- Introduction on Internet
- History
- Web architechture
- Protocol
- Language
- HTML, Javascript, React

History of Internet

What is the "internet"?

- The Internet is a global information network that connects millions of computers It is growing exponentially and provides a unique information resource that is global, diverse and current.

What do we use the internet for?

- Browsing information
- Social media and creating awareness
- For entertainment (music, videos, movies)
- Communicating with people all over the world.
- Sharing information
- E-commerce and banking

Timeline of the Internet

1965 — Lawrence Roberts and Thomas Merril communicate with one another using computers connected via a low-speed dial-up telephone line in Massachusetts and California, creating the first Wide Area Network and laying the groundwork for the internet.

1969 — ARPAnet, the first version of the internet, is created and used to link computers at UCLA and Stanford.

1974 — Robert Kahn and Vinto Cerf publish "A protocol for Packet Network Communication" laying the groundwork for TCP/IP and much larger, interconnected computer networks. The word "internet"is first used.

1986 — NSFNET is founded, creating the backbone and providing the investment needed to create the internet as we know it today.

1989 — The first commercial ISP, the world launches.

1989 — Tim Berners-Lee invents the world wide web, which opens the door for the internet to go mainstream.

1993 — The first web browser available to the public, Mosaic launches.

1994 — Netscape navigator is released in stores.

2000 — Some 300 million people around the world are officially online.

2004 — Facebook launches and the Web 2.0 begins to grow rapidly.

2007 — The iPhone is released, giving rise to the mobile revolution.

2010 — 1.966 billion people are online worldwide.

2022 — 5 billion people are online

Internet in the 1950's

- The Internet has no single "inventor." Instead, it has evolved over time.
- The Internet got its start in the United States more than 50 years ago as a government weapon in the Cold War.

Contd.

- Scientists and researchers used it to communicate and share data with one another through what was known as DARPNET(Defense's Advanced Research Projects Agency Network).
- This was in response to the Soviet Union bombings and a need to create a communication network which was free from interference and disruption.

Internet in the 1960's

- In 1962, a scientist from M.I.T. and DARPA named J.C.R. Licklider proposed : a "galactic network" of computers that could talk to one another. Such a network would enable government leaders to communicate even if the Soviets destroyed the telephone system.
- In 1965, another M.I.T. scientist developed a way of sending information from one computer to another that he called "packet switching." Packet switching breaks data down into blocks, or packets, before sending it to its destination.

Contd.

- Without packet switching, the government's computer network—now known as the ARPANET—would have been just as vulnerable to enemy attacks as the phone system.
- In 1969, ARPANET delivered its first message: a "node-to-node" communication from one computer to another but it crashed the full network. The internet was yet to be born

Internet in the 1970's

- By the end of 1969, just four computers were connected to the ARPANET, but the network grew steadily during the 1970s.
- As packet-switched computer networks multiplied, however, it became more difficult for them to integrate into a single worldwide "Internet."

Contd.

- By the end of the 1970s, a computer scientist named Vinton Cerf had begun to solve this problem by developing a way for all of the computers on all of the world's mini-networks to communicate with one another.
- He called his invention "Transmission Control Protocol," or TCP. (Later, he added an additional protocol, known as "Internet Protocol." The acronym we use to refer to these today is TCP/IP.)

Contd.

- TCP/IP was described to be the "handshake" between computers all over the world. It enabled each computer to have its own identity.

Internet in the 1980's

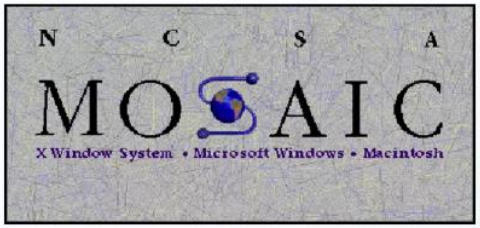
- Cerf's protocol transformed the Internet into a worldwide network. Throughout the 1980s, researchers and scientists used it to send files and data from one computer to another.
- However, this network was still between scientists and researchers from different universities and labs.

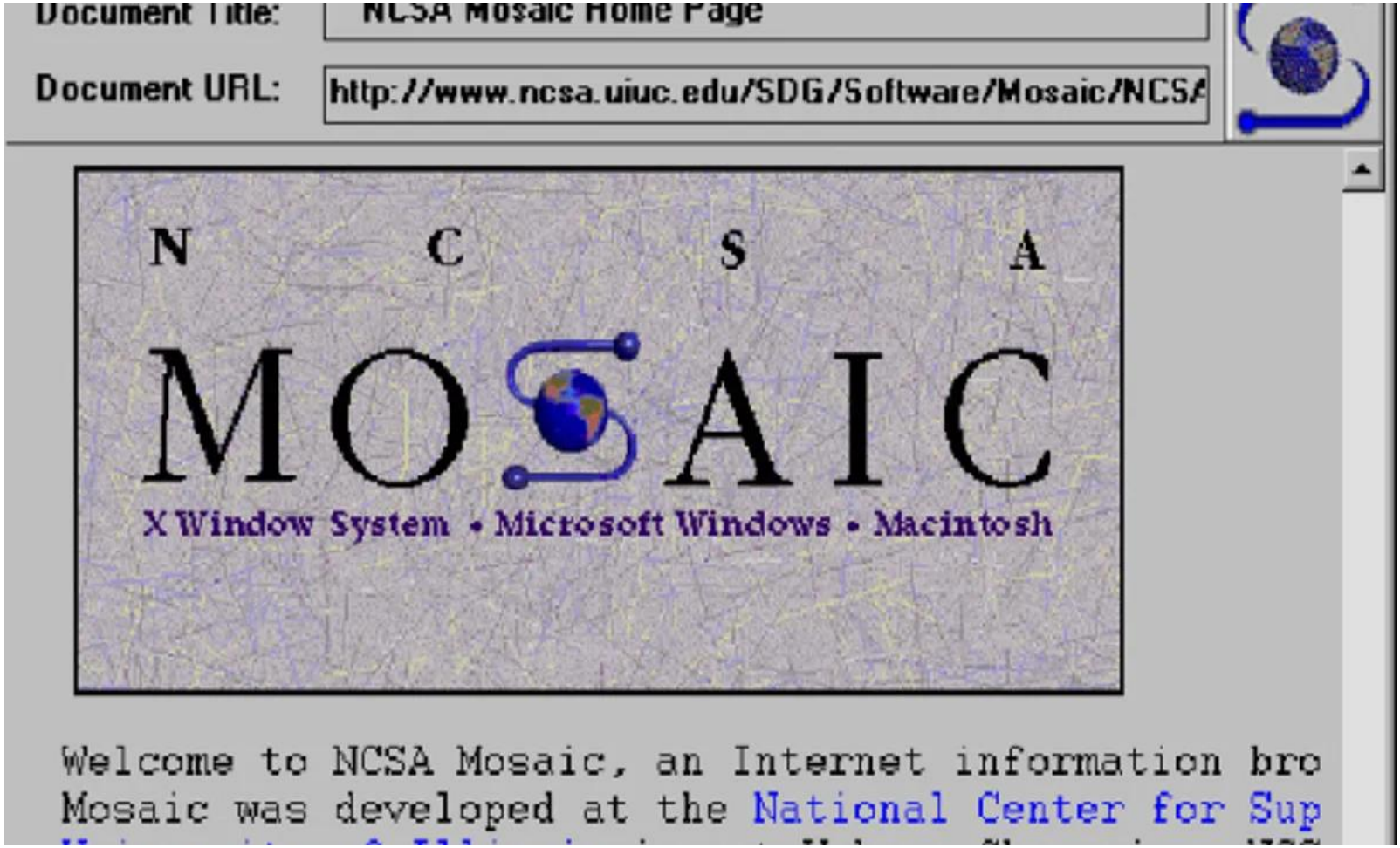
Internet in the 1990's

- However, in 1991 the Internet changed again.
- Tim Berners-Lee introduced the World Wide Web: an Internet that was not simply a way to send files from one place to another but was itself a "web" of information that anyone on the Internet could retrieve.
- Berners-Lee created the first browser and the Internet that we know today.

Contd.

- In 1992, a group of students and researchers at the University of Illinois developed a sophisticated browser that they called Mosaic. (It later became Netscape.)
- Mosaic offered a user-friendly way to search the Web: It allowed users to see words and pictures on the same page for the first time and to navigate using scrollbars and clickable links.



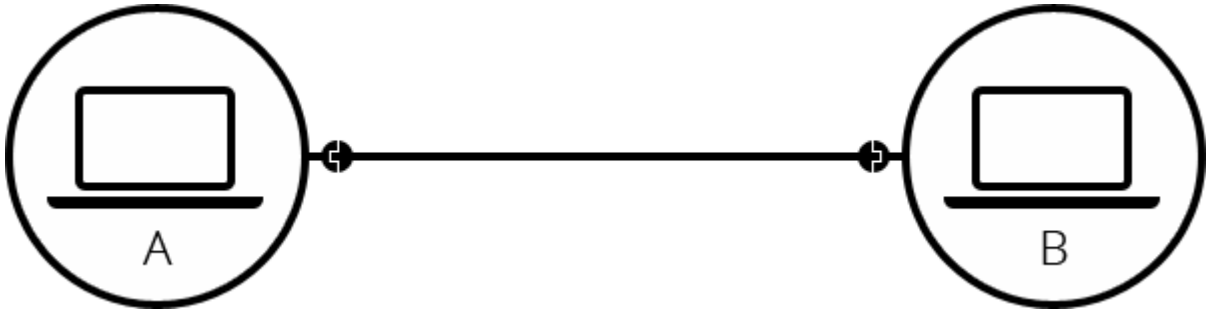

Document Title:
NCSA Mosaic Home Page
Document URL:
http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSA
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
Welcome to NCSA Mosaic, an Internet information bro
Mosaic was developed at the National Center for Sup

Contd.

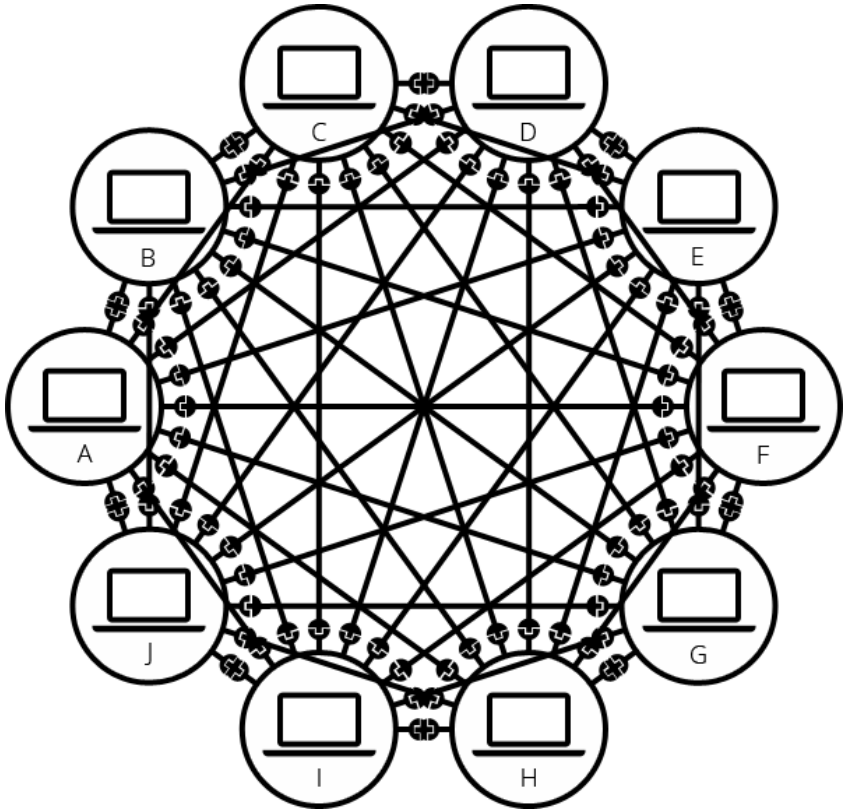
- That same year, Congress decided that the Web could be used for commercial purposes. As a result, companies of all kinds hurried to set up websites of their own, and e-commerce entrepreneurs began to use the Internet to sell goods directly to customers.
- More recently, social networking sites like Facebook have become a popular way for people of all ages to stay connected.

How Internet Works

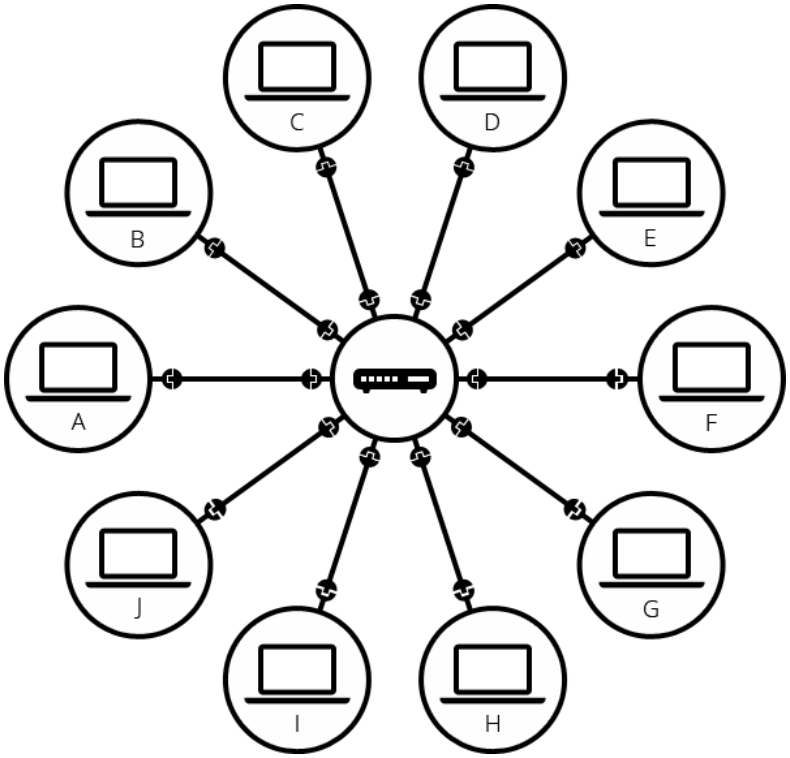
- **A simple network**
- When two computers need to communicate, you have to link them, either physically (usually with an Ethernet cable) or wirelessly
- (for example with Wi-Fi or Bluetooth systems).
- All modern computers can sustain any of those connections.



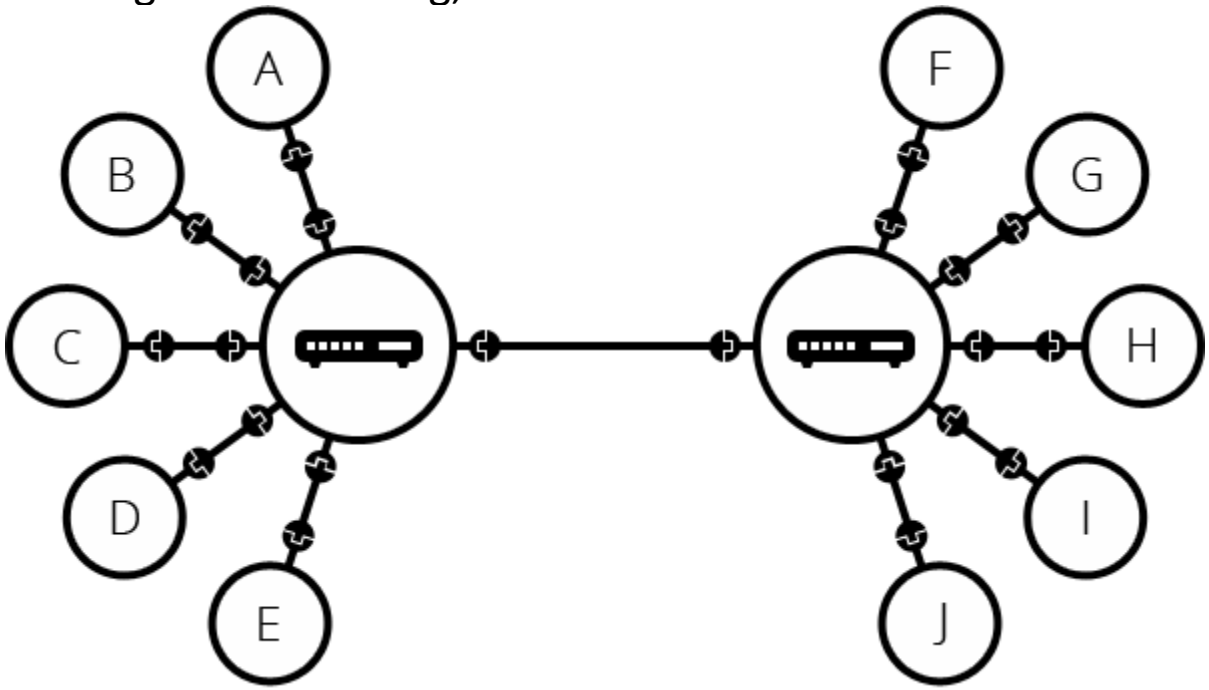
Such a network is not limited to two computers.
You can connect as many computers as you wish.
But it gets complicated quickly. If you're trying to connect, **say, ten computers, you need 45** cables, with nine plugs per computer!



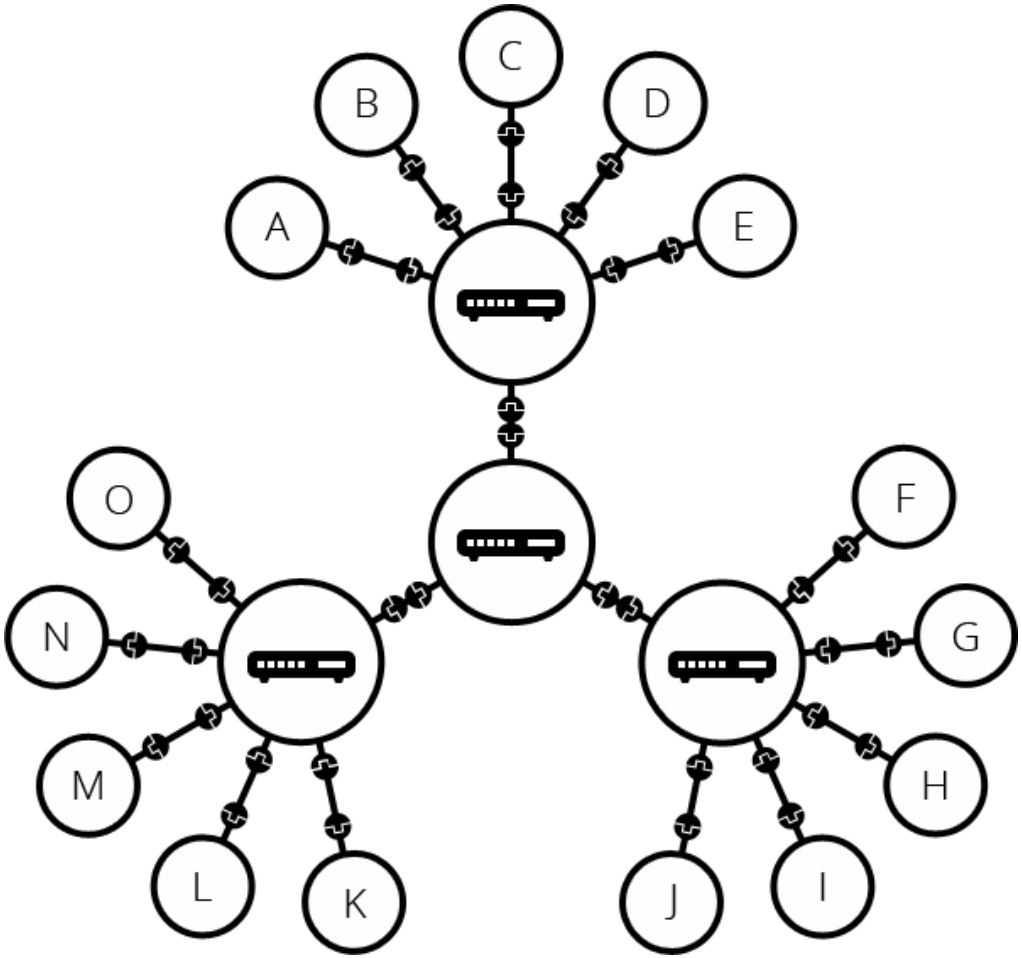
- To solve this problem, each computer on a network is connected to a special tiny computer called a *router*.
- This *router* has only one job: like a signaler at a railway station, it makes sure that a message sent from a given computer arrives at the right destination computer.
- To send a message to computer B, computer A must send the message to the router, which in turn forwards the message to computer B and makes sure the message is not delivered to computer C.



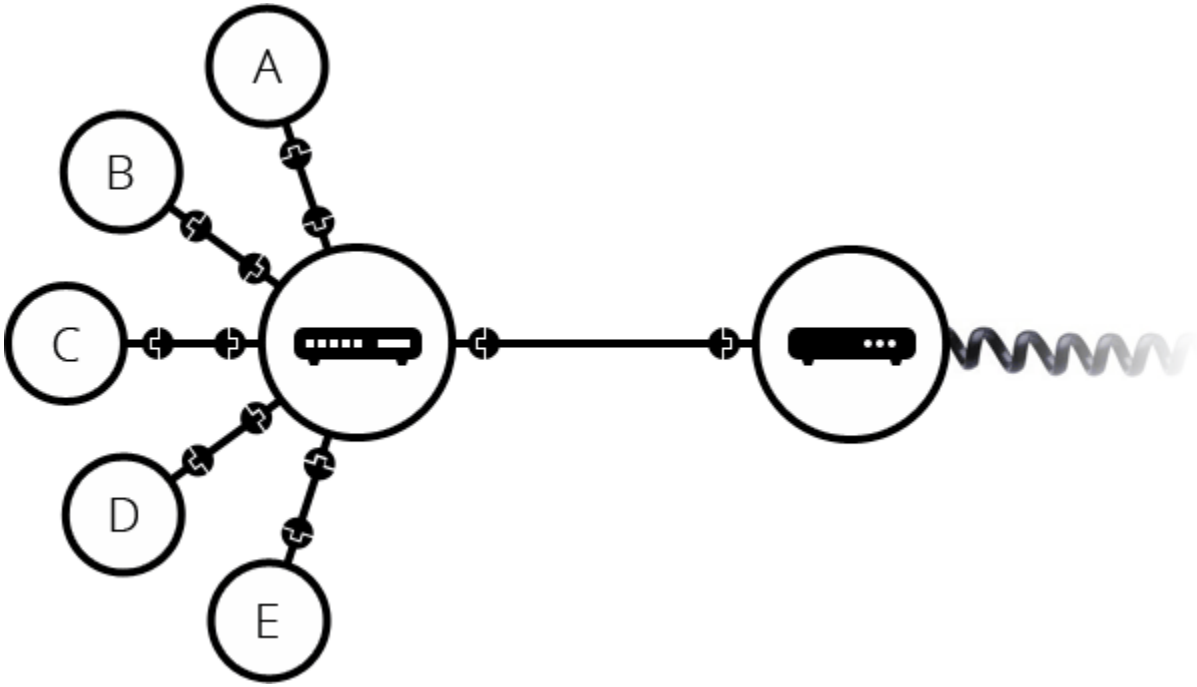
- **A network of networks**
- So far so good. But what about connecting hundreds, thousands, billions of computers?
- Of course a single *router* can't scale that far, but, if you read carefully, we said that a *router* is a computer like any other, so what keeps us from connecting two *routers* together? Nothing, so let's do that.



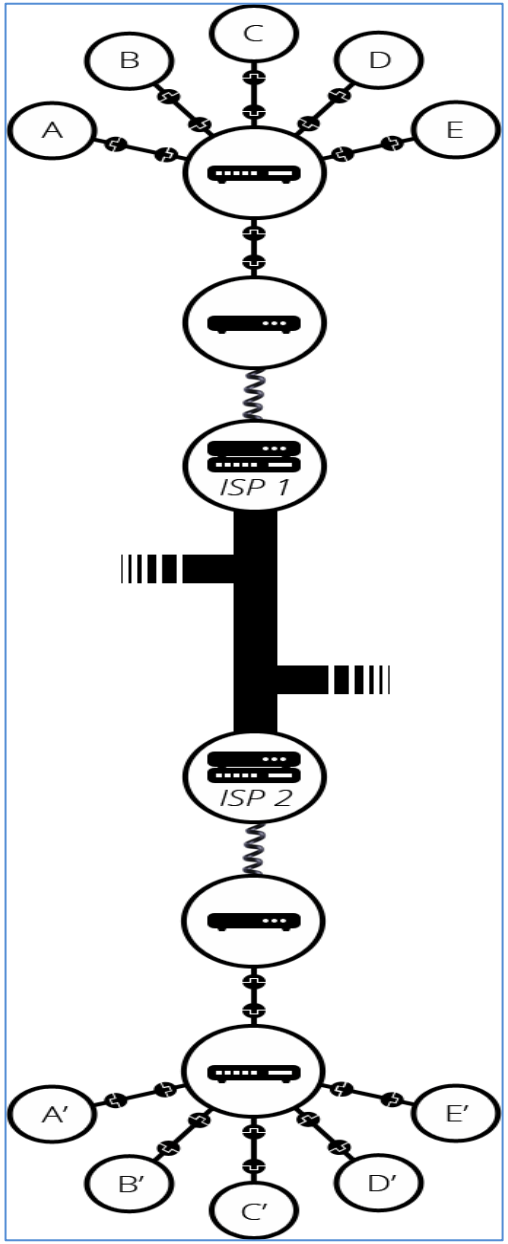
By connecting computers to routers, then routers to routers, we are able to scale infinitely.



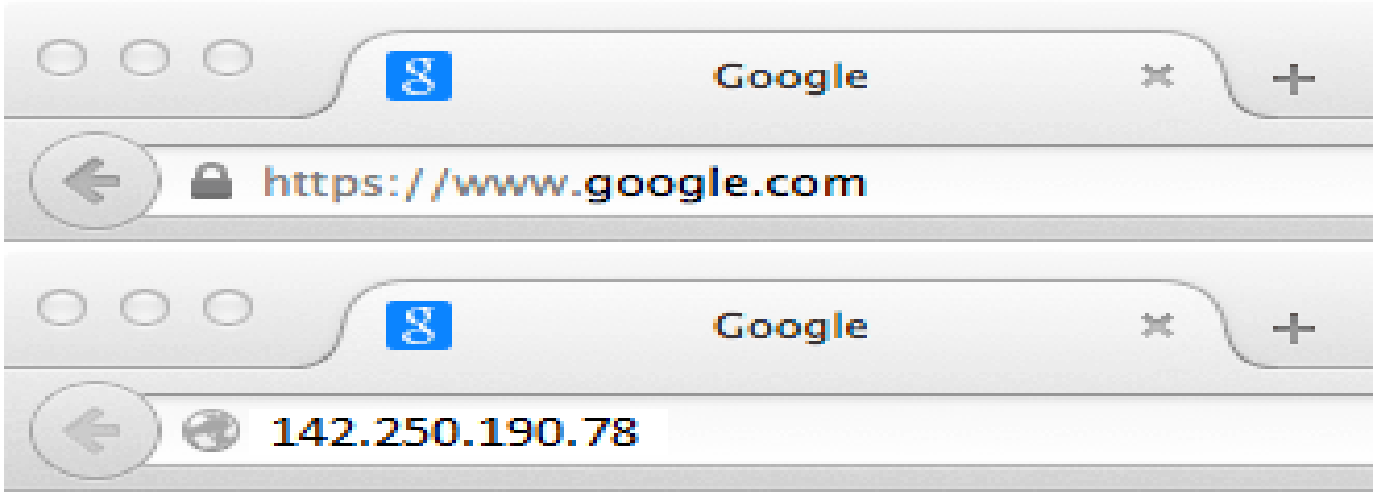
- Such a network comes very close to what we call the Internet, but we're missing something.
- We built that network for our own purposes.
- There are other networks out there: your friends, your neighbors, anyone can have their own network of computers.
- But it's not really possible to set cables up between your house and the rest of the world, so how can you handle this?
- Well, there are already cables linked to your house, for example, electric power and telephone.
- The telephone infrastructure already connects your house with anyone in the world so it is the perfect wire we need.
- To connect our network to the telephone infrastructure, we need a special piece of equipment called a *modem*.
- This *modem* turns the information from our network into information manageable by the telephone infrastructure and vice versa.


A
B
C
D
E

- So we are connected to the telephone infrastructure.
- The next step is to send the messages from our network to the network we want to reach.
- To do that, we will connect our network to an Internet Service Provider (ISP).
- An ISP is a company that manages some special *routers* that are all linked together and can also access other ISPs' routers.
- So the message from our network is carried through the network of ISP networks to the destination network.
- The Internet consists of this whole infrastructure of networks

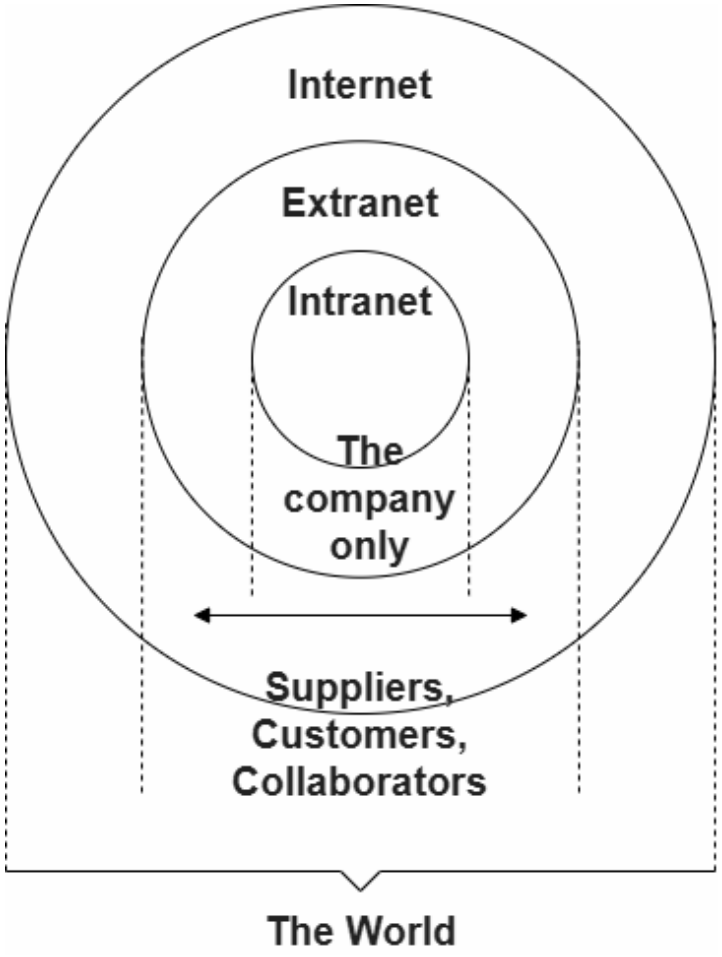

C
B
D
A
E
ISP 1
ISP 2
A′
E′
B′
D′
C′

- **Finding computers**
- If you want to send a message to a computer, you have to specify which one.
- Thus any **computer linked to a network has a unique address** that identifies it, called an "IP address" (where IP stands for *Internet Protocol*).
- It's an address made of a series of four numbers separated by dots, for example: 192.168.2.10.
- That's perfectly fine for computers, but we human beings have a hard time remembering that sort of address.
- To make things easier, we can alias an IP address with a human-readable name called a *domain name*.
- For example (at the time of writing; IP addresses can change) google.com is the domain name used on top of the IP address 142.250.190.78.
- So **using the domain name is the easiest way for us to reach a computer over the Internet.**

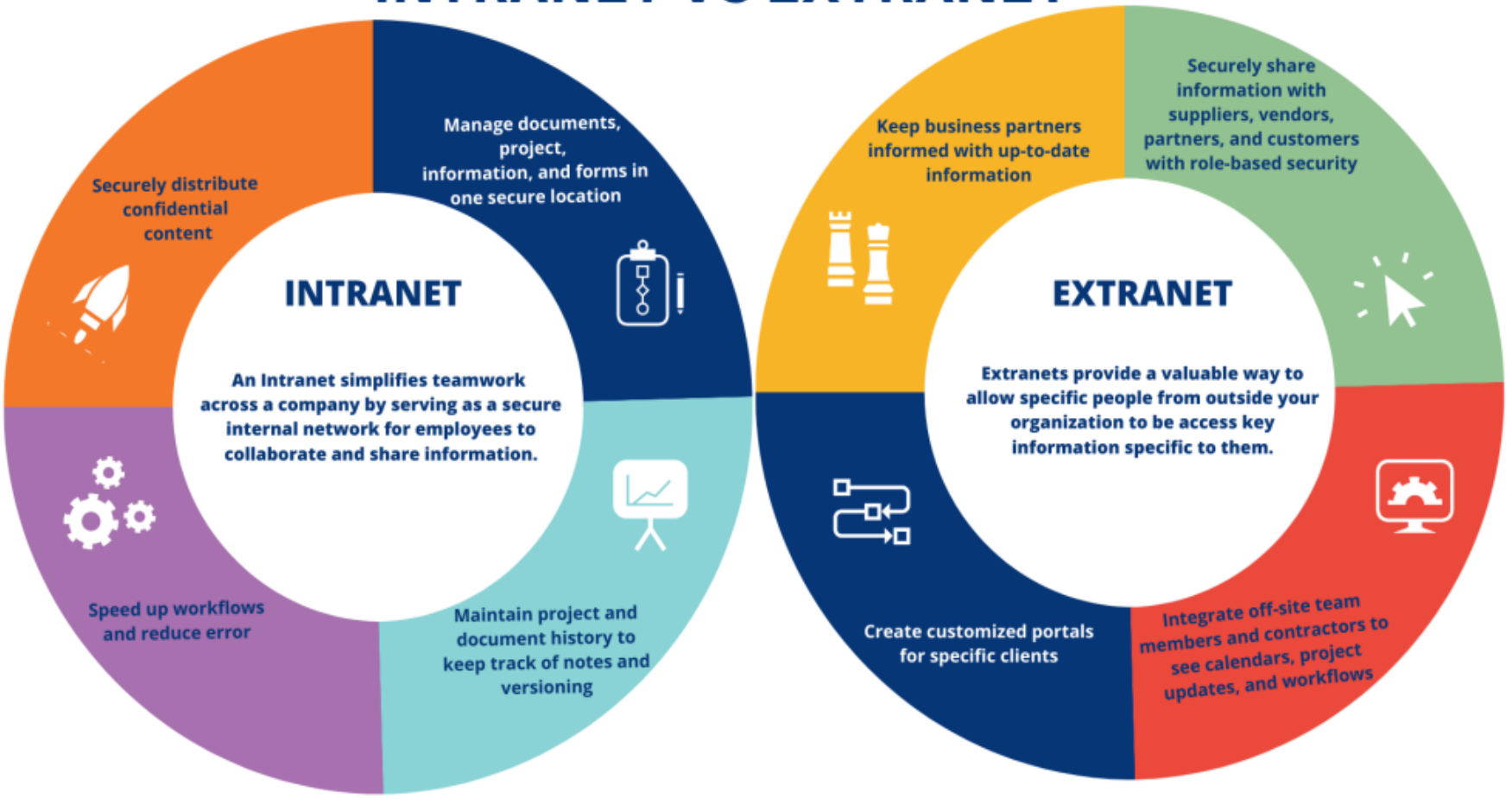

Google
https://www.google.com
Google
142.250.190.78

- **Internet and the web**
- As you might notice, when we browse the Web with a Web browser, we usually use the domain name to reach a website.
- Does that mean the Internet and the Web are the same thing? It's not that simple.
- As we saw, the Internet is a technical infrastructure which allows billions of computers to be connected all together.
- Among those computers, some computers (called *Web servers*) can send messages intelligible to web browsers.
- The *Internet* is an infrastructure, whereas the *Web* is a service built on top of the infrastructure.
- It is worth noting there are several other services built on top of the Internet, such as email and IRC.

- **Intranets and Extranets**
- Intranets are *private* networks that are restricted to members of a particular organization.
- They are commonly used to provide a portal for members to securely access shared resources, collaborate and communicate.
- For example, an organization's intranet might host web pages for sharing department or team information, shared drives for managing key documents and files, portals for performing business administration tasks, and collaboration tools like wikis, discussion boards, and messaging systems.
- Extranets are very similar to Intranets, except they open all or part of a private network to allow sharing and collaboration with other organizations.
- They are typically used to safely and securely share information with clients and stakeholders who work closely with a business.
- Often their functions are similar to those provided by an intranet: information and file sharing, collaboration tools, discussion boards, etc.
- Both intranets and extranets run on the same kind of infrastructure as the Internet, and use the same protocols.
- They can therefore be accessed by authorized members from different physical locations.
-

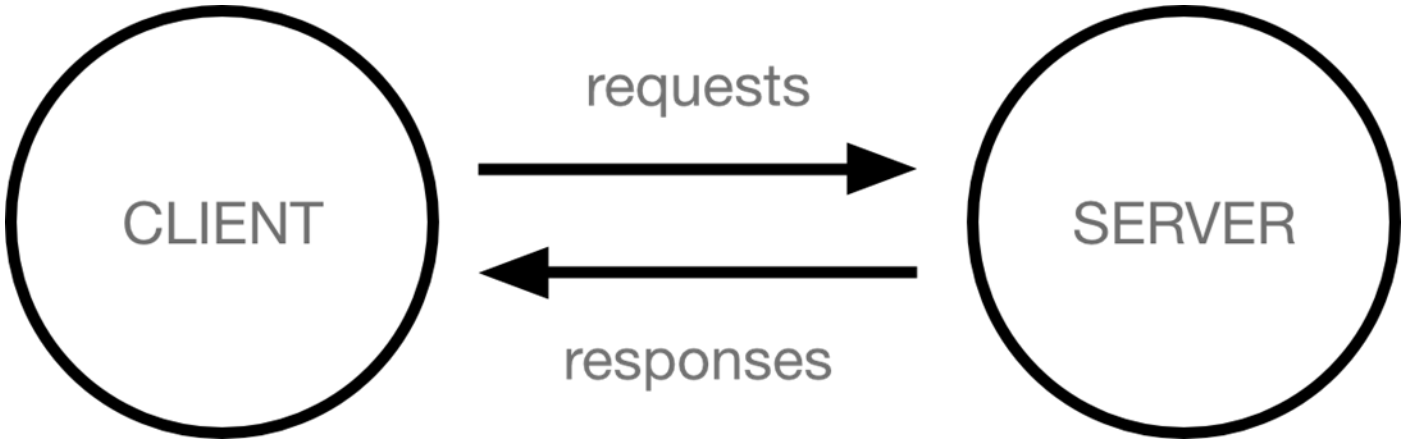

Internet
Extranet
Intranet
The company only
Suppliers, Customers, Collaborators
The World

INTRANET VS EXTRANET

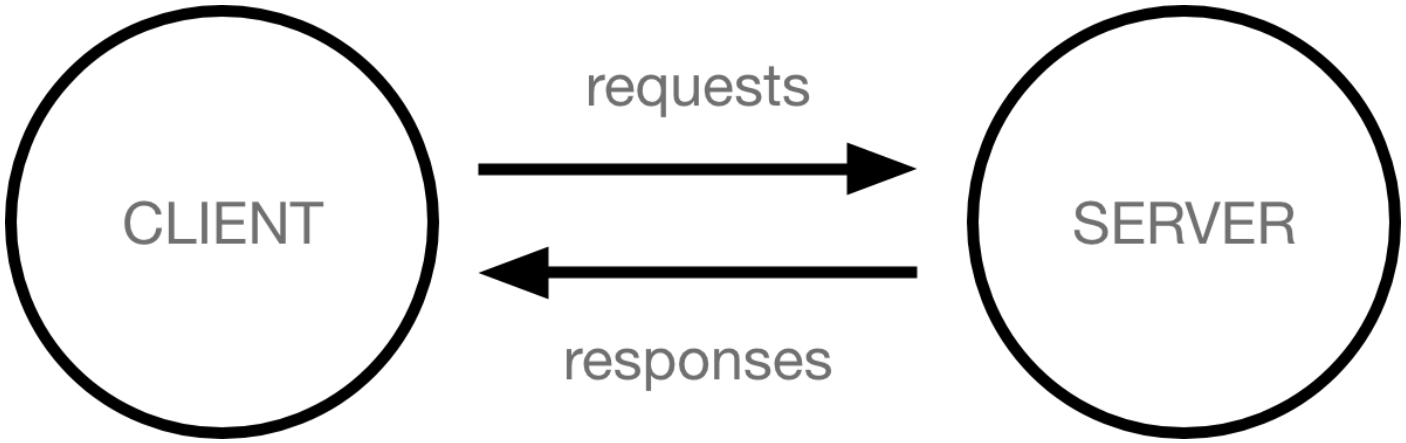


How the web works

- **Clients and servers**
- Computers connected to the internet are called **clients** and **servers**. A simplified diagram of how they interact might look like this:



- Clients are the typical web user's internet-connected devices (for example, your computer connected to your Wi-Fi, or your phone connected to your mobile network) and web-accessing software available on those devices (usually a web browser like Firefox or Chrome).
- Servers are computers that store webpages, sites, or apps.
- When a client device wants to access a webpage, a copy of the webpage is downloaded from the server onto the client machine to be displayed in the user's web browser.
- **The other parts of the toolbox**
- The client and server we've described above don't tell the whole story. There are many other parts involved, and we'll describe them below.
- For now, let's imagine that the web is a road. On one end of the road is the client, which is like your house. On the other end of the road is the server, which is a shop you want to buy something from.
-



In addition to the client and the server, we also need to say hello to:

- **Your internet connection**: Allows you to send and receive data on the web. It's basically like the street between your house and the shop.
- **TCP/IP**: Transmission Control Protocol and Internet Protocol are communication protocols **that define how data should travel across the internet.**
- This is like the transport mechanisms that let you place an order, go to the shop, and buy your goods. In our example, this is like a car or a bike (or however else you might get around).
- **DNS**: Domain Name System is like an **address book for websites**. When you type a web address in your browser, the browser looks at the DNS to find the website's IP address before it can retrieve the website. The browser needs to find out which server the website lives on, so it can send HTTP messages to the right place (see below). This is like looking up the address of the shop so you can access it.

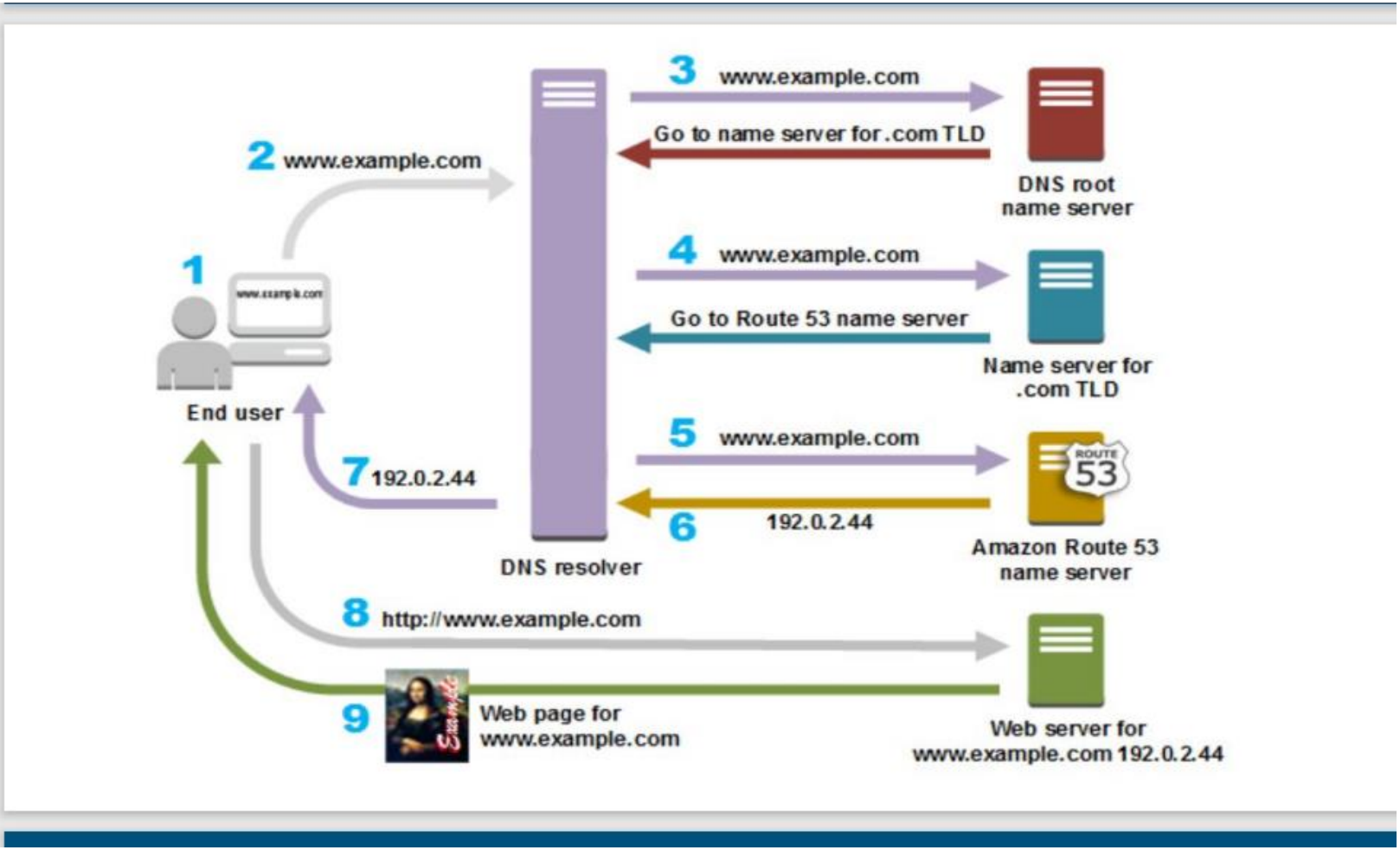
- **HTTP**: Hypertext Transfer Protocol is an application protocol that defines a language for clients and servers to speak to each other. This is like the language you use to order your goods.
- **Component files**: A website is made up of many different files, which are like the different parts of the goods you buy from the shop. These files come in two main types:
  - **Code files**: Websites are built primarily from HTML, CSS, and JavaScript, though you'll meet other technologies a bit later.
  - **Assets**: This is a collective name for all the other stuff that makes up a website, such as images, music, video, Word documents, and PDFs.

- **So what happens, exactly?**
- When you type a web address into your browser (for our analogy that's like walking to the shop):
- The browser goes to the DNS server, and finds the real address of the server that the website lives on (you find the address of the shop).
- The browser sends an HTTP request message to the server, asking it to send a copy of the website to the client (you go to the shop and order your goods).
- This message, and all other data sent between the client and the server, is sent across your internet connection using TCP/IP.
- If the server approves the client's request, the server sends the client a "200 OK" message, which means "Of course you can look at that website! Here it is", and then starts sending the website's files to the browser as a series of small chunks called data packets (the shop gives you your goods, and you bring them back to your house).
- The browser assembles the small chunks into a complete web page and displays it to you (the goods arrive at your door — new shiny stuff, awesome!).

- **Order in which component files are parsed**
- When browsers send requests to servers for HTML files, those HTML files often contain <link> elements referencing external CSS stylesheets and <script> elements referencing external JavaScript scripts. It's important to know the order in which those files are parsed by the browser as the browser loads the page:
- The browser parses the HTML file first, and that leads to the browser recognizing any <link>-element references to external CSS stylesheets and any <script>-element references to scripts.
- As the browser parses the HTML, it sends requests back to the server for any CSS files it has found from <link> elements, and any JavaScript files it has found from <script> elements, and from those, then parses the CSS and JavaScript.
- The browser generates an in-memory DOM tree from the parsed HTML, generates an in-memory CSSOM structure from the parsed CSS, and compiles and executes the parsed JavaScript.
- As the browser builds the DOM tree and applies the styles from the CSSOM tree and executes the JavaScript, a visual representation of the page is painted to the screen, and the user sees the page content and can begin to interact with it.

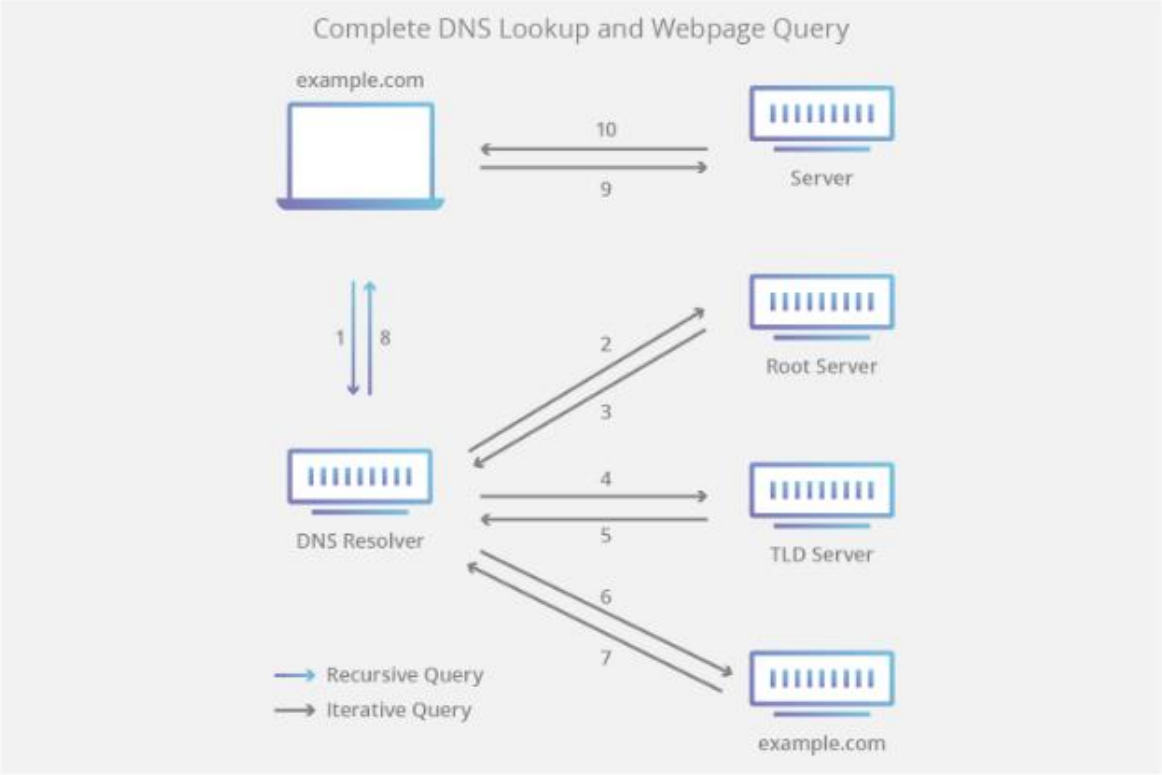
DNS explained

- Real web addresses aren't the nice, memorable strings you type into your address bar to find your favorite websites. They are special numbers that look like this: 63.245.215.20.
- This is called an IP address, and it represents a unique location on the web.
- However, it's not very easy to remember, is it?
- That's why Domain Name Servers were invented.
- These are special servers that match up a web address you type into your browser (like "mozilla.org") to the website's real (IP) address.
- Websites can be reached directly via their IP addresses. You can use a DNS lookup tool to find the IP address of a website.


1
End user
2 www.example.com
DNS resolver
3 www.example.com
Go to name server for .com TLD
DNS root name server
4 www.example.com
Go to Route 53 name server
Name server for .com TLD
5 www.example.com
6 192.0.2.44
Amazon Route 53 name server
7 192.0.2.44
8 http://www.example.com
9
Web page for www.example.com
Web server for www.example.com 192.0.2.44

- **What are the steps in a DNS lookup?**
- For most situations, DNS is concerned with a domain name being translated into the appropriate IP address.
- To learn how this process works, it helps to follow the path of a DNS lookup as it travels from a web browser, through the DNS lookup process, and back again. Let's take a look at the steps.
- Note: Often DNS lookup information will be cached either locally inside the querying computer or remotely in the DNS infrastructure.
- There are typically 8 steps in a DNS lookup.
- When DNS information is cached, steps are skipped from the DNS lookup process which makes it quicker.
- The example below outlines all 8 steps when nothing is cached.
-

- **The 8 steps in a DNS lookup:**
- A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a DNS recursive resolver.
- The resolver then queries a DNS root nameserver (.).
- The root server then responds to the resolver with the address of a Top Level Domain (TLD) DNS server (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
- The resolver then makes a request to the .com TLD.
- The TLD server then responds with the IP address of the domain’s nameserver, example.com.
- Lastly, the recursive resolver sends a query to the domain’s nameserver.
- The IP address for example.com is then returned to the resolver from the nameserver.
- The DNS resolver then responds to the web browser with the IP address of the domain requested initially.
- Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:
- The browser makes a HTTP request to the IP address.
- The server at that IP returns the webpage to be rendered in the browser (step 10).


Complete DNS Lookup and Webpage Query
example.com
10
9
Server
1
8
2
3
Root Server
DNS Resolver
4
5
TLD Server
6
7
Recursive Query
Iterative Query
example.com

HTTP

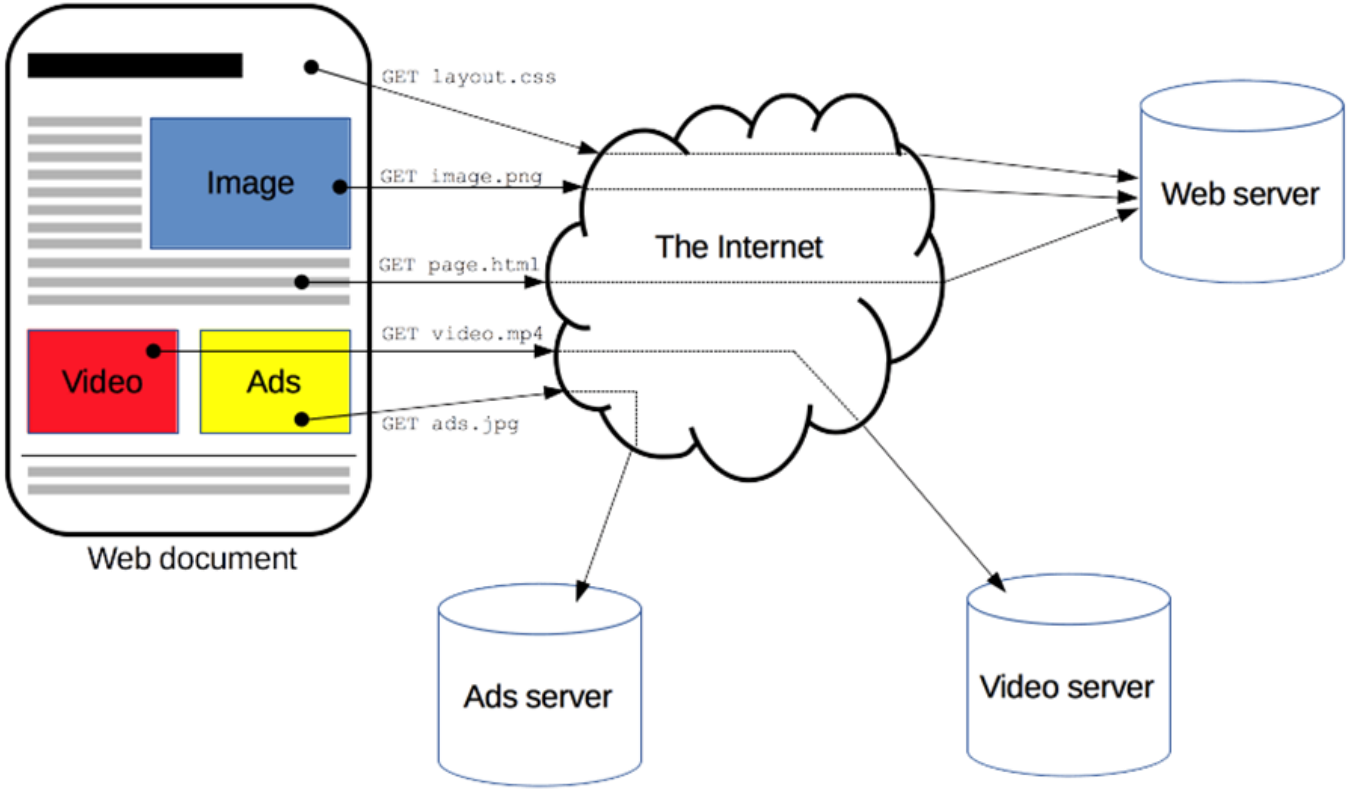
- **Evolution of HTTP**
- **HTTP** (HyperText Transfer Protocol) is the underlying protocol of the World Wide Web.
- Developed by Tim Berners-Lee and his team between 1989-1991, HTTP has gone through many changes that have helped maintain its simplicity while shaping its flexibility.
- Keep reading to learn how HTTP evolved from a protocol designed to exchange files in a semitrusted laboratory environment into a modern internet maze that carries images and videos in high resolution and 3D.

- **Invention of the World Wide Web**
- In 1989, while working at CERN, Tim Berners-Lee wrote a proposal to build a hypertext system over the internet.
- Initially called the *Mesh*, it was later renamed the *World Wide Web* during its implementation in 1990. Built over the existing TCP and IP protocols, it consisted of 4 building blocks:
- A textual format to represent hypertext documents, the *HyperText Markup Language* (HTML).
- A simple protocol to exchange these documents, the *HyperText Transfer Protocol* (HTTP).
- A client to display (and edit) these documents, the first web browser called the *WorldWideWeb*.
- A server to give access to the document, an early version of *httpd*.

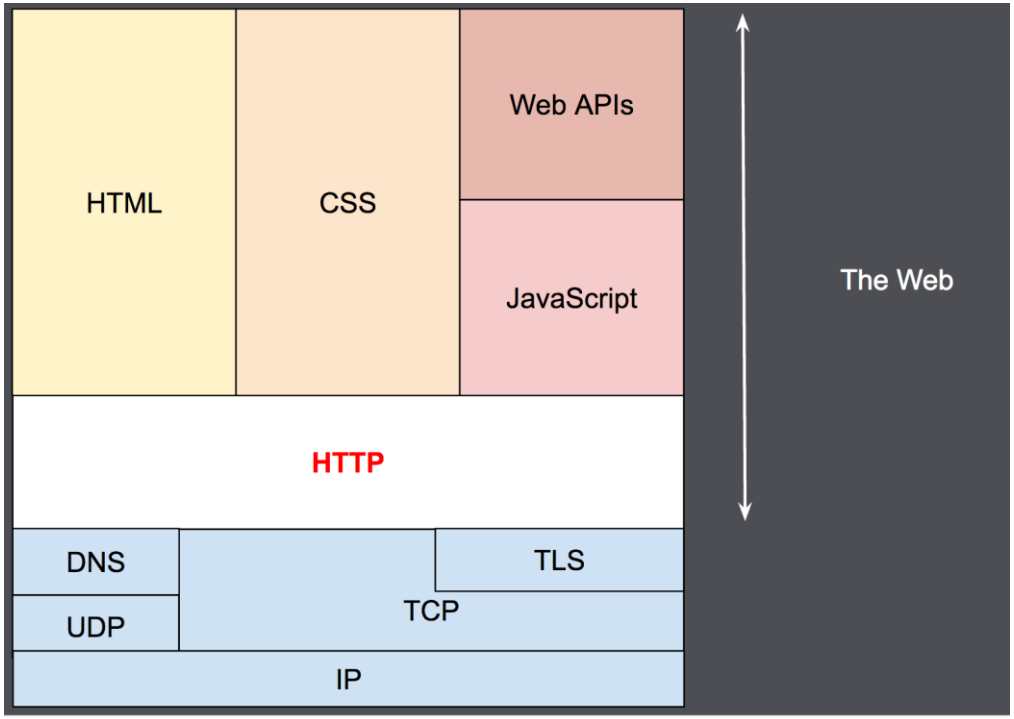
- These four building blocks were completed by the end of 1990, and the first servers were running outside of CERN by early 1991. On August 6, 1991, Tim Berners-Lee posted on the public alt.hypertext newsgroup. This is now considered to be the official start of the World Wide Web as a public project.
- The HTTP protocol used in those early phases was very simple. It was later dubbed HTTP/0.9 and is sometimes called the one-line protocol.

HTTP

- **HTTP** is a protocol for fetching resources such as HTML documents.
- It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser.
- A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.

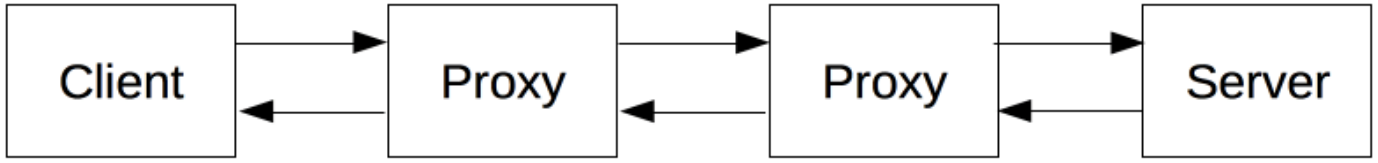

GET layout.css
Image
GET image.png
GET page.html
Video
Ads
GET video.mp4
GET ads.jpg
Web document
The Internet
Web server
Ads server
Video server

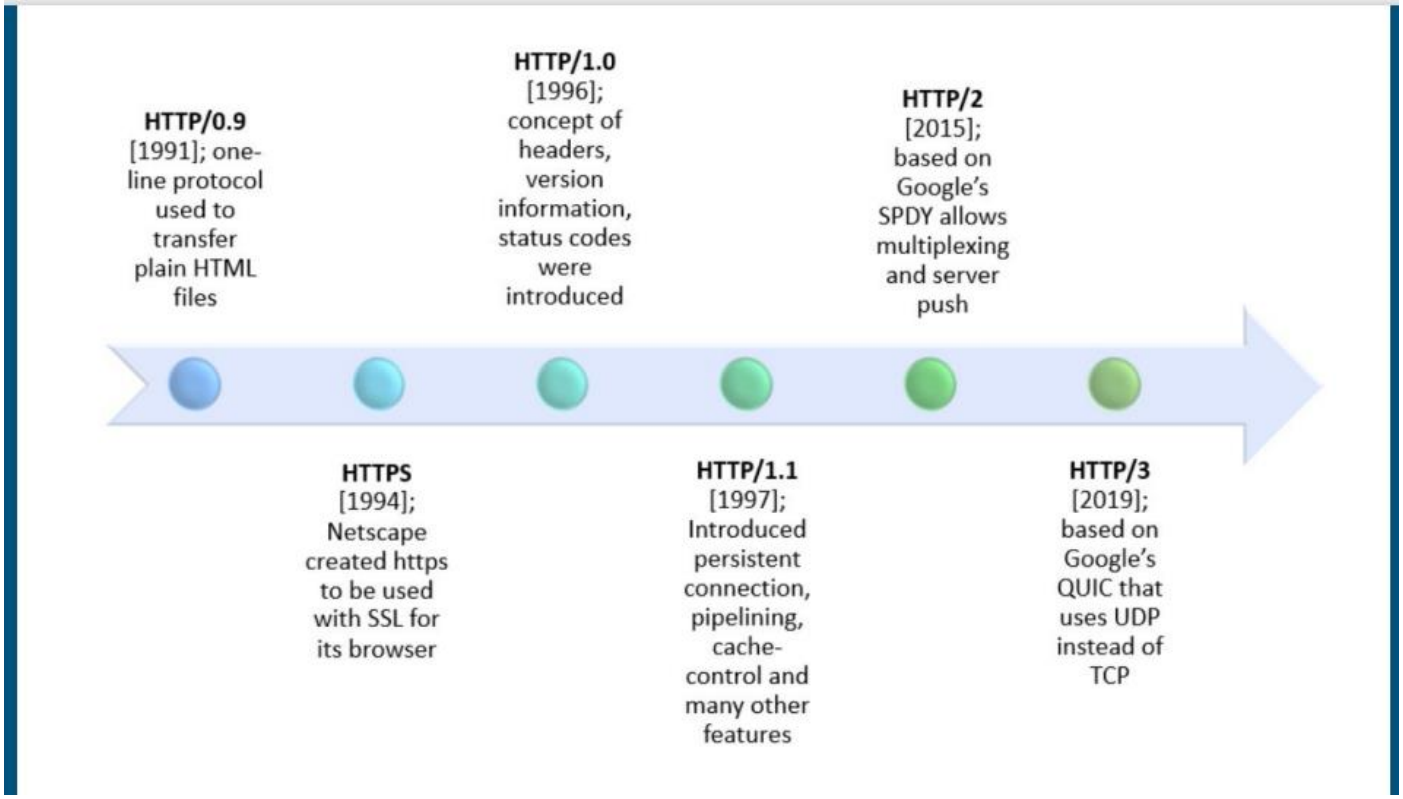
- Clients and servers communicate by exchanging individual messages (as opposed to a stream of data).
- The messages sent by the client, usually a Web browser, are called *requests* and the messages sent by the server as an answer are called *responses*.



- Designed in the early 1990s, HTTP is an extensible protocol which has evolved over time.
- It is an application layer protocol that is sent over TCP, or over a TLS-encrypted TCP connection, though any reliable transport protocol could theoretically be used.
- Due to its extensibility, it is used to not only fetch hypertext documents, but also images and videos or to post content to servers, like with HTML form results.
- HTTP can also be used to fetch parts of documents to update Web pages on demand.

- **Components of HTTP-based systems**
- HTTP is a client-server protocol: requests are sent by one entity, the user-agent (or a proxy on behalf of it).
- Most of the time the user-agent is a Web browser, but it can be anything, for example, a robot that crawls the Web to populate and maintain a search engine index.
- Each individual request is sent to a server, which handles it and provides an answer called the *response*.
- Between the client and the server there are numerous entities, collectively called proxies, which perform different operations and act as gateways or caches, for example.
- In reality, there are more computers between a browser and the server handling the request: there are routers, modems, and more. Thanks to the layered design of the Web, these are hidden in the network and transport layers. HTTP is on top, at the application layer. Although important for diagnosing network problems, the underlying layers are mostly irrelevant to the description of HTTP.




HTTP/0.9
[1991]; one-line protocol used to transfer plain HTML files
HTTPS
[1994]; Netscape created https to be used with SSL for its browser
HTTP/1.0
[1996]; concept of headers, version information, status codes were introduced
HTTP/1.1
[1997]; Introduced persistent connection, pipelining, cache-control and many other features
HTTP/2
[2015]; based on Google's SPDY allows multiplexing and server push
HTTP/3
[2019]; based on Google's QUIC that uses UDP instead of TCP

Versions

- HTTP 0.9 =>Simple
- HTTP 1.0=>Added status
- HTTP1.1=>Modifies with session management
- Reuse connection, Cache control, chunk transfer data
- HTTP/2 more stable
- HTTP/3 =>
- QUIC is designed to provide much lower latency for HTTP connections. Like HTTP/2, it is a multiplexed protocol, but HTTP/2 runs over a single TCP connection, so packet loss detection and retransmission handled at the TCP layer can block all streams.

Status codes

The Status-Code element in a server response, is a 3-digit integer where the first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. There are 5 values for the first digit:

S.N.	Code and Description
1	**1xx: Informational** It means the request has been received and the process is continuing.
2	**2xx: Success** It means the action was successfully received, understood, and accepted.
3	**3xx: Redirection** It means further action must be taken in order to complete the request.
4	**4xx: Client Error** It means the request contains incorrect syntax or cannot be fulfilled.
5	**5xx: Server Error** It means the server failed to fulfill an apparently valid request.

1xx: Information

Message	Description
100 Continue	Only a part of the request has been received by the server, but as long as it has not been rejected, the client should continue with the request.
101 Switching Protocols	The server switches protocol.

Message	Description
200 OK	The request is OK.
201 Created	The request is complete, and a new resource is created .
202 Accepted	The request is accepted for processing, but the processing is not complete.
203 Non-authoritative Information	The information in the entity header is from a local or third-party copy, not from the original server.
204 No Content	A status code and a header are given in the response, but there is no entity-body in the reply.
205 Reset Content	The browser should clear the form used for this transaction for additional input.
206 Partial Content	The server is returning partial data of the size requested. Used in response to a request specifying a *Range* header. The server must specify the range included in the response with the *Content-Range* header.

Message	Description
300 Multiple Choices	A link list. The user can select a link and go to that location. Maximum five addresses .
301 Moved Permanently	The requested page has moved to a new url .
302 Found	The requested page has moved temporarily to a new url .
303 See Other	The requested page can be found under a different url .
304 Not Modified	This is the response code to an *If-Modified-Since* or *If-None-Match* header, where the URL has not been modified since the specified date.

	Match header, where the URL has not been modified since the specified date.
305 Use Proxy	The requested URL must be accessed through the proxy mentioned in the *Location* header.
306 *Unused*	This code was used in a previous version. It is no longer used, but the code is reserved.
307 Temporary Redirect	The requested page has moved temporarily to a new url.

4xx: Client Error

Message	Description
400 Bad Request	The server did not understand the request.
401 Unauthorized	The requested page needs a username and a password.
402 Payment Required	*You can not use this code yet.*
403 Forbidden	Access is forbidden to the requested page.
404 Not Found	The server can not find the requested page.
405 Method Not Allowed	The method specified in the request is not allowed.
406 Not Acceptable	The server can only generate a response that is not accepted by the client.
407 Proxy Authentication Required	You must authenticate with a proxy server before this request can be served.
408 Request Timeout	The request took longer than the server was prepared to wait.

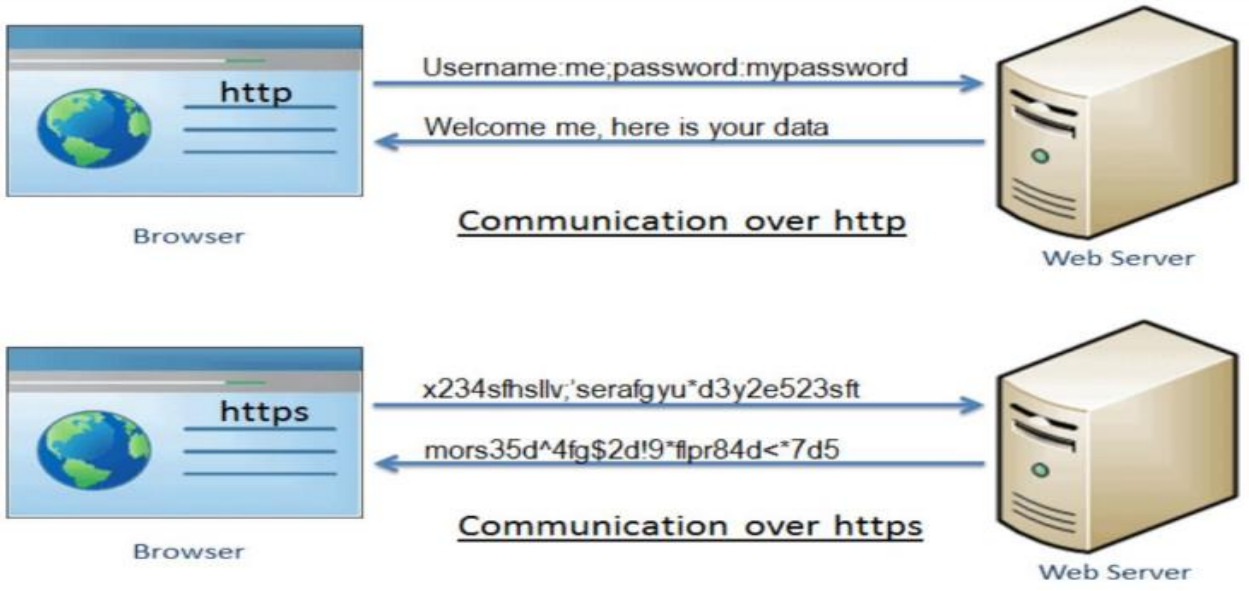
5xx: Server Error

Message	Description
500 Internal Server Error	The request was not completed. The server met an unexpected condition.
501 Not Implemented	The request was not completed. The server did not support the functionality required.
502 Bad Gateway	The request was not completed. The server received an invalid response from the upstream server.
503 Service Unavailable	The request was not completed. The server is temporarily overloading or down.
504 Gateway Timeout	The gateway has timed out.
505 HTTP Version Not Supported	The server does not support the "http protocol" version.

HTTPS

HTTPS stands for Hyper Text Transfer Protocol Secure. It is a protocol for securing the communication between two systems e.g. the browser and the web server.

HTTPS established an encrypted link between the browser and the web server using the Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocols. TLS is the new version of SSL.



Secure Socket Layer (SSL)

SSL is the standard security technology for establishing an encrypted link between the two systems. These can be browser to server, server to server or client to server. Basically, SSL ensures that the data transfer between the two systems remains encrypted and private.

The https is essentially http over SSL. SSL establishes an encrypted link using an SSL certificate which is also known as a digital certificate.

http vs https

http	https
Transfers data in hypertext (structured text) format	Transfers data in encrypted format
Uses port 80 by default	Uses port 443 by default
Not secure	Secured using SSL technology
Starts with `http://`	Starts with `https://`

Web servers – IIS, Apache server

- Nginx
- Apache Tomcat
- Node.js
- Lighttpd
- Cherokee
- Microsoft IIS
- Appweb
- Hiawatha

IIS

Internet Information Services (IIS) 7 and later provide a request-processing architecture which includes:

- The Windows Process Activation Service (WAS), which enables sites to use protocols other than HTTP and HTTPS.
- A Web server engine that can be customized by adding or removing modules.
- Integrated request-processing pipelines from IIS and ASP.NET.

HTTP METHODS

- HTTP methods are HTTP requests that indicates what actions the server should perform.
- These methods allow for a much richer communication between the client and the server.
- These are the common HTTP methods.
- GET
- POST
- PUT
- PATCH
- DELETE
- HEAD
- OPTIONS

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
- https://howdns.works/
- https://howhttps.works/why-do-we-need-https/

HTML

Objectives

- Getting Started With HTML
  - Introducing HTML
  - How Web pages work
  - Introduction to Tags
  - Structure of HTML page
- Creating a Web Page
  - Introduction table, tr, td tags.
  - Paragraphs and Line breaks
  - Headings
  - Creating Ordered and Unordered List
- Text Formatting And Alignment
  - Boldfaces and Italics
  - Font Size, Color and Typeface
  - Special Characters

Objectives

- Using Links & Images
- Style Sheet
  - Purpose of Style Sheet
  - Style Rules
  - Attaching Style Sheet
  - Defining classes
  - Div tag, Span Tag
  - Using Span
- Developing Forms
  - Introducing Forms
  - Using Standard Form Controls

What is World Wide Web?

- The *World Wide Web (Web)* is a network of information resources.
- The Web relies on three mechanisms to make these resources readily available to the widest possible audience:
  - A uniform naming scheme for locating resources on the Web
  - Protocols, for access to named resources over the Web
  - Hypertext, for easy navigation among resources

What is the Web Browser?

- The web browser is an application software to explore www (World Wide Web).
- It provides an interface between the server and the client and requests to the server for web documents and services.
- It works as a compiler to render HTML which is used to design a webpage. Whenever we search anything on the internet, the browser loads a web page written in HTML, including text, links, images, and other items such as style sheets and JavaScript functions.
- Google Chrome, Microsoft Edge, Mozilla Firefox, Safari are examples of web browsers.

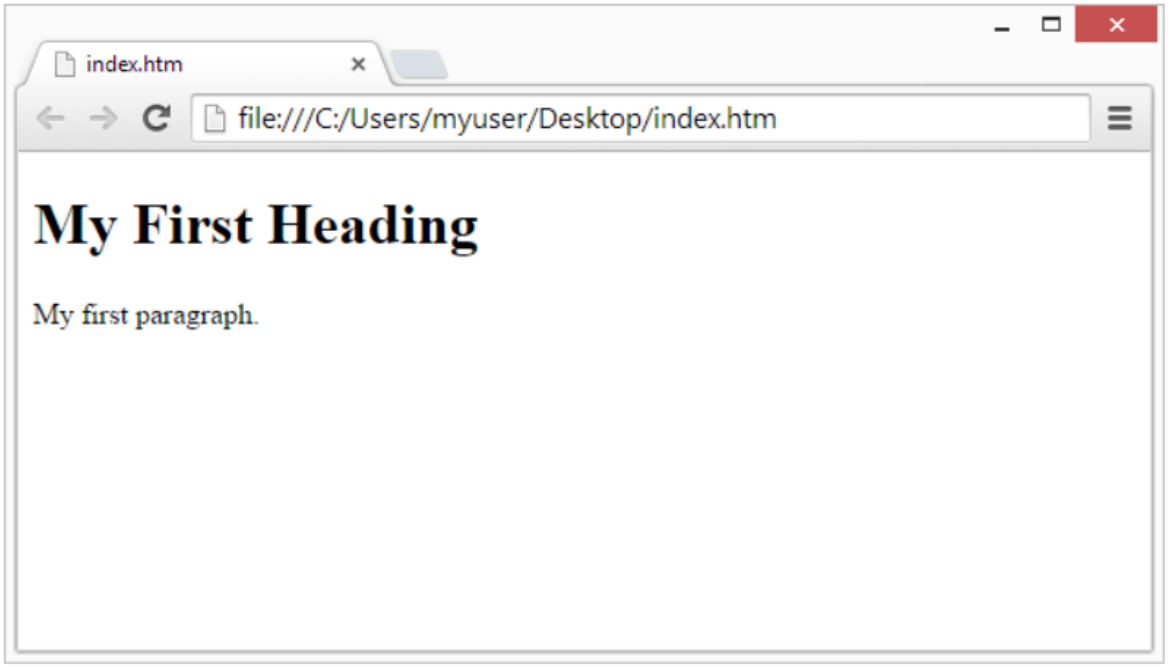
Web page

- A **web page** or **webpage** is a document, commonly written in HTML, that is viewed in an Internet browser.
- A web page can be accessed by entering a URL address into a browser's address bar.
- A web page may contain text, graphics, and hyperlinks to other web pages and files.
- A web page is often used to provide information to viewers, including pictures or videos to help illustrate important topics.
- A web page may also be used as a method to sell products or services to viewers. Multiple web pages make up a website, like our Computer Hope website.
- **When was the first web page created?**
- The first web page was created at CERN by Tim Berners-Lee on August 6, 1991.

Web Browsers

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:



Editor


What are the Top HTML Editor for Developer
BlueGriffon
Microsoft Visual Studio
Sublime Text Editor
Komodo Edit
aptana
STUDIO 3.6
NetBeans
CODEPEN
notepad++
Bluefish
Atom Editor

Introducing HTML

What is HTML

To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand.
The publishing language used by the World Wide Web is HTML (from HyperText Markup Language).

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Introducing HTML

What is HTML

HTML contains special markers, or tags, which define the style and structure of the page.

Tags are also referred to as elements.

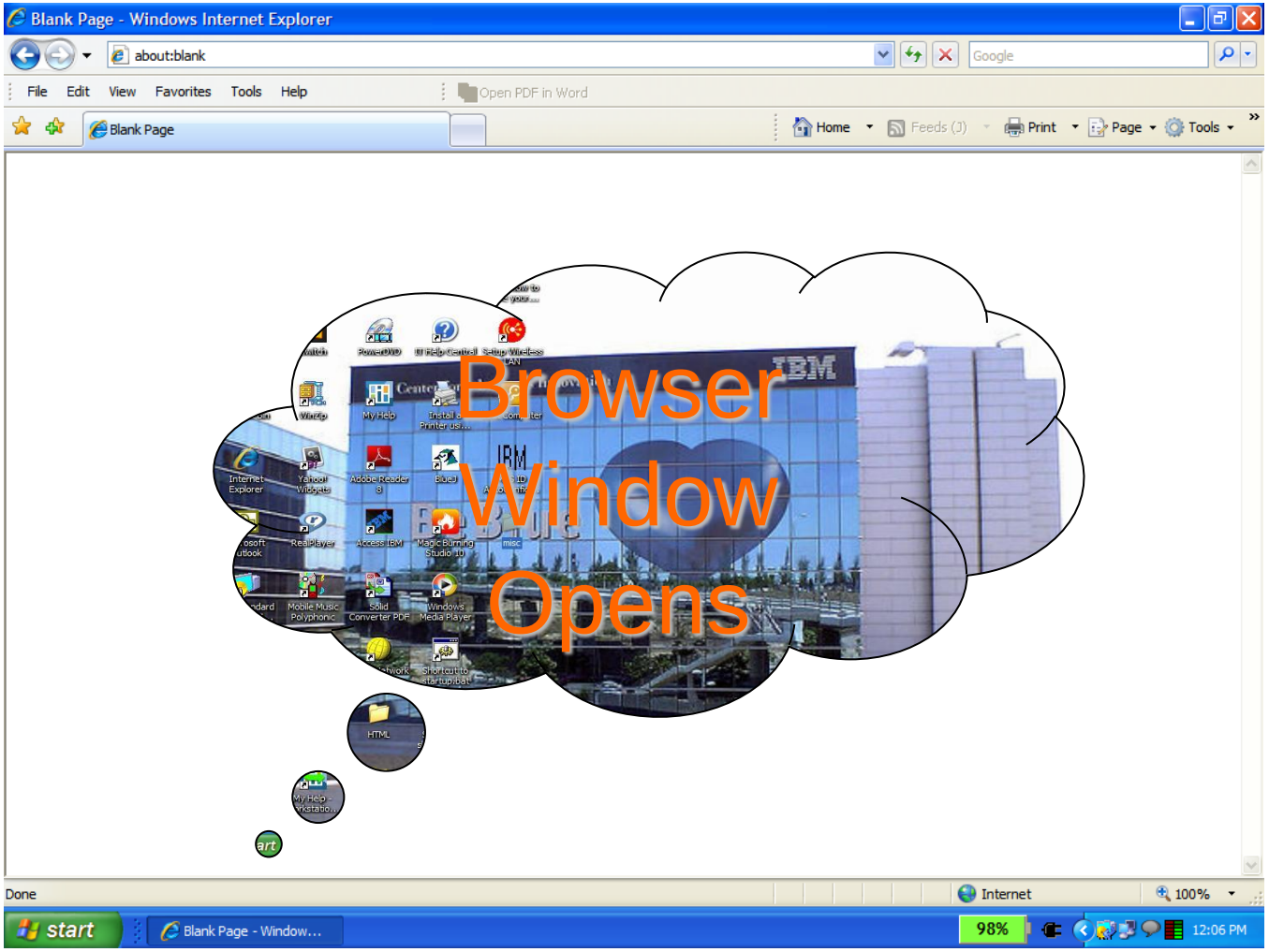

HTML Tags List
Basic Tags
Formatting Tags
Image Tags
Link Tags
List Tags
Form Tags
Table Tags
Media Tags
Meta Tags
Programming Tags
General Tags

Introducing HTML

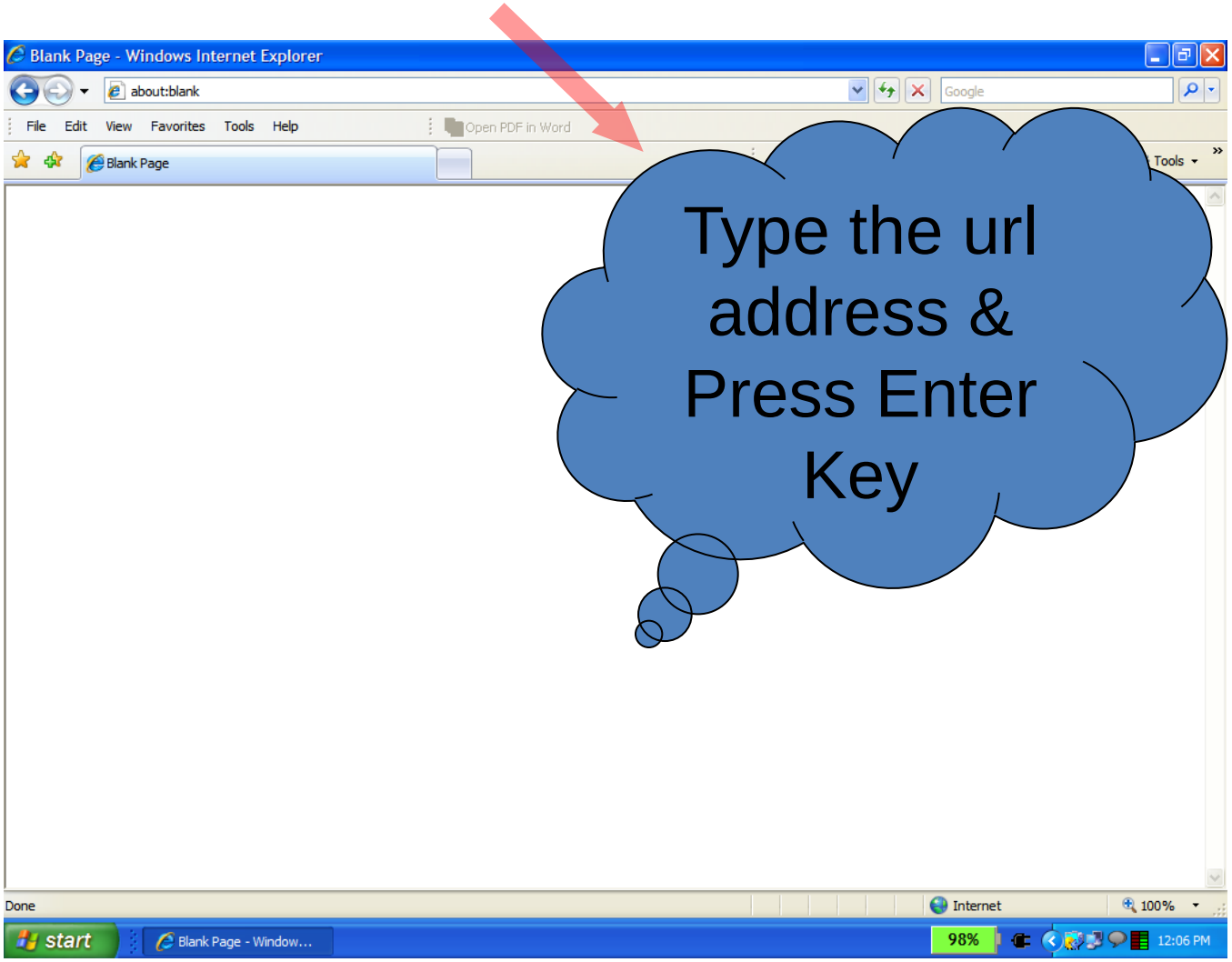
Uses of HTML

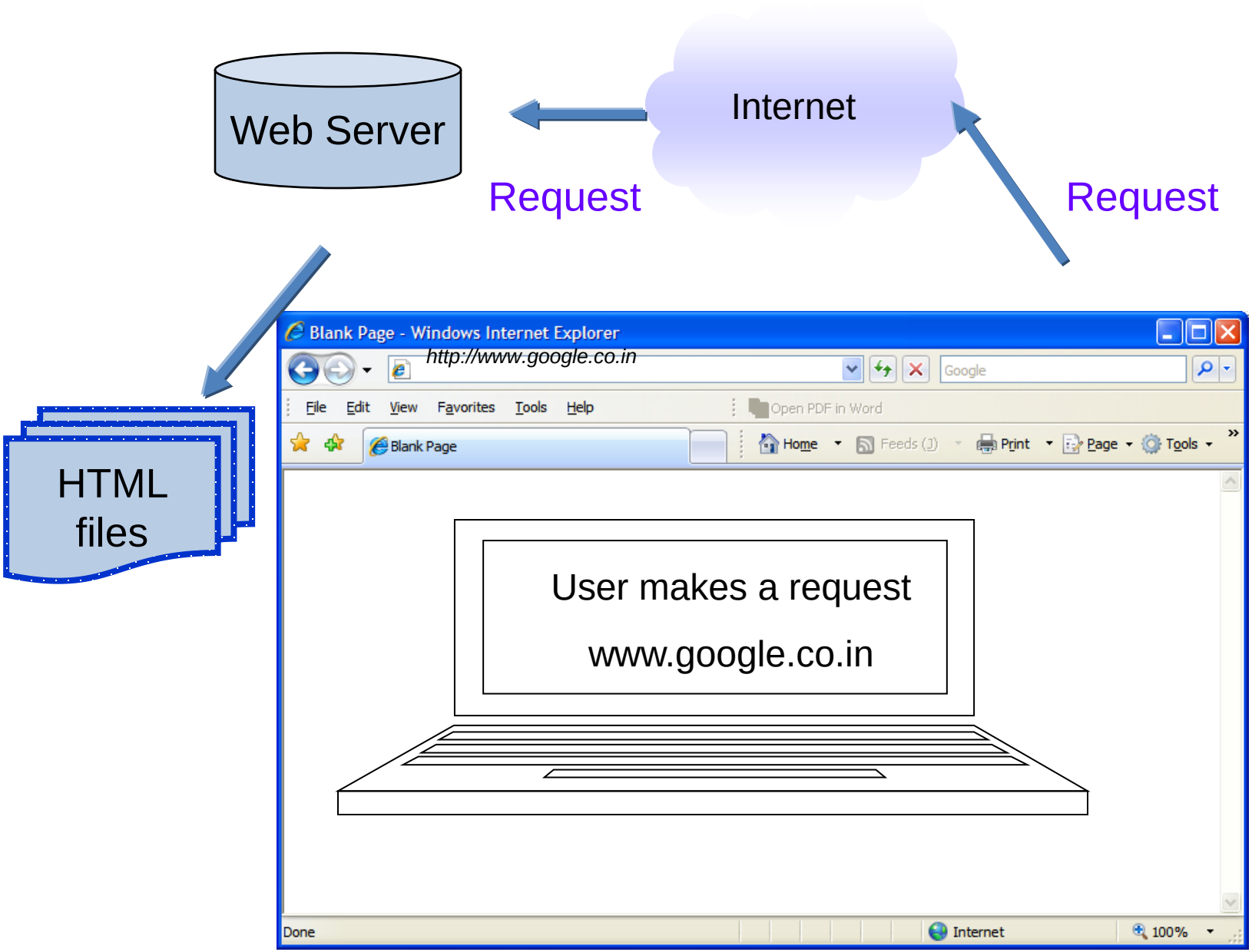
- Publish online documents with headings, texts, tables, lists, photos, etc.
- Retrieve online information via hypertexts links.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

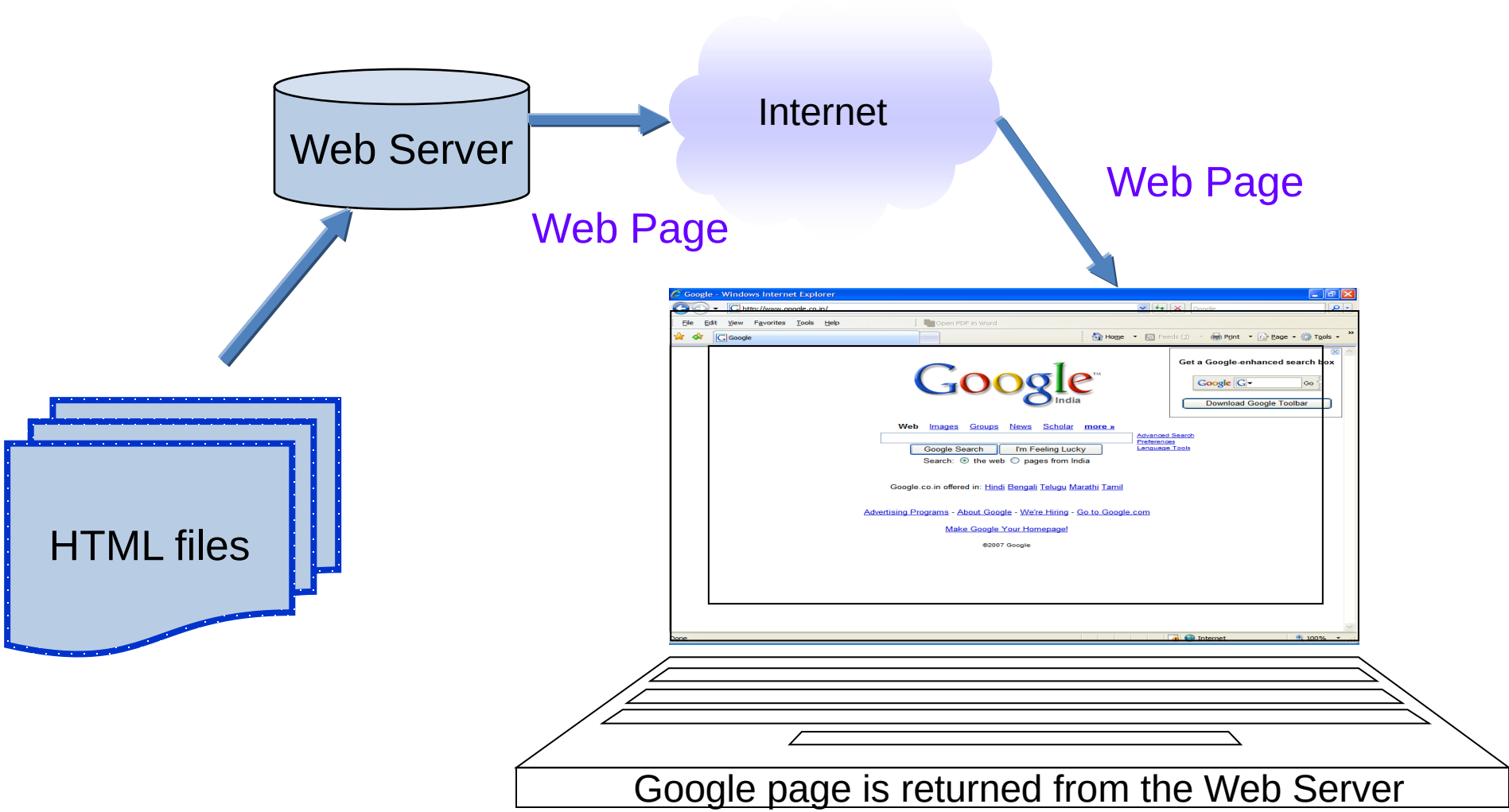

Double Click on the Internet Explorer button
My Documents
ISMA Peer To P...
NetBeans 5.5
My Computer
Learn how to move your ...
My Network Places
Switch
PowerDVD
IT Help Central
Setup Wireless LAN
Recycle Bin
WinZip
My Help
Install a Printer usi...
Lock Computer
Internet Explorer
Yahoo! Widgets
Adobe Reader 8
BlueJ
MTS ID AutoConfig...
Microsoft Outlook
RealPlayer
Access IBM
Magic Burning Studio 10
misc
IBM Standard Software I...
Mobile Music Polyphonic
Solid Converter PDF
Windows Media Player
ISSI EZUpdate
Nero StartSmart
AT&T Network Client
Shortcut to startup.bat
Second Life
Intranet Labor Claiming
HTML
Shortcut to shutdown.bat
R Connect Dialer
My Help - Workstatio...
THREADS
SkillSoft Cours...
start
98%
11:27 AM


Blank Page - Windows Internet Explorer
about:blank
Google
File Edit View Favorites Tools Help
Open PDF in Word
Blank Page
Home
Feeds (J)
Print
Page
Tools
Browser Window Opens
IBM
Done
Internet
100%
start
Blank Page - Window...
98%
12:06 PM

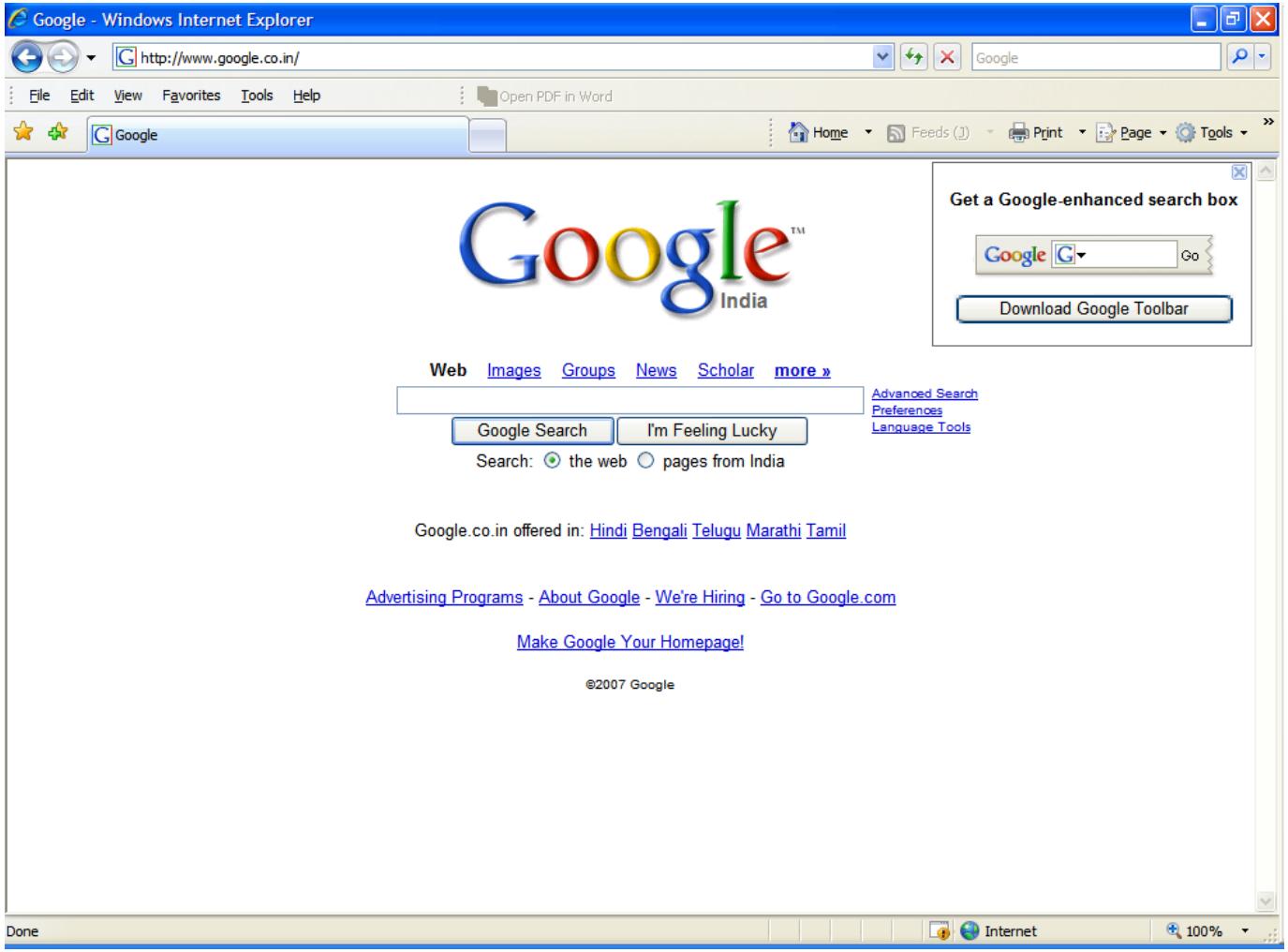
http://www.google.co.in




Web Server
Internet
Request
Request
HTML files
Blank Page - Windows Internet Explorer
http://www.google.co.in
Google
File Edit View Favorites Tools Help
Open PDF in Word
Blank Page
Home
Feeds (J)
Print
Page
Tools
User makes a request
www.google.co.in
Done
Internet
100%



Google page is returned from the Web Server


Google - Windows Internet Explorer
http://www.google.co.in/
File Edit View Favorites Tools Help
Open PDF in Word
Google
Home Feeds (J) Print Page Tools
Get a Google-enhanced search box
Google Go
Download Google Toolbar
Google India
Web Images Groups News Scholar more »
Advanced Search
Preferences
Language Tools
Google Search I'm Feeling Lucky
Search: the web pages from India
Google.co.in offered in: Hindi Bengali Telugu Marathi Tamil
Advertising Programs - About Google - We're Hiring - Go to Google.com
Make Google Your Homepage!
©2007 Google
Done
Internet
100%

Introducing HTML

What is a Tag

A tag is a reference in an HTML document which describes the style and structure of the document.

All tag start with < (less than symbol) and end with > (greater than symbol)

Tags which mark a beginning have no / (front slash).

Tags which mark an ending have a / immediately after <, as in </.

- What is an HTML Element?
- An HTML element is defined by a start tag, some content, and an end tag:
- <tagname> Content goes here... </tagname>
- The HTML **element** is everything from the start tag to the end tag:
- <h1>My First Heading</h1>
- My first paragraph.

Introducing HTML

Example on Tag

<HTML>, <BODY>, <TITLE>	Starting tags / elements
</HTML>, </BODY>, </TITLE>	Ending tags / elements

HTML Page Structure

Below is a visualization of an HTML page structure:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

HTML History

Since the early days of the World Wide Web, there have been many versions of HTML:

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

Introducing HTML

Explanation of the HTML code

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

Start of an HTML document.

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

Indicates to the browser that this is an HTML document.

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

HTML document is divided into two sections.
1. <HEAD> …. </HEAD>

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

HTML document is divided into two sections.
2. <BODY> …. </BODY>

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

Shows the contents in the browser title bar.
Generally placed inside Head tag.

```
<HTML>
<HEAD>
<TITLE> My Home Page </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

Indicates comment tag. Can be placed anywhere in the document.

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Explanation of the HTML code

Signal the end of HTML document.

```
<HTML>
<HEAD>
<TITLE>  My Home Page  </TITLE>
</HEAD>
<!-- Written by Manas K Sahoo -->
<!-- Created:  30-May-2007 -->
<!-- Organization:  IBM -->
<BODY>This is where the text goes.
</BODY>
</HTML>
```

Introducing HTML

Kinds of Tags

1. Range / Container tags

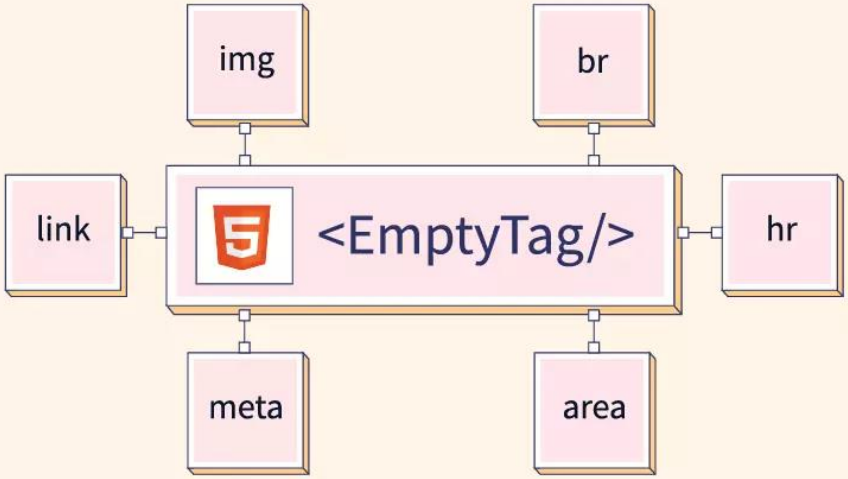
They are called so because they cover a range of text.

Example : <HTML>, <HEAD>, <BODY>, <TITLE>

2. Standalone / Point tags

Those tags which do not need an ending.

Example: <HR>,


img
br
link
<EmptyTag/>
hr
meta
area

Paragraphs
&
Line Breaks

Line Separators

- Tags used for separating one line with other can be classifies as follows:-

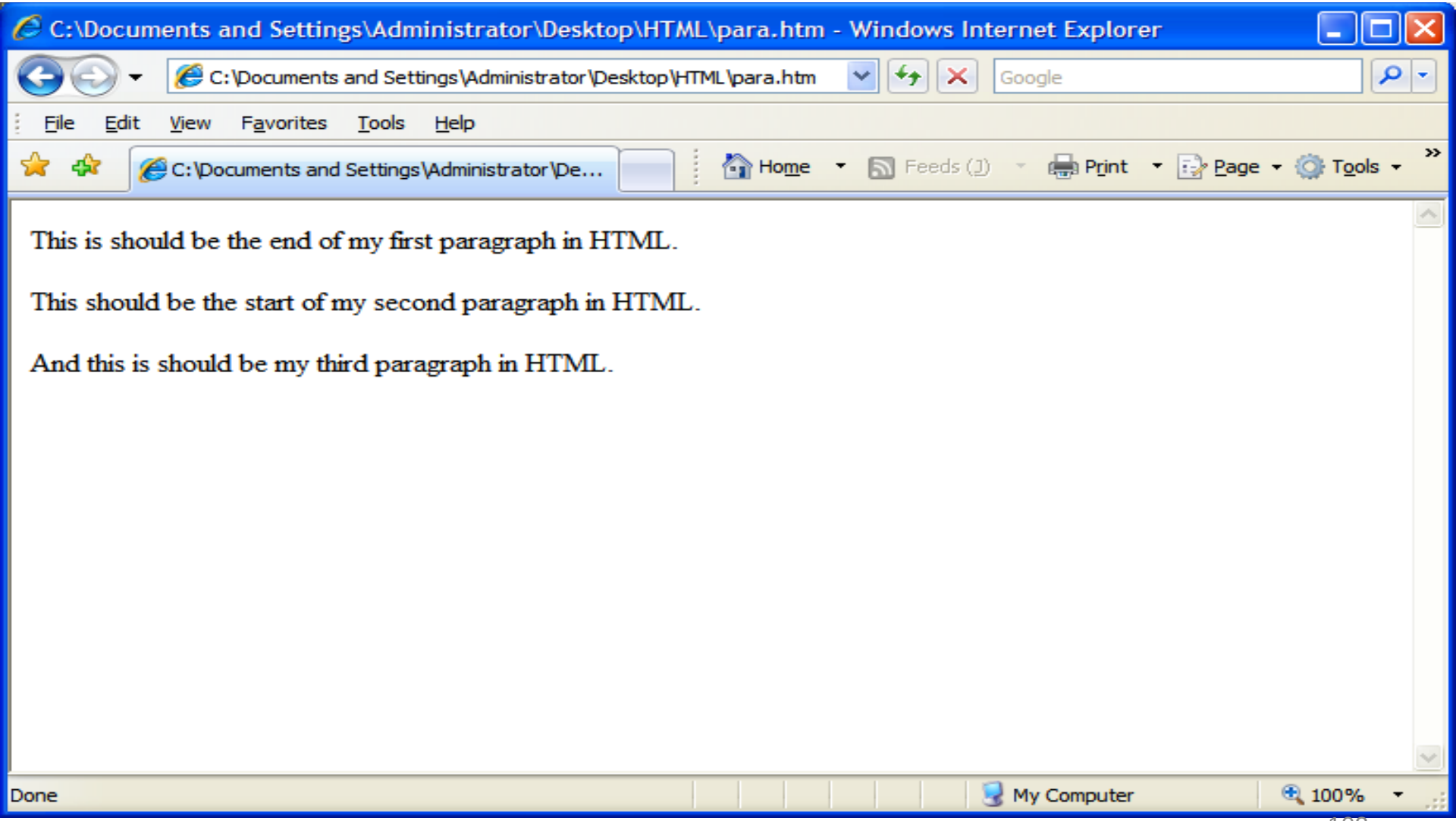
1. <P> tag
2.
 tag
3. <HR> tag

The <P> Tag

<P> tag.

- Normally all text in an HTML document is treated like one long paragraph.
- Indented or blank lines which normally indicate start of paragraphs in text are ignored.
- To separate your text into paragraphs use the
- The <P> tag forces an end of line and forces a blank line before the next paragraph.

Requirement Analysis



Code Snippet To Fulfill the Requirement

```
<p>This is should be the  end of  my first
paragraph in HTML.</p>

<p>This should be the start of my second
paragraph in
HTML.</p>

<p>And this is should be my third paragraph
in
HTML.</p>
```

tag forces an end of line and forces a blank line before the next paragraph.

*** ***This is should be the end of my first paragraph in HTML.

*** ***This should be the start of my second paragraph in HTML.

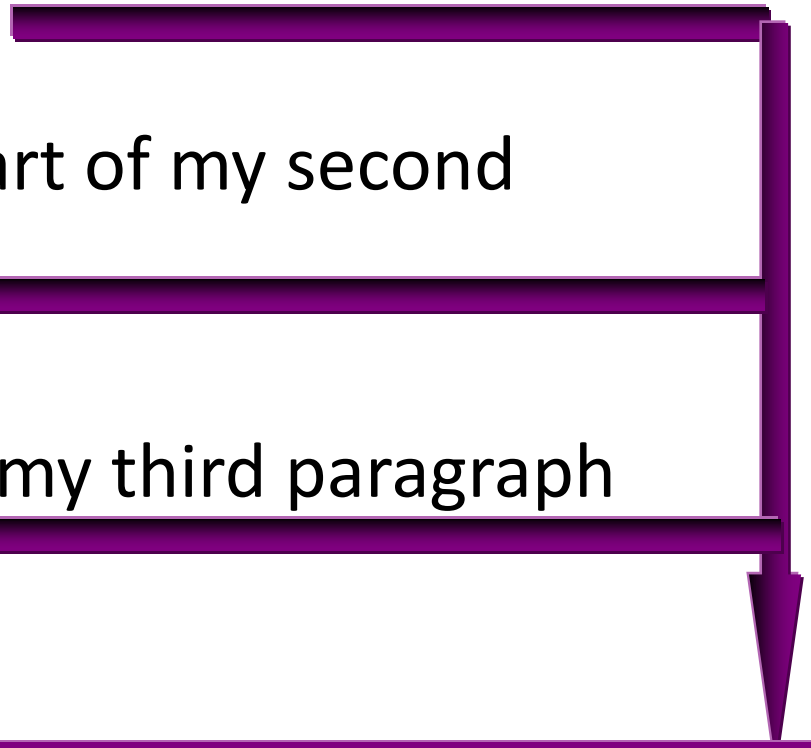
*** ***And this is should be my third paragraph in HTML.

 tag forces an end of line and forces a blank line before the next paragraph.

This is should be the end of my first paragraph in HTML.*** ***

This should be the start of my second paragraph in HTML.*** ***

And this is should be my third paragraph in HTML.*** ***

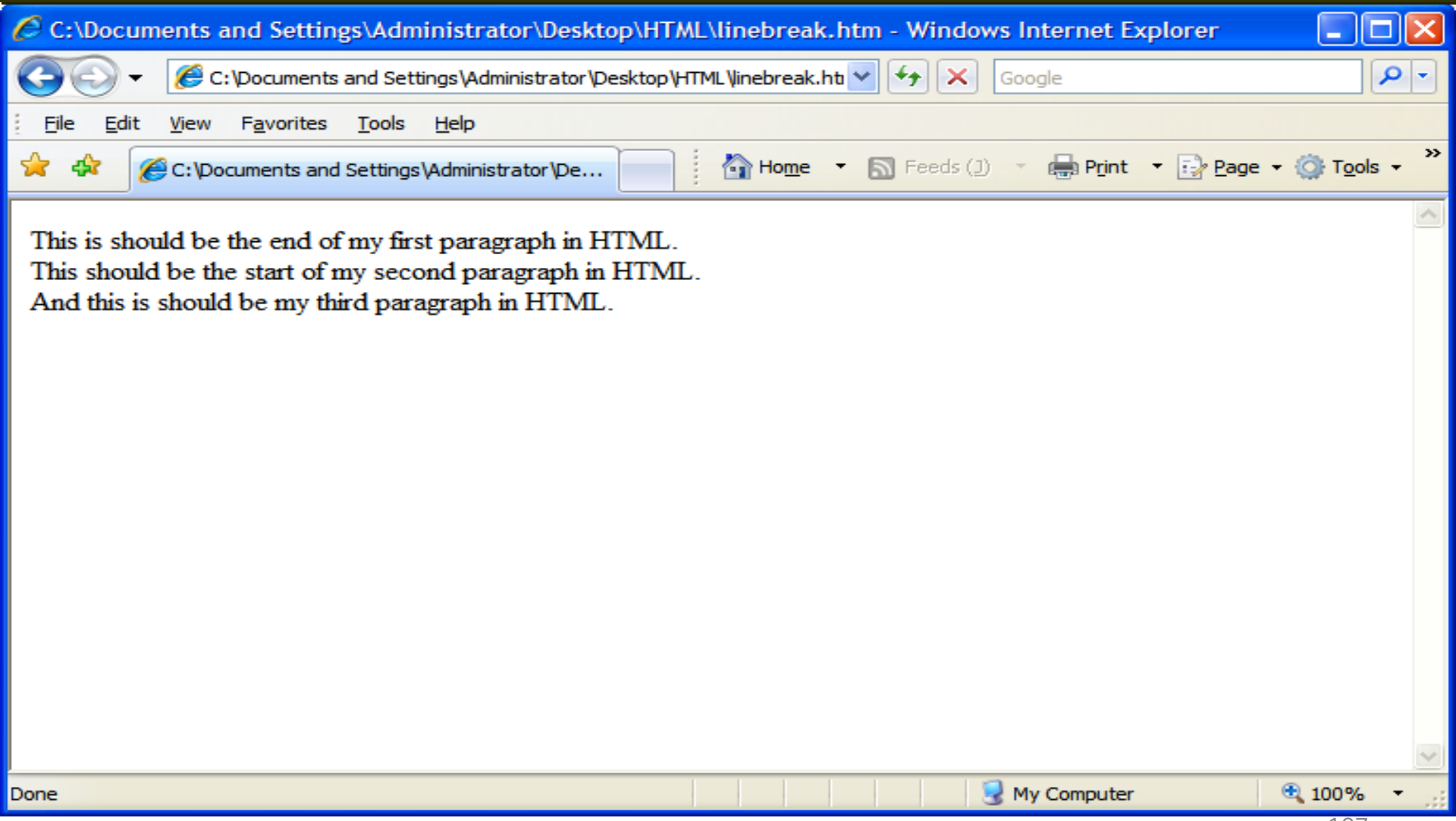


Ends the paragraph tag.

The
 Tag

- Used to control the line breaks.
- Use single spacing among the lines unlike <p> tag that uses double spacing among the paragraph.
- Does not have a closing braces.

Requirement Analysis



Code Snippet To Fulfill the Requirement

This is should be the end of my first paragraph in HTML.

This should be the start of my second paragraph in
HTML.

<p>And this is should be my third paragraph in
HTML.

Forces line break with one line spacing between two lines.

This is should be the end of my first paragraph in HTML.***
***

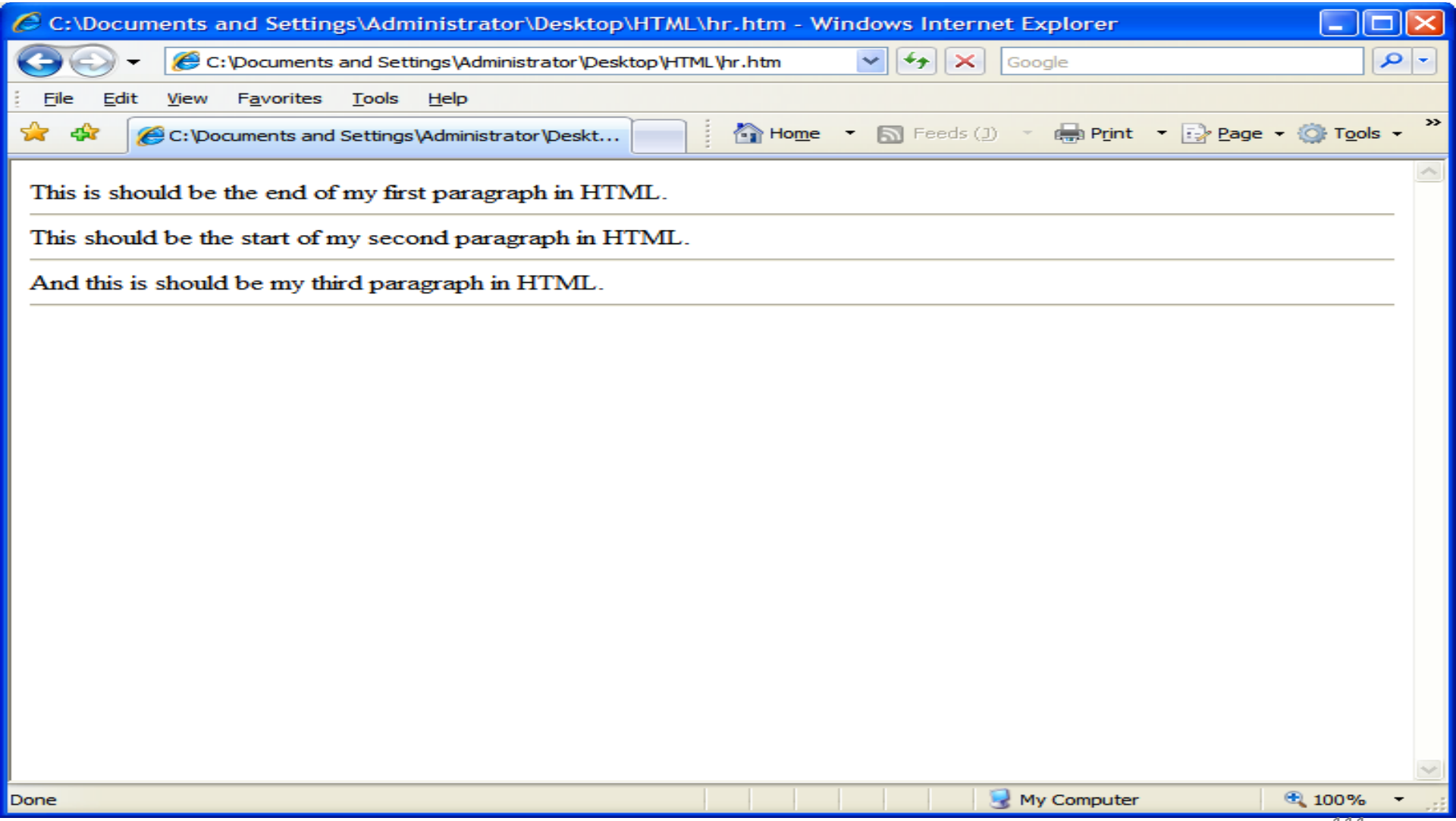
This should be the start of my second paragraph in HTML.***
***

<p>And this is should be my third paragraph in HTML.***
***

The <HR> tag

- The <HR> tag draws a horizontal line across the page.
- It acts a paragraph break.
- There is no need to use the
 before or after the <HR> tag.

Requirement Analysis



Code Snippet To Fulfill the Requirement

This is should be the end of my first paragraph in HTML.<hr>

This should be the start of my second paragraph in
HTML.<hr>

And this is should be my third paragraph in HTML.<hr>

Draws a horizontal line across the page.

This is should be the end of my first paragraph in HTML.***<hr>***

This should be the start of my second paragraph in HTML.***<hr>***

And this is should be my third paragraph in HTML.***<hr>***

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP

Thank you! ☺