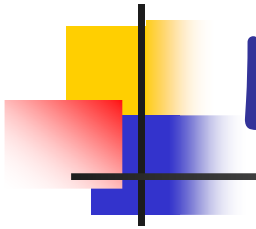
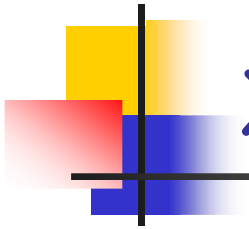


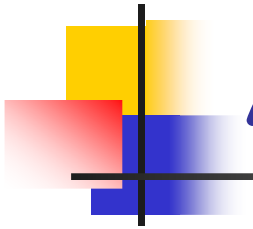
Lecture-6

- Miscellaneous topics. Introduction to
  - XML
  - AJAX
  - HTML DOM



XML - EXtended Markup Language

- XML is a markup language, like HTML
- Designed to carry data
  - Not to display data
- XML tags are NOT predefined.
  - Unlike HTML
  - You must define your own tags
- Self-descriptive
- Represented in plain text.

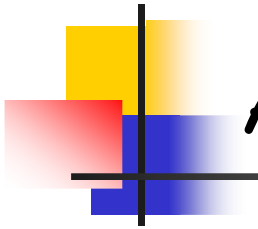


A simple example

```
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Let's meet tomorrow!</body>
</note>
```

Note:

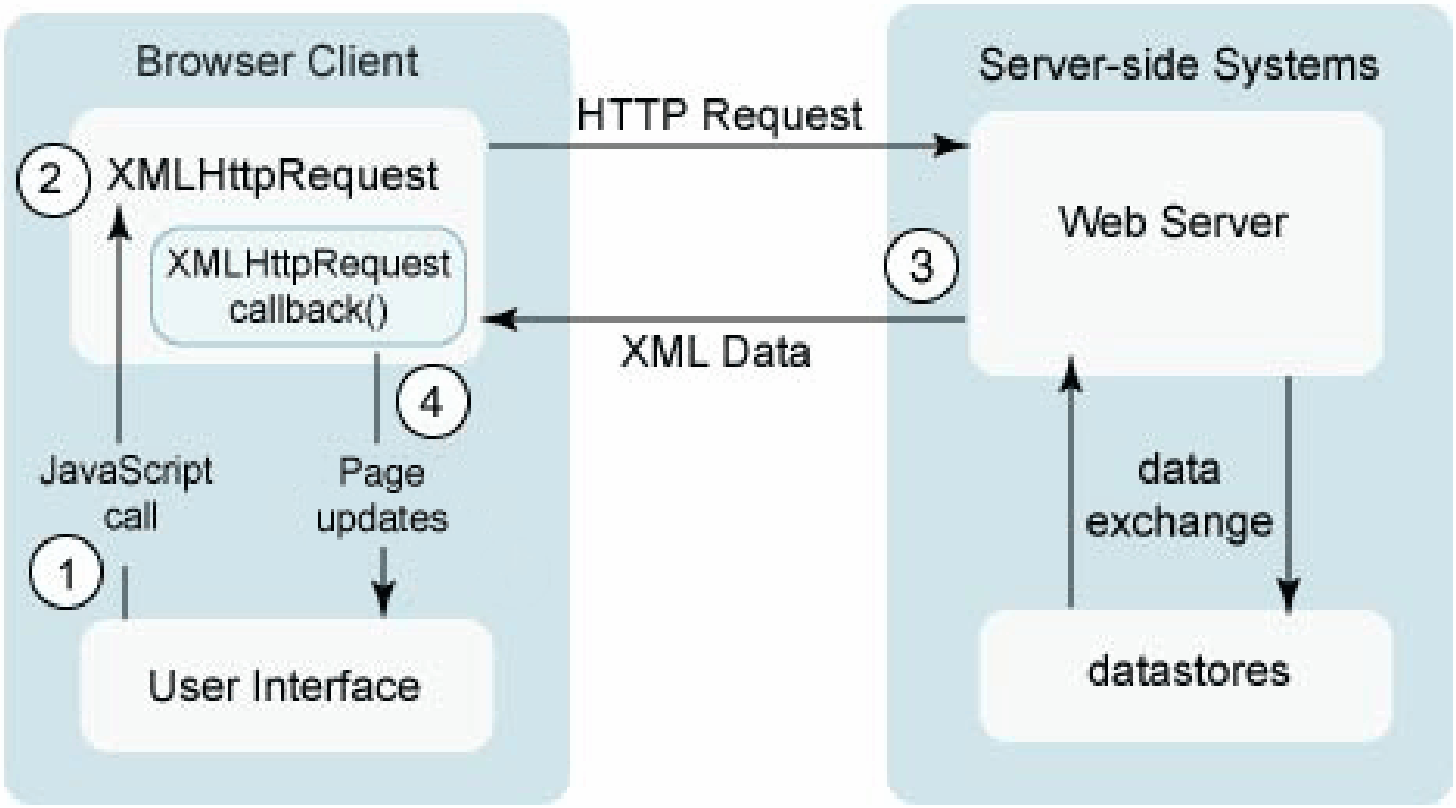
1. User defined tags.
2. Self descriptive



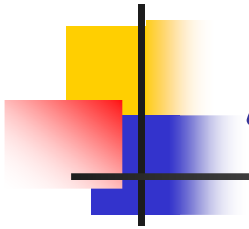
AJAX – Asynchronous Java And XML

- Made popular by Google (with Google Suggest).
- NOT a new programming language
  - A new way to use existing standards.
- Based on JavaScript and HTTP requests.
- With AJAX, JavaScript communicates
  - Directly with the (web) server
  - using XMLHttpRequest object
  - To retrieve data as needed
  - Using Javascript events (e.g., keyPressed)
  - WITHOUT refreshing the page.

How does AJAX work

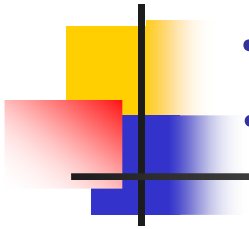


Source: SUN's JAVA web page



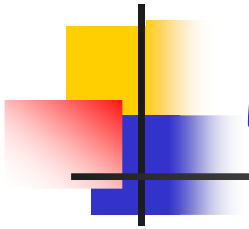
AJAX … contd.

- Note: Data is typically stored in XML format
- XMLHttpRequest
  - The basic data structure interfacing the client with server.
  - Sends a request to a server (e.g., Google suggest server) on any events
    - Like "onKeyup(..)" when the user types any character search key.
  - Receives data from the server
  - Updates the required fields with data received from server.



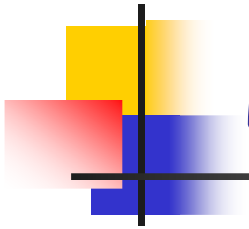
Introduction to HTML DOM

- DOM – Domain Object Model
- HTML DOM
  - Defines objects and properties of HTML elements.
  - A standard object model for HTML
  - A standard programming interface for HTML
  - Platform and language-independent



HTML DOM

- Everything in HTML is a node.
- Entire document – a document node.
- Every HTML tag – an element node.
- Text in the HTML elements – text nodes.
- Every HTML attribute is an attribute node.
- Comments are comment nodes.

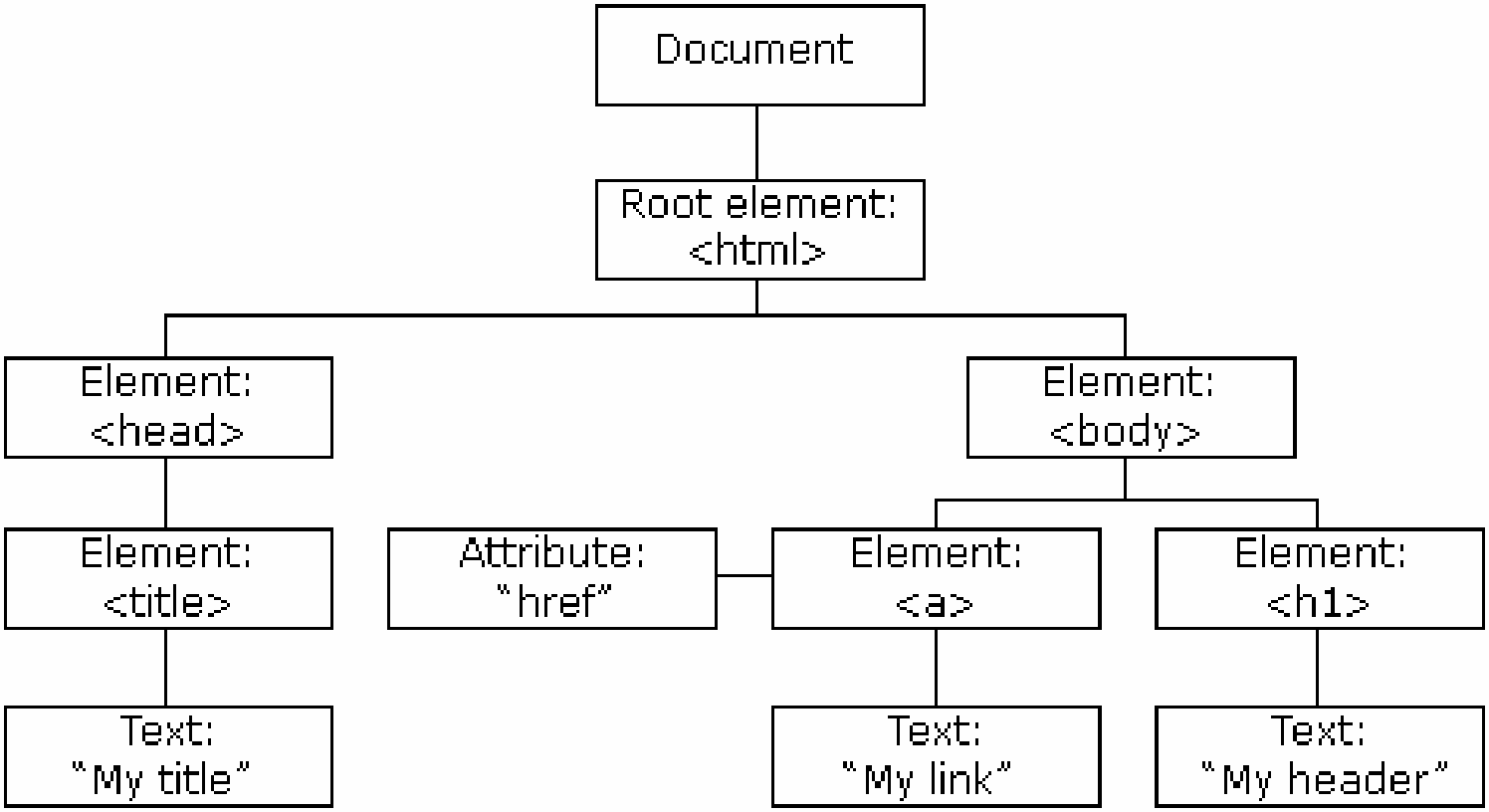


HTML DOM – Object model

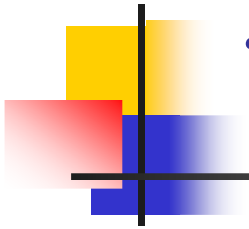
- Each node is an object.
- Objects have methods
- Can use methods to retrieve or change HTML content dynamically.

⇒Basis for Dynamic HTML (DHTML)

HTML DOM Node Tree



Source: W3 schools web page



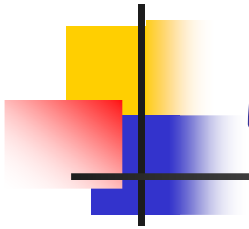
Tree structure … contd.

- Follow the standard "tree" nomenclature
- Top node is called the root
- Every node, except the root, has exactly one parent node.
  - Root has none.
- A node can have any number of children
- Leaf is a node with no children
- Siblings are nodes with the same parent

HTML DOM tree – example

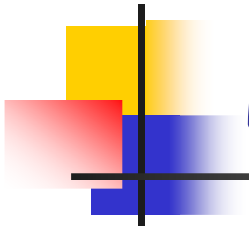
```
<html>
   <head>
     <title>DOM Tutorial
     </title>
   </head>
   <body>
     <h1>DOM Lesson one </h1>
     <p> Hello world! </p>
   </body>
</html>
```

- <html> node is the root node
  - Has no parent node
- Parent node of the <head> and <body> nodes is the <html> node.
- Parent node of the "Hello world!" text node is the <p> node
- <html> node has two child nodes
  - <head> and <body>
- <head> node has one child node
  - <title> node
- <title> node has one child node
  - text node "DOM Tutorial"
- <h1> and <p> nodes are siblings
  - Both child nodes of <body>



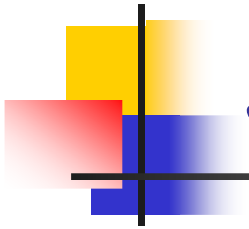
HTML DOM properties

- For any HTML element (node) x,
  - x.innerHTML - the inner text value of x
  - x.nodeName - the name of x
  - x.nodeValue - the value of x
  - x.parentNode - the parent node of x
  - x.childNodes - the child nodes of x
  - x.attributes - the attributes nodes of x



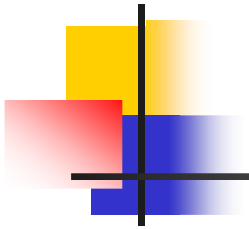
HTML methods

- For any HTML element (node) x
  - x.getElementById(*id*)
    - get the element with a specified id
  - x.getElementsByTagName (*name*)
    - get all elements with a specified tag name. Tag = "body", for example.
  - x.appendChild(*node*)
    - insert a child node to x
  - x.removeChild(*node*)



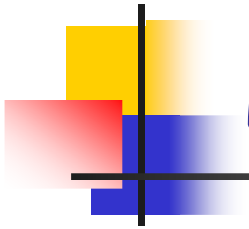
Special properties

- x.firstChild
  - Returns the first child node
- x.lastChild
  - Returns the last child node
- document.documentElement
  - Returns the root node
- document.body
  - Gives access to the body tag.



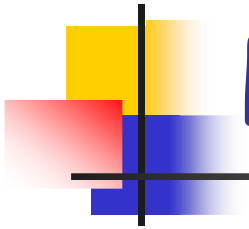
Back to the example …

- document - the current HTML document
- getElementById("intro") - the <p> element with the id "intro"
- childNodes[0] - the first child of the <p> element (the text node)
- nodeValue - the value of the node (the text itself)



HTML nodes

- DOM node properties
  - nodename – read only
  - nodevalue – can be changed
  - nodetype – read only
- Accessing nodes
  - getElementById (<id>)
  - getElementByTagName(<tag>)
  - A combination of the above
    - Using the DOM tree and parent/child relationship.

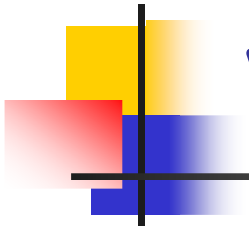


Putting it all together

- Nodes can be accessed.
- HTML content can be changed
  - Background color, image sources
  - Text, font, color of an element
- Add Javascript events to the mix

⇒ Can create a very dynamic web content, animation, etc.

- Basis for today's web pages.



What Javascript CANNOT do

- Javascript cannot
  - Read or write files on client
    - (Other than cookies).
  - Close a window it did not open.
  - Access information (cookies or web content) of other web pages.
  - Access databases, without the use of AJAX and a server side script
  - Cannot write files to servers without the help of server side script.