HTML 5

N Satyanarayana
CDAC

HTTP vs HTML

- HTML: hypertext markup language
  - Definitions of tags that are added to Web documents to control their appearance
- HTTP: hypertext transfer protocol
  - The rules governing the conversation between a Web client and a Web server

Both were invented at the same time by the same person

Many application layer protocols are used on the Internet, HTTP is only one

Protocol	**Application**
HTTP: Hypertext Transfer	Retrieve and view Web pages
FTP: File Transfer	Copy files from client to server or from server to client
SMTP: Simple Mail Transport	Send email
POP: Post Office	Read email

The TCP/IP protocol layers

The application program is king – it gets work done using the lower level layers for communication between the client and server.

Application	Get useful work done – retrieve Web pages, copy files, send and receive email, etc.
Transport	Make client-server connections and optionally control transmission speed, check for errors, etc.
Internet	Route packets between networks
Data link	Route data packets within the local area network
Physical	Specify what medium connects two nodes, how binary ones and zeros are differentiated, etc,

URL

- *<scheme> : //<host> :<port> /<path> ;<parameters> ?<query> #<fragment>*
  - scheme
    - The protocol you are using
  - host
    - Host name or ip number
  - port
    - TCP port number that protocol server is using
  - path
    - Path and filename reference of object on server

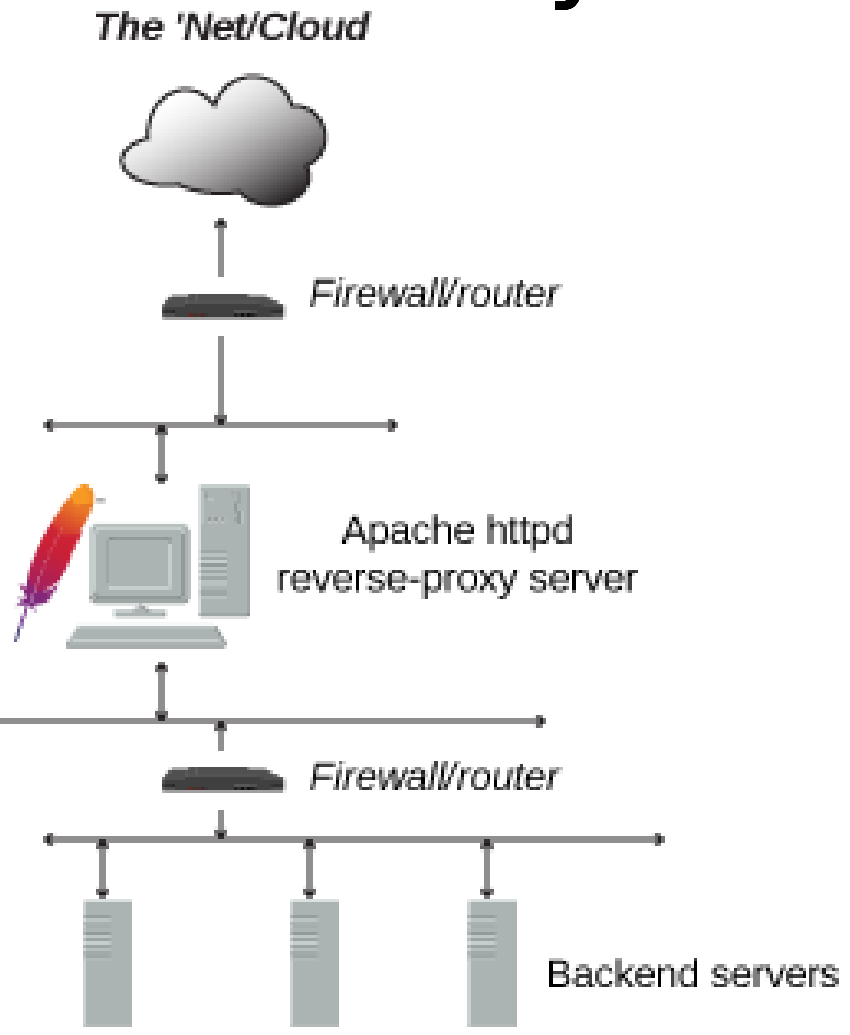
Web Documents

- html
- ASCII text
- Preformatted
  - postscript
- Images
  - GIF
  - JPEG
- Video
  - MPEG
- jsp, asp, php etc

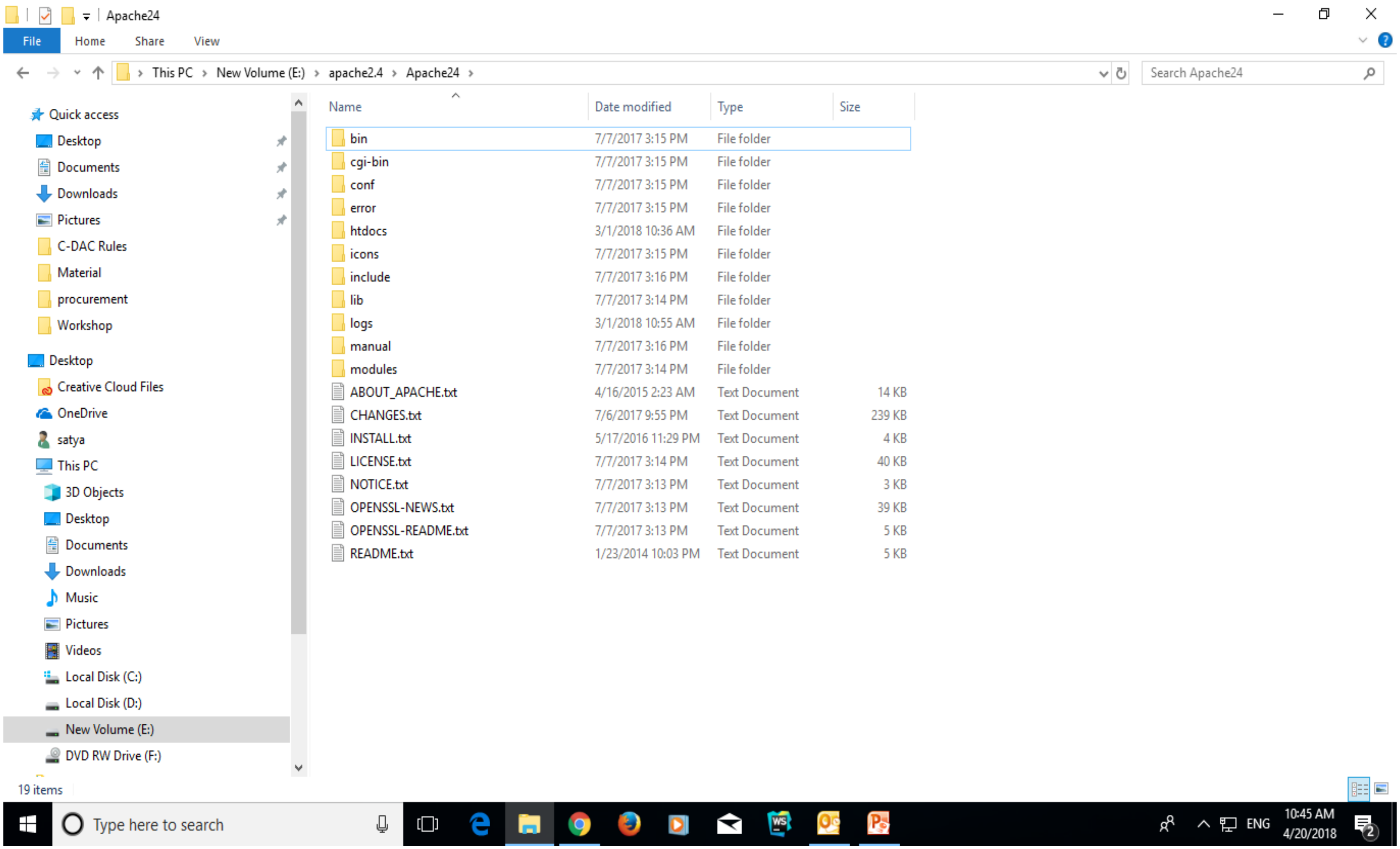
Gateways

- Translates from one protocol or service to another
  - HTTP / database query
  - Database query results / HTTP

Reverse Proxy Server / Gateway



Apache Web Server Folder Structure



Apache24

File | Home | Share | View

This PC > New Volume (E:) > apache2.4 > Apache24 >

Search Apache24

Quick access
- Desktop
- Documents
- Downloads
- Pictures
- C-DAC Rules
- Material
- procurement
- Workshop

Desktop
- Creative Cloud Files
- OneDrive
- satya
- This PC
  - 3D Objects
  - Desktop
  - Documents
  - Downloads
  - Music
  - Pictures
  - Videos
  - Local Disk (C:)
  - Local Disk (D:)
  - New Volume (E:)
  - DVD RW Drive (F:)

Name	Date modified	Type	Size
bin	7/7/2017 3:15 PM	File folder	
cgi-bin	7/7/2017 3:15 PM	File folder	
conf	7/7/2017 3:15 PM	File folder	
error	7/7/2017 3:15 PM	File folder	
htdocs	3/1/2018 10:36 AM	File folder	
icons	7/7/2017 3:15 PM	File folder	
include	7/7/2017 3:16 PM	File folder	
lib	7/7/2017 3:14 PM	File folder	
logs	3/1/2018 10:55 AM	File folder	
manual	7/7/2017 3:16 PM	File folder	
modules	7/7/2017 3:14 PM	File folder	
ABOUT_APACHE.txt	4/16/2015 2:23 AM	Text Document	14 KB
CHANGES.txt	7/6/2017 9:55 PM	Text Document	239 KB
INSTALL.txt	5/17/2016 11:29 PM	Text Document	4 KB
LICENSE.txt	7/7/2017 3:14 PM	Text Document	40 KB
NOTICE.txt	7/7/2017 3:13 PM	Text Document	3 KB
OPENSSL-NEWS.txt	7/7/2017 3:13 PM	Text Document	39 KB
OPENSSL-README.txt	7/7/2017 3:13 PM	Text Document	5 KB
README.txt	1/23/2014 10:03 PM	Text Document	5 KB

19 items

Type here to search

ENG 10:45 AM 4/20/2018

Apache Server Directives

- Reverse Proxying (For single back end server)
  - ProxyPass “/img” “http://www.example.com”
  - ProxyPassReverse “/img” “http://www.example.com”

- Clusters (For load balancing)
  - <**Proxy** balancer://myset>

```
    BalancerMember http://www2.example.com:8080
    BalancerMember http://www3.example.com:8080
    ProxySet lbmethod=bytraffic
</Proxy>
ProxyPass "/images/" "balancer://myset/"
ProxyPassReverse "/images/" "balancer://myset/"
```

SSL Configuration

- Procedure for generating self-signed certificate
  - Generate:
    - openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem
  - Review:
    - openssl x509 -text -noout -in certificate.pem
  - Convert .pem to .crt file
    - openssl x509 -outform der -in {your-cert}.pem -out {your-cert}.crt
    - Replace {your-cert} with the name of the file

HTTP 1.0

- Method
  - Get
    - Returns object
  - Head
    - Returns information about object
  - Post
    - Sends information to be stored on server or as input to script

HTTP 1.0

- Method
  - Put
    - Sends new copy of existing object to server
    - Usually not allowed
  - Delete
    - Deletes object
    - Usually not allowed

HTTP 1.0

- Other information
  - User Agent
    - Kind of browser
  - If-Modified-Since
    - Returns object only if more recent than given date
    - Otherwise returns status code 304

HTTP 1.0

- Other information
  - Accept
    - Mime types which browser can accept
      - Multipurpose Internet Mail Extension
        - » text/plain
        - » text/html
        - » application/postscript
        - » image/gif
        - » image/jpeg
        - » audio/basic
        - » video/mpeg
        - » x-world/x-vrml

HTTP 1.0

- Other information
  - Authorization
    - User password

```
GET /X/Y/Z.HTML      HTTP 1.0
User Agent: Prodigy-WB/1.3e
Accept: text/plain
Accept: text/html
Accept: application/postscript
Accept: image/gif
```

  - Accept: */*

HTTP 1.0

- HTTP response
  - Status line
    - HTTP-version   Status-code   Reason
    - Status-codes 1xx - Informational
      - Reserved for future use

HTTP 1.0

- HTTP response
  - Status line
    - Status-codes 2xx - Success
      - The action was successfully received, understood, and accepted

» 200	OK
» 201	POST command successful
» 202	Request accepted
» 203	GET or HEAD request fulfilled
» 204	No content

HTTP 1.0

- HTTP response
  - Status line
    - Status-codes 3xx - Redirection
      - Further action must be taken in order to complete request
        - » 300 Resource found at multiple locations
        - » 301 Resource moved permanently
        - » 302 Resource moved temporarily
        - » 304 Resource has not modified (since date)

HTTP 1.0

- HTTP response
  - Status line
    - Status-codes 4xx - Client error
      - The request contains bad syntax or cannot be fulfilled

» 400	Bad request from client
» 401	Unauthorized request
» 402	Payment required for request
» 403	Resource access forbidden
» 404	Resource not found
» 405	Method not allowed for resource
» 406	Resource type not acceptable

HTTP 1.0

- HTTP response
  - Status line
    - Status-codes 5xx - Server error
      - The server failed to fulfill an apparently valid request

» 500	Internal server error
» 501	Method not implemented
» 502	Bad gateway or server overload
» 503	Service unavailable / gateway timeout
» 504	Secondary gateway / server timeout

HTTP 1.0

- HTTP response
  - Description of information

• Server	Type of server
• Date	Date and time
• Content-Length	Number of bytes
• Content-Type	Mime type
• Content-Language	English, for example
• Content-Encoding	Data compression
• Last-Modified	Date when last modified
• Expires	Date when file becomes invalid

HTTP 1.0

- Problems
  - HTTP is stateless
    - Each request requires separate TCP connection
    - Server doesn’t remember previous requests

Http 1.0 vs 1.1

- OPTIONS method
  - A way for a client to learn about the capabilities of a server without actually requesting a resource
- Upgrade
  - Switch protocols

How it works?

- Code that resides on server machine but gets downloaded and executed on client browser.
  - Programming Language: HTML
- Code that resides on server machine but gets executed on server machine.
  - Programming Language: JSP, ASP, PHP, JS
- Both the codes reside on server machine in appropriate folders corresponding to a web application server.
  - Web Application Server: Apache, Tomcat, Jboss, IIS

HTML5: What is it?

HTML5 is the new standard for HTML, XHTML, and the HTML DOM (document object model).

The previous version of HTML came in 1999. The web has changed a lot since then.

Most modern browsers have HTML5 support.

History of HTML

1991	HTML first published
1995	HTML 2.0
1997	HTML 3.2
1999	HTML 4.01
2000	XHTML 1.0
2002 -200 9	XHTML 2.0
201 2	HTML5

After HTML 4.01 was released, focus shifted to XHTML and its stricter standards.

XHTML 2.0 had even stricter standards than 1.0, rejecting web pages that did not comply. It fell out of favor gradually and was abandoned completely in 2009.

HTML5 is much more tolerant and can handle markup from all the prior versions.

Though HTML5 was published officially in 2012, it has been in development since 2004.

HTML5: Origins

HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

WHATWG was working with web forms and applications, and W3C was working with XHTML 2.0. In 2006, they decided to cooperate and create a new version of HTML.

HTML5: Ground Rules

Some rules for HTML5 were established:

> New features should be based on HTML, CSS, DOM, and JavaScript
> Reduce the need for external plugins
> Better error handling
> More markup to replace scripting
> HTML5 should be device independent
> Dev process should be visible to the public

HTML5: New Features

> Canvas element for drawing
> Video/audio elements for media playback
> Better support for local offline storage
> New content specific elements, like article,
   footer, header, nav, section
> New form controls, like calendar, date, time, email, url, search

It is all about.....

- Providing powerful capabilities for Web-based applications with more powerful interaction, video support, graphics, more styling effects, and a full set of APIs.

- Evolve as a future Open Web Platform

HTML5: Support

You may well ask: “How can I start using HTML5 if older browsers don’t support it?” But the question itself is misleading.

HTML5 is not one big thing; it is a collection of individual features. You can only detect support for individual features, like canvas, video, or geolocation.

HTML5: Example

HTML5 supports all the form controls from HTML 4, but it also includes new input controls. Some of these are long-overdue additions like sliders and date pickers; others are more subtle...

HTML5: Example

For example, the email input type looks just like a text box, but mobile browsers will customize their onscreen keyboard to make it easier to type email addresses. Older browsers that don’t support the email input type will treat it as a regular text field, and the form still works with no markup changes or scripting hacks.

HTML5: DOCTYPE

The DOCTYPE which comes before the beginning <html> tag is much simpler in HTML 5. Here are some examples of what it looks like now...

```
<!DOCTYPE HTML>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Basic HTML Page Structure

```
<!DOCTYPE HTML>
<html>
<head>
    ....
</head>
<body>
       ....
       <p id=“para” onclick=“changecontent()”>Not Ok</p>
       ....
<script>
      function changecontent() {
            document.getElementById(“para”).innerHTML = “Ok”;
      }
</script>
</body>
</html>
```

Elements of head tag

- title - <title> Yahoo! </title>
- style - <style media=“all” type=“text/css”>
  h1 {color: #000000;}
  </style>
- base - <base href=“www.cdac.in” target=“_blank”>
- link - <link rel=“stylesheet” type=“text/css” href=”demo.css”>
- meta - <meta name="viewport" content="width=device-width, initial-scale=1.0">
- script - <script src=“Demo.js” type=“text/javascript”></script>
- noscript

HTML5

- More focused on semantics of the document structure so that it is more accessible
- Encourages separation of presentation stuff (color, style etc.)
- Lessens developer efforts on HTML coding (e.g., input validation stuff especially)

Background

- HTML was primarily designed as a language for semantically describing scientific documents, although its general design and adaptations over the years have enabled it to be used to describe a number of other types of documents.
- The main area that has not been adequately addressed by HTML is a vague subject referred to as Web Applications.

Source: W3C Technical Specification about HTML5

Contd…

- HTML5 is targeted specifically at applications that would be expected to be used by users on an occasional basis, or regularly but from disparate locations, with low CPU requirements.

- Examples of such applications include
  - online purchasing systems,
  - searching systems,
  - games (especially multiplayer online games),
  - public telephone books or address books,
  - communications software (e-mail clients, instant messaging clients, discussion software),
  - document editing software, etc.

Semantics

- A semantic element describes the meaning of the element
  - Example: ~~<div> and <span>~~
  - Example: <form>, <table>, <img>

- These definitions allow HTML processors, such as Web browsers or search engines, to present and use documents and applications in a wide variety of contexts that the author might not have considered.

Few Use cases

- HTML conveys meaning rather than presentation
  - Same page may be viewed on different devices
  - Quick navigation to parts of the document (speech browsers)
  - Improves indexing of pages by search engines

Authors must not use elements, attributes, or attribute values for purposes other than their appropriate intended semantic purpose, as doing so prevents software from correctly processing the page.

A simple example

(Usage of disk space) ? Progress : meter

- Why?
  - The progress element represents the completion progress of a task.
  - The meter element represents a scalar measurement within a known range, or a fractional value

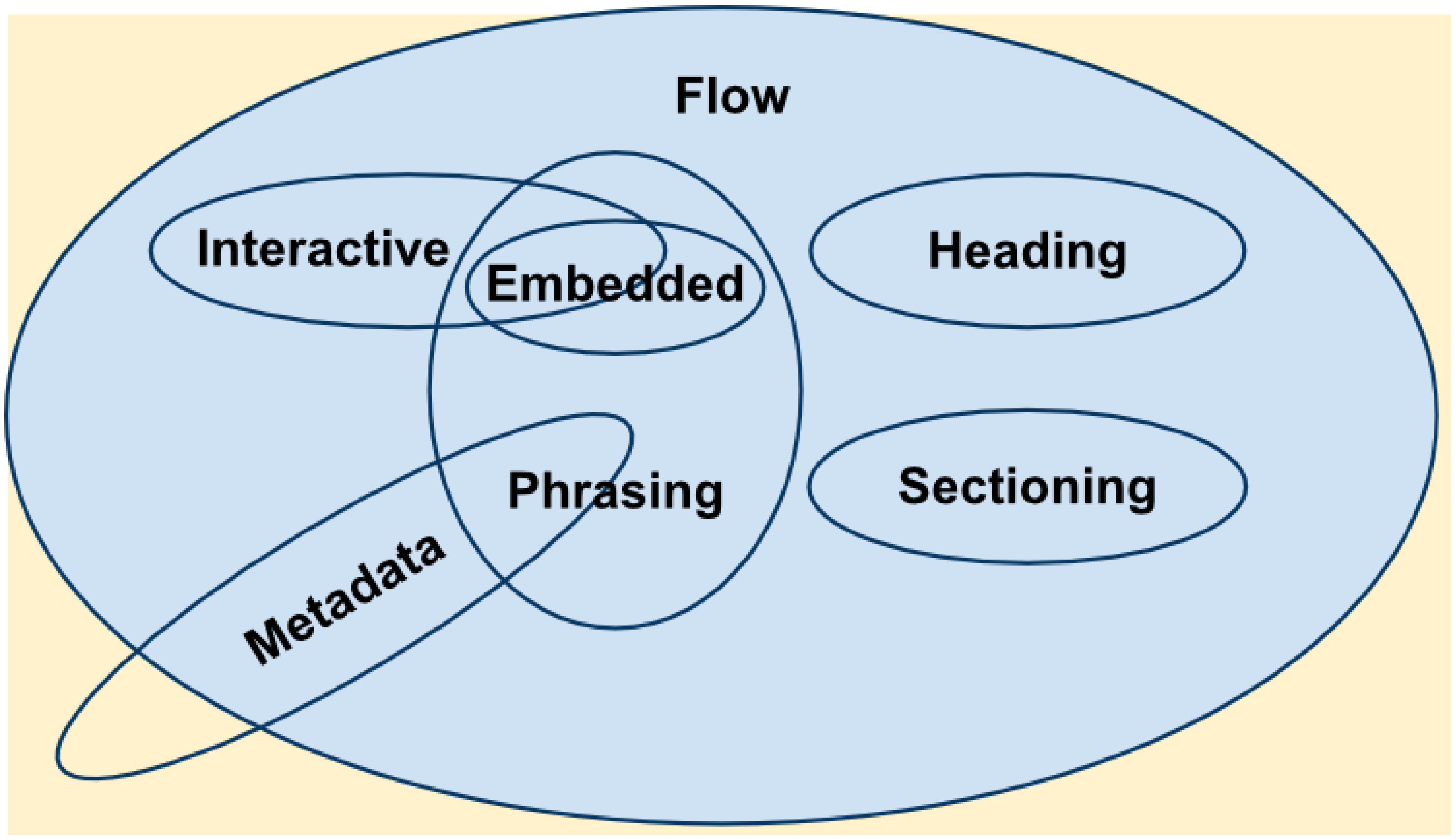
HTML Content

- Each HTML element must abide by rules defining what kind of content it can have.
- These rules are grouped into content models common to several elements.
- HTML conformant document is the one which follows these rules.

Content Categories

- Main content categories, which describe common content rules shared by many elements;
- Form-related content categories, which describe content rules common to form-related elements;
- Specific content categories, which describe rare categories shared only by a few elements, sometimes only in a specific context.

Content Categories



Source: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories

Content Category Description

Heading content

h1, h2, h3, h4, h5, h6

Metadata content

base, link, meta, noscript, script, style, title

Embedded content

audio, canvas, embed, iframe, img, math, object, svg, video

Phrasing content

a*, abbr, area*, audio, b, bdi, bdo, br, button, canvas, cite, code, data, date, datalist, del*, dfn, em, embed, i, iframe, img, input, ins*, kbd, keygen, label, map*, mark, math, meter, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, textarea, time, u, var, video, wbr, *Text**

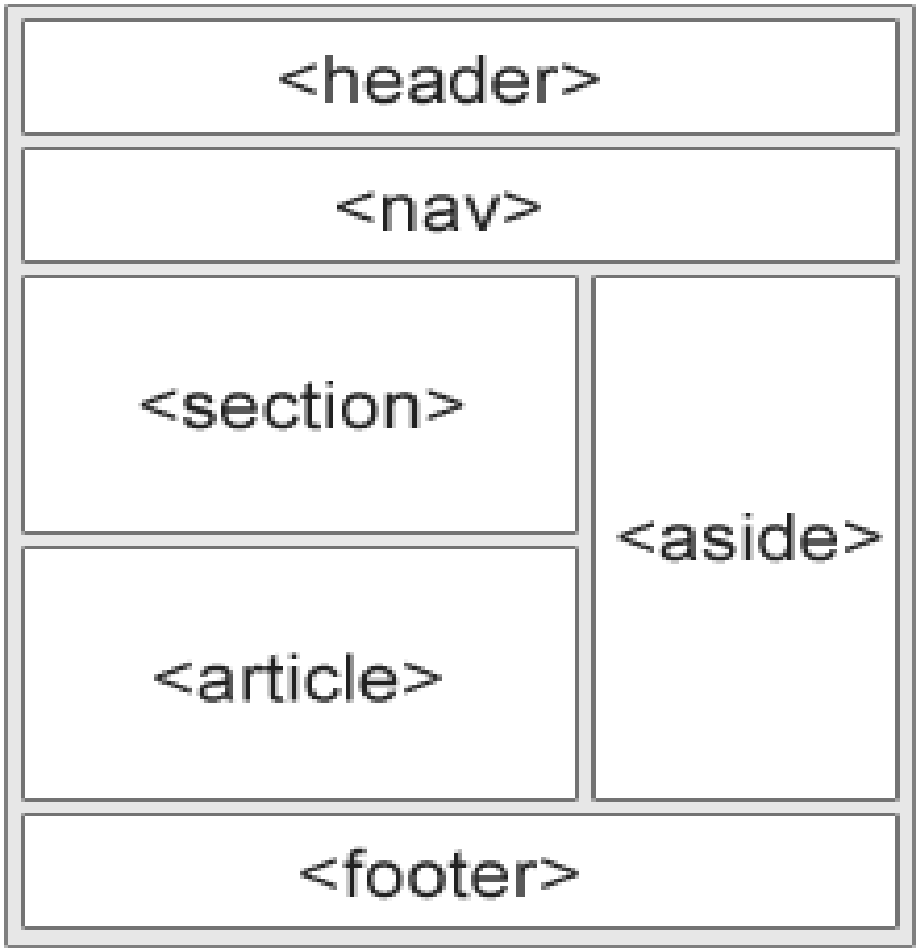
Sectioning content

article, aside, nav, section

Interactive content

a, audio*, button, embed, iframe, img*, input*, keygen, label, object*, select, textarea, video*

Web page structure



HTML5 - New Elements for better document structure

Tag	Description
<header>	Defines a header for the document or a section
<footer>	Defines a footer for the document or a section
<section>	Defines a section in the document
<article>	Defines an article in the document
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.

Contd…

Tag	Description
<nav>	Defines navigation links in the document
<progress>	Defines the progress of a task
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time

Section Element

- A section is a thematic grouping of content, typically with a heading.

```
 <section>
   <h1>Interfaces</h1>
   <p>Java doesn’t support true
multiple inheritance....</p>
 </section>
```

Article Element

- This element specifies independent, self-contained content.
- It should be possible to read it independently from the rest of the website.
  - Blog post, Technical article

Nesting of Elements

- (Nesting of elements) ? Pos~~sible : Not~~ Possible
- <section> elements can have <article> elements
- <article> elements can have <sections> elements

HTML5: MIME types

Every time your web browser requests a page, the web server sends “headers” before it sends the actual page markup. Headers are important, because they tell your browser how to interpret the page markup that follows.

MIME = Multipurpose Internet Mail Extensions

HTML5: MIME types

The most important header is called Content-Type, and it looks like this:

```
Content-Type: text/html
```

“text/html” is called the “content type” or “MIME type” of the page.

HTML5: MIME types

This header is the only thing that determines what a particular resource truly is, and therefore how it should be rendered. Images have their own MIME types (image/jpeg for JPEG images, image/png for PNG images, and so on). JavaScript files have their own MIME type. CSS stylesheets have their own MIME type. Everything has its own MIME type. The web runs on MIME types.

Form Submission

- (Input validation) ? Client Side : Server Side
  - If java script enabled – Client Side
  - If java script not enabled – Server Side

- In HTML5, no java script required for client side input validation

Input Types

- Form
  - Date (Works with chrome but not with Firefox)
  - Number
  - Fixed Increment
  - Placeholder
  - Output – Displays out put return by a script
  - Required

Type can be one of datetime, time, week, month, range with min and max attributes etc.

Media Elements

- HTML5 supports following video formats
  - Ogg
  - Mpeg-4 (Mp4)
  - WebM

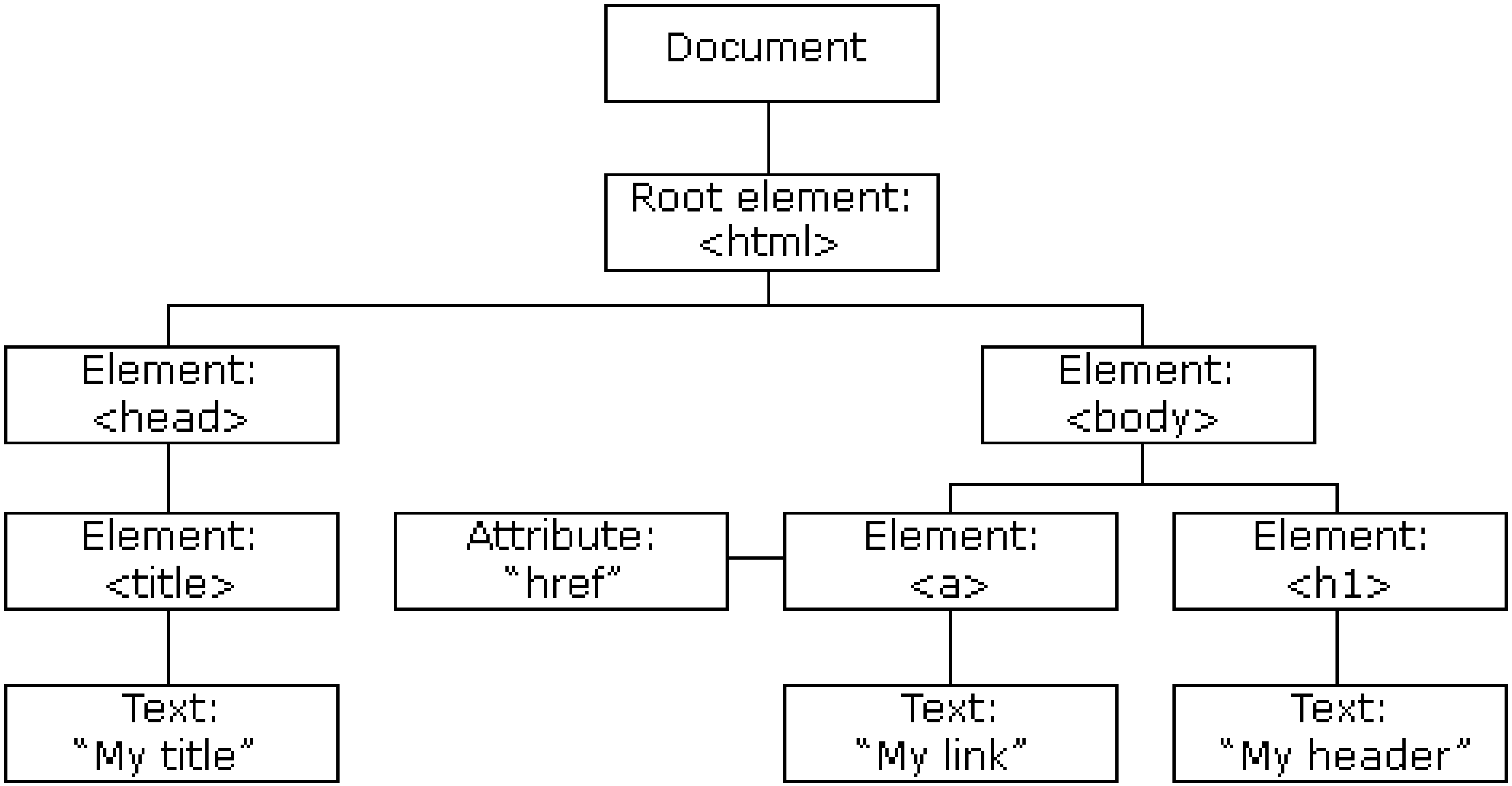
- HTML5 supports following video formats
  - MP3, WAV and Ogg

Media Elements

- Before HTML5 there was no standard for showing videos on a web page
- HTML5 video tag specifies a standard way to embed a video in a web page
- Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

DOM Tree


Document
Root element: <html>
Element: <head>
Element: <body>
Element: <title>
Attribute: "href"
Element: <a>
Element: <h1>
Text: "My title"
Text: "My link"
Text: "My header"

Document Object Model

- The W3C Document Object Model is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of a document.

- In DOM, each HTML element is considered as an object. Through object one can set/get the properties of an HTML element by using corresponding methods and event listeners.

Example

- document.getElementById(“para”).innerHTML = “Ok”;

- Above statement should be run in a script tag
  - <script type=“text/javascript” src=“demo.js” async|defer></script>
  - <noscript>Enable JavaScript in your browser </noscript>

DOM Examples

- Change html element attribute value
  - var para = document.getElementById(‘para’);
  - para.setAttribute(“innerHTML”,”Ok”);
  - para.style.color=“red”;

- Adding event listener
  - document.getElementById(“para”).addEventListener(“click”, displayDate);
  - document.getElementById(“para”).addEventListener(“click”, displayName);

Adding New Element

var lielem = document.createElement(“li”);

var text = document.createTextNode(“DoE”);

lielem.appendChild(text);

document.getElementById(“cdac”).appendChild(lielem);

Event Propagation

- Bubbling
  - In this case, the innermost element’s event will be executed first than the outer one.

- Capturing
  - In this case, the outmost element’s event will be executed first than the inner one.

Cascading Style Sheet

- Used for presentation – how web pages have to be styled or laid out etc.

- CSS Rules usually consists of
  - Set of properties with values
  - Selector to apply the updated values to

- Can be referenced from web page in one of
  - <link rel="stylesheet" href="style.css" /> | external reference
  - <div style="color:red;"> </div> | inline
  - <style> p{color:red;} </style> | using html tag

  ways

Cascading Style Sheet

```
body {
    background-color: powderblue;
}
h1, h2, h3 {
    color: blue;
}
p {
    color: red; #text color
    border: 1px solid blue; #border color
    padding: 30px; #space between text and border
    margin: 20px;  #space outsider border
}
```

Cascading Style Sheet

- To define css style for a specific element use id attribute

```
#p01 {
   color: blue;
}
<p id=“p01”>This is cdac</p>
```

- To define css style for a class of elements use class attribute

```
.para {
  color: blue;
}
<p class=“para”>This is cdac</p>
```

Cascading Style Sheet

- Element Positioning
  - CSS Properties:
    - position: static | relative | absolute | fixed | sticky
    - Helper Properties:
      - top | left | right | bottom | z-index

Note: Helper properties doesn’t work without position or position: static

Cascading Style Sheet

- position: relative;
  - An element’s new position relative to its normal position.
  - When position attribute value is non-static, its helper properties also must be set to see the effect.
- position: absolute;
  - In this case, the element will be removed from the document flow and will get placed relative to its parent element.

Cascading Style Sheet

- Animation Properties

```
div {
position: relative;
animation-name : moveltr;
animation-duration : 10s;
}
@keyframes moveltr {
from { left: 0px; }
to { left: 200px; }
}
```

Application Programming Interfaces

HTML5 Canvas

- Canvas element is used to draw graphics on the fly via scripting
- It is only a container for graphics and use some scripting to actually draw on the canvas

Example: <canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">

- The getContext() method returns an object of type canvas which has several methods to be invoked on it for drawing purpose.

HTML5 Drag and Drop

- Steps:
  - Make an element draggable by setting draggable attribute to true. <img draggable=true>
  - Set an event listener for dragstart that stores the data being dragged.
  - Set an event listener for drop event

Example

Geo Location & Web Storage

- Locate user's geo location
  - Get the location of this PC
  - getCurrentPosition() method is used to get the user's position.
  - See the example code for displaying different types of maps

- HTTP State Maintenance
  - Before HTML5 :
    - The cookie specifications require that browsers must meet the following requirements in order to support cookies:
      - Can support cookies as large as 4,096 bytes in size
      - Can store at least 50 cookies per domain (i.e. per website)
      - Can store at least 3000 cookies in total

Contd…

- HTTP State Maintenance
  - After HTML5
    - Local Storage
      - Can store up to 5MB data
      - Information is never transferred to the server
      - Similar to persistent cookies concept
    - Session Storage
      - Similar to session cookies
      - Difference is information stored would not be shared between new and old window/tab unlike in the case of session cookie.

Web DB

- User agents need to store large numbers of objects locally in order to satisfy off-line data requirements of Web applications.

- Web Storage is useful for storing pairs of keys and their corresponding values

- Limitation with web storage is
  - does not provide in-order retrieval of keys,
  - efficient searching over values,
  - storage of duplicate values for a key.

Web Worker

- Through webworker we can run scripts in background threads without interfering with the user interface.

- Worker can communicate with the JavaScript which initiated the web worker by posting messages to an event handler specified by that code (and vice-versa).

- Sample Program

Contd…

- Worker’s context is different from that of window context, it is represented by DedicatedWorkerGlobalScope or SharedWorkerGlobalScope.

- Rules
  - You can run any JavaScript code inside a worker
  - You can’t manipulate DOM inside a worker
  - Can’t access some default methods or properties of window object
  - For a list of methods and properties that can be accessed from

https://developer.mozilla.org/en-US/docs/Web/API/ Web_
Workers_API/Functions_and_classes_available_to_workers

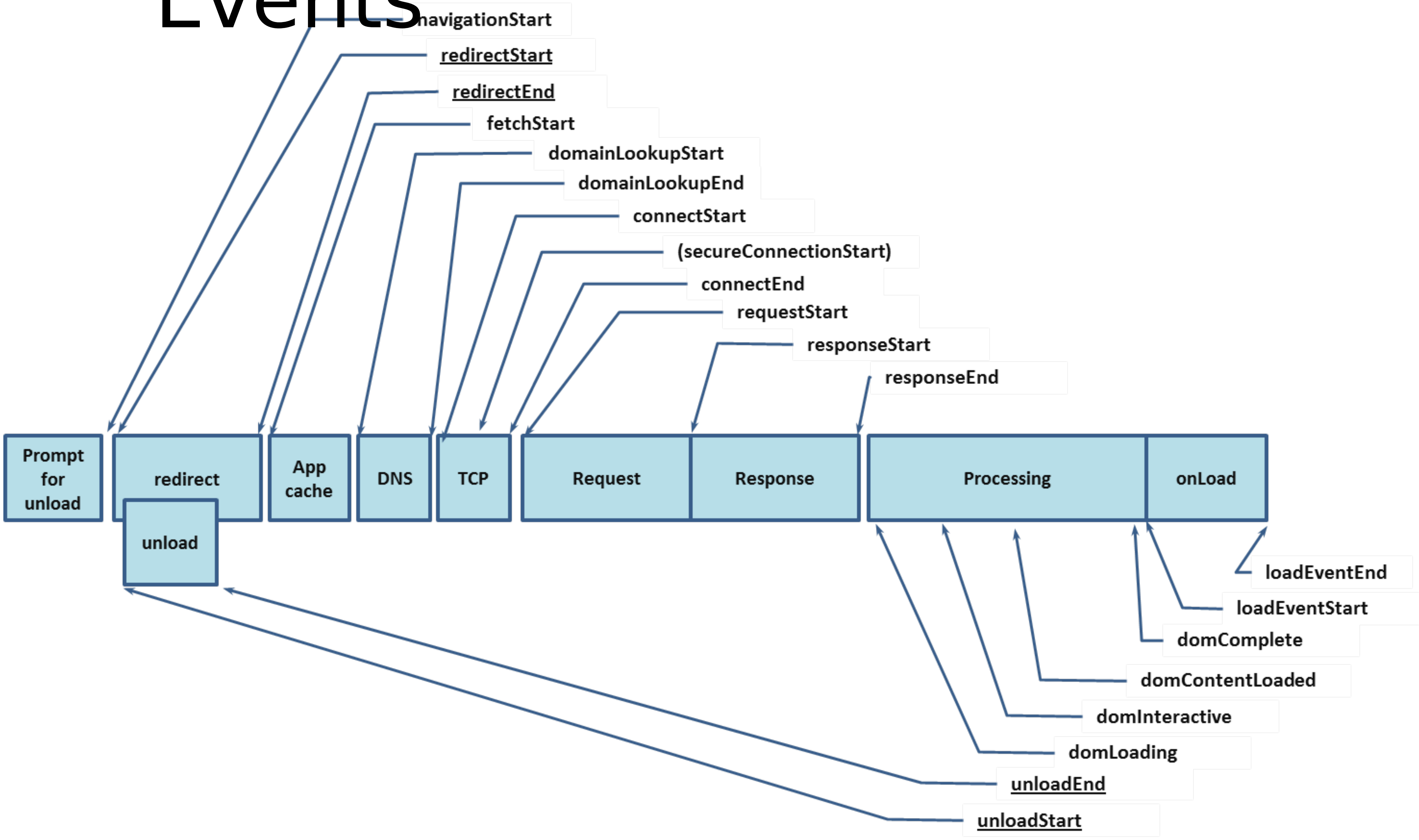
Web API (Few more) Performance Measurement

- User latency is an important quality bench mark for Web applications.

```
<html>
<head>
<script type="text/javascript">
var start = new Date().getTime();
function onLoad() {
var now = new Date().getTime();
var latency = now - start; alert("page loading time: " + latency); }
</script>
</head>
<body onload="onLoad()">
<!- Main page body goes from here. --> </body>
</html>
```

- It does not give information about the time it takes to get the page from the server (details about end to end latency).

Sequence of Page Loading Events



Navigation Timing

```
<html>
<head>
<script type="text/javascript">
   function onLoad() {
   var now = new Date().getTime();
   var page_load_time = now - performance.timing.navigationStart;
   alert("User-perceived page loading time: " + page_load_time);
   }
</script>
</head>
<body onload="onLoad()">
 <!- Main page body goes from here. -->
</body>
 </html>
```

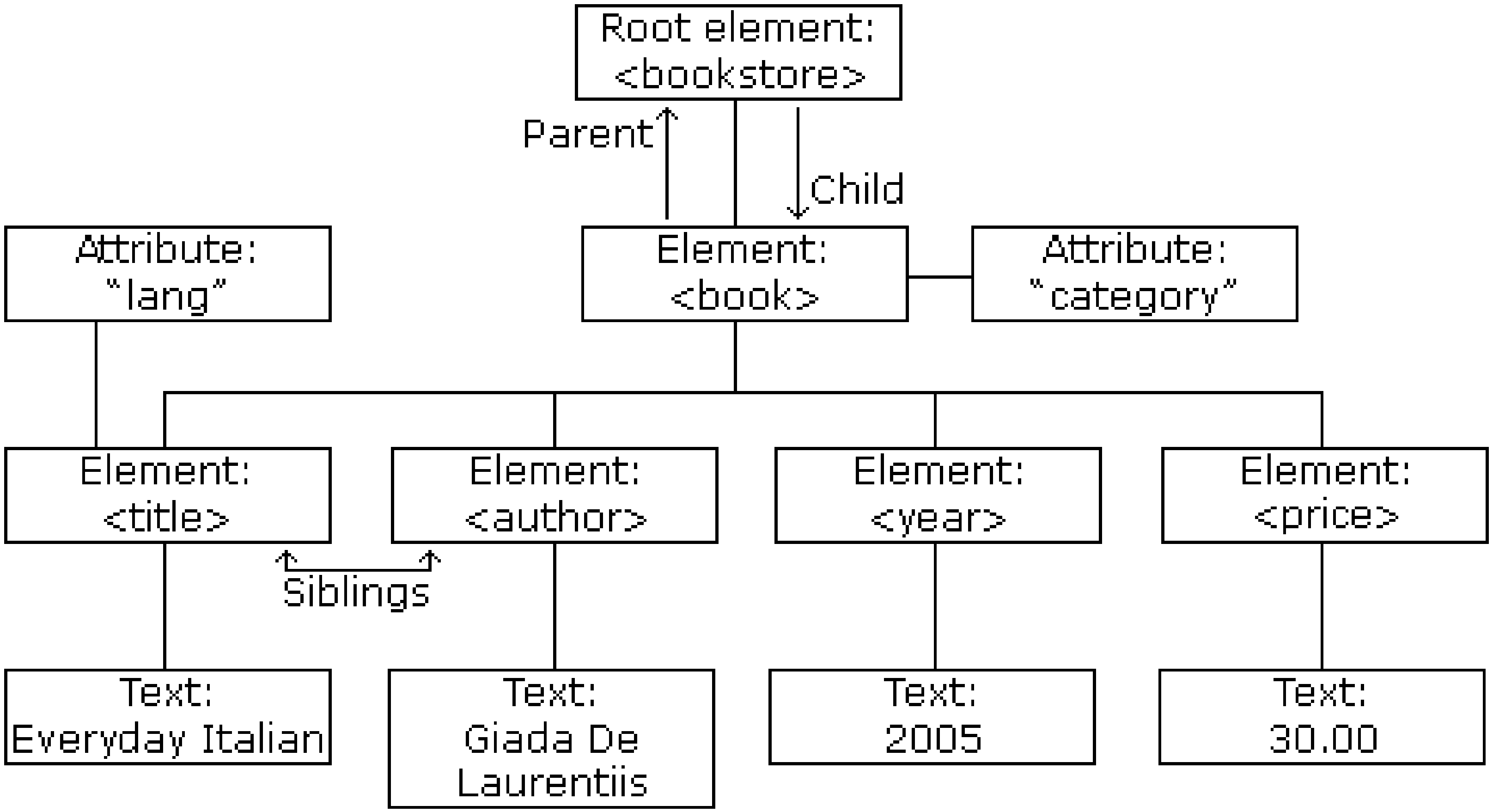
eXtensible Markup Language

- Designed for storing and transporting data
- XML document can be read by both human and machine
- XML does not use predefined tags as in the case of HTML

Sample XML Code

```
<?xml version="1.0" encoding="utf-8"?>
<nitf>
   <head>
        <title>  CDAC  </title>
   </head>
</nitf>
```

XML Tree


Root element:
<bookstore>
Parent
Child
Attribute:
"lang"
Element:
<book>
Attribute:
"category"
Element:
<title>
Element:
<author>
Element:
<year>
Element:
<price>
Siblings
Text:
Everyday Italian
Text:
Giada De Laurentiis
Text:
2005
Text:
30.00

XML Rules

- Attribute values must always be quoted
- XML elements must be properly nested
- Tags are case sensitive
- XML Prolog (<?xml version=“1.0” encoding=“utf-8” ?>) is optional
- The document must contain a root element
- Comments syntax is same as that of

Entity References

- Certain symbols such as >, < etc., cannot be used in XML. Instead corresponding entity reference have to be used to avoid parsing errors.

&lt;	<
&gt;	>
&amp;	&
&apos;	‘
&quot;	“

XML Document

- XML documents can be extended by some more information any time.

- The application using the XML document will not crash or break if it is extended with some more information.

- Example:

  -
    ```
    <note>
      <date>2008-01-10</date>
      <to>CDAC CHN</to>
      <from>CDAC HYD</from>
    </note>
    ```