**HTTP (hypertext transfer protocol)**

* Is an application for distributed, collaborative, hypermedia information systems.
* Is the foundation of data communication for the world wide web
* Development of HTTP was initiated by [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee) at [CERN](https://en.wikipedia.org/wiki/CERN) in 1989.

**CSS (Cascading style sheets)**

* Is a style sheet language used for describing the presentation of a document written in a mark-up language.
* CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications](https://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.
* **Client side service** - refers to operations that are performed by the **client** in a **client**–server relationship in a computer network. Typically, a **client** is a computer application, such as a web browser, that runs on a user's local computer or workstation and connects to a server as necessary.
* **IP address** – 4 decimal separated by dots; unique name. Is a numerical label assigned to each device (e.g., computer, printer) participating in a [computer network](https://en.wikipedia.org/wiki/Computer_network) that uses the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) for communication.
* **DNS (domain name system**) – system for search. Is a central part of the Internet, providing a way to match names (a website you're seeking) to numbers (the address for the website). Anything connected to the Internet - laptops, tablets, mobile phones, websites - has an Internet Protocol (IP) address made up of numbers.

**Internet (internet society) (internetwork) -** Is the global system of interconnected [computer](https://en.wikipedia.org/wiki/Computer) [networks](https://en.wikipedia.org/wiki/Computer_network) that use the [Internet protocol suite](https://en.wikipedia.org/wiki/Internet_protocol_suite) (TCP/IP) to link devices worldwide.It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.

1. **Logically linked together by a global unique address space based on the IP**

* **Intra** – within the boundaries.
* **Network-** interconnection of devices. a group or system of interconnected people or things.
* **Nodes** - devices that are connected. (Ex. Desktop, laptop). is a point of intersection/connection within a network. In an environment where all devices are accessible through the network, these devices are all considered nodes.
* **Interconnections -** is the physical linking of a [carrier](https://en.wikipedia.org/wiki/Common_carrier)'s [network](https://en.wikipedia.org/wiki/Telecommunications_network) with equipment or facilities not belonging to that network. The term may refer to a connection between a carrier's facilities and the equipment belonging to its customer, or to a connection between two (or more) carriers.

**Wired** – physical connection

**Wireless** – radio waves

* **Protocols** – communication standards (representing data). **-** is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols specify interactions between the communicating entities.
* **IPv4 (32 Bits) -** is the fourth version of the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP). It is one of the core protocols of standards-based internetworking methods in the [Internet](https://en.wikipedia.org/wiki/Internet), and was the first version deployed for production in the [ARPANET](https://en.wikipedia.org/wiki/ARPANET) in 1983. Is a [connectionless](https://en.wikipedia.org/wiki/Connectionless_communication) protocol for use on [packet-switched](https://en.wikipedia.org/wiki/Packet-switched) networks.
* **IPv6 (64 Bits) -** is the most recent version of the [Internet Protocol](https://en.wikipedia.org/wiki/Internet_Protocol) (IP), the [communications protocol](https://en.wikipedia.org/wiki/Communications_protocol) that provides an identification and location system for computers on networks and routes traffic across the [Internet](https://en.wikipedia.org/wiki/Internet).

1. **Is able to support communications using the transmission control protocol/ internet protocol (TCP/IP)**
2. **Provides uses or makes accessible, either publicly or privately, high level services layered on the communications**

* **1969** – Originally funded by ARPA (Advanced Research Projects Agency), now DARPA, within the United States Department of Defense, ARPANET was to be used for projects at universities and research laboratories in the US. The packet switching of the ARPANET was based on designs by British scientist Donald Davies and Lawrence Roberts of the by the end of 1969, ARPANET was able to connect to four locations: UCLA, UC Santa Barbara, SRI, and Utah. Using 4 computers.
* **Circuit switch** – establish actual connection. is a method of implementing a [telecommunications network](https://en.wikipedia.org/wiki/Telecommunications_network) in which two [network nodes](https://en.wikipedia.org/wiki/Network_nodes) establish a dedicated [communications channel](https://en.wikipedia.org/wiki/Communications_channel) ([circuit](https://en.wikipedia.org/wiki/Telecommunication_circuit)) through the network before the nodes may communicate.
* **1972** – Email was introduced by **Ray Tomlinson** that made use of the internet.
* **1989** – Tim Berners-Lee, a British scientist at CERN, invented the World Wide Web. The first website at CERN - and in the world - was dedicated to the World Wide Web project itself and was hosted on Berners-Lee's NeXT computer. The website described the basic features of the web; how to access other people's documents and how to set up your own server.
* **Sir Tim Berners-Lee** – (HTTP, HTML, URL, webservers, web client). is an English [computer scientist](https://en.wikipedia.org/wiki/Computer_science), best known as the inventor of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). he implemented the first successful communication between a [Hypertext Transfer Protocol (HTTP)](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) client and server via the Internet sometime around mid-November of that same year. the director of the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium). He is also the founder of the [World Wide Web Foundation](https://en.wikipedia.org/wiki/World_Wide_Web_Foundation), and is a senior researcher and holder of the [founders chair](https://en.wikipedia.org/wiki/Professors_in_the_United_States#Special_academic_ranks_.28tenured.29) at the [MIT Computer Science and Artificial Intelligence Laboratory](https://en.wikipedia.org/wiki/MIT_Computer_Science_and_Artificial_Intelligence_Laboratory)

**WAIS(Wechsler Adult Intelligence Scale)–** indexes; queries information. Is an IQ test designed to measure intelligence and cognitive ability in adults and older adolescents.

**Gopher(protocol)** – hierarchy of information; a [TCP/IP](https://en.wikipedia.org/wiki/TCP/IP) [application layer](https://en.wikipedia.org/wiki/Application_layer)[protocol](https://en.wikipedia.org/wiki/Communications_protocol) designed for distributing, searching, and retrieving documents over the Internet. The protocol was invented by a team led by [Mark P. McCahill](https://en.wikipedia.org/wiki/Mark_P._McCahill)

**Usenet** – Forum (Ex, newsfeed). is a worldwide distributed discussion system available on computers. It was developed from the general-purpose [UUCP](https://en.wikipedia.org/wiki/UUCP) [dial-up](https://en.wikipedia.org/wiki/Dial-up) network architecture.

**Webserver** – hosts website. Is a computer system that processes requests via [HTTP](https://en.wikipedia.org/wiki/HTTP), the basic [network protocol](https://en.wikipedia.org/wiki/Network_protocol) used to distribute information on the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). The term can refer to the entire system, or specifically to the [software](https://en.wikipedia.org/wiki/Software) that accepts and supervises the HTTP requests.

**HTTP**

* Use to access resources on the World Wide Web.
* Invented by Sir Tim Berners-Lee.
* Jointly developed by W3C and IETF.

**HTTP 0.9 (1991) –** This request consists of the word "GET", a space, the [document address](https://www.w3.org/Addressing/BNF.html" \l "1) , omitting the "http:, host and port parts when they are the coordinates just used to make the connection. The response to a simple GET request is a message in hypertext mark-up language ([HTML](https://www.w3.org/MarkUp/) ). This is a byte stream of ASCII characters.

**HTTP 1.0 (RFC 1945, may 1996)**

**HTTP 1.1 (RFC 2068, Jan 1997)**

**HTTP 1.1 (RFC 2616, Jun 1999)**

**HTTP 1.1 (RFC 7540, Jun 2014)**

**HTTP 2 (RFC 7540, May 2015) -** Patterned after SPDY.

* **HTTP runs on top of TCP/IP, using TCP port 80 by default or TCP port 443 for HTTPS (HTTP over SSL/TLS)**
* **HTTP is based on a client- server architecture**
* Clients, a.k.a user agents (UA)
* Web browsers, web crawlers/spiders, ad other end user tools and applications
* **NOTE:** port number – unique number assigned by operating system (0-64K)
* IP address + port number = socket
* **HTTPS** – encrypted communication
* **Servers** – origin servers

-proxy servers, gateways, tunnels

* **HTTP uses a request**- response standard protocols.
* The client sends an HTTP request message to the servers.
* The server processes the request and replies with an HTTP response message.
* HTTP is a stateless communications protocols

Server do not keep information about clients in between requests. “SERVE AND FORGET”

* **HTTP2 “PUSH”**
* HTTP provides support for other functionalities, such as:
* Cache control
* Content media type (MIME) specification
* Language and character set specification
* Content/transfer codings
* Content negotiation
* Client server protocol negotiations

MIME – multipurpose Internet Mail Extensions.

**HTTP Resource Addressing -** Routing the message

* HTTP resources are identified using URLs (RFC 3986) or more specifically, HTTP URLs

**URN** – naming without regard to where it is (Ex. ISBN)

**URL** – how to access (mechanism of finding)

* **Scheme (HTTP or HTTPS) – protocol**
* **Authority**

-user info or authentication credentials (deprecated)

-host (host name – where the resources)

- Domain name (resolved to an IP address using DNS) of the server where the resource reside (or will be created)

- Port number (default- 80)

* **Path to resource** (resolved relative to the document root on the server)
* May refer to a static or dynamic resource (static – html itself; dynamic – scripts)
* **Query** (starts with ? ) (Ex. Products.php?)

**HTTP request message**

* **Request line** (CRLF terminated line connecting three spaces separated values)
* Method
* Request URI
* HTTP protocol version
* **Message Headers** (general, request and or entity headers)
* HTTP 1.1 requires at least the host request headers to be provided (1 header required)(Ex. Fieldname: value)

1. **General headers fields** – can be used by clients and server. This section defines the syntax and semantics of all standard HTTP/1.1 header fields. For entity-header fields, both sender and recipient refer to either the client or the server, depending on who sends and who receives the entity.
2. **Request header fields** – used by clients. A request message from a client to a server includes, within the first line of that message, the method to be applied to the resource, the identifier of the resource, and the protocol version in use.
3. **Response header fields** – used by server. The response-header fields allow the server to pass additional information about the response which cannot be placed in the Status- Line. These header fields give information about the server and about further access to the resource identified by the Request-URI.
4. **Entity header fields** - Entity-header fields define metainformation about the entity-body or, if no body is present, about the resource identified by the request. Some of this metainformation is OPTIONAL; some might be REQUIRED by portions of this specification.

* **Empty line (CRLF)** - refers to Carriage Return (ASCII 13, \r) Line Feed (ASCII 10, \n). They're used to note the termination of a line, however, dealt with differently in today's popular Operating Systems.
* **Message body** a.k.a payload (optional) (message body – entity).

**HTTP response message**

* **Status line** (CRLF – line terminated line consisting of three space-separated values). The first line of a Response message is the Status-Line, consisting of the protocol version followed by a numeric status code and its associated textual phrase, with each element separated by SP characters. No CR or LF is allowed except in the final CRLF sequence.
* **HTTP protocols**
* **Status code**
* **Reason phrase**
* Information (1xx)
* Success (2xx)
* Redirection (3xx)
* Client error (4xx)
* Server error (5xx)
* **Reason phrase** – consumption for human beings (Ex. Success 2xx – 200 OK)
* Message headers (general, response, and /or entity headers)
* Entity line (CRLF)
* Message body (optional) – have body more than request message.

**HTTP Request methods (HTTP verbs)**

* **Standard methods**
* **Get** – most commonly used HTTP method**.**
* Most supported compliant general-method purpose servers. The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI.
* Transfer a current selected representation of the resource identified by the URI.
* **Head** – does not return the body. The HEAD method is identical to GET except that the server MUST NOT return a message-body in the response. The metainformation contained in the HTTP headers in response to a HEAD request SHOULD be identical to the information sent in response to a GET request. This method can be used for obtaining metainformation about the entity implied by the request without transferring the entity-body itself. This method is often used for testing hypertext links for validity, accessibility, and recent modification.
* Represents metadata.
* Used to retrieve metadata about the entity implied by the request without transferring the entity itself. (e.g to test or link validity or resource modification)
* Like GET, must be supported by all general purpose servers.
* **Post** – typically used in submitting HTML for data. The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.
* Perform resource specific processing of entities.
* **Put** – store the enclosed entity in the message body under the specified request URI (i.e the resource identified by the request URI is either created or replaced). The PUT method requests that the enclosed entity be stored under the supplied Request-URI.
* **Extention method eg. Webdav(RFC 4918)**
* **Profind, propatch, mkcol, copy, move, lock, unlock**
* **Options** – request information about the communication.
* **Delete** – remove the resource associated with the request URI. The DELETE method requests that the origin server delete the resource identified by the Request-URI. This method MAY be overridden by human intervention (or other means) on the origin server. The client cannot be guaranteed that the operation has been carried out, even if the status code returned from the origin server indicates that the action has been completed successfully. However, the server SHOULD NOT indicate success unless, at the time the response is given, it intends to delete the resource or move it to an inaccessible location.
* **Trace** – for diagnostic purposes (request/response chain). The TRACE method is used to invoke a remote, application-layer loop- back of the request message. The final recipient of the request SHOULD reflect the message received back to the client as the entity-body of a 200 (OK) response. TRACE allows the client to see what is being received at the other end of the request chain and use that data for testing or diagnostic information. The value of the Via header field (section [14.45](https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.45)) is of particular interest, since it acts as a trace of the request chain.
* request a loop back of the request message (i.e request the server to do echo back to the client the received request message)
* **Filter proxy** – inspection.  is one of these services that modifies either the requests or the returned information. One application of this would be to deny access to a particular web server by simply not requesting pages from that server (based on the server's IP address or name).
* **Chunked encoding** – length of data in hexadecimal. A transfer encoding. Not knowing how long the data. The content can be broken up into a number of chunks; each of which is prefixed by its size in bytes. A zero size chunk indicates the end of the response message. If a server is using chunked encoding it must set the **Transfer-Encoding** header to "chunked".
* **Connect**  - to use to establish a tunnel
* Request the establishment of a tunnel. This specification reserves the method name CONNECT for use with a proxy that can dynamically switch to being a tunnel
* **Safe methods** – methods when used are not modifying anything (get, head, option, trace)
* **Idempotence methods** – can be used several times (results the same when used) (get, ead, options, trace, put, delete)
* **Cacheable methods**

**HTTP Message Headers**

* **General** – may be used by client and servers
* **Cache control** – clients and server whether they can store.
* **Connection** – establish persistent connection.
* **Pragma** – generic header (used in HTTP 1.0)
* **Trailer** – header at the end of the entity.
* **Transfer** – encoding.
* **Upgrade** – different version
* **Via**
* **Warning**
* **Request Headers**

Content negotiation headers

Popular (compressed encoding)

* **Accept**
* **Accept charset**
* **Accept encoding**
* **Accept language**
* **Authorization** – provide authorization credentials
* **Expect** - used in two face operation/request
* Typical value expect: 100- continue
* **From** – email address where it came from.
* **Host** – required header in HTTP 1.0
* **If match**
* **If modified since**
* **If none match**
* **If range**
* **If- unmodified**
* **Max – forwards** – used for troubleshooting.
* **Referrer** – the site where the request originated
* **TE**
* **User agents**
* **Response**
* **Accept –ranges** – server had served resources.
* **Age** - “how old the response is” ; age of resource.
* **Entity (ETag)**
* **Location**
* **Proxy authenticate**
* **www authenticate**
* **reply after**
* **server**
* **vary**
* **Entity**
* **Allow** – what operations that can be done on the entity
* **Content- encoding**
* **Content-language** – important in the case of accessibility
* **Content- length** – how long the data in bytes.
* **Content- location**
* **Content – MDS (deprecated) ­**– message digest version 5
* **Content – range**
* **Content – type** – MIME type
* **Expires**
* **Last- modified**
* **HTTP status Codes**
* **Informational (1xx)**
* 100 continue
* 101 switching protocols – from upgrade request
* **Success (2xx)**
* 200 OK
* 201 Created
* 202 Accepted – received information but no process
* 203 Non- authorative information
* 204 No content
* 205 Reset Content
* 206 Partial content
* **Redirection (3xx)**
* 300 Multiple choices
* 301 Moved permanently
* 302 found
* 303 see other
* 304 not modified
* 305 used proxy
* 306 unused
* 307 temporary redirect
* **Client error (4xx)**
* 400 Bad request – request malformed
* 401 unauthorized
* 402 Payment Required – Ecommerce
* 403 Forbidden – not allowed to access
* 404 not found
* 405 method not allowed
* 406 not acceptable
* 407 Proxy authentication Required
* 408 request timeout
* 409 Conflict
* 410 Gone
* 411 Length Required
* 412 Precondition Failed
* 413 Request Entity too large
* 414 request URI too large
* 415 Unsupported Media Type
* 416 Request Range not satisfiable
* 417 Expectation Failed – Expectation cannot be met
* 426 Upgrade Required (newly implemented
* **Server error (5xx)**
* 500 Internal server error
* 501 Not implemented
* 502 Bad gateway
* 504 Gateway Time-out – busy
* 503 Service Unavailable
* 505 HTTP Version not supported

**HTML –** language to create a webpage.

* Mark –up language; used to mark up documents
* Derived from SGML (XML)
* **Structure and Content – Html**
* **Presentation/ aesthetics**  - how the page looks like – CSS
* **Behavioural –** how it interacts – Client side

**The role:** to provide structure and content

**Inventor:** Sir Tim Berners lee

**Oversees dev. Of HTML** – WWW

**Versions of HTML:** HTML 1.0 – original

HTML 2.0 – standard using RFC 1866

Specification – recommendation – to capture what is currently

HTML 3.0

HTML 3.2 – superseded by 4.0

HTML 4.0 – 1997

HTML 4.01 – 1999

**Ian Hickson** – decided to evolve 4.01

HTML 5.0 – 2016

**Documentation specify HTML standards – W3C Recommendation**

REFERENCES:

http://www.webopedia.com/TERM/H/HTTP.html

http://www.webopedia.com/TERM/W/World\_Wide\_Web.html

<https://en.wikipedia.org/wiki/Federal_Networking_Council>

www.w3c.org