**January 30 – February 2**

**HTTP**

* Application layer communications protocol used to share resources (hypertext/hypermedia) on the World Wide Web.
* Invented by Tim Berners-Lee (Father of Modern Web).
* Jointly developed by the W3C and the IETF.
* Runs on top of TCP/IP, using TCP port 80 by default or TCP port 443 for HTTP (HTTP over SSL/TLS).
* Is based on a client-server architechture.
* Uses a request-response standard protocol.
* The client sends an HTTP request message to the server then the server processes the request and replies with an HTTP response message.
* Is a stateless communications protocol.
* Servers do not keep information about clients in-between requests.

**Versions**

* HTTP 0.9 (1991)
* HTTP 1.0 (RFC 1945, May 1996)
* HTTP 1.1 (RFC 2068 January 1997, RFC 2616 June 1999), RFC 7230 – 7235 (June 2014)
* HTTP 2 (RFC 7540, May 2015), Backwards compatible

**IP Addresses and Socket** – called “network socket”.

**Client-server Architechture**

* Host web resources.
* Ideally it is a computer connected to other computer.
* Most cases it undergo different nodes.

Client

Proxy Server

Gateway

Server

**Web crawlers/spiders** – read different websites and send keywords to the database in order to be used when a client is searching something.

**Server** – any application that uses HTTP; (proxy servers, gateways, tunnels).

**Polling** – ask for the server if there is any new that be needed to retrieve. (every 30 second/minitue/etc.).

**HTTP 2 –** can implement push notifications; real time notifications.

**HTTP provides support for other functionalities, such as:**

* Cache control
* Content media type (MIME – Multipurpose Internet Mail Extensions) specification
* Language and character set specification
* Content/tranfer codings
* Content negotiation
* Client-server protocol negotiations
* Persistent connections
* Request pipelining
* Authentication/authorization

**HTTP Resource Addressing** – HTTP resources are identified using URIs (RFC 3986) or more specifically, HTTP URLs.

* Scheme (http or https)
* Authority
* 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
* Path to resource (resolved relative to the document root on the server).
* Query – typically provided as key = value pairs, with ampersand (&) separators between key/value pairs.
* Fragment identifier