Internet

The internet is a global information system that is logically linked together by globally unique address. It is able to support communications using time TCP/IP. It can also be described like a network of networks that consists of millions of private, public, academic, business, and government networks, of local to global scope. Vinton Cerf is the father of the internet

* Etymology
  + : Inter – Crossing Boundaries
  + : Network – Connection
* Age: 48, started around 1969. First began with the ARPANET

WWW

Allows documents to be connected to other documents by hypertext link

* Age: 28, started around 1989
* Creator: Tim Berners-Lee
* WAIS: Wide Area Informaation Servers, a specialized database
* GOPHER (protocol): designed for searching, retrieving, and indexing information in services
* USENET: Newsgroup

HTTP stands for Hypertext Transfer Protocol. It is a stateless communications protocol application layer used to access resources (hypertext/hypermedia) on the WWW, invented by Tim Berners-Lee. It was developed by W3C (World Wide Web Consorcium) and IETC (Internet Engineering Task Force)

## HTTP versions

* HTTP 0.9 (1991)
* HTTP 1.0 (RFC 1945, May 1996)
* HTTP 1.1 (RFC 2068, Jan 1997, RFC 2616 Jun 1999) RFC 7230 – Jun 2014
* HTTP 2 RFC 7540 May 2015 (push protocol)

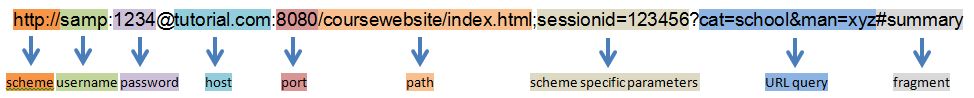
## HTTP Fundamentals

HTTP usually runs on top of TCP/IP using TCP using port 80 by default, or TCP port 443 for HTTPS (HTTP over SSL/TLS (Transport Layer Security)). HTTP resources are identified using URIs (Uniform Resource Identifier) specifically,HTTP URLs

## HTTP Resource Addressing

* Scheme - ex. http: or https:
* Authentication Information Protocol (optional)
* Host - The host name identifies the host that holds the resource. For example, www.example.com. A server provides services in the name of the host, but hosts and servers do not have a one-to-one mapping.
* Port number (optional) - Most servers use the well-known port numbers for HTTP and HTTPS , so most HTTP URLs omit the port number.
* Path - The path identifies the specific resource in the host that the web client wants to access. For example, /software/htp/cics/index.html.
* Scheme (optional)-The scheme identifies the protocol to be used to access the resource on the Internet. It can be HTTP (without SSL) or HTTPS (with SSL).
* URL-encoded query (optional) -The query string is usually a string of name and value pairs; for example, term=bluebird. Name and value pairs are separated from each other by an ampersand (&); for example, term=bluebird&source=browser-search.
* Bookmark/Fragment identifier (optional)

#### Example



HTTP is based on client-server architecture

## Client

Clients,AKA User Agents(UA) host web resources (HTML, CSS, JavaScript)

* Client requests to the server
* User agents – accessing resources
* Examples: Web browsers, web crawlers/spiders, email clients, webtools & app

## Server

The server process the requests and replies with an HTTP response message.

* Origin server – actual pages or CSS
* Proxy Servers
* Gateways
* 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 communications protocol

* Servers do not keep information about clients in between requests
* Web applications effect session tracking using mechanism such as cookies on URL-encoded session information to keep track of related client requests

### Important notes:

* HTTP 1.1 , uses a pull protocol where once a client sends a requests, the server will give what is needed.
* HTTP 2 , uses the push protocol where a server can push information.
* Stateless communications protocol ( servers do not keep information about clients in-between requests)
* Cookie - A very small text file
* Cache - Local storage/copy of resource that is fetched from a server

### Acronyms:

* TLS - Transport Layer Security
* Poll - frequent Requests
* MIME - Multipurpose Internet Mail Extensions (text/html)
* DNS - Domain Name System

## HTTP Provides support for other functionalities such as:

### Cache control

* Proxies can be caching servers
* Note: Locality of Memory Principle

### Content media type(MIME) specification

* Tell you what type of content it is. Example: text/html

### Language and character set specification

### Content/transfer codings

* Example: Compression

### Content negotiation

* Example: Mobile and desktop has different content even if it's the same website

### Client server protocol negotiation

* If connections have different protocols. Example: HTTP 1.0 and HTTP 2
* Connections have to negotiate which protocol to use

### Persistent connections

* Introduced in HTTP 1.1
* In HTTP 1, connections where non-persistent so once a request has been fulfilled the connection would cut off. This made sense before since websites back then have little content. However nowadays the non-persistent connection cannot be used anymore. because of overhead. More sites have more requests and for each request there will be a transfer, which called for the need of persistent connections. So when asking for a request, it will inform the server not to close the connection.

### Request pipelining

* Introduced in HTTP 1.1
* There is still overhead in waiting for one request to be finished in a persistent connection. Request pipelining enables you to fetch the next instruction EVEN IF the first one isn't done yet.
* HTTPS
* Status line
  + HTTP protocol version
  + Status code
  + Reason phrase
* Response headers
* Empty line
* Message body

GET

* Most commonly used HTTP method
* Can be combined with conditional and/or range request headers to effect conditional and/or partial resource retrieval
* Must be supported by all general-purpose servers.

HEAD

* Identical to GET, except the message body is not included in the response

POST

* Request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line.

OPTIONS

* Request for information about the communication options available on the request/response chain identified by the Request-URI. This method allows the client to determine the options and/or requirements associated with a resource, or the capabilities of a server, without implying a resource action or initiating resource retrieval.

TRACE

* Request the server to “echo” back to the client the received request
* Typically used for testing/diagnostics of the request chain

PUT

* Request the server to store the enclosed entity in the message under the specified request URI

DELETE

* Request the server to delete the resource identified by the request URI

CONNECT

* Reserved for use of tunneling proxy servers

Idempotent Methods

* The methods GET, HEAD, PUT and DELETE share this property.
* D
  + DARPA -Defense Advanced Research Projects Agency
    - - an Agency from the US defence task to invent technology to be used by the military.
    - - working on packet switched vs circuit switch
    - - store and forward
  + Discussion System- Collection of newsgroup where users post messages and distributed via usenet servers
  + DNS- Domain Name System
* E
  + Email - Electronic Mail
* H
  + HTTP (Hypertext Transfer Protocol) - It is a stateless communications protocol application layer used to access resources (hypertext/hypermedia) on the WWW, invented by Tim- Berners Lee.
* I
  + Internet – global information system that is logically linked together by globally unique address
  + IOT- Internet of things
* L
  + LAN - Local Area Network
* M
  + MIME- Multipurpose Internet Mail Extensions (text/html)
* P
  + Pocketswitch- circuit switch – calling somebody
  + Poll- frequent Requests
* T
  + TCP – Transmission Control Protocol
  + TLS- Transport Layer Security
  + Transceiver- has both transmitter and a receiver
* U
  + URL – Uniform Resource Locator, addresses a specific document
  + Usenet - Worldwide distributed discussion system available on computer

## What is HTML?

HTML is a markup language devised to allow website creation. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another.

### HyperText Markup Language

* HyperText is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.
* Markup is what HTML tags do to the text inside them. They mark it as a certain type of text (italicised text, for example).
* HTML is a Language, as it has code-words and syntax like any other language.
* The <!DOCTYPE html> declaration defines this document to be HTML5.
* The <html> element is the root element of an HTML page.
* The <head> element contains meta information about the document.
* The <title> element specifies a title for the document.li>
* The <body> element contains the visible page content.
* The <h1> element defines a large heading.
* The <p> element defines a paragraph.

### HTML Versions

* HTML 1.0 - 1991
* HTML 2.0 - November 24, 1995; published as RFC 1866.
* HTML 3.0 - HTML 3.0 builds upon HTML 2.0 and provides full backwards compatibility.
* HTML 3.2 - January 14, 1997; Published as a W3C Recommendation.
* HTML 4.0 - December 18, 1997; Published as a W3C Recommendation; supports more multimedia options, scripting languages, style sheets, better printing facilities, and documents that are more accessible to users with disabilities.
* HTML 4.01 - December 24, 1999; This document obsoletes previous versions of HTML 4.0. This enhances the functionality and interoperability of the Web.
  + Strict – deprecated
  + Transitional – still make use the deprecated
  + Frameset – using frames
* HTML 5 (working draft)- October 28, 2014; In this version, new features are introduced to help Web application authors, new elements are introduced based on research into prevailing authoring practices, and special attention has been given to defining clear conformance criteria for user agents in an effort to improve interoperability.
* HTML 5.1 - November 1, 2016; The W3C started working on the draft of HTML 5.1 with the goal of fixing some of the issues that were left open in HTML5.
* HTML 5.2 - February 28, 2017; A working draft; The W3C aims to produce an HTML 5.2 Recommendation in late 2017 that would obsolete HTML 5.1.

### Important Notes:

* World Wide Web Consortium (W3C) - An international community that oversees the development of HTML and open standards to ensure the long-term growth of the Web
* W3C Recommendation – documents specify the HTML standards.
* Tim Berners-Lee –invented HTML.

## Document Object Model (DOM)

It is a programming interface for HTML and XML documents. It provides a structured representation of the document and it defines a way that the structure can be accessed from programs so that they can change the document structure, style and content.

### Hierarchy

### Window object

Top of the hierarchy. It is the outmost element of the object hierarchy and represents an open window in a browser.

### Document Objects

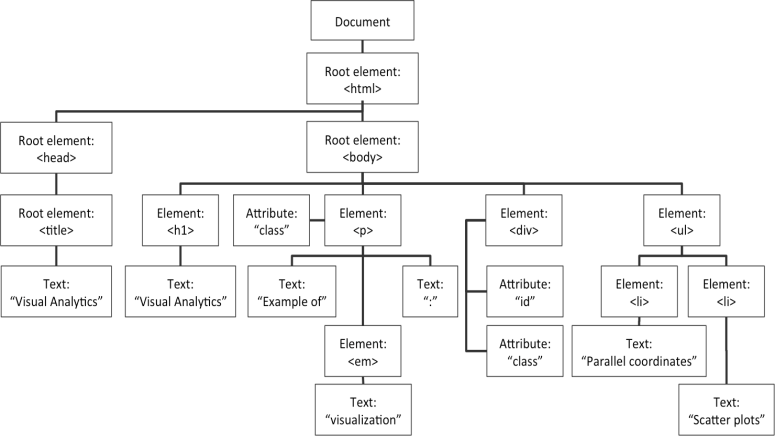
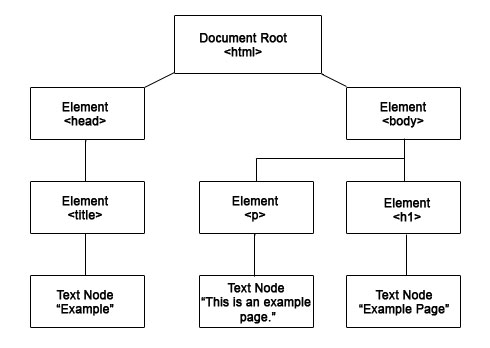
In the HTML DOM (Document Object Model), everything is a node. Each node has a type, corresponding to what kind of HTML element it represents. Each node has different properties and methods available, depending on its type. The base type of all dom nodes is Node, which contains methods for traversing the tree.

* Document - represents the current type
* document.getElementId
* document.getElementbyClassName
* document.getElementbyName
* document.getElementbyTagName
* document.querySelector
* document.querySelectorAll
* document.body.matches('body')
* nodeName - element tag itself
* p.nodeName   
  -- "p"
* p.finalChild   
  -- "hello"
* p.nodeValue   
  -- null
* nodeValue - elements dont usul=ally have a value
* p.parentNode --  <body> ... </body>
* document.body.childNodes   
  -- [text, p#p, text, comment, text]
* p.childNodes  
  -- [text]
* document.body.firstElementChild --  <p id = "p"> helloworld </p>

### Traversing the DOM object:

the top node is called the root (or root node).  
Every node has exactly one parent, except the root (which has no parent).  
A node can have any number of children.  
Siblings (brothers or sisters) are nodes with the same parent.

#### Example

<html>   
  <head>   
   <title>Example</title>   
  </head>   
  <body>   
   <h1>Example Page</h1>   
   <p>This is an example</p>   
  </body>   
</html>   
**DOM Representation**

CSS

Language to specify the presentation aspects of structured documents. Hakon Wium Lie (CHSS) and Bert Bos (SSP). They were working on their own versions but decided to work together.

* Versions
  + CSS 1 (1996)
  + CSS 2.1 (2011)
  + CSS 3
* CSS Preprocessors: (Extensions)
  + SASS, Less
* CSS Frameworks:
  + Bootstrap, Materialize
* Others:
  + 960 Grid System
  + Foundation

HTML/XHTML Stylesheets

1. Author Styles (original styles)
2. User Styles
   * Can be provided by the user, may be styled different due to a person's disability or preference.
   * CSS can let the user override the original style, however user styles are being faced out. Example: Chrome
3. User Agents
   * Browsers have a default user agent stye, and it differs from each browser.
   * If you want a style to be common across browsers, make use of "ResetCSS"

## Syntax (CSS statements)

### At-Rules

* @charset
* @import
* @media
* @font-face
* @keyframes
* @page
* @namespace
* @document (Firefox only supports this at-rule)

### CSS Rule Sets (or style rules)

#### Introduction

A rule or "rule set" is a statement that tells browsers how to render particular elements on an HTML page. A rule set consists of a selector followed by a declaration block.

* selector - The selector "selects" the elements on an HTML page that are affected by the rule set. The selector consists of everything up to (but not including) the first left curly bracket.
* declaration block - The declaration block is a container that consists of anything between (and including) the curly brackets. Whitespace inside a declaration block is ignored - so it can be used to lay out rules in any way you want.
* property declaration - The property is the aspect of that element that you are choosing to style.
* property name
* property value - The value is the exact style you wish to set for the property.

#### Simple Selectors

A simple selector is either a type selector or universal selector followed immediately by zero or more attribute selectors, ID selectors, Class selectors or pseudo-classes, in any order. The simple selector matches if all of its components match

1. Universal selector - The universal selector, written "\*", matches the name of any element type.

Example: \*{ }

1. Type selector- A type selector matches the name of a document language element type. A type selector matches every instance of the element type in the document tree.

Example: p{ }

1. Grouping selector- When several selectors share the same declarations, they may be grouped into a comma-separated list.

Example: h1, h2, h3{ }

1. ID selector- ID selectors match an element instance based on its identifier. A CSS ID selector contains a "#" immediately followed by the ID value, which must be an identifier.

Example: #nav{ }

1. Class selector- authors may use the period (.) notation as an alternative to the ~= notation when representing the class attribute.

Example: .hide{ }

1. Descendant selector- Descendant selectors express such a relationship in a pattern. A descendant selector is made up of two or more selectors separated by white space. A descendant selector of the form "A B" matches when an element B is an arbitrary descendant of some ancestor element A.

Example: div.content em{ }

1. Child selector- A child selector matches when an element is the child of some element. A child selector is made up of two or more selectors separated by ">".

Example: div.content > em{ }

1. Adjacent sibling selector- Adjacent sibling selectors have the following syntax: E1 + E2, where E2 is the subject of the selector. The selector matches if E1 and E2 share the same parent in the document tree and E1 immediately precedes E2, ignoring non-element nodes (such as text nodes and comments).

Example: h1 + p{ }

1. Attribute selector - allows authors to specify rules that match elements which have certain attributes defined in the source document.

Example: div[class=note] { }

1. Pseudo Class Selectors
   * Dynamic pseudo-classes
     + :hover
     + :focus
     + :active

Example: ul.menu > li{ } ul.menu > li:hover{ }

* + Structural pseudo classes
    - :root
    - :first-child
    - :last-child
    - :nth-child()
    - :only child()
    - :nth-of-type()
    - :first-of-type
    - :last-of-type
    - :nth-last-of-type()
    - :nth-last-child()
    - :only-of-type
    - :empty

Example: li:first-child{ }

* + Link pseudo classes
    - :link
    - :visited

Example: a:link, a:visited{ }

* + Language pseudo class
    - :lang

Example: :lang(fr) > q { quotes: '« ' ' »'; }

* + UI Element States pseudo classes
    - :enabled (CSS3)
    - :disabled (CSS3)
    - :checked (CSS3)
    - :indeterminate (CSS3)
  + Target pseudo class
    - :target

Example: :target { outline: solid red }

* + Negation pseudo class
    - :not()

Example: input:not([type="button"]) { border-color: blue; }

1. Pseudo Elements
   * ::first-letter
   * ::first-line
   * ::before
   * ::after

Example: p::before { content:"Paragraph" }

1. Combinators
   * descendant combinator
   * child combinator
   * ::before
   * sibling combinator

Example: p::before { content:"Paragraph" }

CSS Rule Precedence

1. BY Origin and Importance
   * - author/ user/ user agent important declaration.
   * - author/ user/ user agent normal declaration.
   * - When an important rule is used on a style declaration, this declaration overrides any other declarations.
2. Example:
3. table td {height: 50px !important;}   
   .myTable td {height: 50px !important;}   
   #myTable td {height: 50px !important;}
4. By Specifity
   * - greater inline style.
   * - greater number of id selector, attribute selector & pseudo-classes-greater number of id selector, attribute selector & pseudo-classes.
   * - greater number of type selectors & pseudo element.
5. Example:
   * html body div#pagewrap ul#summer-drinks li.favorite {
   * class= "bd"> color: red;
   * font-weight: bold;
   * }
6. By Order
7. Which one is latter specified.

CSS DECLARATION

1. Properties : shorthand properties and vendor - specific extensions (aka vendor prefixes)
2. Values
   * keywords
   * members (integers, etc.)
   * length, angles, duration, frequency
3. Length Units
   * font-relative: em, ex,ch, rem
   * font-relative: em, ex,ch, rem
   * viewport - percentage: vw, vh, vmin, vmax
   * absolute lengths: em, mm, q, in, pt, pc, px
   * angle units: deg, grad, rad, turn
   * duration units: (add notes)
   * frequency units: hg, khg
   * resolution units: dpi, dpcm, dppx

## What is JavaScript?

JavaScript is a cross-platform, object-oriented scripting language. It is a small and lightweight language. Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

### 3 ways to call javascript

1. External Linked Script   
   <head>   
     <script>link</script>   
   </head>
2. Embedded Script   
   <script>   
    document.getElementById("demo").innerHTML = "My First JavaScript";   
   </script>
3. Inline Script   
   <button onclick="console.log("document.getElementById("demo").innerHTML = "My First JavaScript";");>CLICK ME!

noscript

The <noscript> tag defines an alternate content for users that have disabled scripts in their browser or have a browser that doesn't support script.

The <noscript> element can be used in both <head> and <body>.

The HTML <script> element is used to embed or reference an executable script.

### Ways to load the Javascript files

1. defer - This Boolean attribute is set to indicate to a browser that the script is meant to be executed after the document has been parsed   
   <script defer="defer">
2. async - indicate that the browser should, if possible, execute the script asynchronously. It has no effect on inline scripts (i.e., scripts that don't have the src attribute).   
   <script src="demo\_async.js" async></script>

#### Example

<script>   
 console.log('embedded script ... ');  
 // global   
 function callname(){   
 console.log('you called me')   
<script>

Type system and Variable scoping rules

1. Dyanamic (aka loose or weak) typing
2. Global (aka top-level) or local scopes
3. Data Types
4. Primitive types
   * Numbers (decimal, hexadecimal notation)
   * Booleans (true, false)
   * Strings (Single or Double quote delimited)
   * Undefined and Null
5. Composite (object) types
   * Core JavaScript Objects
     + Objects, Number, Boolean, String, Date, Math, Global, RegExp, Error
     + Arrays (Array)
     + Functions (Function, Arguments)
6. DOM Objects
   * Anchor, Applet, Attr, Comment, DOMException, DOMImplementation, DocumentFragment, Element, Form, Image, Input, Layer, Link, Node, Option, Select, Style, Text, TextArea

## Javascript Functions

Window Object - represents an open window in a browser, This defines dynamic property of an object.

 Example:   
  let a = 10;

### JavaScript Data Types

JavaScript variables can hold many data types: numbers, strings, objects and more:

* String   
  A string(or a text string) is a series of characters like "John Doe". Strings are written with quotes. You can use single or double quotes.
* Numbers JavaScript has only one type of numbers.Numbers can be written with, or without decimals.
* Booleans   
  Booleans can only have two values: true or false.   
   Example   
     var x = true;   
    var y = false;
* Arrays JavaScript arrays are written with square brackets. Array items are separated by commas.  Example   
    var cars = ["1", "2", "3"];

### JavaScript Primitives

* A primitive value is a value that has no properties or methods.
* A primitive data type is data that has a primitive value.

#### JavaScript defines 5 types of primitive data types:

1. string
2. number
3. boolean
4. null
5. undefined

Function Declarations   
 Example:   
  function myFunction(a, b) {   
   return a \* b;   
  }

Function Expressions - A JavaScript function can also be defined using an expression.  
 Example:   
  var x = function (a, b) {return a \* b};

## Array

* a = []   
  -- []
* a = [1, 2, 3]   
  -- [1, 2, 3]
* a = new Array (1, 2, 3)   
  -- [1, 2, 3]
* a = new Array (10)   
  -- undefined x 10

### Functions

* Abstraction or blocks
* All functions return a value

#### Example

functions greater(v1, v2){   
if (v1, v2){   
 return v1;   
} else if (v2 > v1) {   
 return v2;   
}

/\* functions can be nested \*/

function outer(){   
  // some code here..   
}   
function inner(){   
  // inner function code here..   
 }   
  // more codes..   
}

JavaScript-is a full-fledged dynamic programming language that, when applied to an HTML document, can provide dynamic interactivity on websites. It was invented by Brendan Eich, co-founder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation.

DOM- Document Object Model- defines the logical structure of documents and the way a document is accessed and manipulated.

HTML-Hypertext Mark-up Language - is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user.

CSS - Cascading Stylesheet - is a style sheet language used for describing the presentation of a document written in a markup language.

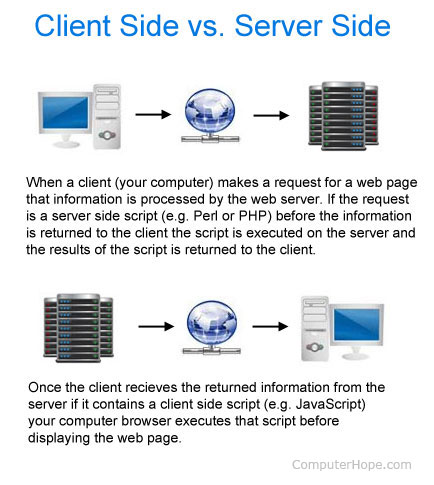
XHTML - Extensible HyperText Mark-up Language - is a markup language written in XML. A hybrid between HTML and XMLspecifically designed for Net device displays.

WHATWG - Web Hypertext Application Technology Working Group - is a community of computer professionals and general users with an interest in improving HTML led by Ian Hickson

ARIA- Accessible Rich Internet Applications - Accessible Rich Internet Applications (ARIA) defines ways to make Web content and Web applications (especially those developed with Ajax and JavaScript) more accessible to people with

ECMAScript - European Computer Manufacturer Association Script- ECMAScript is a standard script language, developed with the cooperation of Netscape and Microsoft and mainly derived from Netscape's JavaScript, the widely-used scripting language that is used in Web pages to affect how they look or behave for the user.

SSS or "Server Side Scripting" is a way of designing sites in a way where processes and users can be processed from the originating server. Server-side scripts provide an interface to the user and are used to limit access to proprietary data and help keep control of the script source code.



https://www.computerhope.com/jargon/s/server-side-scripting.htmBasics of SSS

Runs on a server, embedded in the site’s code

Designed to interact with back-end permanent storage, like databases, and process information from the server to access the database—like a direct line from user to database

Facilitates the transfer of data from server to browser, bringing pages to life in the browser, e.g., processing and then delivering a field that a user requests or submits in a form

Runs on-call. When a webpage is “called up,” or when parts of pages are “posted back” to the server with AJAX, server-side scripts process and return data

Powers functions in dynamic web applications, such as user validation, saving and retrieving data, and navigating between other pages

Plays a big role in how a database is built from the ground up and managed afterwards—an example of how roles often overlap in all aspects of development

Build application programming interfaces (APIs), which control what data and software a site shares with other apps

Mister Website's Backend Process

The backend consists of three main parts:

Server

Database

Backend Web Application

The **server** is a powerful computer that can run backend software, database houses, and the middleware. Imagine if you are updating your facebook profile. Server-side scripts will help you get the updated information and process it on the server using the **application** which then leads to updating the **database.**

What is PHP?

PHP or in recursive acronym *Hypertext Preprocessor* is an open source scripting language that is for web development and can be embedded in HTML.

Sample Code

<!DOCTYPE HTML>   
 <html>  
<head>  
  <title>PHP</title>   
   </head>  
  <body>  
  < ?phpecho'PHP SCRIPT';?>   
  </body>  
</html>

History

PHP was created by Rasmus Lerdorf in the year 1994 and in the present time PHP is now produced by The PHP Development Team. Before, PHP stands for **Personal Home Page** but in present time it is already known for its recursive acronym Hypertext Preprocessor.

PHP Are used in:

Server-side Scripting: Server-side scripting is the target field for PHP. PHP parser, web server, and a web browser are the three things to make it work. The users need to run the web server as long as it has a PHP that is already installed.

Command line Scripting: Running php does not require to use any server or browser. The only thing to use PHP without a browser and a server is the PHP Parser. It is a type of usage that is ideal for scripts.

Node.js is a Javascript free and open source cross-platform for server-side programming that allows users to build network applications quickly. Node.JS is a system that uses event-driven programming to build scalable applications and network programs. Node.js basically allows the developer to run JavaScript in the back end, away from the browser.

## Brief History of Node.JS

Node.js was originally written in 2009 by Ryan Dahl by making use of Google's V8 VM, the same runtime environment for JavaScript that Google Chrome uses. The initial release supported only Linux and Mac OSX. Its development and maintenance was led by Dahl and later sponsored by Joyent

Node.js has become wildly popular, with coders everywhere using it to create APIs and build a new matrix of interoperability across the Internet. It is a runtime system for creating (mostly) server-side applications. It's best known as a popular means for JavaScript coders to build real-time Web APIs.

## Node.JS's Backend Process

Node.js allows you to run JavaScript code in the backend, outside a browser. In order to execute the JavaScript you intend to run in the backend, it needs to be interpreted and, well, executed. This is what Node.js does, by making use of Google's V8 VM, the same runtime environment for JavaScript that Google Chrome uses.

Node.js ships with a lot of useful modules, so you don't have to write everything from scratch, like for example something that outputs a string on the console. Node.js operates on a single-thread, using non-blocking I/O calls, allowing it to support tens of thousands of concurrent connections

In the database side, data gets queued through some kind of cache or message queuing infrastructure and digested by a separate database batch-write process, or computation intensive processing backend services, written in a better performing platform for such tasks. Similar behavior can be implemented with other languages/frameworks, but not on the same hardware, with the same high, maintained throughput.

## Node.JS Example

### Once you install Node.JS into the computer, it can automatically be run. Here is an example of a simple NodeJS code:

// Call the console.log function.  
console.log("Webtek is Awesome!");

Save the file, and execute it through Node.js:  
node webtek.js

### Here is an example of a more complicated Node.JS code:

// Load the http module to create an http server.  
var http = require('http');  
// Configure our HTTP server to respond to all requests.  
var server = http.createServer(function (request, response) {  
response.writeHead(200, {"Content-Type": "text/plain"});  
response.end("Web Technology is Awesome!\n");  
});  
// Listen on port 8000, IP defaults to 127.0.0.1  
server.listen(8000);  
// Put a friendly message on the terminal  
console.log("Server running at http://127.0.0.1:8000/");

You can run this by saving it in a file called "example-console.js" and running it with node example-console.js

## Express.JS and Koa.JS framework

Node.js builds server apps with a lightweight, efficient JavaScript framework called Express.js. There are other frameworks it’s compatible with, but Express.js is best known as another quarter of the MEAN software stack. Newer to the scene and following in Express’ footsteps is the Koa.js framework, a callback-less lightweight framework written by the same author as Express. Get a rundown of more Node frameworks in this 15 Frameworks to Know for Next-Level Node Development article.