January 20,2018

* WWW

= Collection of text-based web resources, applications that can be accessed via the internet.

= invented by Sir Tim Berners-Lee(1989)

= came up with HTML, HTTP, URL

= from CERN

= client-server architecture

= applications: -Webserver

* Webclient (e.g Browsers)
* Two web resources
  + Static
    - Exists already in the web
  + Dynamic
    - Produced on-the-fly (e.g Google Search)
* Webspiders

= collects/ crawls through a specific information and harvests it

* Internet

= International Network

= Global

* Hosts

= the devices (computer, cellphones, laptops, etc)

* FTP

= File Transfer Protocol

* URL

= way of addressing

* http://

= shows what type of protocol you are using

* DNS

= allows to map the hostname to an IP address

* Fiddler

= web debugger proxy

= eavesdrop in a web

= retrieve-response interface

* HTTP is based on a client-server architecture
  + Servers
    - Origin servers
    - Proxy servers
    - Gateway servers
  + Clients

= A.K.A user agents(UA)

* + - Web servers
    - Web crawlers/spiders
    - Other end user tools and applications
* HTTP

= is a stateless communications protocol

= servers do not keep information about clients in between requests

= serve and forget

January 24, 2018

//may diagram sa papel

* Physical layer = hardware
* Application layer

= apps e.g browsers, web

= doesn’t count itself with the physical layer

//may diagram ulit d2

* HTTP

= by Tim Berners-Lee (1989)

= joined by W3C and IETF

* Version
  + History

-HTTP 0.9(1991)

>W3.org/protocols/http/asImplemented.html

-HTTP 1.0 (RFC 1945, May 1996)

>tools.ietf.org/html/rfc1945

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

>tools.ietf.org/html/rfc2616

-HTTP 2 (RFC 7540 May 2015)

>SPDY=for faster transfer of resources(A Google Work)

>SeverPush= to lessen page-load time

>Pipelining= for HTTP 1.1, request wait response (HOLB-Head of Line Blocking)

=for HTTP 2, request, request, request

* HTTP typically functions on top of TCP/IP using TCP port 80 by default, or TCP port 443 for HTTPS (HTTP over SCL/TLS)

January 27, 2018

* Cache

= a copy of something saved in a local storage

* Content media type (MIME) specification

= MIME = Multipurpose Internet Mail Extension

= specifies the type of format a file has

* Content negotiation

= having multiple file formats in which a client may ask a specific form as you comply to what they prefer

* Language and Character set specification
* Content/ Transfer Coding

= e.g compression(zip, rar)

* Client-server protocol negotiations
* Persistent connections
* Request pipelining/multiplexing

= HOLB(Head of Line Blocking)

= to shorten latency

* Authentication/Authorization
* HTTP Resource Addressing
  + URI

= Uniform Resource Identifier

* + Parts
    - Scheme(http or https)
    - Colon
    - Authority

= user info

= host- domain name

= port number

* + - Path

= static/dynamic(document root)

* + Query

= typically provided as key

= clear with ampersand separator between key/value pairs (?id=1234), may be URL encoded

* + Fragment identifier

= A.K.A bookmark (#)

=e.g. <http://usr:pwd@server.org:81/info/profile.php?id=1234#addr>

* HTTP Request Message
  + Request Line

= Consists of 3 space-separated values

>Method >Request Target >Protocol Version

\* Message Headers

>General >Request >Entity

\* Empty Line

\* Message Body aka Payload

* HTTP Response Message
  + Status Line

>Protocol Version >Status Code > Reason Phrase

\* Message Headers

>General >Response >Entity

\*Empty Line

\*Message Body aka Payload

* Request Methods
  + GET

= transfer a selected representation of a target source, then retrieved resource is returned in the message body as an entity

= most commonly used HTTP method

* + HEAD

= identical to GET except that it doesn’t send a message body in response

= used to retrieve metadata about the target resource

* + POST

= typically used in submitting HTML form data

* + PUT

= create or replace the state of the target resource with the state defined by the representation enclosed in the request message payload

* + DELETE

= removes the association between the target resource and current functionality

* + OPTIONS

= request information about the communication options available for the target resource

= the request target may be \*

* + TRACE

= requests a remote, application-level loop back of the request message

= used for testing/diagnostics of the request/response chain

* + CONNECT

= establishment of a tunnel to the destination origin server

= used to create an end-to-end virtual connection, with one or more proxies, which can be secured using TLS.

* Method Properties

\* Safe Methods

= client does not request, and does not expect, any state change on the server (GET, HEAD, OPTIONS….)

\* Idempotent Method

= intended effect on the server of multiple identical requests with the method with the method and is the same as the effect of a single such request (GET, HEAD, OPTIONS, TRACE, PUT, DELETE)

\* Cacheable Methods

= the response to a method is allowed to be stored for future reuse (GET, HEAD, POST…)

* HTTP Message Headers
  + General Header Fields (RFC 2616)
  + Request Header Fields (RFC 2616)
  + Response Header Fields (RFC 2616)
  + Entity Header Fields (RFC 2616)
* HTTP Status Codes
  + Information (1xx)
  + Success (2xx)
  + Redirection (3xx)
  + Client Error (4xx)
  + Server Error (5xx)
* HTTP Extensions

-http can be extended by defining new request methods, message headers, and/or status codes, and then implementing servers and/or clients that make use of the new functionality

\* Request Methods (PROPFIND, MKCOL (4918 RFC)

\* Message Headers

\*Status codes

* HTML Form Data Submission
  + If(POST) = the query/url is shorter in the browser, the long is now a payload of the request (use fiddler) though it is idempotent and unsafe(verify upon refresh)
* Content Negotiation

= the client and server can negotiate about a data tey can exchange

* Range Request
* HTTP Extensions

= can be extended by defining new request methods, message headers and defining codes and implementing servers/clients that make use of the new functionality.

* WEBDAV (RFC 4918)

-Distributed Authoring and Versioning

-PUT, DELETE, COPY

Feb 17, 2018

* SGML

= Standard Generalized Mark-up Language

= used to define a language

= used to mark up documents

* Historically: HTML used to specify the structure and content of web pages, as well as their presentations.
* Modern: Solely for semantic specification of web page structure and content, with presentational aspects being delegated to style sheets
* Initially developed at CERN
* Versions: Pre-standarzation:
  + HTML tags: by Tim Berners Lee(late 1991)
  + HTML Internet Draft
* Standards: HTML 2.9(RFC 1866. Nov 1995)
* HTML 3.2 (W3C Recommendation. Jan 1997)
* HTML 4.0 (W3C Recommendation. Dec 1997)
* HTML 4.1 (W3C Recommendation. Dec 1999)
  + Introduced deprecation of presentational elements and attributes in favour of style sheets (Strict, Transitional, Frameset DTDs) // Document Type Definition
* XHTML

= reformulation of HTML in XML

= intended to facilitate the introduction of new HTML elements and/or attributes, provide interoperability between HTML- and XML-based systems, and leverage the use of XML tools in creating, editing, and parsing HTML

= sought to move the Internet content-development community from the days of malformed, non-standard HTML markup into the well-formed and valid world of XML

* + - XHTML 1.0 (W3C Recommendation, Jan 2000)
    - XHTML 1.1 Module-Based XHTML (W3C Recommendation, May 2001)

//\* DOM = Death of Model

February 23, 2018

* HTML 5

= arose from the effort to evolve HTML (instead of replacing it) to address backward compatibility issues that hindered the adoption of XHTML, as well as to address the emerging needs to modern web applications

= work on HTML5 was initially undertaken by the WHATWG(led byApple, Mozilla, and Opera) in 2004, with the W3C signifying interest to participate in the effort in 2006; the first draft of HTML5 was published in 2008(edited by Ian Hickson)

= HTML5 incorporated specifications from HTML4, XHTML1, and the DOM, and included various extensions of existing APIs and introduced new APIs as well

* + - HTML5 (W3C Recommendation, Oct 2014)//the first standard for HTML 5
    - HTML 5.1 (W3C Recommendation, Nov 2016)//minor revisions
    - HTML 5.2(W3C Recommendation, Dec 2017)//minor revisions
    - HTML Living Standard(WHATWG)//last updated 23 February 2018
    - Difference Between HTML4 and HTML5

// there are still old installing formats that are to be used in the new HTML that is why we are covering the old versions

* HTML Document Structure

> root element(html)

= declarative header section containing document metadata, delimited by the head element

// Metadata included are head element, title element, base element, link element, meta element, style element- most important when you are to publish data(documents) in the web

= document body containing the document’s actual content, delimited by either the body element or the frameset element (obsolete in HTML5)

* HTML Document Components
  + Document Type Declaration
    - HTML 4.01 DOCTYPEs
    - XHTML 1.0 DOCTYPEs
    - HTML5 DOCTYPEs

-Elements (HTML 4.01, XHTML 1.0, HTML5)

= Tags

=\* Content

-Content Models

// defines what can be contained in an element e.g <p> <body>

-Void Elements

//elements that doesn’t have contents e.g <br> <img><area>

=\*Attributes and Attribute Values (HTML 4.01, XHTML 1.0, HTML5)

-Global Attributes

//because they apply practically to the whole html e.g accesskey, class, contenteditable, dir, draggable, hidden, id, lang, spellcheck, style, tabindex, title, translate

-Custom Non-Visible Data Attributes

//has been standardized to start with “data-“ (without ung “ “)

-WAI-ARIA Attributes

//Web Availability Initiative arya. Assistive Technology. The “attributes” that gives components for assists.

//WebContentAccesibilityGuidelines= guidelines especially for PWD (search google)

//<DOCTYPE html>= case insensitively= is used nowadays to indicate whether you’re claiming that your document is in line to the standards

//Strict, Transitional, Frameset = 3 flavours of HTML

// DOM = Document Object Model Tree

html->

head-> body->

<-Title

-Element-Specific Attributes //pertain to specific properties

=\*MathML, SVG

//Mathematical Markup Language= allows you to have your mathematical markups e.g delta, alpha. //Scalable Vector Graphics= -bitmapped and –vector. Bitmapped= 1920x1080 in 1 byte=used for high quality images. Vector=for…like… you know… a house?... for like simple images, logos and icons. You can interact with it using scripts, which means u can use it for animations etc.

* Character References

//rep of characters that are not available in your keyboard, and also for characters that are used specifically only for HTML. e.g <p> a<b>c<p> 🡨 not good. Therefore there are &lt for less than etc.

* Comments

=<!-- …. -- >

* HTML PREPROCESSORS
  + =A language that allows you to write html in a certain way that gets a ……. Search google…e.g PUG, SLIM