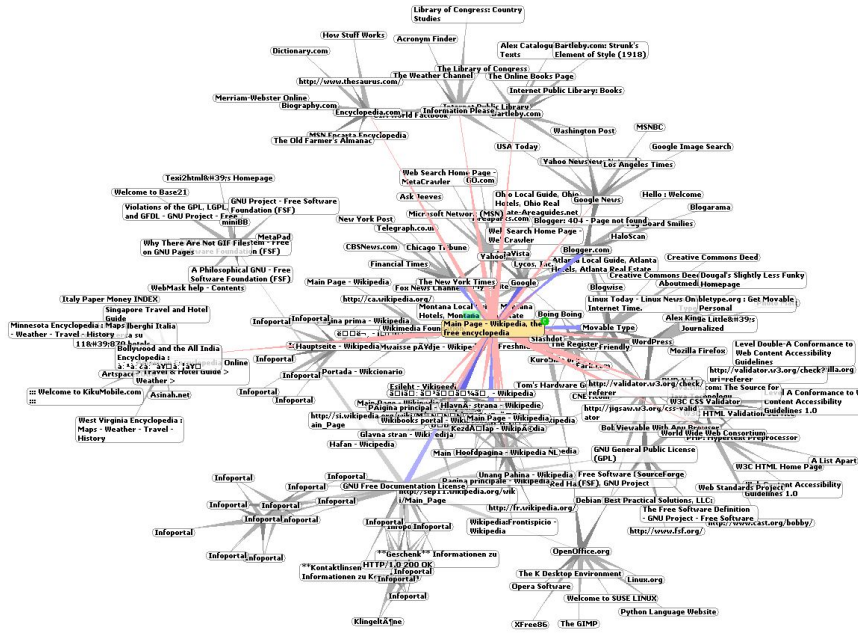
Basic Client Side

BWA Slides #1

Content

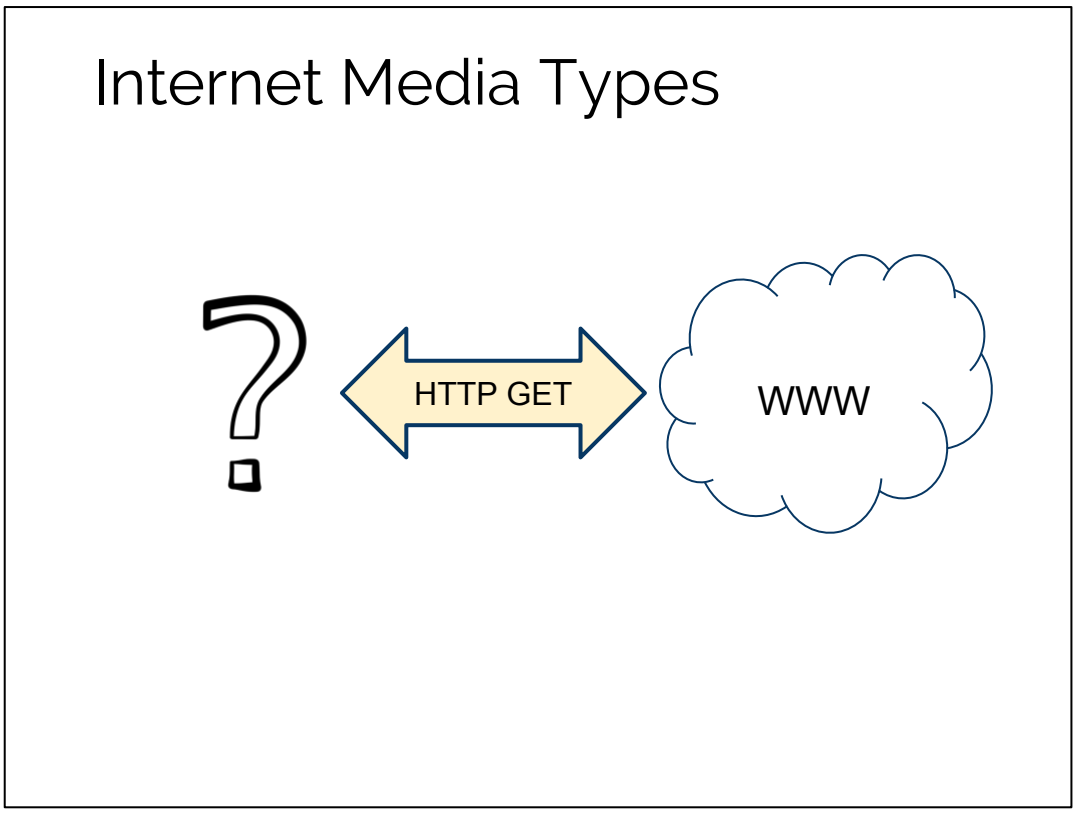
- WWW
- Internet Media Types (MIME)
- HTML5,
- URI
- Document Object Model, DOM
- Basic CSS
- Bootstrap
- DOM Events
- Intro to JavaScript
- The Browser
- Chrome Developer Tools

World Wide Web



The World Wide Web (www, W3) is an information space based on;

- Internet
- The Hypertext transfer protocol, HTTP
- A publishing language, the HyperText Markup Language, HTML
- Documents and resources identified with Uniform Resource Identifiers, URIs
- Hyperlinks between documents and resources



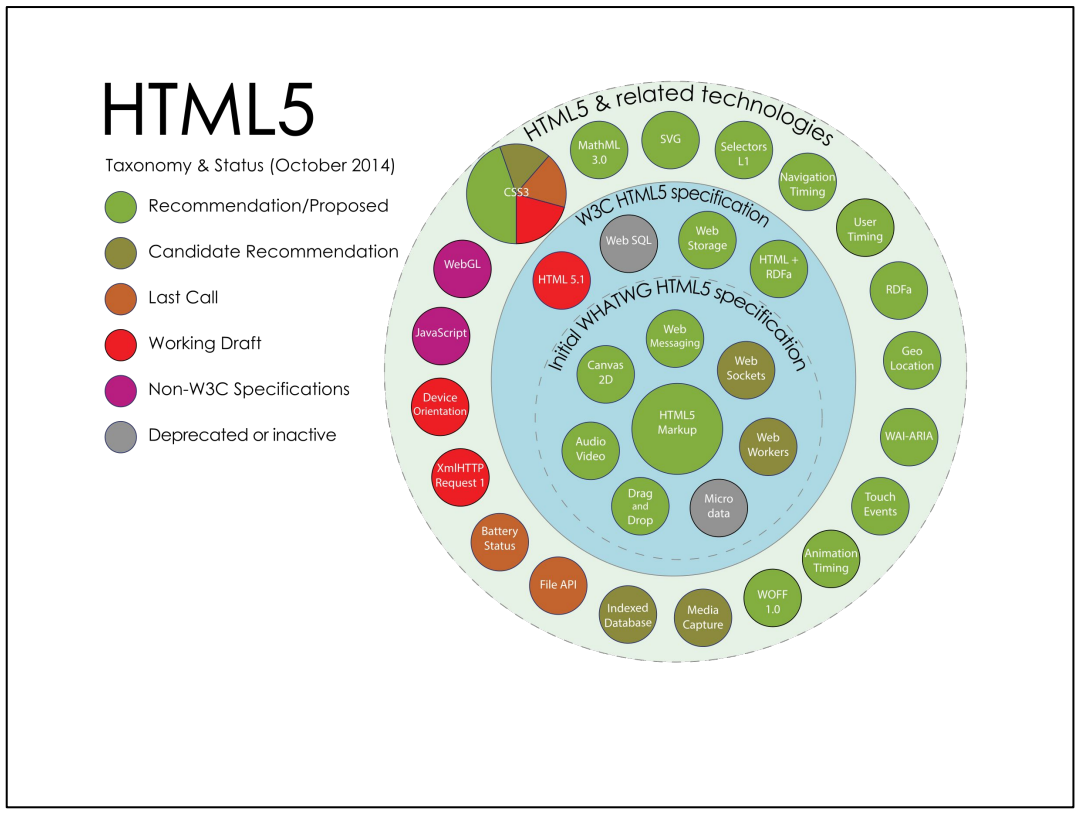
HTTP is capable of carrying arbitrary "labeled" content. The mechanism used to label such content is a media type, consisting of a top-level type and a subtype and optional parameters (previously known as MIME, some still say...), see RCF 6838

- Used in request header (Accept) and response header (Content-Type)

Some common types: text/html, text/css, text/plain, text/xml, image/png, application/pdf,
application/xhtml+xml. Full list of med media types.

"...when a client computer requests a Web page from the server, the client computer uses media types to tell the server what type of files the client computer will accept. Conversely, when the server sends files back to the client computer, the server uses media types to identify what type of files it is sending back". See Media Types

Plugins see HTML5 specification upcoming ...



HTML and related technologies described by web specifications/recommandations (web standards)

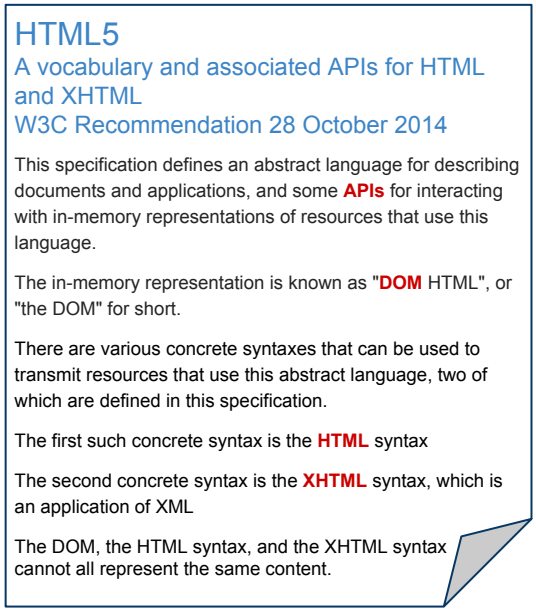
- HTML5 often used as an umbrella concept involving many standards
- HTML5 is not same as HTML4

Producers of web standards;

- Web Hypertext Application Technology Working Group (WHATWG)
- World Wide Web Consortium (W3C)
- Ecma International

We will mostly use the W3C HTML5 recommendation

W3C HTML5 Recommendation



HTML5 Recommendation;

- Design notes
- Specification describes two languages, HTML5 and XHTML5!
  - XHTML is HTML with the "rules" for XML …
    - …well formedness (i.e. (self)closing tags) and more
- We mainly use HTML5 but in some situation we need XHTML5.
- Recommendation uses word normative (and non-normative)
  - Normative: "The way it should be"

Many dependencies on other specifications

HTML Syntax

```
<!DOCTYPE html>
<html>
 <head>
  <title>Sample page</title>
 </head>
 <body>
  <h1>Sample page</h1>
  <p>This is a <a href="demo.html">simple</a>
                                        sample.</p>
  <!-- this is a comment -->
 </body>
</html>
```

HTML documents must consist of the following parts, in the given order (text is allowed inside element's, attribute values, and comments);

- Optionally, a single "BOM" (U+FEFF) character.
- Any number of comments and space characters.
- A DOCTYPE.
- Any number of comments and space characters.
- The root element, in the form of an html element (optional but if not present often automatically inserted, see below).
- Any number of comments and space characters.

Comments; <!-- *some text* --> (not nested, no "--" inside and more …)

Elements

- There are five different kinds of elements; void elements, raw text elements, escapable raw text elements, foreign elements, and normal elements.
- Tags are used to delimit the start and end of elements in the markup. The content must be placed in between start and end tag.
- Void elements only have start tag (i.e. no content).
- Some tags are optional (avoid!!)
- Attributes for an element are expressed inside the element's start tag (if any).
- Attribute values are a mixture of text and character references (char's not on keyboard, see below), except with the additional restriction that the text cannot contain an ambiguous ampersand
- Attribute values; Empty syntax, Unquoted syntax, Single quoted syntax,

- double quoted syntax
- Attributes values sometimes must have specific data type (although written as a string)
- Case sensitivity: Sometimes … (tag names insensitive).
    - We use lowercase.

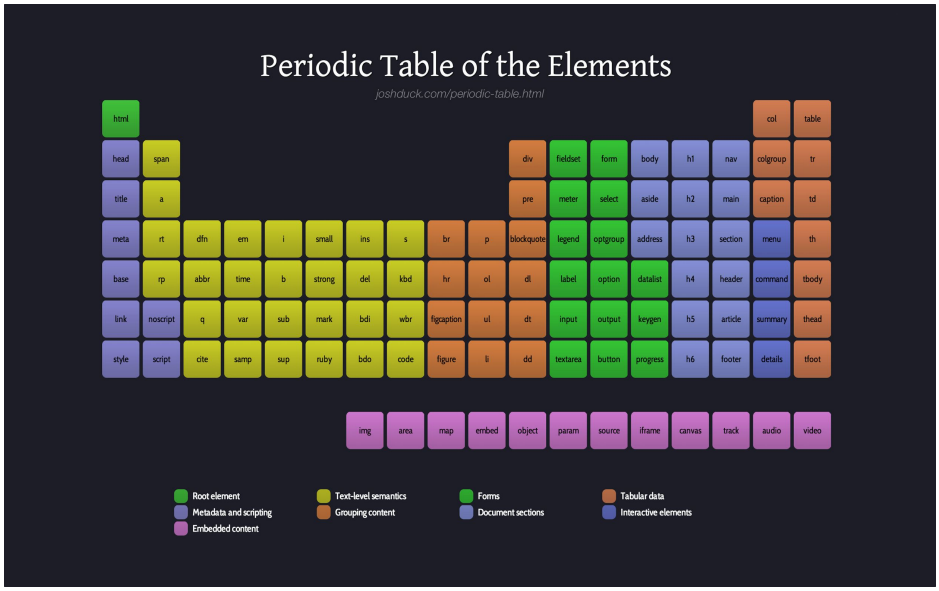
Head and Body elements

- The head element represents a collection of metadata for the Document.
    - Metadata important for searchengines … or not …
- The body element represents the (visible) content of the document.

Character entities

- Reserved characters in HTML must be replaced with character entities.
- Characters, not present on your keyboard, can also be replaced by entities.

HTML Elements



Not exactly as in recommendation but nice overview.
See also MDNs HTML element reference and w3schools HTML Reference

Some Global Attributes

Id

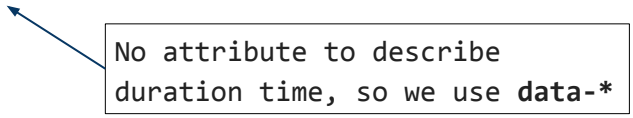
```
<section id="content"> ... </section>
```

Class

```
<article class="important"> ... </article>
```

Custom Attribute

```
<li data-length="2m11s">Beyond The Sea</li>
```



Global attributes

- id, Element id. The value must be unique amongst all the IDs in the element's home subtree and must contain at least one character.
- class, The attribute, if specified, must have a value that is a set of space-separated tokens representing the various classes that the element belongs to i.e. specify set belonging for elements.
- data-*, (attribute prefixed with data-) intended to store custom data private to the page or application, for which there are no more appropriate attributes or elements. These attributes are not intended for use by software that is independent of the site that uses the attributes.

HTML Semantics

4.9.1 The `table` element

Categories:

Flow content.
Palpable content.

Contexts in which this element can be used:

Where flow content is expected.

Content model:

In this order: optionally a `caption` element, followed by zero or more `colgroup` elements, followed optionally by a `thead` element, followed optionally by a `tfoot` element, followed by either zero or more `tbody` elements or one or more `tr` elements, followed optionally by a `tfoot` element (but there can only be one `tfoot` element child in total), optionally intermixed with one or more script-supporting elements.

Content attributes:

Global attributes
`border`
`sortable` - Enables a sorting interface for the table

Tag omission in text/html:

Neither tag is omissible

Allowed ARIA role attribute values:

Any role value.

Allowed ARIA state and property attributes:

Global aria-* attributes
Any `aria-*` attributes applicable to the allowed roles.

DOM interface:

IDL

```
interface HTMLTableElement : HTMLElement {
             attribute HTMLTableCaptionElement? caption;
  HTMLElement createCaption();
  void deleteCaption();
             attribute HTMLTableSectionElement? tHead;
  HTMLElement createTHead();
  void deleteTHead();
             attribute HTMLTableSectionElement? tFoot;
  HTMLElement createTFoot();
  void deleteTFoot();
  readonly attribute HTMLCollection tBodies;
  HTMLElement createTBody();
  readonly attribute HTMLCollection rows;
  HTMLElement insertRow(optional long index = -1);
  void deleteRow(long index);
             attribute DOMString border;
};
```

Element definition

Elements, attributes, and attribute values in HTML are defined to have certain meanings (semantics).

- 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 (browsers, search engines).

Browsing context

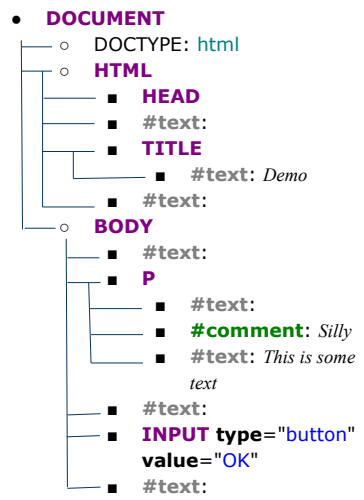
- A tab or window in a Web browser typically contains a browsing context, as does an iframe or frames in a frameset.
  - iframe etc. will create nested browsing contexts
  - NOTE: This is a security risk, … clickjacking.
- In the browsing context the “Document” is displayed (normally in this course the HTML the page)

Document vs DOM

HTML Document

```
<!DOCTYPE html>
<html>
    <head>
        <title>Demo</title
    </head>
    <body>
        <p>
            <!-- Silly  -->
            This is some text
        </p>
        <input type="button"
    value="OK" />
    </body>
</html>
```

HTML DOM

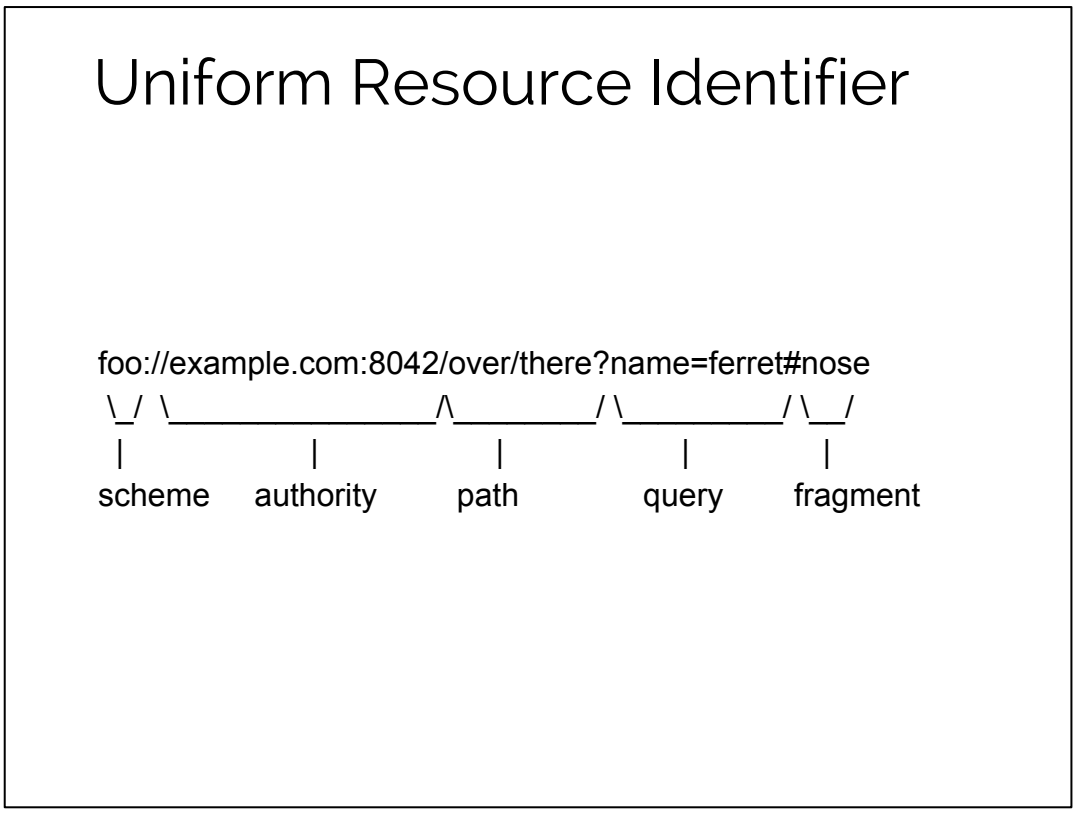


[Document](#)

- Text document written in (X)HTML syntax (textual data, UTF8 encoded)
- Composed of elements with attributes etc.

[DOM](#)

- In memory tree structure composed of nodes (representing the document)
- [Node tree](#) (specified in [W3C DOM 4](#) recommendation, not in HTML5)
- Nodes have properties (attributes)
    - NOTE: Some nodes have mutual state
- Nodes have methods
    - HTML5 recommendation specifies DOM API for all nodes
    - Nodes are objects implementing any of interfaces: Document, DocumentFragment, DocumentType, Element, Text, Processing Instruction, or Comment interface (or subinterfaces)
    - APIs for elements described using "web interface definition language" code, from the [WebIDL specification](#)
- Test with [Live DOM viewer](#)



Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. Defined in RFC3986 RFC3987. (RFC)

URI composed of (simplified);

- Scheme: The protocol (http, https, ws, …)
- Authority: The server name or address, optionally port and user info. Creates a scope for the remaining parts of the URI.
- Path: (hierarchically) identifies resources within the scope of the scheme and authority. Terminated with ? or #.
    - If there is something following the resource else no ? or #
- Query: (non hierarchically) identification of resources. Possibly multiple name/value pairs using ampersand (name1=value1&name2=value2&...)
- Fragment: Starting with #. Given a primary resource (page), identifies secondary resource given an identifier (label in page). Fragment part is local (not sent with any request)
- URI vs URL
    - Many authors use URL and/or URI interchangeably (or as equivalent).

An URI reference is either an URI or an, a relative reference.

- A relative reference consists only of the path, and optionally, the resource, but no scheme or server.
- If using a relative reference, resolve the relative reference, to find the resource

- Relative reference vs URI, pros and cons ?
    - "In general, it is considered best-practice to use relative URIs, so that your website will not be bound to the base URI of where it is currently deployed. For example, it will be able to work on localhost, as well as on your public domain, without modifications."
    - True for static pages, but for Web applications?... more to come ...

Percent Encoding: If using URI reserved characters for some other purpose must percent encode

HTML Links

Anchor

```
<a href=”https://developer.mozilla.org”>MDN</a>
```

Area

```
<area shape="rect" coords="25,25,125,125" href="red.html"
alt="Red box."/>
```

Link

```
<link rel="stylesheet" type="text/css" href="theme.css"/>
```

Links are a conceptual construct, created by a, area (with href attribute), and link elements, that represent a connection between two resources, one of which is the current Document. There are two kinds of links in HTML:

- Hyperlinks (a and area): Exposed to the user by the user agent (browser) so that the user can cause the user agent to navigate to those resources, e.g. to visit them in a browser or download them.
- Links to external resources (link): These are links to resources that are to be used to augment the current document, generally automatically processed by the user agent (browser find style sheets).

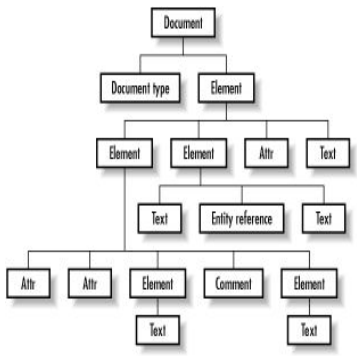
Some attributes;

- href a valid URI
- download (a and area only) link intended for downloading
- rel Relationship between the document containing the hyperlink and the destination resource
- type (link only) Hint for the type of the referenced resource

Area tag normally used with image maps

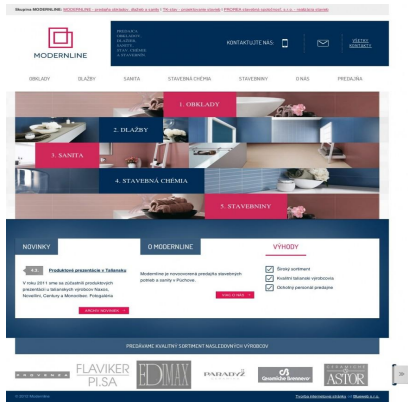
Cascading Style Sheets

The structure (HTML/DOM)



CSS

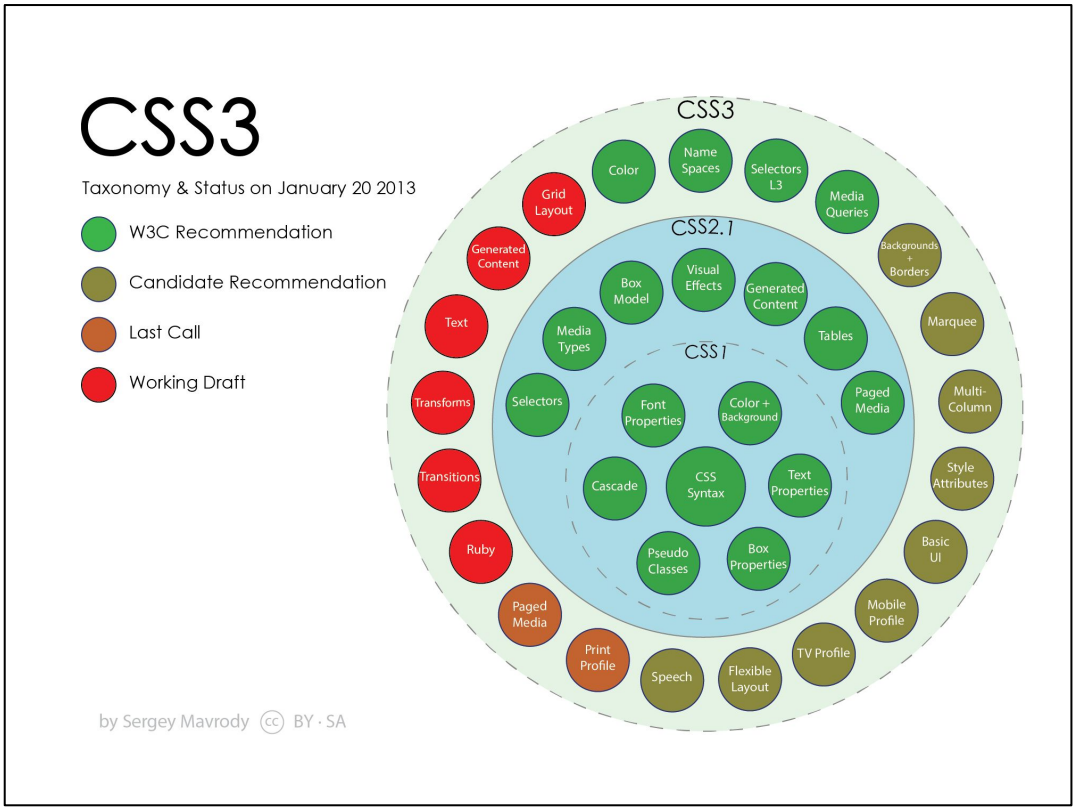
The view




```
<link href="resources/css/default.css" rel="stylesheet" />
```

[Rendering]

- User agents are not required to present HTML documents in any particular way. Recommendation provides a set of suggestions for rendering HTML documents that, if followed, are likely to lead to a user experience that closely resembles the experience intended by the documents' authors
- In general, user agents (browsers) are expected to support Cascading Style Sheets (CSS), and many of the suggestions in recommendation are expressed in CSS terms.



Cascading Style Sheets (CSS) is a language for describing the presentation semantics of markup documents

- To produce a (nice) 2D view of the DOM tree (or XML DOM-tree)
  - Possible to
    - do layout (positioning)
    - set styles (fonts, colors, backgrounds,...)
    - animations, ...
- In files*.css
- Linked to HTML document using <link> tag. Downloaded when browser hits tag (and cached, more later).
- Cascade
  - There can be many CSS involved: author, user, user agent (browser), ... may overlap !!
  - The "cascade" resolves the interaction

CSS3 is an umbrella specification

- Builds on CSS 2.2 (level 2, revision 2) and a plethora of modules
- CSS3 Overview
  - "Unlike CSS 2, which is a large single specification defining various features, CSS3 is divided into several separate documents called "modules". Each module adds new capabilities or extends features defined in CSS 2, preserving backward compatibility."

CSS allows authors to move style information to a separate style sheet resulting in

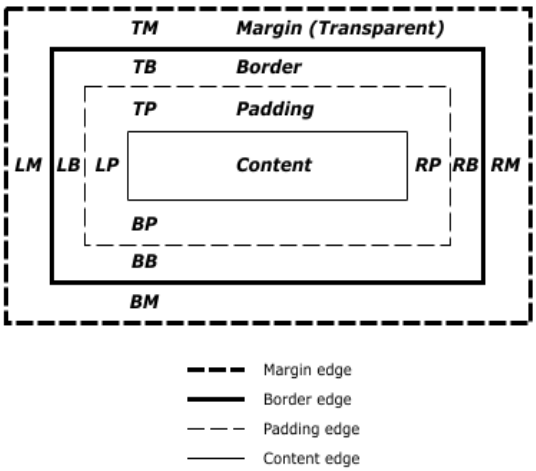
considerably simpler HTML markup

- Much easier to maintain, possible to change “look” without changing structure (HTML)

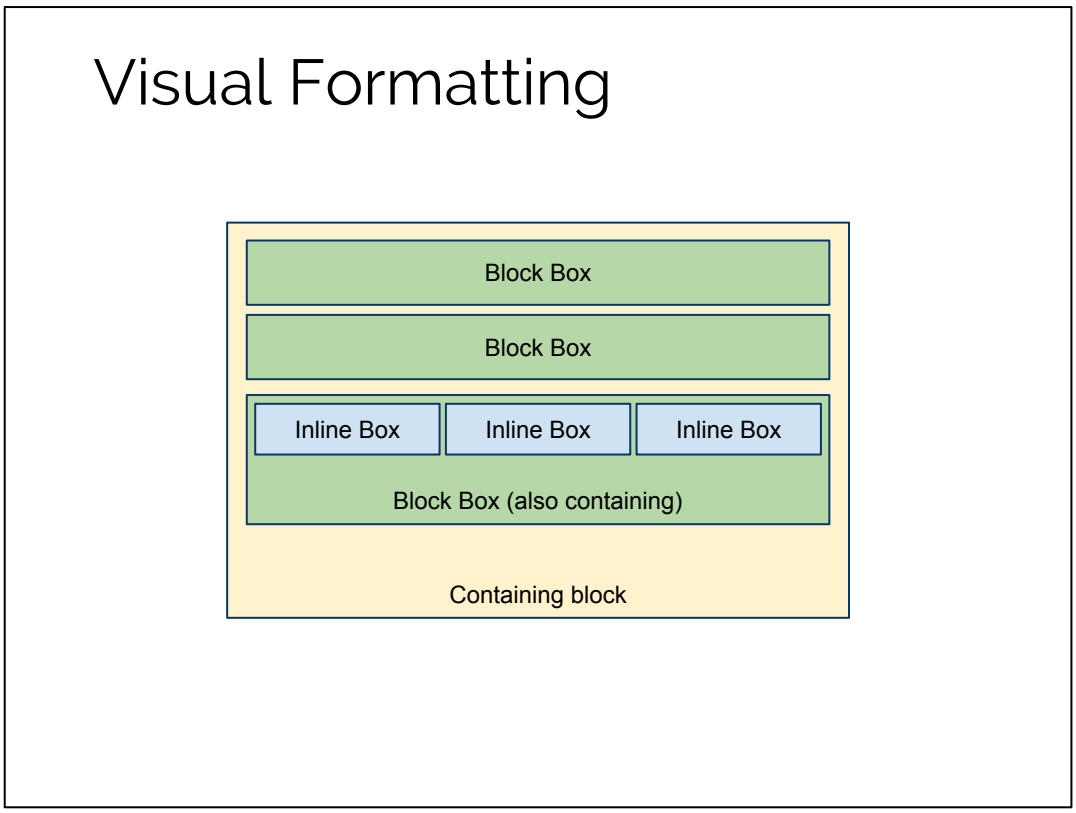
All HTML elements may have style attribute (inlinestyle).

- We don't use! Separation of concerns!
- We use id and class in conjunction with external style sheets (CSS), more to come…

The Box Model



The CSS box model describes the rectangular boxes that are generated for HTML elements in the document tree and laid out according to the visual formatting model.



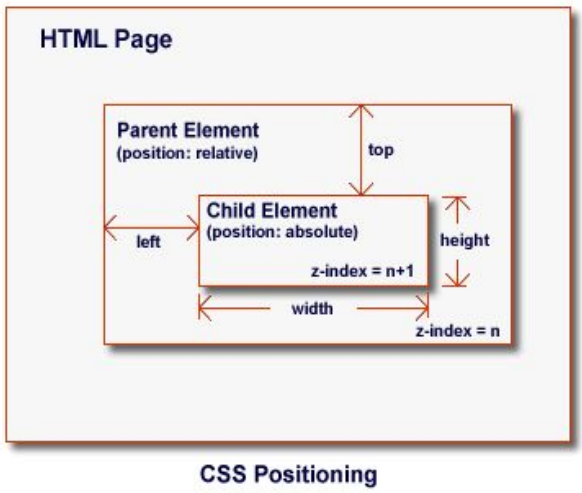
Visual Formatting

- Many box positions and sizes are calculated with respect to the containing block. In general, generated boxes act as containing blocks for descendant boxes
- Block boxes are laid out one after the other, vertically, beginning at the top of a containing block. Will have a newline.
- Inline boxes are distributed in lines of containing block
- This may be altered using positioning schemas, upcoming …

Some HTML elements generating block boxes

- article, aside, h1, h2, h3, h4, h5, h6, hgroup, nav, section, address, blockquote, center, div, figure, figcaption, footer, form, header, hr, legend, listing, p, plaintext, pre, xmp

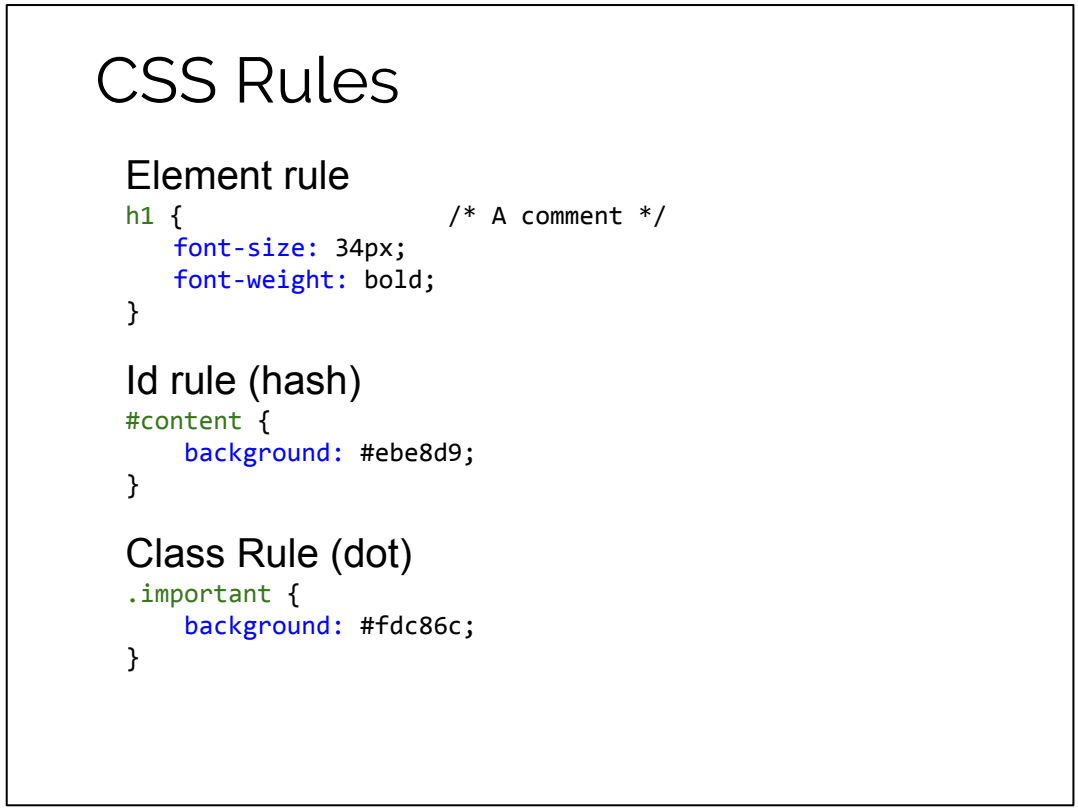
Positioning Schemas



CSS Positioning

In CSS 2.2, a box may be laid out according to three positioning schemes:

- Normal flow. In CSS 2.1, normal flow includes block formatting of block-level boxes, inline formatting of inline-level boxes, and relative positioning of block-level and inline-level boxes.
- Floats. In the float model, a box is first laid out according to the normal flow, then taken out of the flow and shifted to the left or right as far as possible. Content may flow along the side of a float.
- Absolute positioning. In the absolute positioning model, a box is removed from the normal flow entirely (it has no impact on later siblings) and assigned a position with respect to a containing block.



Syntax

- Case insensitive
- Comments begin with the characters "/*" and end with the characters "*/"
- A rule set (also called "rule") consists of a selector followed by a declaration block.
- Selectors are patterns used to determine which style rules apply to elements in the document tree.
- A declaration block starts with a left curly brace ({) and ends with the matching right curly brace (}). In between there must be a list of zero or more semicolon-separated (;) declarations.
- A declaration is either empty or consists of a property name, followed by a colon (:), followed by a property value. Around each of these there may be white space.
- The properties.
- Vendor specific extensions
- There are types for values.
- Illegal parts ignored (so if error … no nice rendering, … watch out!)

CSS refactoring

CSS @Rules

@Rules

```
@import "print-main.css" print;
@import url("fancyfonts.css") screen;
@media print {
  body { font-size: 10pt }
}
```

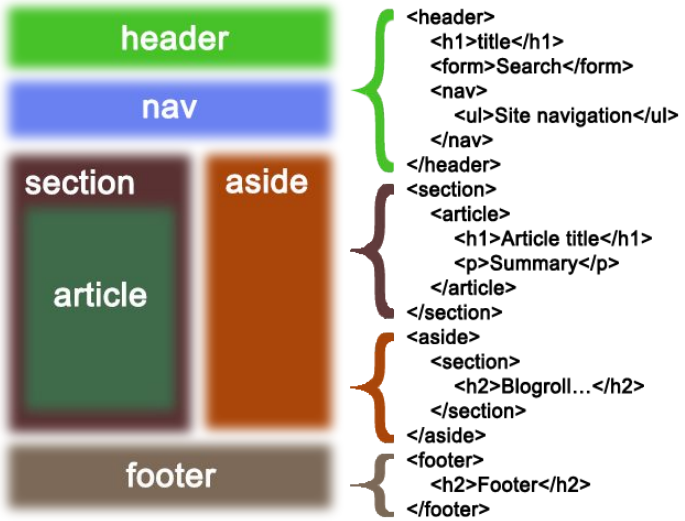
Some special rules

- The '@import' rule allows users to import style rules from other style sheets. Any @import rules must precede all other rules
- An @media rule specifies the target media types (separated by commas) of a set of statements (delimited by curly braces).

Also media queries

- A media query consists of a media type and zero or more expressions that check for the conditions of particular media features. Among the media features that can be used in media queries are 'width', 'height', and 'color'.

Basic Page Structure with HTML/CSS



Creating a basic page style (layout and styles)

- HTML document using section tags
- CSS style sheet to normalize the default browser style (normalize.css or similar recommended)
- CSS stylesheet (default.css) containing positioning rules for the section tags …
- … and colors, fonts, etc. for remaining tags

Front End User Interface Frameworks

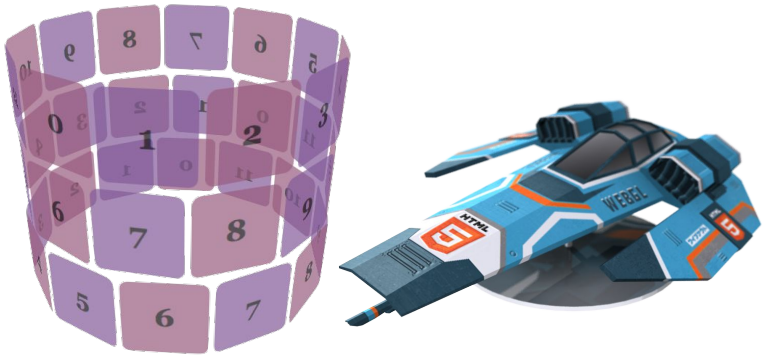
```
<body class="container">


<nav class="navbar navbar-default" role="navigation">
    <a class="navbar-brand" href="/home">Home</a>
  ...
</nav>
```

We're not graphical designers …

- CSS very tedious …
- Better use some client side framework
- Possibly Bootstrap
  - Bootstrap components …
  - …mostly using the class-attribute

NOTE: Need some JavaScript to function, more to come …

Web Graphics



Graphics and animations with HTML/CSS (i.e. no need for plugins like Flash?!)

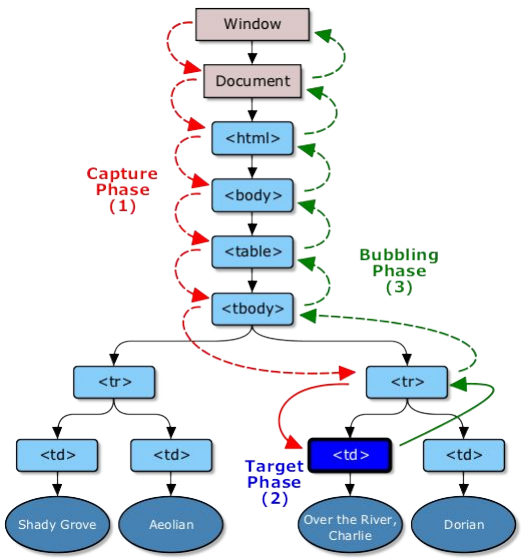
- Canvas (tutorial)
- 2D breakout game using pure JavaScript (click result to run)
- Transforms
- Transitions

Full games, OpenGL in the browser

- WebGL (need browser support and JavaScript)

Not much covered in course … possibly own explorations during project(?)

DOM Events



DOM Events

- Goal is the design of an event system which allows registration of event listeners and describes event flow through a tree (DOM) structure.

Event handler content attributes

- May be specified on any HTML element:
- Attributes like: onmousedown, onmouseenter, onmouseleave, onmousemove, onmouseout, etc
- Event handler content attributes (values), when specified, must contain valid JavaScript code

Simplifies summary:

- We attach JavaScript functions as event listeners.
- When event fires JS function called (similar to Swing)

NOTE: Capture, bubbling makes it possible to use same listener for many elements

Intro to JavaScript

f.js file

```
function removeTable() {
    var div =
              document.getElementById("table");
    while (div.firstChild) {
        div.removeChild(div.firstChild);
    }
}
```

HTML Page

```
<script type="text/javascript" src="f.js"></script>
<input type="button" value="Remove Table"
                        onclick="removeTable();" />
```

For now we just observe

- JavaScript is a scripting language executed in the browser (downloaded and executed when browser hits <script> tag in page, cached!)
  - See Page Loading later
- We use the <script> tag in the <head> section to download the JS code (code in separate file).
  - It possible to inline JavaScript code in HTML pages. We avoid!
- Scripts are not statically typed (no types in code)
- Most constructs should be familiar: variables, assignments, references, if, while, ...
- It's possible to create functions in JavaScript and registers as listeners (for some HTML element)
- The browser environment supplies JavaScript objects implementing the DOM APIs. Some examples are window, history and document.
- JavaScript functions have access to the DOM API objects. Possible to manipulate DOM!

JavaScript functions

- A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().
- Function names may contain letters, digits, underscores, and dollar signs.
- The parentheses may include variable number of parameter

- The code to be executed, by the function, is placed inside curly brackets: {}
- If return statement in body, will return some value else a void-function
- Function executes statement by statement (like Java)
- … more later

The Browsers

A web browser (a browser) is a software application for retrieving, presenting and traversing information resources on the World Wide Web.

"... the browser's user interface is not specified in any formal specification, it is just good practices shaped over years of experience and by browsers imitating each other." //How browsers work

Browser functionality (for end user)

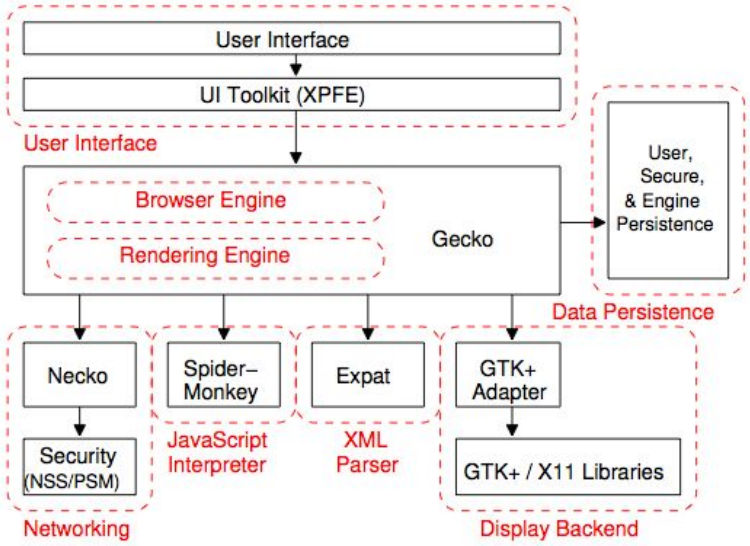
- Address bar for inserting the URI. If URI entered into the address bar of the browser
  Back and forward buttons, navigating to previous or next URL (browser has a history list of URL)
- A refresh and stop buttons for refreshing and stopping the loading of current documents
  Home button that gets you to your home page
- Bookmarking options
- Tabs, to be able to browse more resources

Browser cache: The browser will (for efficiency reasons) cache pages, images, etc.

- More on browser behaviour when using forms ...

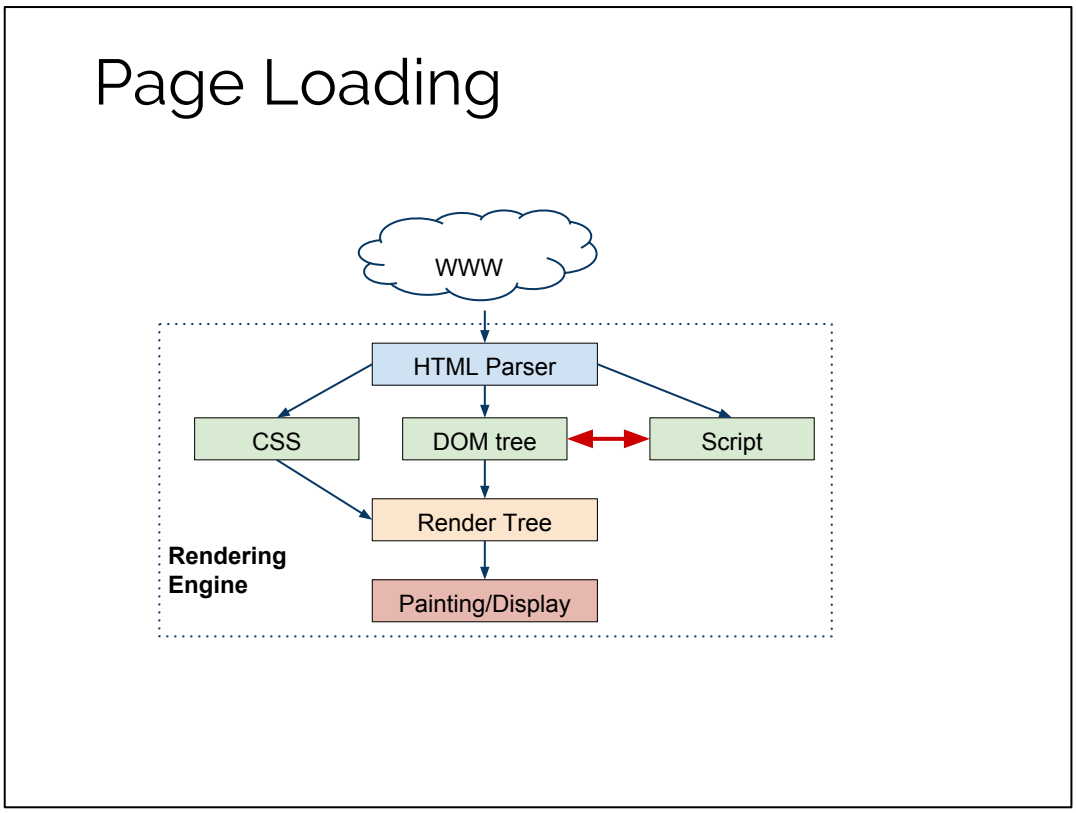
Quirksmode

Mozilla Browser Architecture



Top level components for Mozilla Firefox Browser

- The user interface
- Gecko: Browser and render engine responsible for displaying the requested content. For example if the requested content is HTML, it is responsible for parsing the HTML and CSS and displaying the parsed content on the screen. Other common browser/render engine is WebKit.
    - Microsoft has it's own proprietary Trident (or upcoming Spartan)
- Data persistence: This is a persistence layer. The browser needs to save all sorts of data on the hard disk
- Networking: Used for network calls, like HTTP requests.
- JavaScript interpreter: Used to parse and execute the JavaScript code.
- XML parser
- Display backend: Used for drawing basic widgets like combo boxes and windows



General flow (from How Browsers Work)

- The rendering engine will start parsing the HTML document and turn the tags to DOM nodes in a tree called the "content tree".
- It will directly execute the JavaScript code when encountered. Script may alter DOM during construction (we avoid this using special constructs)!!!
- It will parse the style data, both in external CSS files and in style elements. The styling information together with visual instructions in the HTML will be used to create another tree - the render tree.
- The render tree contains rectangles with visual attributes like color and dimensions. The rectangles are in the right order to be displayed on the screen.
- After the construction of the render tree it goes through a "layout" process. This means giving each node the exact coordinates where it should appear on the screen.
- The next stage is painting - the render tree will be traversed and each node will be painted using the UI backend layer.

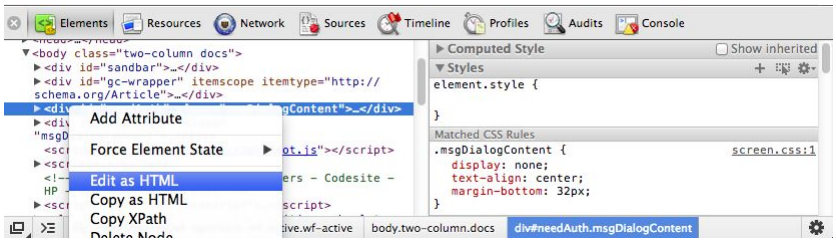
It's important to understand that this is a gradual process. For better user experience, the rendering engine will try to display contents on the screen as soon as possible. It will not wait until all HTML is parsed before starting to build and layout the render tree. Parts of the content will be parsed and displayed, while the process continues with the rest of the contents that keeps coming from the network.

Some authors put the <script> tag last in page, to be executed when DOM fully

constructed

- Also attribute async defer

Chrome Developer Tools



We need a client side inspection and debugging environment.

- Chrome Developer Tools, included in Chrome will be used by me …
- … other similar in other browsers.
- For now use
  - Check that resources exists (are downloaded)
  - Inspect DOM