Internet – internetwork, a global network or a global information system that is logically linked together by a globally unique address space based on IP.

* Began in 1969
* ARPANET –
* DARPA –
* 1972 – electronic mail/email

Inter – outside boundary

Intra – within a boundary

Network – connecting devices together

Nodes – different devices connected to the network.

IOT – internet of things

Network Protocol – standardized

WWW – a system made up of internet servers (1989) by Tim Berners Lee

WAIS – wide area information server, do thing document by document.

Gopher (protocol) – almost similar with WAIS

Usenet – forum ex. Stack overflow

CERN – center for European nuclear research

Web servers – web program that host web resources

Client app – browsers

Semantic Web – a more intelligent web

HTTP – application layer communication protocol used to access resources (hypertext/hypermedia) on the WWW.

* Invented by Tim Berners-Lee
* Jointly developed by the W3C and IETF
* Version history:

HTTP 0.9 (1991)

HTTP 1.0 (RFC 1945, May 1996)

HTTP 1.1 (RFC 2068 Jan. 1997, RFC 2616 June 1999) RFC 7230-7235 June 2014

HTTP 2 (RFC 7540 May 2015)

SPDY – protocol of Google

* HTTP Fundamentals – http runs on the 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
  + Client aka User Agents (UA):
    - Web browsers, web crawlers/spider, other end user tools and applications
* Servers:
* Origin servers –
* Proxy servers –

IANA –

ICANN –

HTTP – uses a request – response standard protocol

* The client sends an HTTP request message to the server
* The server processes the request and replies with an HTTP response message

HTTP – is a stateless communication protocol

* Push – server can now initiate, server contact the client
* Servers do not keep information about clients in-between request (only keeps access log).
* Pull –
* Polling – periodically check the new information manually or use script and it is not very efficient.

HTTP – provides support for other functionalities such as:

* Cache control – local storage
* Content media type (MIME) specification
* Language and character ser specification
* Content or transfer codings
* Content negotiation
* Client server protocol negotiations
* Persistent connections
* Request pipelining
* Authentication or authorization

MIME – multipurpose internet mail extensions (ex. Text, img…)

Google analytics – search engine, know the traffic of your website

Additional protocol:

* Server push – request something the server response and also request something. (server volunteer to request associated resources)
* HTTP Resource Addressing

HTTP Resource – are identified using URI’s or more specifically HTTP URL’s.

URN – name ex. ISBN an international book number or serial number, UPC a universal product code

URL – locator ex. Links

* Scheme – http/https
* Authority – information about the server, port, authentication
  + - User information or authentication credentials (deprecated)
    - Host – domain name (resolved to an IP address using DNS) of the server where the resource resides or will be created.
    - Port number – optional
* Path to resource (resolved relative to the document root on the server)

Relative URL –

Absolute URL –

* + Query – typically provided as key = value pairs, with ampersand separators between key or value pairs. – may be URL encoded
  + Fragment identifier – start with hash tag or pound sign ex. Page1.1#xyz…
* HTTP Request Message
* Request Line
  + Method – HEAD, GET, POST, PUT, DELETE, TRACE, CONNECT…
    - WebDAV - extension
  + Request URI – target
  + HTTP Protocol Version
* Message Headers (general, request, response, entity) ex. fieldname:value
  + General – used by either client or server
  + Request – client
  + Response – server
  + Entity – content

HTTP 1.1 – requires at least the Host request header to be provided

* Empty Line (CRLF)
* Message Body (payload) – optional
  + DNT – do not track, an extension
  + Upgrade – Insecure – Request – also an extension
  + Blank line after the Accept- Language – means terminate
* HTTP Response Message
* Status Line (CRLF terminated line consisting of 3 space separated values)
  + HTTP protocol version
  + Status Code – always the same
  + Reason Phrase – can change
  + Status Code
    - Information(1XX)
    - Success(2XX)
    - Redirection(3XX)
    - Client Error(4XX)
    - Server Error(5XX)
    - Message Headers (general, request and entity header)
    - Empty Line – terminate
    - Message Body – (payload) optional
  + Request header – ex. HOST, user-agents
  + General header – ex. connection both request and response
  + HTML, Script, CSS – text file
* HTTP Request Methods
* Standard Methods
  + GET – transfer of a current selected representation of the resource identified by the Request URI.
  + HEAD – same as GET, except that the entity is not included – used to retrieve metadata about that entity implied by the request without transferring the entity itself – like GET, must be supported by all general purpose servers.
    - Composer – you can compose a request and modify GET to HEAD
    - Code rot
    - Link rot –
      * Link checker – look for the value and try fetching.

Fiddler – a transparent proxy

* POST – perform resource specific processing of the entities enclosed in the message.
* PUT – store the enclosed entity in the message body under the specified Request URI (ei. The resource identified by the Request URI is either created or replaced using enclosed entity)

Dav- APACHE WAMP

* Extension Methods
* WebDAV (RFC 4918)
  + - PROPFIND, PROPATCH, MKCOL
  + OPTION - special syntax is to add \* - means asking server a general information
  + TRACE – request a loop back of the request message (ei. Request the server to echo back to the client the received request message) – typically used for testing or diagnostics of the request or response chain.
    - Transfer-Encoding: chunked – section of the data
  + CONNECT – request the establishment of a (encryption) tunnel to the destination origin server and if successful there after restrict it behavior to blind forwarding of packets, in both directions, until the tunnel is closed.

Fiddler – man in the middle – intercept

Categorize – safe method, not safe, idempotent method, cacheable method

* Safe Method – no changes has done – GET, HEAD, OPTION, TRACE
* Not safe – POST, PUT, DELETE
* Idempotent – GET, HEAD, OPTION, TRACE, DELETE, PUT – no changes even if you retrieve several times, same result – POST is not idempotent because POST can make changes
* Cacheable methods
* HTTP Message Headers

4 Categories

* General Header Fields
* Request Header Fields
* Response Header Fields
* General Header Fields
  + Cache control – (ex. Cache-control: max-age=0)
  + Connection – (ex. Connection: keep-alive)
  + Date –
  + Pragma –
  + Trailer -
  + Transfer encoding -
  + Upgrade – 101 switching protocols
  + Via -
  + Warning -
* Request Header Fields
  + Accept -
  + Accept Charset -
  + Accept Encoding -
  + Accept Language –
  + Authorization -
  + Expect –
  + From -
  + Host -
  + If – Match -
  + If – Modified – Since -
  + If – None – Match -
  + If – Range -
  + If – Unmodified – Since -
  + Max – Forwards -
  + Proxy – Authorization -
  + Range -
  + Referer -
  + TE -
  + User Agent -

Dedicated hosting –

Shared virtual hosting –

Stale – under If – None – Match

Etag – can be disabled

Precondition Failed – 412 – operation failed since it is already created.

* Response Header Fields
  + Accept – Ranges