**WWW - World Wide Web**

**1.1 World Wide web Introduction**

**Web browser**

**-** Visualize pages on web document.

- Contains link, button, video, image

**internet :**

- hardware - global network of machines , and devices that facilitate interaction

**nodes -** computes

**edges -** connection

**www :**

- software - application on top of internet - using hardware infra structure

- also protocol for communication

- composed of document that are linked to each other through web-browser

**nodes** - documents/content/resources

**edges**- links between documents

**Addresses**

URI - uniform Resource Identifier - unique

URL - uniform resource Locator - type of URI. location on www and protocol

*http://www.example.com/home/inde.html?a=12&b=34*

*http :* protocol

*www.example.com :* host name

*example.com :* domain name

*home/inde.html :* file path

*a=12&b=34 :* query parameter

**Protocol:**

Structure of content of communication between devices

**Contents on www**

**static:** Contents does not change, -who, when-html,css

**dynamic:** content changes - user, event -js

request - receive / send - how display - how generate dynamic content

**1.2 HTTP Client Request, Server Response**

**View content on WWW:**

1980's - text web browsers

1095s - mosaic web browser - images and links between pages

**web browser :**

software displays content and let you navigate

component:

Rendering engine -

decides what to display and how to display on web browser.

presentation, ordering, content

html, css

Java script Engine-

dynamic part - create modify

run code

Web browser works -

http protocol -

web browser communicates - request, transfer documents

client -(laptop, mobile, fridge)

server - server /big computer

connected via connections , devices

client initiates - http request

server - listens - decides response and content and sends http response

client again - decides what to do with the content from server response

http:

plain text, human readable protocol on the web

client server model

request + client info

response + server info



**http request**

verb - Action client hoping to perform

get - requesting resource

head - just header info, not any resource

post -create new content on server

argument - name of the resource requested

protocol - http/1.1

client information + other information

Ex:

Request Verb URI HTTP Version

----- --------------------------- -----------

GET /example/index.html HTTP/1.1 ------------> Request Line

Host: www.edx.org |

User-Agent: Mozilla/4.0 |-----------> Request header

Accept-Language: en-us |

Content-Length: 9 |

-------------> Blank Line Separator

a= 12 & b=34 -------------> Request Message Body

**HTTP Response -**

1st line

Protocol and status code

1XX - Information only

2XX - Success

3XX - Client redirect

4XX - client error

5XX - server error

Most common status codes

200 OK - request succeeded , returning contents requested

404 Not Found - requested resource does not exist

500 Server Error - Error on server side while processing request

2nd line

header information - info about server / info of content

3rd line

blank separator

4th line

response body - content of page requested

Ex:

HTTP Version Status Code

---------- --------

HTTP/1.1 200 OK -----------------------------------> Response Line

Date: Fri, 06 Apr XXXX 09:30:00 GMT |

Server : Apache/1.4 |

Last-Modified: Wed, 04 Apr XXXX |

Connection: close |------------> Response Headers

Content-Type: text/html |

Content-Length: 228 |

--------------> Blank Line Separator

<!DOCTYPE html> <html><Head> --------------> Response Body

**HTML**

**1.3 HTML Basic**

**HTML -**

Structure content / data on WWW

html document - elements -tags - can be nested

plain text - human readable

structure

doesn't decide how to display - web browser decides

**Tags**

<!DOCTYPE html> - version of html

<html> - all contents within

<head>- info of document

<title> title bar

<link> link to other document ex: css -styling

<meta> provide additional information Ex: keyword used by search engine

<script> link to dynamic content Ex: Javascript

<body> - content to be displayed Ex: image, text

**HTTP and HTML -**

browser request for content

client - http request

server - response - content - plain text -html

browser - how to display

Browser - View source Code

html page - right click - view source -html source

**1.4 Important HTML Tags**

<p> : separate paragraph

<h1><h6> : headings

<b> : bold <strong>

<i> : italic <em>

<hr> : horizontal line

<br> : break line

<div> : division. content goes together. useful in styling

<span> : text part. smaller content than div. styling

<!-- comments -->

&nsbp; : single white space

&lt; &gt; : less than / greater than

&amp; : & symbol

&copy; &reg; : copy and registration symbol

**1.5 HTML Attributes**

HTML Attributes -

additional information to tag

name value pair

Ex: <p name="value">

title : title of element. see title when hover over element

<h1 title= "heading">

style: how the content will appear. multiple sub key value pair

<h1 style =" color:red; text-transform:capitalize">

background-color: red, #012345

font-family: verdana, courier

font-size: 12px, 4in, 200%

text-align : center, left, right

id : (#) unique identifier to html element within html tag within html page can be used to formatting

class : (.) subgroup of elements

html - style tag

**1.6 CSS - Style - CSS**

- http response : html

- html looks for css

- inline / same html file / separate file

- separate file - client makes another request for css -http response

- html apply css

inline css:

<h1 style="color:red:>heading1</h1>

advantage : easy to use when want something just once

disadvantage: mixing content html and css

same html :

<style>

h1 { -----> CSS selector

color: red; -----> property: value pair

}

</style>

Advantage: separate content in head, easy to use

Disadvantage : head might get big, html specific

External CSS:

separate file - link within the head

.html

<head>

<link rel="stylesheet", type="text/css", href = "movie-styles.css" />

</head>

.css

h1 {

color:red

}

Advantage : separate content and html , can use css file for multiple pages

disadvantage : need to manage files

css selectors:

external selector :

h1 { color:red; }

all <h1>

class selector :

.address { ... }

<div class="address"> ... </div>

id selector:

#section1 { ... }

<p id="section1"> ... </p>