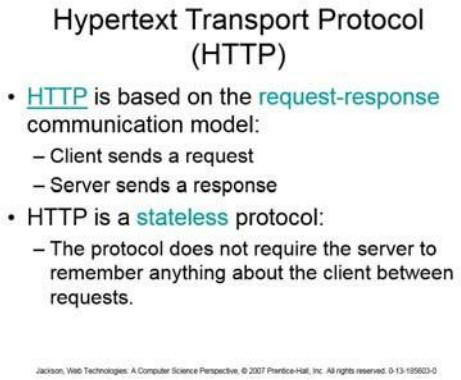
**Mid Term - We will cover these chapters**

1- Chapter 1: HTTP
2- Chapter 2: HTML
3- Chapter 3: CSS
4 Chapter 4: JavaScript
5- Chapter 5: DOM

**Questions Type:**
**These are just examples of the major questions types:**
**MCQs, True/False, Short Answer, What is the output of the program or the function, identify the error in the code, write a code.**

- Chapter 1:
  - HTTP request / response (URI, MIME)
  - Caching
  - Definition of Web Server


HTTP request: ← !important

- Start line
  - Example: GET / HTTP/1.1
  - Three space-separated parts:
    - HTTP request method
    - Request-URI
    - HTTP version
  - Uniform Resource Identifier (URI)
    - Syntax: *scheme* : *scheme-depend-part*
      - Ex: In http://www.example.com/
        the scheme is http
    - Request-URI is the portion of the requested URI that follows the host name (which is supplied by the required Host header field)
      - Ex: / is Request-URI portion of http://www.example.com/

- URI's are of two types:
  - Uniform Resource Name (URN)
    - Can be used to identify resources with unique names, such as books (which have unique ISBN's)
      - Scheme is urn
  - Uniform Resource Locator (URL)
    - Specifies location at which a resource can be found
    - In addition to http, some other URL schemes are https, ftp, mailto, and file.
- Common request methods:
  - GET
    - Used if link is clicked or address typed in browser
    - No body in request with GET method
  - POST
    - Used when submit button is clicked on a form
    - Form information contained in body of request
  - HEAD
    - Requests that only header fields (no body) be returned in the response

- Header field structure:
  - *field name* : *field value*
  - Syntax
    - Field name is not case sensitive
    - Field value may continue on multiple lines by starting continuation lines with white space
    - Field values may contain MIME types, quality values, and wildcard characters (*'s)
  - MIME type← !important
    - Convention for specifying content type of a message
    - In HTTP, typically used to specify content type of the body of the response
    - MIME content type syntax:
    - *top-level type* / *subtype*
    - Examples: text/html, image/jpeg.
    - Example header field with quality values:
      accept: text/xml,text/html;q=0.9,
      text/plain;q=0.8, image/jpeg,
      image/gif;q=0.2,*/*;q=0.1
    - *Quality value applies to all preceding items*
    - Higher the value, higher the preference
    - Note use of wildcards to specify quality 0.1 for any MIME type not specified earlier

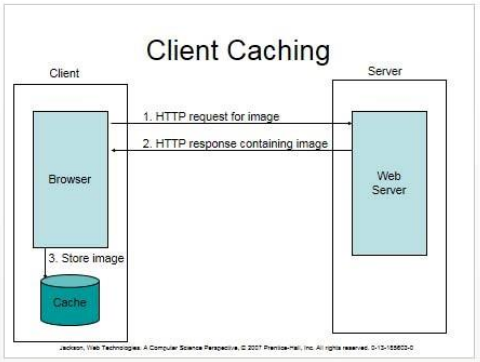
Response: ← !important

- Status line
  - Example: HTTP/1.1 200 OK

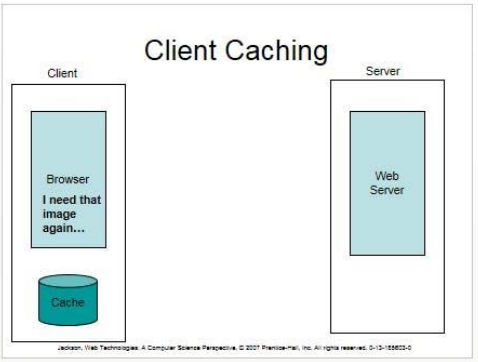
- Three space-separated parts:
    - HTTP version
    - status code
    - reason phrase (intended for human use)
- Status code ← !important
    - Three-digit number
    - First digit is class of the status code:
        - 1=InformaΘonal
        - 2=Success
        - 3=RedirecΘon (alternate URL is supplied)
        - 4=Client Error
        - 5=Server Error
    - Other two digits provide additional information

- Header fields
    - Common header fields:
        - Connection, Content-Type, Content-Length
        - Date: date and time at which response was generated (required)
        - Location: alternate URI if status is redirection
        - Last-Modified: date and time the requested resource was last modified on the server
        - Expires: date and time after which the client's copy of the resource will be out-of-date
        - ETag: a unique identifier for this version of the requested resource (changes if resource changes)

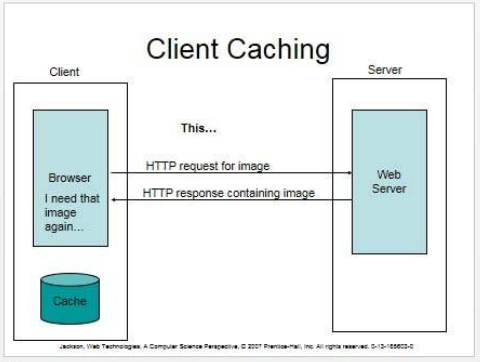
- Caching : A cache is a local copy of information obtained from some other source


Client Caching
Client
Server
1. HTTP request for image
2. HTTP response containing image
Browser
Web Server
3. Store image
Cache

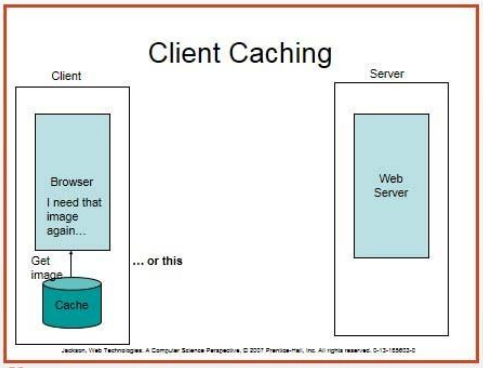
49


Client Caching
Client
Server
Browser
I need that image again…
Web Server
Cache

50


Client Caching
Client
Server
This…
Browser
I need that image again…
HTTP request for image
HTTP response containing image
Web Server
Cache

51


Client Caching
Client
Server
Browser
I need that image again…
Web Server
Get image
… or this
Cache

52

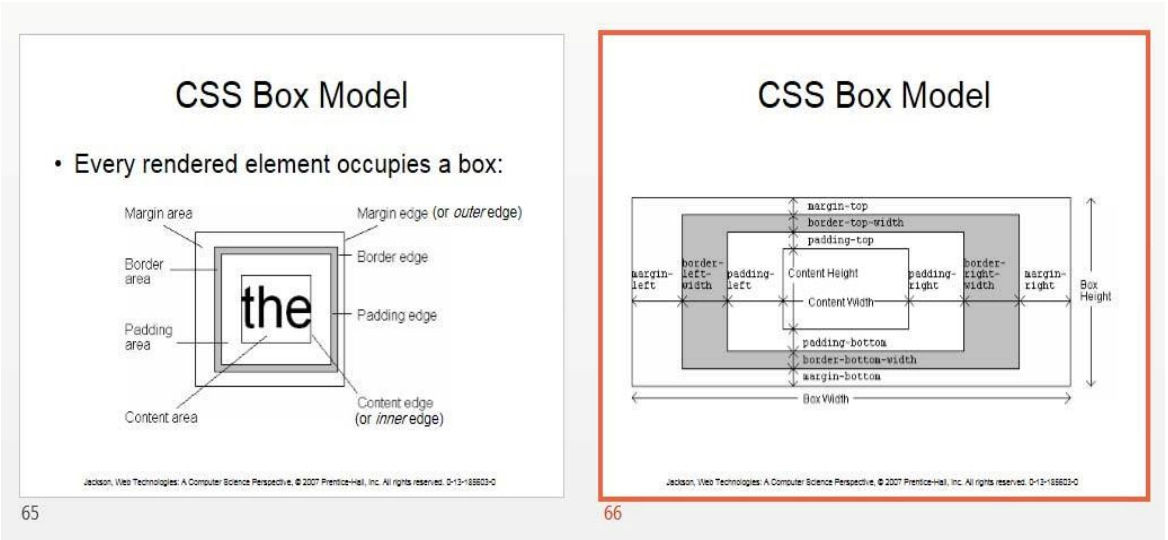
Solve caching problem ?
Slide 47 to 57.

- Chapter 2:
  - HTML vs. XHTML← !important
    - HTML allows some tag omissions (e.g., end tags)
    - XHTML element and attribute names are lower case (HTML names are case-insensitive)
    - XHTML requires that attribute values be quoted
  - HTML tags: div, p, span, table, ol, ul, img, a, form, input,

See the text file in mid folder. ← !important

- Chapter 3:
  - Different methods of embedding CSS
    - Inline, in the style attribute. Ex.  Red color .
    - Embed in the style attribute:
      <style>
      P { color: red}
      </style>
    - Linked or externalin another file: Ex.
      <head>
      <link rel="stylesheet" type="text/css" href="theme.css">
      </head>
  - Selector Strings (id, class, …). ← !important
    - To represent a class selector string we can use a dot in the beginning of the class name. Ex .aParagraph
    - To represent an ID selector string we can use a hash (#) in the beginning of the class name. Ex #aParagraph
    - We can apply multiple classes in a class attribute but we cant do the same for ID's
  - Color, font-style, font-size.
    - Style attriputes to set color
      - Font color = color:red or color:#00ff00.
      - Background color = background-color: red.
      - Font-style : underline , oblique, italic, initial , inherit , hΣp://www.w3schools.com/cssref/pr_font_font-style.asp
      - Font size: read about em in the slides – very important! In short it is a percentage of parent size, if parent is 1 cm size then 2 em mean child is 2 cm.
        hΣp://www.w3schools.com/cssref/pr_font_font-size.asp

- CSS Box Model (with all properties in it).




Also slides from 67 – 74

- Flow Layout (Normal, Relative, Absolute, Float).
  - In normal flow processing, each displayed element has **a corresponding box**
  - html element box is called initial containing block and corresponds to entire document
  - Boxes of child elements are contained in boxes of parent
  - Sibling block elements are laid out one on top of the other
  - Sibling inline elements are one after the other
  - **What is a "block element"?**
    - Element with value block specified for its display property
    - User agent style sheet (not CSS) specifies **default values;** typical block elements include html, body, p, pre, div, form, ol, ul, dl, hr, h1 through h6
    - Most other elements **except li and table-related** have inline specified for display
  - position: static (initial value), relative, or absolute
    - Element is positioned if this property not static
    - Properties **left, right, top, bottom** apply only to positioned elements
    - Primary values are auto (initial value) or CSS length
  - float: none, left, or right
    - Applies to elements with static and relative positioning only

Slide 14 to 25

Who to write a class or id?

What is the difference between class and id?

Slide 65 & 66

Flow layout ( normal ) very important.

No code for flow layout, just the Differences and definitions

- Chapter 4:
  - Window.alert, .prompt
    - 
  - Functions, var, if, loop, operators.
  - Regular Expression.
  - Object methods, constructors not important

how to use 1.alert 2. Prompt.
How to make regular and expression?
/^ format $/

- Chapter 5:

- Events: onclick, onmouseover, onmouseout, onfoucs, onblur
- Document tree: switching example, collapse/expand
- Event PropagaΘon: event listener, bubble (slide 96 example)
- Settimeout, setinterval

write the code?

Explain what the code do?

*document = no write.

Finlay

*Assignments 1 & 2 are important.

*What is the output?

What are the functions?

Write all errors.