**Hyper-Text Transfer Protocol (HTTP)**

* Protocol-access resources (hypertext/ hypermedia) on the World Wide Web (WWW)
* Sir Tim Berners-Lee
* Jointly developed by the W3C and the IETF
* Version History:
* HTTP 0.9 (1991)
* HTTP (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)
* SPDY – Protocol designed by Google

**HTTP Fundamentals**

* 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 Browser, Web Crawlers/Spiders, other End User tools and Application
* Servers:
* Origin servers
* Proxy servers, gateway, tunnels
* 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
* Servers do not keep information about client in-between requests
* HTTP provides support for other functionalities such as:
* Cache control
* Content media type (MIME) specification
* MIME- Multipurpose Internet Mail Extensions
* e.g. text/html (.xls), img/jpg (.jpg)
* Language and character set specification
* Content/Transfer coding
* Content negotiations
* Client-server protocol negotiations
* Locality of Reference Protocol
* Persistent connection
* Request pipelining
* Request one after another
* Authentication/Authorization
* etc.

**HTTP Resource Addressing**

* HTTP resources are identified using URIs (RFC 3986), or, more specifically
* HTTP URIs:
* Scheme (http/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 wil be created)
* Port number
* Path to resource (resolved relative to the document root on the server)
* May refer to a static or dynamic resource
* Static – script programs running on servers
* Dynamic – server as is
* Query
* Typically provided as key=value pairs, with ampersand (&) separators between key/value pairs
* May be URL encoded
* Fragment Identifier
* # - identify id
* + - space
* & - separation keys
* e.g

products.php?id=123&color=red&size=xL