**History of the World Wide Web**

In 1989, World Wide Web started by an English Physicist named Tim Berners-Lee. An organization called World Wide Web Consortium (W3C) at Massachusetts Institute of Technology (MIT) founded and still headed by Tim Berners-Lee and also participated in by several companies who are willing to create standards and ways to improve the quality of the Web. W3C also aims to provide access to the World Wide Web on the principle of no patent, no royalty due and easily adoptable by everyone. W3C has set the standards for HTML, XHTML, CSS and many other internet applications.

**Birth of (WWW) World Wide Web**

* 1989
  + Tim Berners-Lee Opinion
    - Tim Berners-Lee is an English Physicist who had been working since 1980 in Geneva for the (CERN) Conseil européen pour la recherche nucleaire. Anyhow, he was concerned and frustrated in sharing between many scientists at CERN for the reason that lack of information. And also the different platforms being used in communicating with each other becomes a barrier. So Tim Berners-Lee proposed his idea to create a new information system build on shared hypertext documents.
* 1990
  + Development of the Project
    - A basic software program that allows editing and viewing of hypertext documents.
* 1991
  + The Expansion of World Wide Web
    - The World Wide Web becomes accessible by all the employees in CERN and it became the tool to share informations.
* 1993
  + The World Wide Web Continue Beyond CERN
    - WWW was made available and free to all, free of patents and royalties, opening up the Web to commercial use.

**Three Corner Stones**

It was October of 1990 when Tim Berners-Lee created the three corner stones that stood as the foundation of the modern day's World Wide Web. These three fundamental technologies are the things which people may have seen on parts of web browsers.

1. **Hypertext Transfer Protocol or HTTP** was created for the World Wide Web and it is liked for its speed and simplicity. HTTP protocol requests the HTML document from the server and serves it to the browser. HTTP is responsible for the getting of linked resources from across the web. HTTP Protocol serves as the set of standards that allow people or users of the WWW to exchange data and information that are found on web pages. The "http://" in front of the address whenever users access the WWW tells the browser to make a connection to or communicate over HTTP. Some of the browsers in the modern world doesn't require "http://" in the front of the URL because of the reason that it is considered as the default method of communication. But it is also needed to separate it from other protocols like FTP that is why it is still kept in the browsers.

**Hypertext Transfer Protocol Secure or HTTPS** is a protocol that uses HTTP on a connection which is encrypted by the transport-layer security. To protect transmitted data from eavesdropping HTTP is used. For conducting financial transactions the HTTPS is the default protocol. HTTPS can also protect the users from censorship by an ISP or a government.

1. **Hypertext Markup Language or HTML** is the markup language for the web for constructing electronic documents which are called pages. Each of pages that are displayed on the WWW contains a series of connections to other pages named "Hyperlinks". A browser would not know how to load images or display text as elements or other elements without the existence of HTML. HTML code always make sure the proper or right formatting of images and texts so that Internet browser will display these things as they are intended to look. People could think of HTML as the bones which represents the basic structure of a web page and CSS as its skin that represents the appearance when it is overlaid in the HTML.
2. **Uniform Resource Locator or URL** is the unique address that determines where any given document that is living on the web or documents that are accessible over the Internet. It is a subset of URI and it is used to identify or locate to each resource through describing its primary access mechanism on the WWW. URL can be described as <protocol>://<node>/<location>. Example of URL would be https://www.facebook.com/, that is the URL for the Facebook Website.

**WWW vs. Internet**

WWW

"The Web exists because of programs which communicate between computers on the Net. The Web could not be without the Net. The Web made the net useful because people are really interested in information (not to mention knowledge and wisdom!) and don't really want to have know about computers and cables." - Tim Berners-Lee

The software part, which the World Wide Web can be called. It is a group of software services provided by and running on the internet. The information inside the Internet is the World Wide Web. It is the collection of HTML pages that are offered on the internet. To access the information from the various servers then the WWW uses HTTP Protocol. The Hyper Text Transfer Protocol (HTTP) is the one in charge with the documents, linking of files, and other resources inside the World Wide Web. Through sending Web pages that is organized in the form of websites is the way on how the information is sent and these web pages are connected with each other by hyperlinks. By the Uniform Resource Locator address, one can determine the web pages and other pieces of information in WWW.

Internet

The hardware part, which the Internet can be called. The Internet is identified as the collection of all hardware infrastructure existing in the network including routers, cables like fiber-optic, computer system, servers, bridges, satellites , cellular towers , copper wires, wireless connections and other more. Communications between its devices and a large variety of interactions , these are the things that the Internet network are fully supporting. No one ever owns the Internet even the government. They have no rights and authority over its operations, it is free and it is an open broadcast medium of hardware networking. Massive network of network, the Internet. It is an interconnection between a lot of smaller computer networks distributed around the world. By the use of underground cables, sub-oceanic cables ,satellite, links, over ground cables and wireless signals and other more, these networks are interlinked with each other. The Internet Protocol which is dealing with the transmission in packets and data as a whole governs the Internet. In order to view the World Wide Web, you have to access the Internet.

**NOTE: GUMAWA NA AKO NG COMPARISON TABLE, KAHIT ICOPY MO NALANG ITO , MAY NAKAREADY NA RIN AKONG CSS**

\<table id="comparison">

<tr>

<th>Internet</th>

<th></th>

<th>World Wide Web</th>

</tr>

<tr>

<td>Originated on 1969 but the network was opened to commercial interests in 1988</td>

<td>Estimated Year of Origin</td>

<td>English scientist Tim Berners-Lee invented the World Wide Web in 1989 but it was fully commercialized on April 1993</td>

</tr>

<tr>

<td>Internet Protocol</td>

<td>Works on the basis of/Governed by</td>

<td>Hyper Text Transfer Protocol</td>

</tr>

<tr>

<td>ARPANET</td>

<td>First version's name</td>

<td>NSFnet</td>

</tr>

<tr>

<td>Independent of the WWW<br>Internet is the base</td>

<td>Dependency</td>

<td>Requires the Internet to exist</td>

</tr>

<tr>

<td>Hardware</td>

<td>Nature</td>

<td>Software</td>

</tr>

<tr>

<td>The superset of WWW</td>

<td>Set</td>

<td>The subset of Internet</td>

</tr>

<tr>

<td>By IP Addresses, computing devices are identified</td>

<td>Identifier</td>

<td>By Uniform Resource Locator, Information pieces are identified</td>

</tr>

<tr>

<td>Computers, servers, routers, cables, bridges, cellular towers, wireless networks, copper wires, network of Computers, satellites and fiber-optic cables </td>

<td>Comprises</td>

<td>Text, images, video, audio, folders, files and documents stored in different computers</td>

</tr>

<tr>

<td>Connect Computers</td>

<td>Function</td>

<td>Connect People</td>

</tr>

</table>

**NOTE : DNS 🡪 WALA PANG GUMAGAWA**

**HTTP Version History**

HTTP Version Control

HTTP 0.9

* 1995
* The original version
* Connection Nature: terminated immediately after response
* Response: entity-body
* No HTTP headers
* Methods: GET URI

HTTP 1.0

* May 1996
* RFC-1945
* Achieved large dissemination
* Protocol: ASCII
* Connection Nature: terminated immediately after the response
* Methods:
  + GET uri http-version
  + HEAD uri http-version
  + POST uri http-version
* Standard status codes
  + 200 OK
  + 201 Created
  + 202 Accepted
  + 204 No Content
  + 301 Moved permanently
  + 302 moved temporarily
  + 304 Not modified
  + 400 Bad request
  + 401 unauthorized
  + 403 Forbidden
  + 500 Internal server error
  + 501 Not implemented
  + 502 bad gateway
  + 503 Service unavailable

HTTP 1.1

* January 1997
* RFC-2616
* Enhanced speed
* Enhanced cache management
* Connection Nature: long lived
* Protocol : text
* Additional methods:
  + PUT
  + DELETE
  + TRACE
  + OPTIONS
  + CONNECT
* Retrieve through alias in the DNS

HTTP 2.0

* May 2015
* RFC-7540
* Foundation was SPDY protocol
* Included header compression
* Added Status code:
  + 421 Misdirected Request
* Added server push
* Added pipelining feature for sending multiple requests

**REQUEST**

**Request Message:**

**CRLF (Carriage Return Line Feed) -** contains three values which are separated by spaces.

**Request CRLF**

1. Method
2. Request Target
3. Protocol Version

**Process**:

* + - 1. Message Body
      2. Empty Line
      3. Payload

\*HTTP 1.0 has at least one required header which is request header

\*Message headers: General, Request, Entity Header

\*Payload - found after the header, blank line means empty payload

**Request Methods**

1. GET – retrieves, transfers a selected resource into the message body aka entity. Commonly used method (RFC 7231, Sec. 4.3.1) (RFC 2616, Sec. 9.3).
2. HEAD – identical to get but doesn’t get the resource. It has no payload, only headers and metadata (RFC 7231, Sec. 4.3.2) (RFC 2616, Sec. 9.4).
3. POST – non-safe and non-idempotent method. It is utilized for processing resources from the payload given by a request (RFC 7231, Sec. 4.3.3) (RFC 2616, Sec. 9.5).

* Typically used in HTML form data.
* Any data that is being submitted is already in the payload aka message body and the information will not be in the URL.

\*Search-friendly URL – doesn’t contain any query.

1. PUT – utilized to modify all existing models of the resource being targeted by basing it from the payload contained by the request (RFC 7231, Sec. 4.3.4) (RFC 2616, Sec. 9.6).

* Web servers, by default, do not allow put request since it may affect the information.

1. DELETE – logically removes the relationship existing between the resource being targeted and its existing functions. It does not necessarily delete the true resource hence it may be modified by a human intervening from the server which has the true copy (RFC 2616, Sec. 9.7) (RFC 7231, Sec 4.3.5).
2. OPTIONS – allows you to query a particular resource on the server itself. Its response is usually other request methods. It is another way of ensuring the integrity of a resource.

\*A request may be noted as an asterisk (\*) wherein it represents a global or general resource (RFC 7231, Sec. 4.3.7) (RFC 2616, Sec. 9.2).

1. TRACE – echoes back to the user or client what is happening to the request or received request message. It can manipulate the message (RFC 7231, Sec. 4.3.8) (RFC 2616, Sec. 9.8).

* Typically used for testing, diagnosing the request, response chain and troubleshooting.

\*Response chain – set of nodes from client to server.

1. CONNECT – set ups the connection to the requested server of the resource being targeted (RFC 7231, Sec. 4.3.6) (RFC 2616, Sec. 9.9).

\*Link Rot – linking to different pages but these pages aren’t controlled by you so some links may not exist anymore and if not updated, the website will degrade due to dead links. Solution: Update.

**Method Properties:**

1. Safe Methods – used only for read-only purposes. It can’t modify the resource but only retrieve it (RFC 7231, Sec 4.2.1).

* GET, HEAD, OPTIONS, TRACE

1. Idempotent Methods – it refers to the semantics of the method itself (RFC 7231, Sec 4.2.2).

* GET, HEAD, OPTIONS, TRACE, PUT, DELETE

1. Cacheable Methods – specifies that a certain resource given or sent by the server may be stored and retrieved for future purposes (RFC 7231, Sec 4.2.3).

* GET, HEAD, POST

**RESPONSE**

**Response Message:**

**Response CRLF:**

1. Protocol Version
2. Status Code
3. Reason Phrase

Example: HTTP/1.1 200 OK

**Process**:

1. Message Body
2. Empty Line
3. Payload

**Message Headers**

Categories:

1. General Header Fields – these are fields which can be utilized by both the client and server (RFC 2616, Sec. 4.5).

* Cache-Control – specifies the different rules and guidelines that must be observed and upheld by each and every cache-related operation in the request or response chain (RFC 2616, Sec. 14.9).
* Trailer – gives indication if a certain field in the header is made available in the message’s trailer which contains transfer-coding (RFC 2616, Sec. 14.40).
* Connection – gives the sender options for persistent connection (RFC 2616, Sec. 14.10).
* Upgrade – gives the client options on additional functional features of the communication protocols (RFC 2616, Sec. 14.42).
* Date – specifies the details of message on when and what time it was sent or received (RFC 2616, Sec. 14.18).
* Via – utilized by both gateways and proxies to specify certain protocols which will serve as bridging the user agent and server as well as between the server and client with regards to requests and responses, respectively (RFC 2616, Sec. 14.45).
* Pragma – identifies rules and guidelines that may correspond to the receiver present in the response or request chain. It originated from HTTP 1.0 and it also represents any new functionality not yet standardized i.e. not yet listed to any RFC’s (RFC 2616, Sec. 14.32).
* Warning – identifies supplementary information with regards to the state of a message sent or received (RFC 2616, Sec. 14.46).
* Transfer-encoding – specifies if the message body has been edited or transformed. This is important so that the message can be safely sent (RFC 2616, Sec. 14.41).

1. Request Header Fields - these are header fields which can only be seen in client-generated request messages (RFC 2616, Sec. 5.3).

* Accept – indicates what types of media are allowed to be made available for the response (RFC 2616, Sec. 14.1).
* Accept-Charset - indicates what types of set of characters are allowed to be made available for the response (RFC 2616, Sec. 14.2).
* Accept-Encoding – analogous with the method Accept with the exception that it limits the coding of the content that are made available to the response (RFC 2616, Sec. 14.3).
* Accept-Language – limits the preferred natural languages (RFC 2616, Sec. 14.4).
* Authorization – contains the credentials of the user-agent (RFC 2616, Sec. 14.8).
* Expect – expectations of the server which are indicated by the client (RFC 2616, Sec. 14.20).
* From – contains the email address of the user (RFC 2616, Sec. 14.22).
* Host – indicates the Internet host as well as the resource’s number of the port that is being queried (RFC 2616, Sec. 14.23).
* If-Match – used to compare resources through their entity tags. If it matches, the response is a GET request (RFC 2616, Sec. 14.24).
* If-Modified-Since – used to compare resources especially if the resource may have been updated and if not, the server will not receive any representation of the resource (RFC 2616, Sec. 14.25).
* If-None-Match – opposite of the IF-Match header (RFC 2616, Sec. 14.26).
* If-Range – sends parts of the entity if it is not changed, else, it sends the whole entity (RFC 2616, Sec. 14.27).
* If-Unmodified-Since – another way to compare resources and works as If-Modified-Since header’s polar opposite (RFC 2616, Sec. 14.28).
* Max-Forwards – delimits the number of either gateways or proxies that are able to send request by making use of the TRACE and OPTIONS methods (RFC 2616, Sec. 14.31).
* Proxy-Authorization – authorizes the client with the use of proxy which will require the client to authenticate itself (RFC 2616, Sec. 14.34).
* Range – size (RFC 2616, Sec. 14.35).
* Referer – not a typo in this circumstance. It contains the URI address of the resource (RFC 2616, Sec. 14.36).
* TE – lists the accepted transfer-encoding’s extension (RFC 2616, Sec. 14.39).
* User-Agent – lists the user-agent’s information (RFC 2616, Sec. 14.43).

1. Response Header Fields – these are header fields which can only be seen on the server’s side. This requires authentication (RFC 2616, Sec. 6.2).

* Accept-Ranges – lists the acceptable range of requests of resource as indicated by the server (RFC 2616, Sec. 14.5).
* Age – estimated time duration it took for the server to process and create the response and send it to the receiver (RFC 2616, Sec. 14.6).
* ETag – means entity tag which is used to identify which entity a certain resource belongs to (RFC 2616, Sec. 14.19).
* Location – utilized for diverting to another address and not the address which was requested (RFC 2616, Sec. 14.30).
* Proxy-Authenticate – challenges the credentials contained in the Proxy-Authentication (RFC 2616, Sec. 14.33).
* Retry-After – may be utilized for the response 503, Service Unavailable, to let the client know the amount of time the service is unavailable (RFC 2616, Sec. 14.37).
* Server – lists the information with regards to the server (RFC 2616, Sec. 14.38).
* Vary – specifies available header fields which correspond to requests (RFC 2616, Sec. 14.44).
* WWW-Authenticate – a definite inclusion in response 401, Unauthorized, messages. It challenges the credentials with regards to the URI requested (RFC 2616, Sec. 14.47).

1. Entity Header Fields – gives description to the entity which is in the payload (RFC 2616, Sec. 7.1).

* Allow – indicates to the receiver the valid methods applicable to a resource (RFC 2616, Sec. 14.7).
* Content-Encoding – gives information on the type of media of the entity-body (RFC 2616, Sec. 14.11).
* Content-Language – the natural language of which the entity is presented as (RFC 2616, Sec. 14.12).
* Content-Length – corresponds to the entity’s size according to its body or length of body (RFC 2616, Sec. 14.13).
* Content-Location – where the resource is located (URI) (RFC 2616, Sec. 14.14).
* Content-MD5 – used for message integrity check (MIC) contained in the entity-body (RFC 2616, Sec. 14.15).
* Content-Range – specifies the entity-body’s total length (RFC 2616, Sec. 14.16).
* Content-Type – corresponds to what kind of media the body of the entity is (RFC 2616, Sec. 14.17).
* Expires – corresponds to the date and time the response will expire (RFC 2616, Sec. 14.21).
* Last-Modified – date and time the entity was last changed (RFC 2616, Sec. 14.29).
* Extension-header – this is the message header (RFC 2616, Sec. 7.1).
* For a Full list go to iana.org <a href=http://www.iana.org/assignments/message-headers/message-headers.xhtml</a>

**HTTP Status Codes**

**I. Informational 1XX**

Informational status codes indicate connection status or progress regarding the request.

1. 100 (Continue) – the continue status code implies that the introduction to a request has been accepted.
2. 101 (Switching Protocols) – the switching protocols status code denotes that the server understands the client’s request and is ready to fulfill it.
3. 102 (Processing) – the processing status code informs the client that the server has accepted the request but has not finished it yet.

**II. Success 2XX**

Success status codes indicate that the client's request has been received and accepted.

1. 200 (OK) – the OK status code shows that the request was successful.
2. 202 (Accepted) – the accepted status code points out that a specific request is accepted even if it is only being processed.
3. 203 (Non-Authoritative Information) – the non-authoritative information status code indicates a disclaimer that a server obtained information from a third party server.
4. 204 (No Content) – the no content status code informs that the request to the server is successfully processed but cannot return anything because there is no content.

**III. Redirection 3XX**

Redirection status codes indicate that you must do something in order to get to the destination URL.

1. 301 (Moved Permanently) – the moved permanently status code points out that the URL location that has been redirected.
2. 302 (Found) - the found status code informs the location of the temporary URL because of changes in files in the server of the web.
3. 304 (Not Modified) – the not modified status code indicates that the URL is not available at the moment.

**IV. Client Error 4XX**

Client Error status codes indicate that something wrong is being done by the client.

1. 400 (Bad Request) – the bad request status code indicates that there are syntactical errors that cannot be understood by the server.
2. 401 (Unauthorized) – the unauthorized status code resembles status 403(Forbidden) but in order to enter, it just needs authentication.
3. 403 (Forbidden) – the forbidden status code indicates that it will not permit you to enter because there are no protocols for authorization.
4. 404 (Not Found) – the not found status code means that what you’re looking for is temporarily unavailable or it cannot be found.

**V. Server Error 5XX**

Server Error status codes indicate that there is an error while a client’s request is being processed.

1. 501 (Not Implemented) – the not implemented status code implies that the server does not know the process needed to satisfy a request.
2. 502 (Bad Gateway) – the bad gateway status code indicates that a server was acting as a gateway and received an invalid response.

**NOTE : EXTENSIONS 🡪 WALA PANG GUMAGAWA**

**NOTE : RFC’s 🡪 WALA PANG GUMAGAWA**

HTML HISTORY AND VERSION

**What is HTML?**

* The term HTML is an acronym for HyperText Markup Language. It is a markup language used for creating Web pages in World Wide Web. It is actually a product or application of SGML (Standard Generalized Markup Language) which is a specialization of something more general for the markup language.
* SGML is a language that define a markup language.
* Basically, HTML provides the structure, lay-out, content and format of a Web pages.
* Initially it was developed at :
  + CERN
  + IETF
  + W3C
  + WHATWG
* HTML is used purely for semantic specification of structure in Web pages.
* History Version:
  + Pre – standard
    - Late 1991 by Tim Berners-Lee , HTML tags
    - IETF in mid 1993 by Tim Berners-Lee and Dan Conolly, HTML Internet Draft
    - Late 1993 by Dave Raggett, Internet-Draft, "HTML+Hypertext Markup Format
  + Standard
    - HTML 2.0
      * Form Based File Upload
      * HTML Tables
      * Client Side Image Maps
      * Internalizations of HTML
    - HTML 3.2
    - HTML 4.0
    - HTML 4.1

**HTML BASIC STRUCTURE**

DOCTYPE

<!DOCTYPE HTML> which is usually found at the very top of the page before any HTML tags, is called a document type declaration that identifies the document and tells the browser that it is in HTML5 version. This doc type should appear only once at the beginning of the code and the declaration is not case sensitive. <!DOCTYPE HTML> helps the browsers to display the pages of the web in a correct manner or way.

HTML Root Element

The <HTML> serves as the root element that defines the whole HTML document. It holds all of the code and wraps the content in the HTML document. It identifies which the web page is written in HTML. It usually contains each of <head> and <body> elements. The </HTML> tag is the sign that it is the end of the web page. When a web developer used element something like this, <HTML lang="en">, it means that it declares that language of the document is English.

**ELEMENTS AND TAGS**

To mark up or create the content, elements and tags always work together. HTML elements role is to identify the importance of a tag. Tag determines the start and the end of an element, the thing that is inside of each element is called content. Not all of the elements are required to have an end tag like <br>. For each and every element, it has its meaning like example <p>Content</p>, the <p> element is used whenever a web developer will write a paragraph.

There are two main parts for every HTML document. The <head> and <body> elements have both end tags. The <head> element is responsible in containing the meta information of the HTML document and the <title> element that is the only element required to be put inside the <head> tags which indicates the title for the document or simply specifies the web page title. This <head> element could also include a lot of HTML elements that that cannot be seen as parts of the page when rendered by the browser. These elements are either meta data or are the things that is helping in pulling the external resources like JavaScript files or CSS style sheets. The HTML elements which can be used by a web developer that can be put inside the <head> element are the <base>, <link>,<noscript>,<style>,<meta>,<title> and the <script>.

The <body> element serves as a container of the content of a web page that defines the HTML document body, it is considered as the information that a person want to display in a web page. The <body> element should be only one therefore all of the things that a user can see whenever the browser is rendered is contained in the <body> element.

**COMMENTS**

Web developers can easily understand each other specially when one of them will give a guide through the use of comments, comments in HTML starts with <!-- and ends with -->. These comments are ignored and not seen by users whenever the web page is rendered.

References

Internet vs World Wide Web. (n.d.). Retrieved February 16, 2018, from https://www.diffen.com/difference/Internet\_vs\_World\_Wide\_Web

Gil, P. (2017, November 2). What's the Difference Between the Internet and the Web? Retrieved February 16, 2018, from https://www.lifewire.com/difference-between-the-internet-and-the-web-2483335

Kumar, L., & Says, R. C. (2016, December 17). Difference Between Internet and WWW: Discussion on www vs Internet. Retrieved February 16, 2018, from https://techwelkin.com/difference-internet-vs-www

Lee, T. B. (n.d.). Frequently asked questions. Retrieved February 16, 2018, from https://www.w3.org/People/Berners-Lee/FAQ.html

History of the Web. (n.d.). Retrieved February 16, 2018, from https://webfoundation.org/about/vision/history-of-the-web/

Lumsden, A. (2012, September 25). A Brief History of the World Wide Web. Retrieved February 16, 2018, from https://webdesign.tutsplus.com/articles/a-brief-history-of-the-world-wide-web--webdesign-8710

What is HTML (HyperText Markup Language)? (2017, December 20). Retrieved February 16, 2018, from https://www.computerhope.com/jargon/h/html.htm

What is HTTP (HyperText Transfer Protocol)? (2017, October 11). Retrieved February 16, 2018, from https://www.computerhope.com/jargon/h/http.htm

What is URL (Uniform Resource Locator)? (2017, October 30). Retrieved February 16, 2018, from https://www.computerhope.com/jargon/u/url.htm

RFC2616 Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys, J.

Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. June 1999.

(Format: TXT=422317, PS=5529857, PDF=550558 bytes) (Obsoletes

RFC2068) (Obsoleted by RFC7230, RFC7231, RFC7232, RFC7233, RFC7234,

RFC7235) (Updated by RFC2817, RFC5785, RFC6266, RFC6585) (Status:

DRAFT STANDARD) (DOI: 10.17487/RFC2616)

RFC7231 Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. R.

Fielding, Ed., J. Reschke, Ed.. June 2014. (Format: TXT=235053

bytes) (Obsoletes RFC2616) (Updates RFC2817) (Status: PROPOSED

STANDARD) (DOI: 10.17487/RFC7231)

**NOTE : BRITANNY PAKI APA FORMAT**

References: https://tools.ietf.org/html/rfc7231#page-48 (accessed on February 21, 2018) https://httpstatuses.com/ (accessed on February 21, 2018)

https://www.websitepulse.com/blog/what-is-5xx-status-code (accessed on February 22, 2018)

https://www.websitepulse.com/blog/what-is-4xx-status-code (accessed on February 22, 2018)

https://www.websitepulse.com/blog/what-is-3xx-http-status-code (accessed on February 22, 2018)

https://www.websitepulse.com/blog/what-is-2xx-http-status-code (accessed on February 22, 2018)

Penland, J. (2017, July 03). Document Tags. Retrieved February 24, 2018, from <https://html.com/document/>

(n.d.). Retrieved February 24, 2018, from <https://www.w3schools.com/html/html_elements.asp>

HTML 5.2. (n.d.). Retrieved February 24, 2018, from https://www.w3.org/TR/html52/semantics.html#semantics