666CC	6666FF
669900	669933	669966	669999	6699CC	6699FF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF
990000	990033	990066	990099	9900CC	9900FF
993300	993333	993366	993399	9933CC	9933FF
996600	996633	996666	996699	9966CC	9966FF

Hex code oooooo to FFFFFF

999900	999933	999966	999999	9999CC	9999FF
99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF

HTML Colors - RGB Values

- Color value is specified using the rgb( ) property
- Property reads three values: Red, Green & Blue
- The value can be an integer between 0 and 255 or a percentage.
- All the browsers does not support rgb() property of color so it is recommended not to use it.

Color	Color RGB
	rgb(0,0,0)
	rgb(255,0,0)
	rgb(0,255,0)
	rgb(0,0,255)
	rgb(255,255,0)
	rgb(0,255,255)
	rgb(255,0,255)
	rgb(192,192,192)
	rgb(255,255,255)

Text formatting in HTML

- <Hn> …..</Hn>
  - Used to generate headings, $1 \leq n \leq 6$.
  - Six different levels of headings.
  - <H1> is the largest, <H6> is the smallest.
-
  - Paragraph marker, used to separate text into paragraphs.
  - End tag is optional.
  - A series of paragraphs tags …. with no intervening text is treated as a single

Contd…

-
  - Used to indicate that the text following <br> should begin on the next line.
  - The amount of line spacing is less than that of <p> break.
- Example:
  - This is the first line.</br>
  - This is the second line.</br>
  - This is the third line.

Contd.

- <HR>
  - Produces a horizontal line, which can be used to delimit sections.
  - Length of the line can be specified.
- Example:

```
<hr size="20">
<hr width="75%"> 75% of window size
<hr align="right" width="120">
```

- Across full width of browser, 20 pixels thick, 75% of available page width, and 120 pixels right-justfied.

Contd.

- <address>…..</address>
  - Supplies the contact information of the user/person/author……
  - Generally formatted in italics.
- Example:

```
<address>
        SDM Engineering College, Dharwad Karnataka India
</address>
```

Appearance of Text

- ….
  - Displays the enclosed text in bold.
- ……
  - Displays the enclosed text in italics.
- <Cite>….</Cite>
  - Tells the browser that this is a citations.
  - Usually displayed in italics.

Contd.

- _{….}
  - Displays the enclosed text as a subscript.
- ^{….}
  - Displays the enclosed text as superscript.
- ….
  - specifies the style of the enclosed text.
- Attributes:

  color=color name

  face= typeface

  size=value [1 to 7; 3 is the default]

  [can also specify as +n or -n]

Contd.

- …..
  - Centers the enclosed elements horizontally on the page.
- …. [if align attribute is used then closing tag is required.]
  - Used to align a paragraph
  - Option can be left, right or center.

Lists

- There are a number of tags for building lists.
  - Serves the purpose of improving the readability of the text.
- Depending on the way the list items are displayed, lists may be divided into three types:
  1. Numbered list
  2. Unnumbered list
  3. Definition list

Numbered List

- Numbered or order lists are used when the sequence of the item is important.
- Specified using the tag:

  <OL>…….</OL>

- The individual items in the list are specified using the <LI> tag.
- Example:

```
<OL>
    <LI> First item of the list
    <LI> Second item of the list
    <LI> Third item of the list
</OL>
```

Contd.

- The list items are numbered sequentially as 1,2,3……
- Attributes:

  type=1 | A |a |I | I
- Specifies the style of numbering.

  1,2,3……. Or A,B,C……or a,b,c or I,II,III……or i,ii,iii,……

Example

```
<html>
    <head>
        <title>Bulleted Lists </title>
    </head>
    <body text=white bgcolor=blue>
     <h2> Departments</h2>
     <h3>
        <OL>
            <LI> CSE
            <LI> ISE
            <LI> E&C
        </OL>
     </h3>
    </body>
</html>
```

Unnumbered List

- Used to display a set of related items that appear in no particular order.
  - Each item is indented with a preceding bullet symbol.
- Specified using the tag:

  <UL>…..</UL>
- The individual items in the list are specified using the <LI> tag.
- Attribute:

  type=disc|circle|square

Contd.

- The <LI> items can contain multiple paragraphs, specified using <P>.
- Example:

```
<UL>
    <LI> First item of the list
    <LI> Second item of the list
    <LI> Third item of the list
</UL>
```

Example:

```
<html>
    <head>
        <title>Bulleted Lists </title>
    </head>
    <body text=white bgcolor=blue>
     <h2> Departments</h2>
     <h3>
        <UL>
            <LI> CSE
            <LI> ISE
            <LI> E&C
        </UL>
     </h3>
    </body>
</html>
```

Example:

```
<html>
    <head>
        <title>Bulleted Lists </title>
    </head>
    <body text=white bgcolor=blue>
     <h2> Departments</h2>
     <h3>
        <UL type = square>
            <LI> CSE
            <LI type= circle> ISE
            <LI> E&C
        </UL>
     </h3>
    </body>
</html>
```

Example: Numbered and Unnumbered

```
<html>
    <head>
        <title>Bulleted Lists </title>
     </head>
     <body text=white bgcolor=blue>
      <h2> Programming language hierarchy</h2>
      <h3>
          <OL type = A>
              <LI> C
              <LI type= circle> C++
              <LI> Java
          </OL>
      </h3>
     </body>
</html>
```

Definition List

- Specified using the tag:

  <DL>…….</DL>

- Each definition comprises of a definition term (<DT>) and a definition description(<DD>).
- Web browsers format each definition on a new line and indent it.

Contd…

- Example:

```
<DL>
 <DT>TCP
   <DD> Transmission Control Protocol
 <DT> UDP
    <DD> User Datagram Protocol
 <DT> IP
      <DD> Internet Protocol
</DL>
```

Example Definition list

```
<html>
   <head>
       <title> Bulleted Lists </title>
    </head>
    <body text=white bgcolor=blue>
     <h2> Programming language hierarchy</h2>
     <h3>
        <DL>
           <DT>TCP
             <DD> Transmission Control Protocol
            <DT> UDP
              <DD> User Datagram Protocol
            <DT> IP
                 <DD> Internet Protocol
         </DL>

     </h3>
    </body>
</html>
```

Nesting of Lists

- Any list can be nested within another list:
  - When unnumbered lists are nested, the browser automatically uses a different bullet symbol for each level.
  - When numbered lists are nested, by default, every level is numbered using the Arabic numerals(1,2,3,…….).

Example:

```
<html>
   <head>
     <title> Listing Nesting </title>
   </head>
    <body text=white bgcolor=gray>
     <H3> Programming languages :
        <UL>
           <LI> Java <BR>
              <UL>
                 <LI> object-oriented
                  <LI> Platform independent
                   <LI>can be embedded within HTML
               </UL><BR>
<LI> Perl <BR>
                    <UL>
                        <LI> a scripting language
                         <LI> Powerful string handling capabilities
                         <LI> widely used for writing CGI scripts
                      </UL>
                  </UL></H3>
              </body>
</html>
```

Example:

```
<html>
   <head>
     <title> Listing Nesting </title>
   </head>
    <body text=white bgcolor=gray>
     <H3> Programming languages :
        <OL>
           <LI> Java <BR>
              <UL>
                 <LI> object-oriented
                  <LI> Platform independent
                   <LI>can be embedded within HTML
               </UL><BR>
                <LI> Perl <BR>
<OL>
                    <LI> a scripting language
                     <LI> Powerful string handling capabilities
                     <LI> widely used for writing CGI scripts
                  </OL>
             </OL></H3>
          </body>
</html>
```

Hyperlinks

- The power of HTML comes from its ability to link text and/or image to another document or section of a document.
- These links are called hyperlinks.
  - Browser by default highlights the hyperlinks with color and/or underline.

How to specify hyperlinks

- Specified using the anchor tag:

<A>……..</A>

→ Requires an attribute called "HREF" which specifies the path of the resource to be linked.

- Example:

```
<A HREF="image.jpg">portrait</A>
<a href=www.bing.com>search </a>
<a href="mailto:xyz@abc.com"> Mail me</a>
<a href=slides/web.pdf>download pdf</a>
```

Relative Vs Absolute URLs

- Relative URL
  - Provides a link to another document relative to the location of the current document.
  - Commonly used when referring to documents residing on the same website.
- Example:
  - ProgramsList
  - Cricket
- These kind of links are called relative links.

Contd.

- Absolute URL
  - The complete path name of the document in the server is specified.
  - Necessary when linking to documents on other servers.
- Example:

VIII sem syllabus doc

Linking to specific section

- Anchors can be used to go to a particular section in a document.
  - Within the same (or different) document.
- Two steps involved:

1. Define a named section in a document.

```
<a name="Programming"><h2>Cprogramming<h2></a>
```

2. Provide a link to the named section

```
<a href="#Programming">Functions chapter</a>
```

```
<a href="http://www.Programming.in/CProgramming.html"> Programming fundamental concepts </a>
```

Hyperlinks (Example)

```
<html>
    <head>
        <title>Link to  other sites</title>
     <body text=white bgcolor=gray link=red vlink=green> Favorite Search
Engines:
          <ol>
              <li> <a href=http://www.google.com> Google</a>
               <li> <a href="http://www.bing.com/"> Bing</a>
               <li> <a href="http://www.yahoo.com"> Yahoo </a>
           </ol>
       </body>
   </html>
```

Inline Images and other Documents

- Supported image formats:
  - JPEG, GIF, PNG
- Specified using the standalone tag:

  <IMG>
- Attributes of <IMG>
  - SRC: specifies the URL of the image file
  - HEIGHT: height (in pixels) to be set aside for the image.
  - WIDTH: width (in pixels) to be set aside for the image.

Contd.

- The HEIGHT and WIDTH attributes tells the browser to set aside appropriate space(in pixels) for the image as it downloads the rest of the file.
  - Some browsers stretch or shrink an image to fit into the allotted space.
- Example:

```
<IMG SRC=3dimage.gif>
<IMG src=3dimage.gif height=80 width=150>
```

Example

```
<html>
    <head>
      <title> An Image </title>
    </head>
    <body>
     <img src=3dimage.gif align=bottom width=48 height=48> In the text<p> An image
      <img src=3dimage2.gif align=middle width=48 height=48> In the text<p> An image
       <img src=3dimage3.gif align=top width=48 height=48> In the text<p> An image
      </body
</html>
```

Example

```
<html>
    <body>
       Click on the image below to go to google.
         <p>
          <a href="http://www.google.com"> <img align=middle width=90 height=90
   src=3dimage2.gif> </a>
     </body>
</html>
```

HTML FORMS

Introduction

- Standard way to communicate the information from a Web browser to the server is through a *form*.
- Modeled on the *paper forms* that people frequently are required to fill out.
- Forms are represented on web pages using HTML.
- A form basically contains web components.
  - Real-life examples:
    - search engines
    - On-line purchase of items
    - Registration
- The form allows a user to fill up the blank entries and send it back to the owner of the page.
  - Called submitting the form.

Form Example

FORM Tags

- Several tags are used in connection with forms:
  - <form>……..</form>
  - <input>
  - <textarea>…….</textarea>
  - <select>……..</select>

Contd.

- <FORM>……</FORM>
  - HTML forms contain form elements.
  - Form elements are different types of input elements, checkboxes, radio buttons, submit buttons and more.
- Attributes:
  - METHOD
  - ACTION

Contd.

- Method:
  - Indicates how the information in the form will be sent to the web server when the form is submitted.
- Two possible values:
  - POST: causes a form's contents to be parsed one element at a time.
  - GET: concatenates all field names and values in a single large string.
- POST is the preferred method because of string size limitations in most systems.

HTTP methods

Method Name	Description
GET	Retrieves data from a web server by specifying parameters in the URL portion of the request.
POST	A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.
PUT	Replaces all current representations of the target resource with the uploaded content.
DELETE	Removes all current representations of the target resource given by a URI.

Contd.

- ACTION
  - Specifies the URL of a program on the origin server that will be receiving the form's inputs.
  - Traditionally called Common Gateway Interface (CGI)
  - The specified program is executed on the server, when the form is submitted.
    - Output sent back to the browser.

Syntax

- <form method="POST" action="cgi-bin/myprog.pl">

  </form>

INPUT Tag

- This tag defines a basic form element.
- Attributes:
  - Type
  - Name
  - Size
  - Maxlength
  - Value
  - Src
  - align

Contd.

- Type:
  - Defines the kind of element that is to be displayed in the form.
    - TEXT: defines a text box, which provides a single line area for entering text.
    - RADIO: radio button, used when a choice must be made among several alternatives (clicking on one of the buttons turns off all other in the same group)
    - CHECKBOX: similar to the radio buttons, but each box here can be selected independently of the others.

Contd.

- password: similar to a text box, but characters are not shown as they are typed.
- submit: creates a box labeled Submit; if clicked, the form data are passed on to the designated CGI script.
- reset: Creates a button labeled Reset; if clicked, clears a form's contents.

Contd.

- Name:
  - Specifies a name for the input field.
  - The input-handling program(CGI) in reality receives a number of (name, value) pairs.
- Size:
  - Defines the number of characters that can be displayed in a text box without scrolling.
- Maxlength:
  - Defines the maximum number of characters a text box can contain.

Contd.

- Value:
  - Used to submit a default value for a text.
  - Can also be used for specifying the label of a button(renaming “submit”)
- Src:
  - Provides a pointer to an image file.
  - Used for clickable maps.
- align:
  - Used for aligning image types:
  - align=top|middle|bottom

Contd.

- <textarea>……..…..</textarea>
  - Can be used to accommodate multiple text lines in a box.
- Attribute:
  - Name: name of the field.
  - Rows: number of lines of text that can fit into the box.
  - Cols: width of the text area on the screen

Select tag

- <select>……..</select>
  - Used along with the tag <option>
  - Used to define a selectable list of elements.
  - The list appears as a scrollable menu or a pop-up menu (depending on the browser).
- Attributes:
  - Name: Name of the field
  - Size: specifies the number of option elements that will be displayed at a time on the menu.(If the actual number exceeds the size, a scrollbar will appear).

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>HTML select Tag</title>
    </head>
    <body>
        <form>
            <select name="dropdown">
            <option value="Data Structures" selected>Data Structures</option>
            <option value="Data Mining">Data Mining</option>
            </select>
        <input type="submit" value="Submit" />
        </form>
    </body>
</html>
```

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>HTML select Tag</title>
    </head>
    <body>
        <form>
            <select name="dropdown">
            <option value="Data Structures" selected>Data Structures</option>
            <option value="Data Mining">Data Mining</option>
            </select>
        <input type="submit" value="Submit" />
        </form>
    </body>
</html>
```

HTML Table

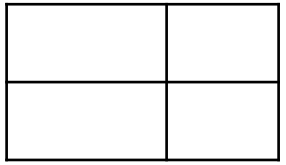
Introduction

- Tables are made up of cells, arranged into rows.
- Usage tags <TABLE>,<TR>,<TD>.
- Example:

```
<table>
    <tr>
        <td>web1.0</td><td>Web2.0</td>
    </tr>
    <tr>
        <td>web3.0</td><td>Web4.0</td>
    </tr>
</table>
```

The Table Tags

- <TABLE>……<TABLE>
  - Defines the beginning and the end of a table.
  - Can have several attributes:
    - bgcolor=#rrggbb or color name
    - rules=all | cols | rows | none
    - border = number
    - height=number, percentage



Contd.

- <TR>……</TR>
  - Defines a row of cells within a table.
  - Can have several attributes:
    - bgcolor=#rrggbb or color name
  - align= left | center | right | justify
- <CAPTION>……</CAPTION>
  - Provides a caption for the table.
  - Must immediately follow the “table” tag, and precede all other tags.

Contd.

- <TD>……..……..</TD>
  - Defines a table data cell
  - A table cell can contain any content. Like an image, or even another table
  - Can have several attributes:
  - bgcolor=#rrggbb or color name
  - Colspan=number
- Specifies the number of columns the current cell should span(default is 1).
  - Rowspan=number

Contd.

- <TH>……..</TH>
  - Defines a table header cell
  - Browsers generally display the contents of the header cells in bold, and centered.
  - Same attributes as the <TD> tag.

Example

```
<table>
   <tr>
      <td colspan=2> First Row , First cell</td>
    </tr>
    <tr>
      <td> Second Row, second cell</td>
      <td> Third Row, Third cell</td>
    </tr>
</table>
```

Example

```
<table border =1>
    <caption> My Table </caption>
    <tr>
        <th>Name</th>
        <th>Marks</th>
    </tr>
</table>
```

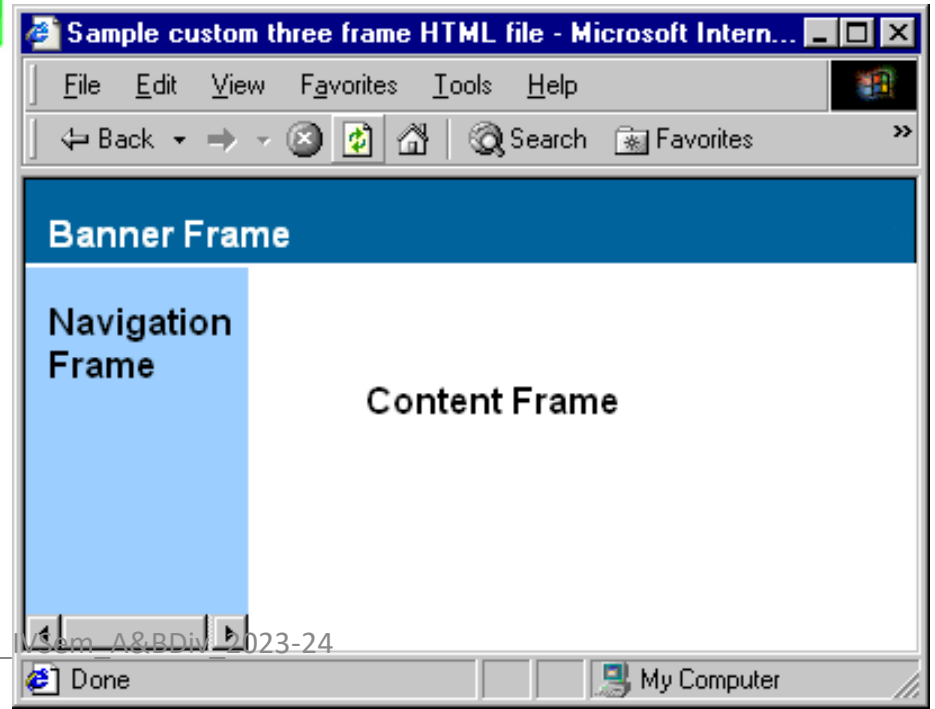
```
<tr>
        <th>xyz</th>
        <th>35</th>
    </tr>
     <tr>
        <th>PQR</th>
        <th>75</th>
    </tr>
```

HTML FRAMES

Introduction

- What are frames?
  - A method for dividing the browser window into smaller sub-windows.
  - Each sub-window displays a different HTML document
- How does a page with a frame look like?

A web page that uses Frames



The Frame Tags

- <frameset>……..</frameset>
  - Defines a collection of frames or other framesets.
  - Attributes:
  - cols=list of lengths(number, %, or *)
  - rows=list of lengths (number ,%, or *)
- Establishes the number and sizes of columns(vertical frames) or rows (horizontal frames) in a frameset.
- In a number of pixels, percentage of total area, or relative values(*) based on available space.

Contd.

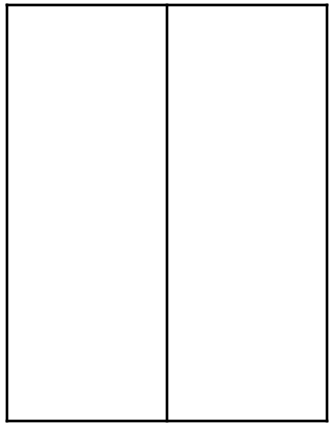
- <iframe>
  - An IFrame (Inline Frame) is an HTML document embedded inside another HTML document on a website. The IFrame HTML element is often used to insert content from another source, such as an advertisement, into a Web page
- Attribute:
  - frameborder=1 |0
  - src=url
  - scrolling=yes|no|auto
  - marginwidth=number
  - marginheight=number
  - name=text

Contd.

- <NOFRAMES>……..</NOFRAMES>
  - The HTML <noframes> tag is used to handle the browsers which do not support the <frame> tag. This tag is used to display alternate text messages.
- Specifies the contents to be displayed by browsers that cannot display frames.
  - Ignored browsers that support frames.

Example

```
<html>
  <head>
    <title> Document with frames</title>
   </head>
   <body>
    <frameset cols="*">
          <iframe src="leftpage.htm"></iframe>
           <iframe src="rightpage.htm"></iframe>
     </frameset>
      <noframes>
            Browser does not support frames.
      </noframes>
    </body>
</html>
```

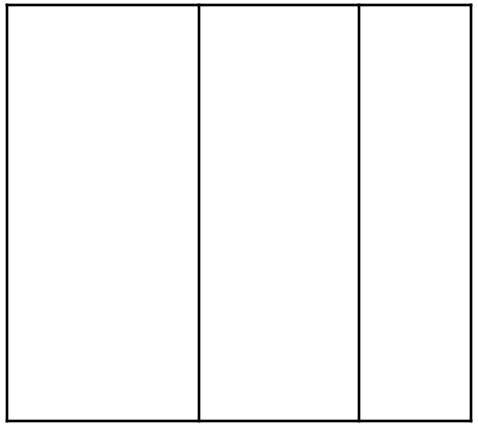


Example

```
<html>
<body>
<frameset cols=“*”>
  <iframe width="100%" height="300px" src="Webpage_OnFrame.htm"
  name="iframe_a"></iframe>
</frameset>
<p><a href="http://www.sdmcet.ac.in" target="iframe_a">SDM College link</a></p>
<p>When the target of a link matches the name of an iframe, the link will open in the
  iframe.</p>
</body>
</html>
```

Example

```
<html>
  <head>
    <title> Document with frames</title>
   </head>
   <body>
    <frameset cols="*">
         <iframe src="leftpage.htm">
          <iframe src="rightpage.htm">
     </frameset>
      <noframes>
           Browser does not support frames.
      </noframes>
    </body>
</html>
```



Linking to a Framed document

- When a hyperlink is clicked, by default, the new page is loaded into the same frame.
- We can load the linked page into a new frame also.

M **E** **N** **U**	**On clicking menu related content is displayed in this frame.**

Contd.

- Assign a name to the targeted frame.
  - <frame src="somepage.html" name="abc">
- Specify this frame in a hyperlink as follows:
  - …

- The document newpage.html will load into the frame name "abc".

Disadvantages of Frames

- Smaller devices cannot cope with frames often because their screen is not big enough to be divided up.
- Your page will be displayed differently on different computers due to different screen resolution.
- The browser's back button might not work as the user hopes.
- There are still few browsers that do not support frame technology.
- Not suitable for search engine.
- Difficulty in finding URL.
- Back Button issue.
- Reduced amount of usable web page space.
- Printing issue

Frames Vs iFrames

Feature	Frame	IFrame
Syntax	<frame src="URL">	<iframe src="URL">
Purpose	Divide a web page into sections with separate HTML documents.	Embed external content within a specific area of a web page.
Compatibility	Deprecated in HTML 4.01, no longer supported in HTML5	Supported in HTML 4.01 and HTML5
Document Isolation	Each frame can have its own URL and separate HTML content.	Content within an iframe is typically hosted on a different domain from the parent page.
Security	Less secure than iframes, as frames can access the parent document	More secure than frames, as iframes are sandboxed by default
Usability	Can be difficult to use and maintain, as frames can be nested	More user-friendly than frames, as iframes are easier to manage
SEO	Frames can impact SEO, as search engines may not crawl the frames within a page	iframes do not impact SEO, as search engines crawl the content within the iframe

Few HTML5 standard elements

- HTML5 introduces a number of new elements and attributes that helps in building modern websites

1. Semantic Elements – These are like <header>, <footer>, and <section>.

2. Forms 2.0 – Improvements to HTML web forms where new attributes have been introduced for <input> tag.

3. Canvas – This supports a two-dimensional drawing surface that you can program with JavaScript.

4. Audio & Video – You can embed audio or video on your web pages without resorting to third-party plug-ins.

HTML Entities

- An HTML entity is a representation of 'string' that begins with an ampersand (&) and ends with a semicolon (;)
- Helps in representing reserved characters on web pages.
- Used to differentiate from HTML code.
- Browsers will always truncate spaces in HTML pages.
- Example:
  - If you leave 10 spaces in your text, the browser will remove 9 of them.
- To add real spaces to your text, you can use the " " character entity

HTML Entities

Symbol	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
₹	Rupees	&#8377;	&#x20B9
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
¼	One-quarter	&frac14;	&#xBC;
½	One-half	&frac12;	&#xBD;

HTML DOCTYPE

- It is a Document type declaration
- It is an instruction to the web browser about what version of HTML the page is written in.

```
<!DOCTYPE html>
```


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN“ "http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

HTML DOCTYPE

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">

HTML 1.0

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN" "http://w3.org/TR/xhtml-basic/xhtml-basic10.dtd">

HTML, XHTML, HTML5 Comparisons

HTML	XHTML
Uppercase or lowercase tags	Lowercase tags only
All elements don't have to be closed	All tags have to be closed
Empty elements don't require closing	Empty elements must be closed
Does not require DOCTYPE declaring	Requires DOCTYPE declaration
Attribute values do not require quotes	Attribute values require quotes
HTML	**HTML5**
Understood by all browsers	Not fully functional with all browsers
Not originally built to handle video	Built for audio & video
Does not support local offline storage	Supports local offline storage
Not built for today's internet needs	Supports content specific tags
Does not support creation of a canvas for drawing	Supports creation of a canvas for drawing

Introduction to XML

Introduction

- eXtensible Markup Language
- Developed from SGML
- A *meta-markup* language
- Deficiencies of HTML and SGML
  - Many complex features that are rarely used
- HTML is a markup language, XML is used to define markup languages
- Markup languages defined in XML are known as *applications*
- XML can be written by hand or generated by computer
  - Useful for data exchange

Contd…

- XML is designed to store and transport data.
- XML became a W3C Recommendation on February 10, 1998.
- XML is not a replacement for HTML.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. Users can define their tags.
- XML is platform-independent and language-independent.
- The main thing that makes XML truly powerful is its international acceptance
- Many organizations use XML interfaces for databases, programming, office application mobile phones, and more.
- It is due to its platform-independent feature.

HTML Vs XML

	HTML	XML
Definition	Markup language for displaying web pages in a web browser. Designed to display data with focus on how the data looks	Markup language defines a set of rules for encoding documents that can be read by both humans and machines. Designed with focus on storing and transporting data.
Date when invented	1990	1996
Extended from	SGML	SGML
Type	Static	Dynamic
Usage	Display a web page	Transport data between the application and the database. To develop other mark up languages.
Processing/Rules	No strict rules. Browser will still generate data to the best of its ability	Strict rules must be followed or processor will terminate processing the file
Language type	Presentation	Neither presentation, nor programming
Tags	Predefined	Custom tags can be created by the author
White Space	Cannot preserve white space	Preserves white space

How can XML be used?

- XML can keep data separated from your HTML
- XML can be used to store data inside HTML documents
- XML can be used as a format to exchange information
- XML can be used to store data in files or in databases
- XML was designed to describe data.
- XML is a software- and hardware-independent tool for
  - Carrying information.

Features of XML

- XML separates data from HTML
  - xml file can be used for storing data
- XML simplifies data sharing
  - XML data is stored in plain text format. This provides a software- and hardware-independent way of storing data.
- XML simplifies data transport
  - One of the most time-consuming challenges for developers is to exchange data between incompatible systems over the Internet.
  - Exchanging data as XML greatly reduces this complexity, since the data can be read by different incompatible applications.
- XML increases data availability
  - Different applications can access your data, not only in HTML pages, but also from XML data sources.
- XML can be used to create new internet languages
  - WSDL, RDF and OWL

The Syntax of XML

- Levels of syntax
  - *Well-formed documents* conform to basic XML rules.
  - *Valid documents* are well-formed and also conform to a *schema* that defines details of the allowed content.
- Well-formed XML documents
  - All begin tags have a matching end tag
    - Empty tags
  - If a begin tag is inside an element, the matching end tag is also
  - There is one *root* tag that contains all the other tags in a document
  - Attributes must have a value assigned, the value must be quoted
  - The characters <, >, & can only appear with their special meaning
  - http://www.w3.org/TR/2006/REC-xml-20060816/#sec-well-formed is the official definition

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<sdm>
    <dept>
        <cse id="cs">
            <course>
                <ug name="BE">
                    <faculty id="ugfaculty">
                        <name>xyz</name>
                        <qualification>abc</qualification>
                        <yearofexperience>2</yearofexperience>
                        <areaofinterest>asd</areaofinterest>
                    </faculty>
                </ug>
                <pg name="pg">
                    <faculty id="pgfaculty">
                        <name>qwe</name>
                        <qualification>tyu</qualification>
                        <yearofexperience>4</yearofexperience>
                        <areaofinterest>jkl</areaofinterest>
                    </faculty>
                </pg>
            </course>
        </cse>
        <ISE>
            <course>
                <ug></ug>
                <pg></pg>
            </course>
        </ISE>
    </dept>
</sdm>
```

XML Comments

- XML comments are just like HTML comments.
  - Syntax:-

    <!-- Write your comment-->

Example:-

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--Students marks details-->
<students>
    <student>
        <name>abc</name>
        <marks>35</marks>
    </student>
    <student>
        <name>xyz</name>
        <marks>60</marks>
    </student>
</students>
```

Introduction to DTD

XML Validation

- A well formed XML document can be validated against DTD or Schema
- A well-formed XML document is an XML document with correct syntax
- Checking Validation:
  - An XML document is called "well-formed" if it contains the correct syntax.
  - A well-formed and valid XML document is one which have been validated against Schema.

Well formed XML

- It must begin with the XML declaration.
- It must have one unique root element.
- All start tags of XML documents must match end tags.
- XML tags are case-sensitive.
- All elements must be closed.
- All elements must be properly nested.
- All attribute values must be quoted.
- XML entities must be used for special characters.

DTD

- DTDs check the validity of the structure and vocabulary of an XML document against the grammatical rules of the appropriate XML language.
- A DTD defines the legal elements of an XML document
- DTD grammars are a set of rules that define:
  - A set of *elements* (tags) and their *attributes* that can be used to create an XML document.
  - *how* these elements can be *combined/embedded*
  - DTDs can't define element content types.
  - It has one data type i.e., (#PCDATA)
- DTD defines the document structure with a list of legal elements and attributes.
- DTD and XML schema both are used to form a well formed XML document.

Internal/External DTD

- DTD declarations either
  - Internal XML document
  - External DTD file, after linked to a XML document.
- Internal DTD:
  - DTD rules are written within xml document using declaration.
  - Scope of this DTD within this document.
- External DTD:
  - Rules in a separate file (with .dtd extension).
  - DTD file is linked to a XML document.

Rules to define ELEMENTS

- Cardinality operators

"*"	May occur 0 or more times
"+ "	May occur one or more times
"? "	May occur zero times or once
No marking	One time

Example: Internal DTD

```
<?xml version="1.0"?>
  <!DOCTYPE employee [
        <!ELEMENT employee (name, designation, email)>
    <!ATTLIST employee
        id CDATA  #REQUIRED>
        <!ELEMENT name (#PCDATA)>
        <!ELEMENT designation (#PCDATA)>
    <!ATTLIST designation
        discipline CDATA  #IMPLIED>
        <!ELEMENT email (#PCDATA)>
  ]>

<employee id="1">
    <name>David</name>
    <designation discipline="Web developer" >Team Lead</designation>
    <email>David@hotmail.com</email>
</employee>
```

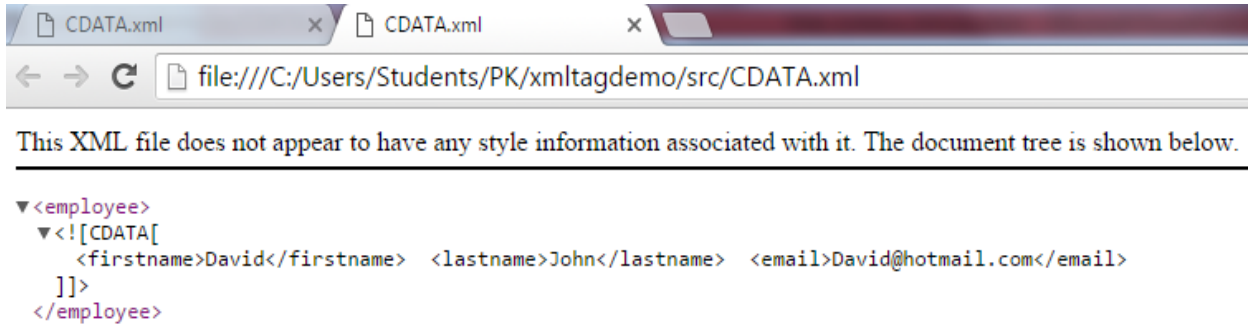
employee.xml

CDATA: Unparsed Character data

→ CDATA contains the text which is not parsed further in an XML document.

→ Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
<![CDATA[
  <firstname>David</firstname>
  <lastname>John</lastname>
  <email>David@hotmail.com</email>
]]>
</employee>
```



Result:

PCDATA: Parsed Character Data

- XML parsers are used to parse all the text in an XML document.
- PCDATA stands for Parsed Character data.
- PCDATA is the text that will be parsed by a parser.
- Tags inside the PCDATA will be treated as markup and entities will be expanded.

External DTD

- External DTDs are shared between multiple XML documents.
- Any changes are updated in the DTD document, which reflects in all XML documents.
- Two types of External DTD:
  1. Private DTD
  2. Public DTD
  - Private DTD: Private DTD identified by SYSTEM keyword. Access for single or group of users.

    Syntax: <!DOCTYPE root_element SYSTEM "dtd_file_location">

Contd…

- **Public DTD**: Public DTD identify by the PUBLIC keyword. Any user can access
  - Syntax:

    <!DOCTYPE root_element PUBLIC "dtd_name" "dtd_file_location">
- Example:
  - <!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

Example: validating xml against DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT employee (firstname,lastname,email)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

employee.dtd

```
<?xml version="1.0"?>
<!DOCTYPE employee SYSTEM "employee.dtd">
<employee>
  <firstname>david</firstname>
  <lastname>John</lastname>
  <email>david@hotmail.com</email>
</employee>
```

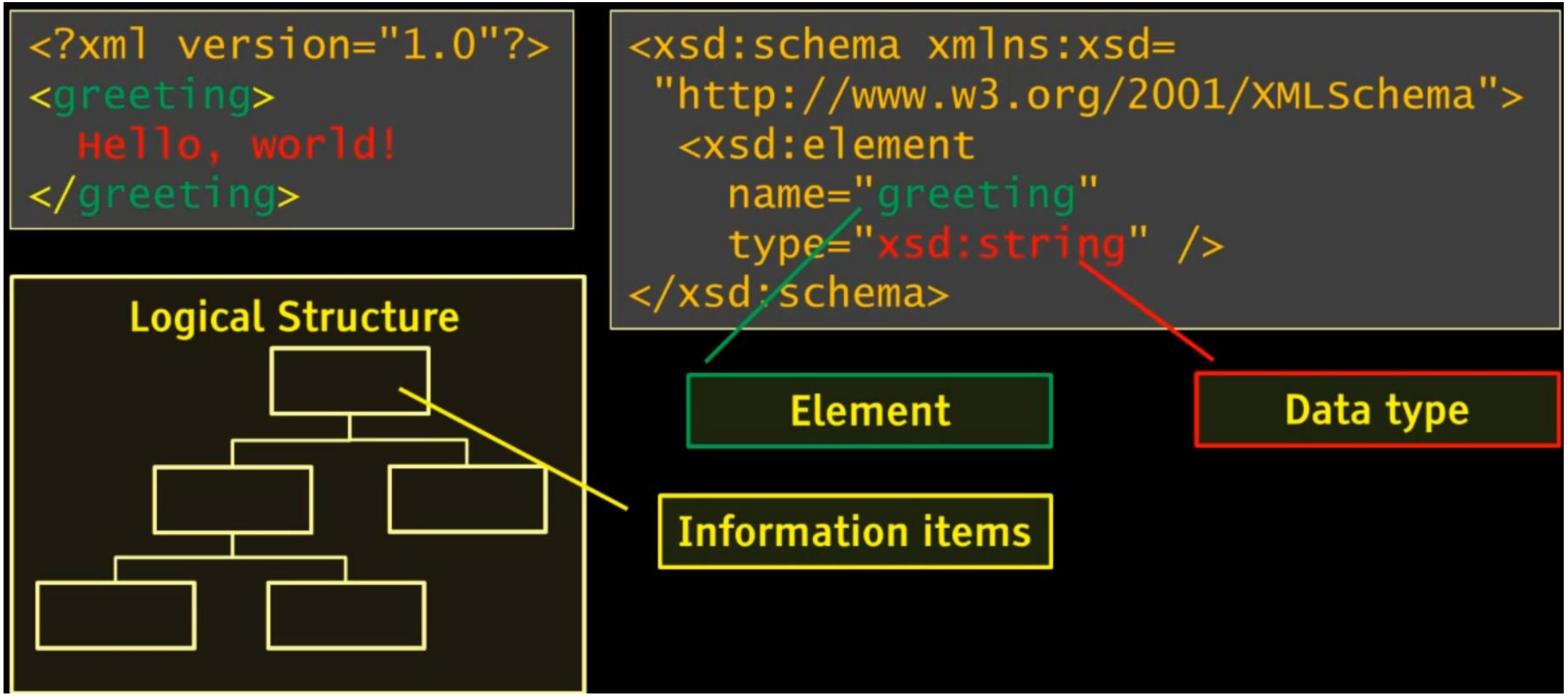
employee.xml

Introduction to XSD

Introduction

- An XML Schema describes the structure of an XML document.
- XML standards are defined by XML Schemas.
- XML Schema can be manipulated with XML DOM.
- Reuse your Schema in other Schemas.
- Create your own data types derived from the standard types.

Introduction



XML schema

- An XSD defines the following rules:
  - The structure of XML

    Ex:- Which elements in which order

    How many times

    With which attributes

    How they are nested
  - XML file is relatively free set of elements and attributes without an XSD
  - XSD check the validity of XML structure.
  - Uses namespaces to allow for reuses of existing definitions
  - It supports a large number of built-in data types and definition of derived data types

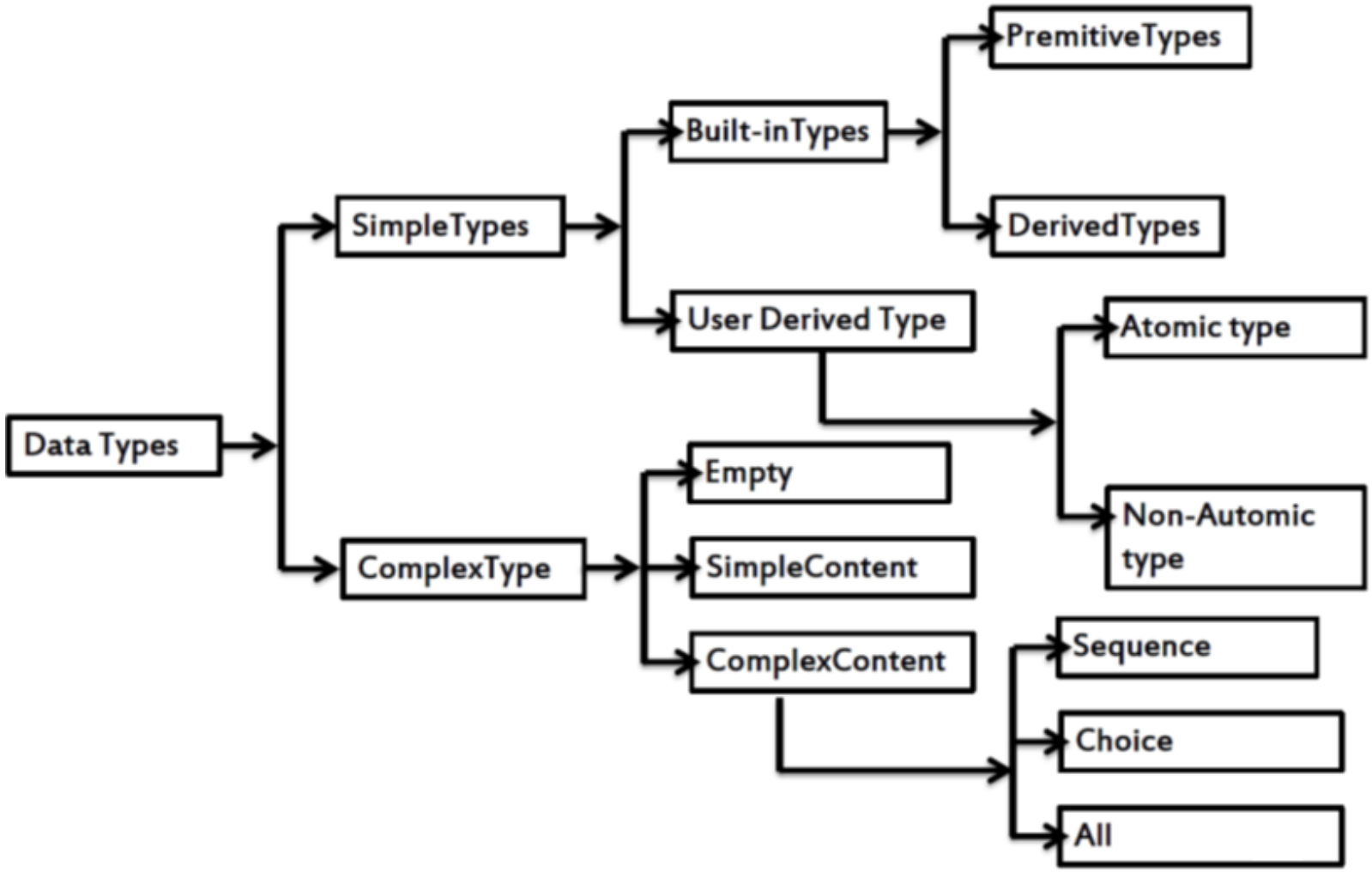
Why XML Schema?

- XML Schema is an XML-based and more powerful tool to validate and define the XML structure.
- XML Schemas uses XML Syntax.
- XML Schemas are extensible because they are written in XML format.
- Reuse your Schema in other Schemas
- XML Schema supports data types.
- Validate the correctness of data
- Ensures a mutual understanding of the content.
- It is an alternative to DTD.

DTD Vs. XSD

- XSD uses XML namespaces whereas DTD does not support XML namespace.
- No limitations in XSD for defining cardinality ratio on any given element.
- XSD supports many data types, whereas DTD has limited number of data types.
- XSD supports user defined data types, whereas DTD doesn't support.

XSD: Data Types Web



XSD: Data Types

- Primitive Types(19):

String, Boolean, Decimal, float, Double, Duration, DateTime, Time ,Date, GYearMonth, GYear, GDate, GDay, GMonth, hexbinary, base64Binary, anyURL, QName, NOTATION

- Derived Types(25):

NormalizedString, token, language, NMTOKEN, NMTOKENS, Name, NCName, ID, IDREF, IDREFS, ENTITY, ENTITIES, integer, nonPositiveInteger, negativeInteger, long, int, short, byte, nonNegativeInteger, unsignedLong, unsignedInt, unsignedShort, unsignedInt, unsignedShort, unsignedByte, positiveInteger.

XML Schema syntax

```
<schema id=ID attributeFormDefault = qualified|unqualified
elementFormDefault = qualified|unqualified
blockDefault = (#all|list of (extension|restriction|substitution))
finalDefault= (#all|list of (extension|restriction|list|union))
targetNamespace = anyURI version = token xmlns = anyURI any attributes >

((include|import|redefine|annotation)*,(((simpleType|complexType|
group|attributeGroup)|element|attribute|notation),annotation*)*)

</schema>
```

Attribute	Description
id	Optional. Specifies a unique ID for the element
attributeFormDefault	Optional. The form for attributes declared in the target namespace of this schema. The value must be "qualified" or "unqualified". Default is "unqualified". "unqualified" indicates that attributes from the target namespace are not required to be qualified with the namespace prefix. "qualified" indicates that attributes from the target namespace must be qualified with the namespace prefix
elementFormDefault	Optional. The form for elements declared in the target namespace of this schema. The value must be "qualified" or "unqualified". Default is "unqualified". "unqualified" indicates that elements from the target namespace are not required to be qualified with the namespace prefix. "qualified" indicates that elements from the target namespace must be qualified with the namespace prefix
blockDefault	•Optional. Specifies the default value of the block attribute on element and complexType elements in the target namespace. The block attribute prevents a complex type (or element) that has a specified type of derivation from being used in place of this complex type. This value can contain #all or a list that is a subset of extension, restriction, or substitution:extension - prevents complex types derived by extension •restriction - prevents complex types derived by restriction •substitution - prevents substitution of elements •#all - prevents all derived complex types
finalDefault	•Optional. Specifies the default value of the final attribute on element, simpleType, and complexType elements in the target namespace. The final attribute prevents a specified type of derivation of an element, simpleType, or complexType element. For element and complexType elements, this value can contain #all or a list that is a subset of extension or restriction. For simpleType elements, this value can additionally contain list and union:extension - prevents derivation by extension •restriction - prevents derivation by restriction •list - prevents derivation by list •union - prevents derivation by union •#all - prevents all derivation
targetNamespace	Optional. A URI reference of the namespace of this schema
version	Optional. Specifies the version of the schema
xmlns	A URI reference that specifies one or more namespaces for use in this schema. If no prefix is assigned, the schema components of the namespace can be used with unqualified references
anyattributes	Optional. Specifies any other attributes with non-schema namespace

XML Namespaces

- XML namespaces are used to group all the elements as a single unit.
- Elements are accessed with fully qualified name by specifying the namespace and element name.
- Namespaces are created in XSD using "**targetNamespace**" attribute in **schema** element.

XML Namespaces

A qualified element is one that has been associated with a namespace

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.example.org/Sample"
    xmlns:tns="http://www.example.org/Sample" elementFormDefault="qualified">
    Add Simple | Complex Type elements
</schema>
```

All schema elements are referred using this namespace

All user defined elements are referred using this namespace

All user defined elements are accessed using short name **tns** instead of using namespace and elements defined by user in schema tag.

XML Namespaces

- XML Namespaces provide a method to avoid element name conflicts.
- Example:

```
<table>
  <tr>
    <td>Web</td>
    <td>Java</td>
  </tr>
</table>
```

```
<table>
  <name>sdm</name>
  <width>8000</width>
  <length>12042</length>
</table>
```

Resolving name conflicts using prefix

```
<h:table>
  <h:tr>
    <h:td>Web</h:td>
    <h:td>Java</h:td>
  </h:tr>
</h:table>
```

```
<f:table>
  <f:name>sdm</f:name>
  <f:width>8000</f:width>
  <f:length>12042</f:length>
</f:table>
```

XML Namespaces: Attribute

- Namespace can be defined by an xmlns attribute in the start tag of an element.
- Syntax: xmlns: prefix="*URI*"
- A Uniform Resource Identifier (URI) is a string of characters which identifies an Internet Resource.

Contd…

```
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Web</h:td>
    <h:td>Java</h:td>
  </h:tr>
</h:table>

<f:table xmlns:f="http://www.w3schools.com/furniture">
  <f:name>sdm</f:name>
  <f:width>8000</f:width>
  <f:length>12042</f:length>
</f:table>
```

- Name conflicts in XML can easily be avoided using a name prefix.
- In the above xml there is no name conflict as each element has a different name.

- XML validating open source tool: **https://www.freeformatter.com/xml-formatter.html**
- XML Schema Validator : **https://www.liquid-technologies.com/online-xsd-validator**

XSD: Attribute

- Simple elements cannot have attributes.
- If an element has attributes, it is considered to be of a complex type. But the attribute itself is always declared as a simple type.
- Example:
  - <College cname="SDMCET">SDM college</College>

XSD:

  - <ed: **attribute** name="cname" **type**="ed: string"/>

XSD: Attribute values

- **Attributes** may have a **default value** OR a **fixed value** specified.
- **Default value** – If attribute does not contain any value, default value will be assigned to the attribute.
- Example:
  - <ed: **attribute** **name**="cname" **type**="ed: string“ **default** = “sdm”/>
- **Fixed value** –Attribute value cannot be changed once it is assigned.
- Example:
  - <ed: **attribute** **name**="cname" **type**="ed: string“ **fixed** = “SDMCET”/>

XML Namespaces: Attribute

→ Namespace can be defined by an **xmlns** attribute in the start tag of an element.

→ Syntax: xmlns: prefix="*URI*"

→ A **Uniform Resource Identifier** (URI) is a string of characters which identifies an Internet Resource.

XSD: Elements

schema	Defines the root element of a schema
complexType	Defines a complex type element
simpleType	Specifies the constraints and information about the values of attributes or text-only elements
sequence	Specifies that the child elements must appear in a sequence. Each child element can occur from 0 to any number of times.

Free Online Tools For Developers

I created this website to help developers by providing them with free online tools. These tools include several formatters, validators, code minifiers, string escapers, encoders and decoders, message digesters, web resources and more.

I will add new tools on a regular basis, so be sure to add this site to your bookmarks.

If you encounter a bug, I would very much appreciate that you send me an email (freeformatter@gmail.com) that explains the nature of your bug. Please include details like which browser version you're using and the steps to reproduce the bug. Other comments are welcome.

-MrForms

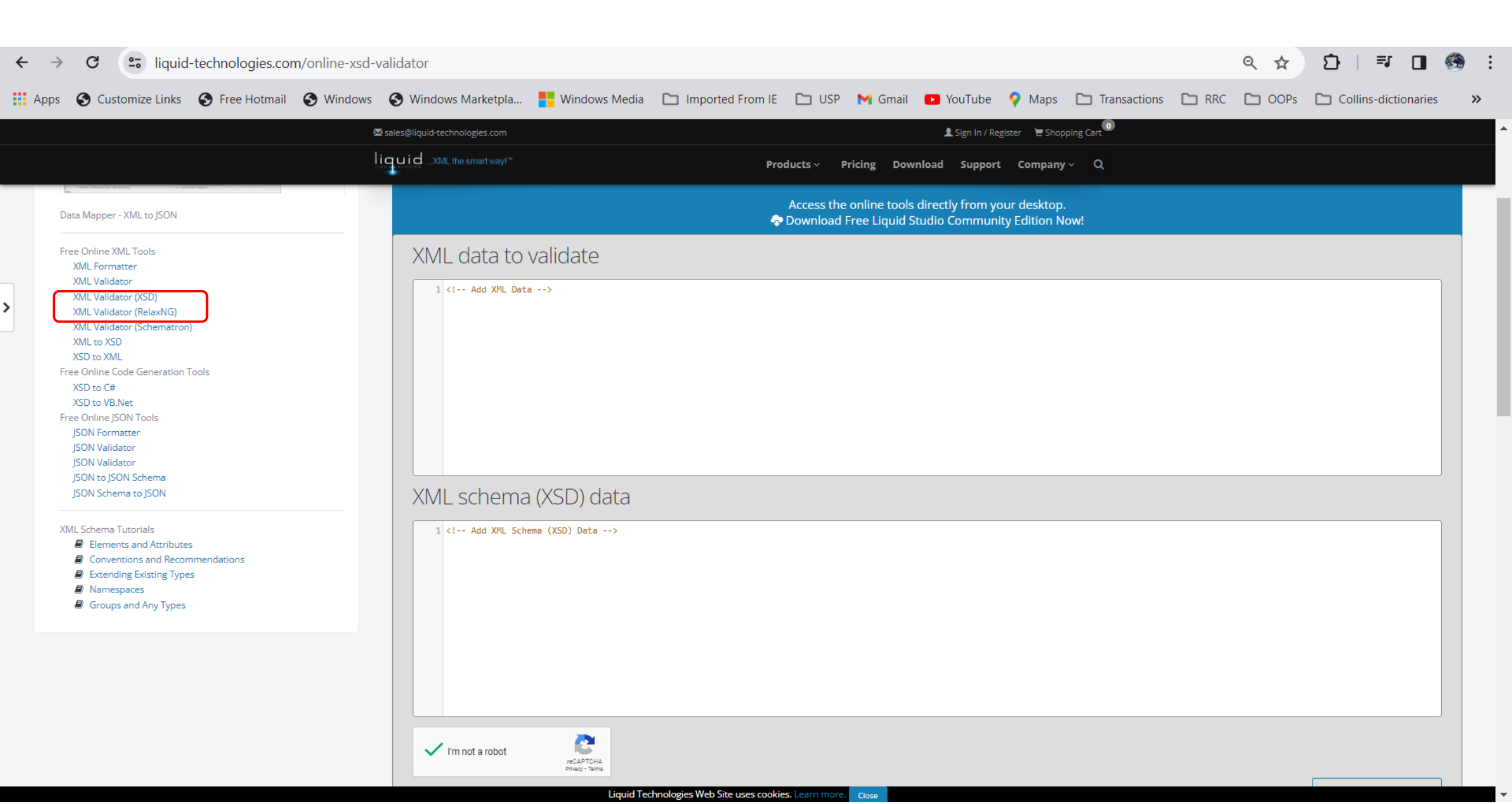
-List of tools-

JSON Formatter / Beautifier

Formats a JSON string/file with your desired indentation level creating an object tree with color highlights. You can now clearly identify object constructs (objects, arrays and members). The JSON tree that is created can be navigated by collapsing the individual nodes one at a time if desired.

- Formats your JSON string/file with choice 6 indentation levels: 2 spaces, 3 spaces, 4 spaces, compact mode, JavaScript escaped and tab separated
- Creates a tree representation of the JSON objects for easy navigation
- Color highlights the different construct of your JSON objects
- Supports copy-paste or file upload

HTML Formatter / Beautifier


liquid-technologies.com/online-xsd-validator
Apps
Customize Links
Free Hotmail
Windows
Windows Marketpla...
Windows Media
Imported From IE
USP
Gmail
YouTube
Maps
Transactions
RRC
OOPs
Collins-dictionaries
sales@liquid-technologies.com
Sign In / Register
Shopping Cart
liquid XML the smart way!™
Products
Pricing
Download
Support
Company
Data Mapper - XML to JSON
Free Online XML Tools
XML Formatter
XML Validator
XML Validator (XSD)
XML Validator (RelaxNG)
XML Validator (Schematron)
XML to XSD
XSD to XML
Free Online Code Generation Tools
XSD to C#
XSD to VB.Net
Free Online JSON Tools
JSON Formatter
JSON Validator
JSON Validator
JSON to JSON Schema
JSON Schema to JSON
XML Schema Tutorials
Elements and Attributes
Conventions and Recommendations
Extending Existing Types
Namespaces
Groups and Any Types
Access the online tools directly from your desktop.
Download Free Liquid Studio Community Edition Now!
XML data to validate
1 <!-- Add XML Data -->
XML schema (XSD) data
1 <!-- Add XML Schema (XSD) Data -->
I'm not a robot
reCAPTCHA
Privacy - Terms
Liquid Technologies Web Site uses cookies. Learn more.
Close