- Design Patterns
- Describe programming techniques to solve common problems.
- Design patterns are not standards to be followed, merely designs of solutions that have worked previously.
- Collective Wisdom
- Design patterns cut development time.
- Many problems we face everyday may already have been solved by someone else.
- That is where design patterns are useful.

- Design Patterns
- In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design.
- A design pattern is not a finished design that can be transformed directly into code.
- It is a description or template for how to solve a problem that can be used in many different situations.
- Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved.
- Algorithms are not thought of as design patterns, since they solve computational problems rather than design problems.
- Design patterns are for avoiding Reinventing Wheels

- AJAX Patterns
- Ajax has been around since 2005.
- Communication Patterns
- Hidden Frame techniques
- Aynchronous XMLHttp calls

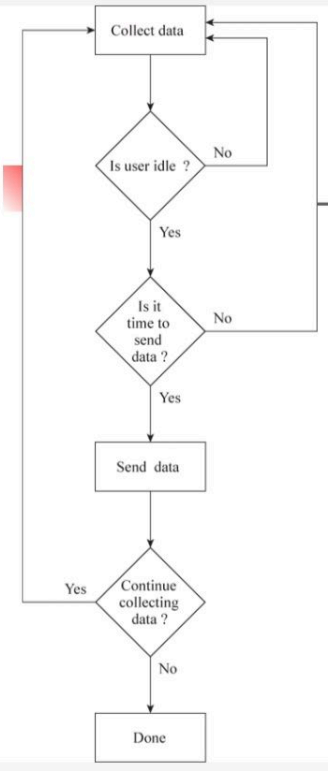
- Predictive Fetch
- Predicting what the user will do next
- Fetch on Demand
- Predictive Fetch pattern is relatively simple

Introduction

- Solves - when to send user data to the server.
- Traditional - each click – request to the server - aware of what the client is doing
- Consider user typing a letter..
- Create a large number of requests in a short time – also UI may slow down

Solution

- Buffer the data to be sent to the server on the client
- Send the data at predetermined times
- A client-side function is called to begin the buffering of data.
- User's status is checked to see if user is idle or not
- If the user is still active, data continues to be collected



Difference between Flex and HTML

- Flex is part of Adobe's Flash Platform.

1.The Flash Player: Flash Player is a browser plug-in which allows us to deploy web based applications to Windows, Mac, Linux, Android, and Blackberry.
2.Adobe AIR: Adobe AIR is a runtime that allows us to deploy native applications to Windows, Mac, Android, iOS, and Blackberry.
3.Flash Professional: Flash Professional is a tool for developing timeline based animations.
4.Flash Builder: Flash Builder is an IDE to help programmer's write advanced code.
5.Adobe Flex: Flex is the Software Development Kit that helps programmers build, debug, and deploy Enterprise applications with the Flash Platform. Flex includes a UI Component library, a SWF compiler, a command line debugger, an application profiler.

Flash and flex are composed of the same genes. They are built under the same technology. Flex is like the brother of flash, the new generation type. Flex has more added features. In short, you can do anything with flex even without flash. Flex is a more detailed application. In flash, you can also do anything flex can do. You will just undergo a long process, but in the end you will get there.

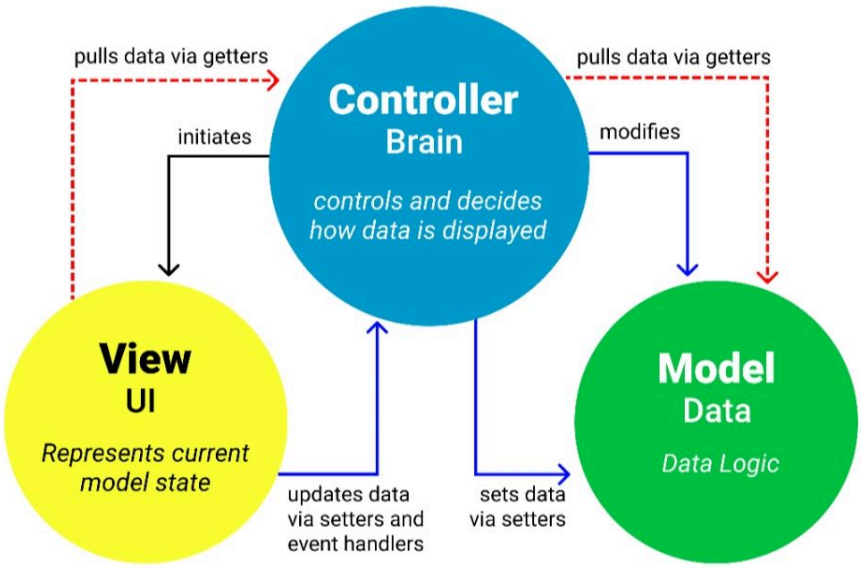
CENGAGE Learning

Differences contd..

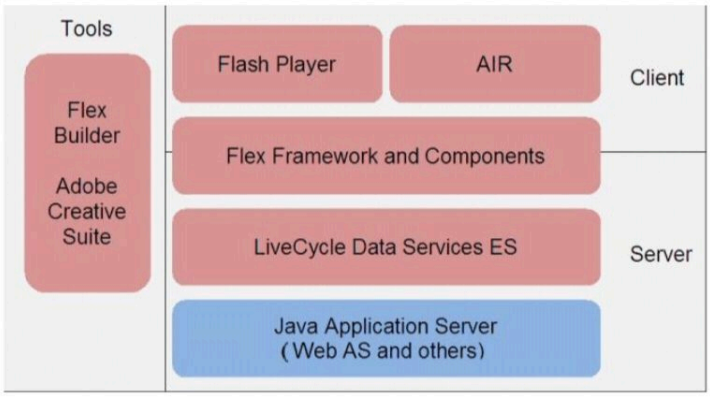
Flash provides you with great animation controls. You can be at ease making your animated graphics since flash has easier animation controls than flex. It also provides you quality sound control. And because flash was developed earlier for the Web world, it is more familiar to Web designers. Since flash is user friendly, most designers can now do a little programming with the help of flash. But most often they can only make it work a little since designing and programming are two very different skills.

CENGAGE Learning

MVC Architecture Pattern



Flex Architecture



An AIR file is **an archive file that contains the application files, which are all of the files contained in the project's bin folder**. In this simple example, those files are the SWF and application XML files. You distribute the AIR package to users who then use it to install the application.

CENGAGE Learning

Flex components/tools

- Component library provides user interface controls
- Simple buttons, checkboxes, and radio buttons
- Complex data grids and combo boxes
- Developers use provided components to design complex layouts and use (or modify) the skins for

Multi-Stage Download

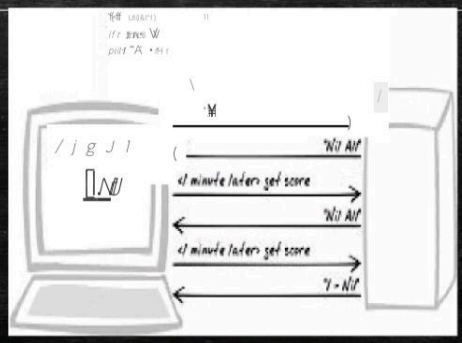
- Popularized by Microsoft's start.com
- When you first visit start.com, it is a very simple page with a search box in the middle
- A series of requests is being fired off to fill in more content on the page
- Within a few seconds, the page jumps to life as content from several different locations is pulled in and displayed

Periodic Refresh

- It is a design pattern
- Describes the process of checking for new server information in specific intervals
- This approach is also called *polling*
- Used to increase user experience
- Notify users of updated information

Polling Mechanism

Requires the browser to keep track of when another request to the server should take place



Real World Examples

- ESPN : Update online scoreboards automatically
- Gmail : Notify users of the new mail
- XHTML Live Chat : Implement a chat room
- Magnetic Ajax Demo : Re-creates online the experience of magnetic poetry
- White Chat & Lace Chat
- JotSpot Live Wiki
- Claude Hussnet's Portal

- EventDispatcher methods

- Here is a summary of the methods in EventDispatcher for ActionScript 3.0. Many of these methods are similar to the methods in the ActionScript 2.0 version:
- addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void
- removeEventListener(type:String, listener:Function, useCapture:Boolean = false): void
- dispatchEvent(event:Event):Boolean
- hasEventListener(type:String):Boolean
- willTrigger(type:String):Boolean

- **addEventListener():** Adds an event handler function to listen to an event so that when that event occurs, the function will be called.
- **removeEventListener():** Removes an event handler added to a listeners list using addEventListener. The same first 3 arguments used in addEventListener must be used in removeEventListener to remove the correct handler.
- **dispatchEvent():** Sends the passed event to all listeners in the listeners list of an object that relates to the event type. This method is most commonly used when creating custom events.
- **hasEventListener():** Determines whether or not an object has listeners for a specific type of event.
- **willTrigger():** Determines whether or not an object or any of its parent containers have listeners for a specific type event. This is much like hasEventListener but this method checks the current object as well as all objects that might be affected from the propagation of the event.

Button, Value selectors, Text components, List based controls

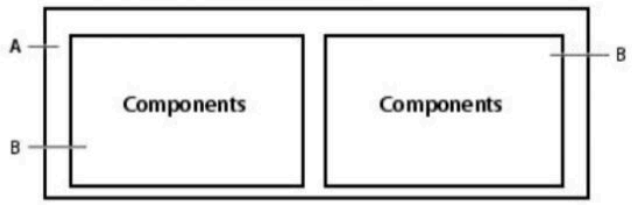
- The Button control is a commonly used rectangular button.
- Button controls look like they can be pressed, and have a text label, an icon, or both on their face. You can optionally specify graphic skins for each of several Button states.
- You can create a normal Button control or a toggle Button control. A normal Button control stays in its pressed state for as long as the mouse button is down after you select it. A toggle Button controls stays in the pressed state until you select it a second time.
- Buttons typically use event listeners to perform an action when the user selects the control. When a user clicks the mouse on a Button control, and the Button control is enabled, it dispatches a click event and a buttonDown event. A button always dispatches events such as the mouseMove, mouseOver, mouseOut, rollOver, rollOut, mouseDown, and mouseUp events whether enabled or disabled.
- You define a Button control in MXML by using the <mx:Button> tag, as the following example shows. Specify an id value if you intend to refer to a component elsewhere in your MXML, either in another tag or in an ActionScript block. The following code creates a Button control with the label "Hello world!":

About layout containers and navigator containers

- Flex defines two types of containers:
- Layout containers
- Control the sizing and positioning of the child controls and child containers defined within them. For example, a Grid layout container sizes and positions its children in a layout similar to an HTML table.
- Layout containers also include graphical elements that give them a particular style or reflect their function.
- The DividedBox container, for example, has a bar in the center that users can drag to change the relative sizes of the two box divisions.
- The TitleWindow control has an initial bar that can contain a title and status information.

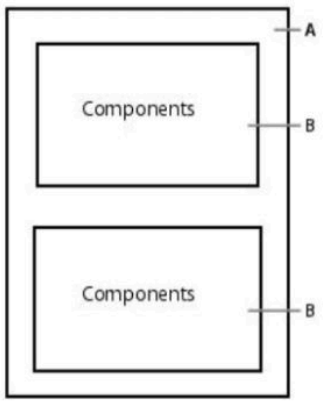
Navigator containers

- Control user movement, or navigation, among multiple child containers.
- The individual child containers, not the navigator container, control the layout andpositioning of their children.
- For example, an Accordion navigator container lets you construct a multipage formfrom multiple Form layout containers.
- Although you can create an entire Flex application by using a single container, typicalapplications use multiple containers.



A. Parent HBox layout container *B*. *Child VBox layout container*

- A VBox container arranges its children in a single vertical stack, or column, and oversees the layout of its own children. The following image shows the preceding example with the outermost container changed to a VBox layout container:



A. Parent VBox layout container *B*. *Child VBox layout container*

Prototype Ajax methods and properties

option	description
method	how to fetch the request from the server (default "post")
parameters	query parameters to pass to the server, if any
asynchronous (default true), contentType, encoding, requestHeaders	

options that can be passed to the Ajax.Request constructor

Prototype Ajax methods and properties

event	description
onSuccess	request completed successfully
onFailure	request was unsuccessful
onException	request has a syntax error, security error, etc.

events in the Ajax.Request object that you can handle

Basic Prototype Ajax template

property	description
status	the request's HTTP error code (200 = OK, etc.)
statusText	HTTP error code text
responseText	the entire text of the fetched page, as a String
responseXML	the entire contents of the fetched page, as an XML DOM tree (seen later)

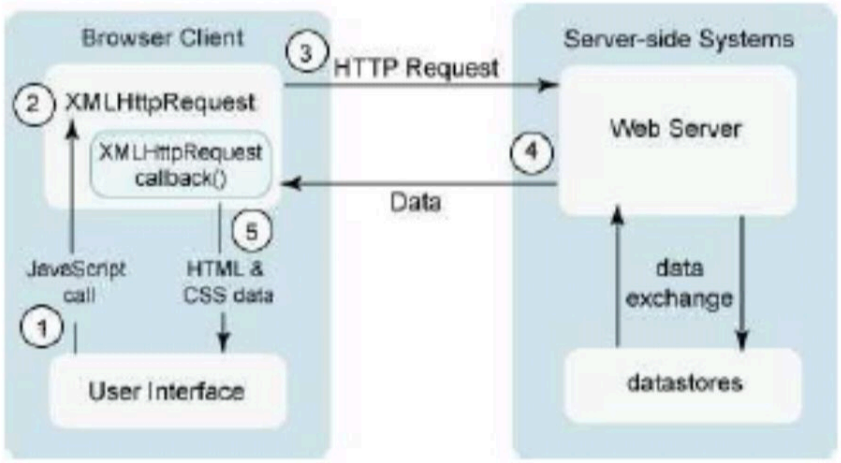
```
function handleRequest(ajax) {
        alert(ajax.responseText);
}
```

JS

A typical Ajax request

1. user clicks, invoking an event handler
2. handler's code creates an XMLHttpRequest object
3. XMLHttpRequest object requests page from server
4. server retrieves appropriate data, sends it back
5. XMLHttpRequest fires an event when data arrives
   - this is often called a callback
   - you can attach a handler function to this event
6. your callback event handler processes the data and displays it

A typical Ajax request

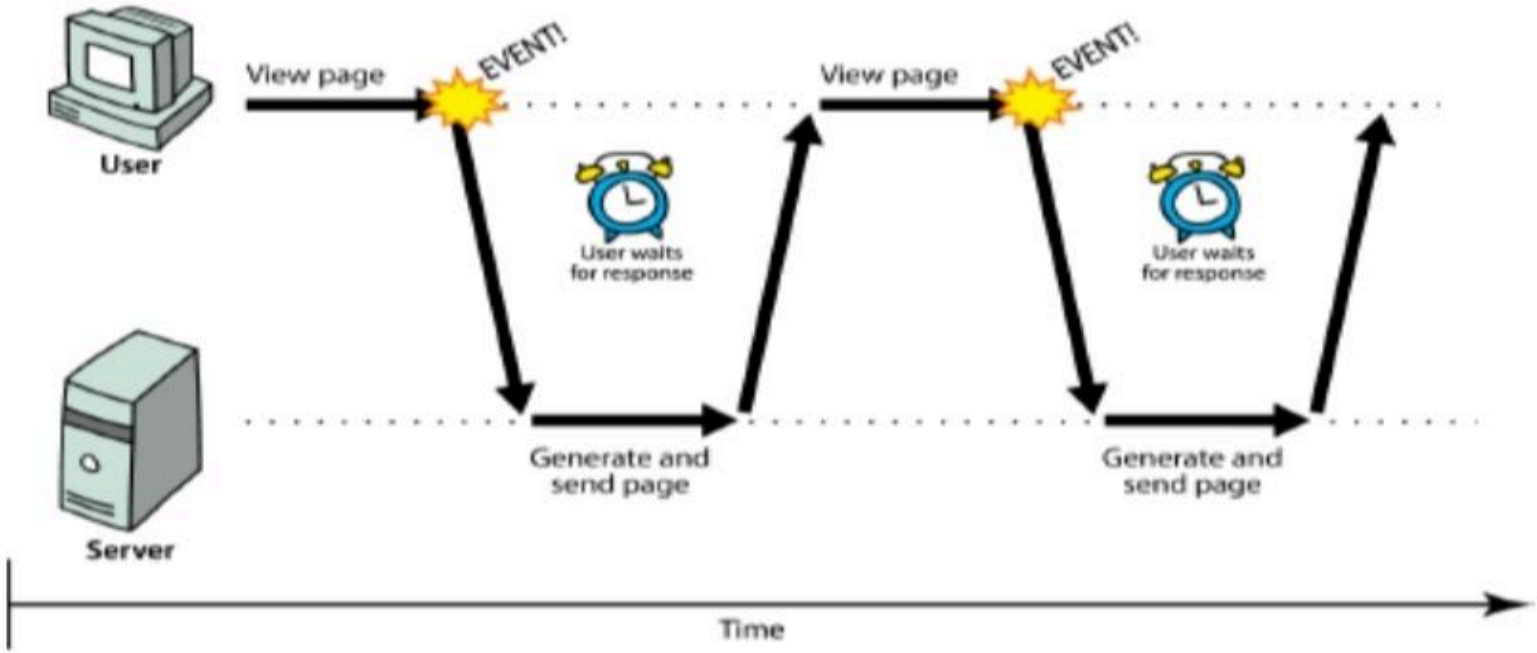


Prototype's Ajax model

```
new Ajax.Request("url",
{
        option : value,
        option : value,
        ...
        option : value
}
);                                                    JS
```

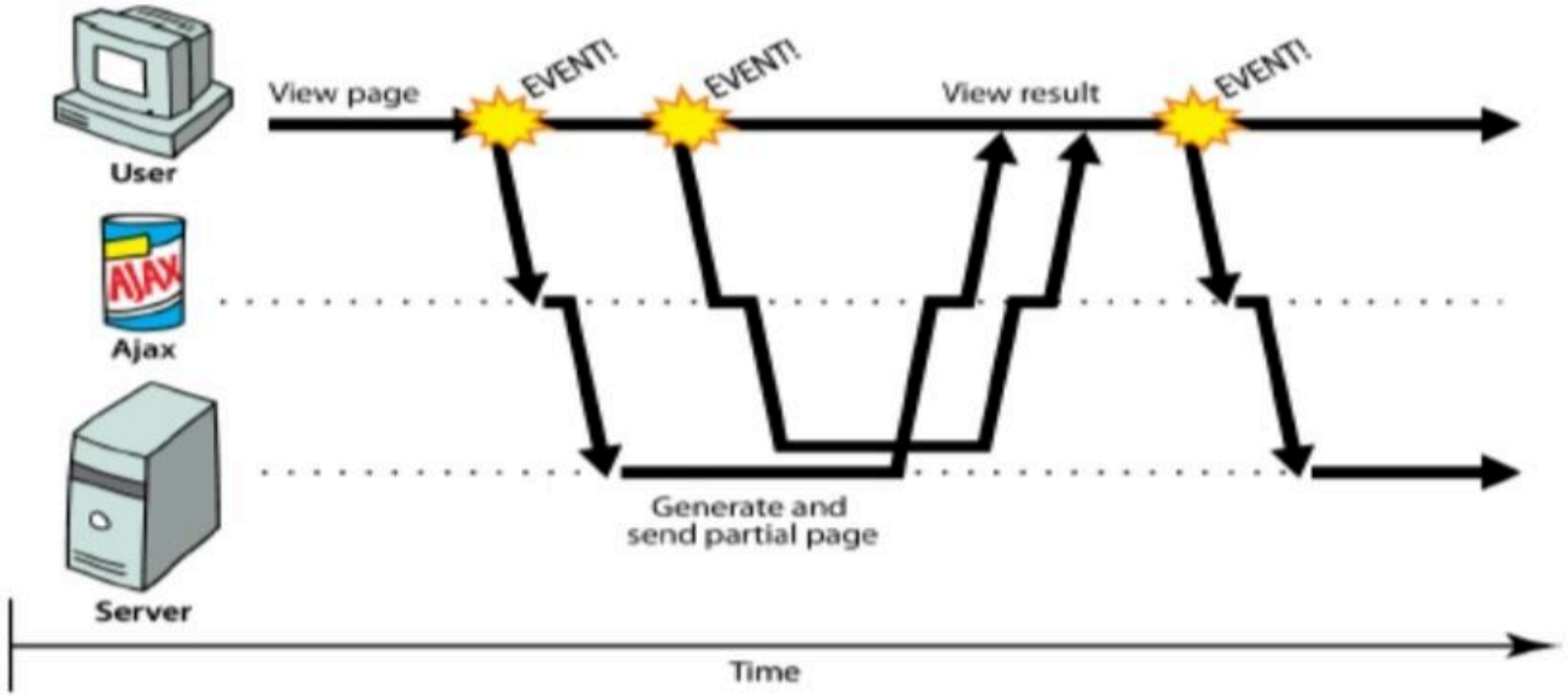
- construct a Prototype Ajax.Request object to request a page from a server using Ajax

Synchronous web communication



- synchronous: user must wait while new pages load
  - the typical communication pattern used in web pages (click, wait, refresh)

Asynchronous web communication



- **asynchronous**: user can keep interacting with page while data loads
  - communication pattern made possible by Ajax

93

Useful Web 2.0 Tools

- Weblogs
- Wikis
- Real Simple Syndication (RSS)
- Aggregators
- Social Bookmarking and Networking
- Online Photo Galleries
- Audio/video-casting

*Not the only web 2.0 tools! Keep educational uses in mind when looking at these.

JavaScript framework

- A JavaScript framework is a collection of JavaScript code libraries that provide a web developer with pre-written code for routine programming tasks.
- Frameworks are structures with a particular context and help you create web applications within that context.
- Frameworks provide a template that handles common programming patterns.
- Each time we build an application, need not to write code for every single feature from scratch. Instead, we can build upon an existing feature set.

FRAMEWORK VS. LIBRARY

- A JS framework is a full toolset that helps shape and organize your website or application.
- A JS library, on the other hand, is a collection of pre-written code snippets that are less about shaping your application and more about providing a use-as-needed library of features.

POPULAR JAVASCRIPT FRAMEWORKS

- FRONT-END FRAMEWORKS
  - REACT
  - ANGULAR
  - VUE
- BACK-END FRAMEWORKS
  - EXPRESS
  - NEXT.JS

Insert Data Into a Database Table

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
 {
 die('Could not connect: ' . mysql_error());
 }
 mysql_select_db("pmg" ,$con);

mysql_query("INSERT INTO student (Rollno, Name, Mark)
VALUES (1, 'Raju', 35 )", $con);
mysql_close($con);
?>
```


Insert Data From a Form Into a Database

```
<html>
<body>
<form action="insert.php" method="post">
Roll No: <input type="text" name="rollNo" />
Name : <input type="text" name="name" />
Mark : <input type="text" name="mark" />
<input type="submit" />
</form>
</body>
</html>
```


Insert.php

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
 {
 die('Could not connect: ' . mysql_error());
 }
mysql_select_db("pmg", $con);
$sql="INSERT INTO student (rollno, name, mark)
VALUES
('$_POST[rollno]','$_POST[name]','$_POST[mark]')";
if (!mysql_query($sql,$con))
 {
 die('Error: ' . mysql_error());
```

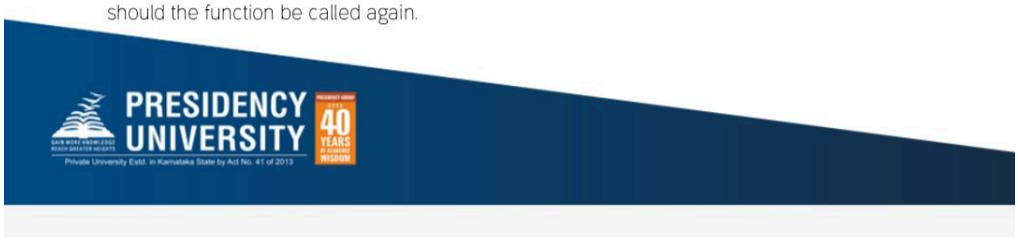
```
$temp = 'The date is ';
echo longdate(time());

function longdate($timestamp)
{
  return $temp . date("l F jS Y", $timestamp);
}
?>
```

Global variables

- To declare a variable as having global scope, use the keyword global.
  Example: global $is_logged_in;
- You can able to access this global variable from any function, without the scope.
- It is visible (hence accessible) throughout the program.

Static variables

A static variable will not lose its value when the function exits and will still hold that value should the function be called again.




```
<?php
   function keep_track() {
      STATIC $count = 0;
      $count++;
      print $count;
      print "<br />";
   }

   keep_track();
   keep_track();
   keep_track();
?>
```

This will produce the following result −

```
1
2
3
```

They can be initialized only with predetermined values

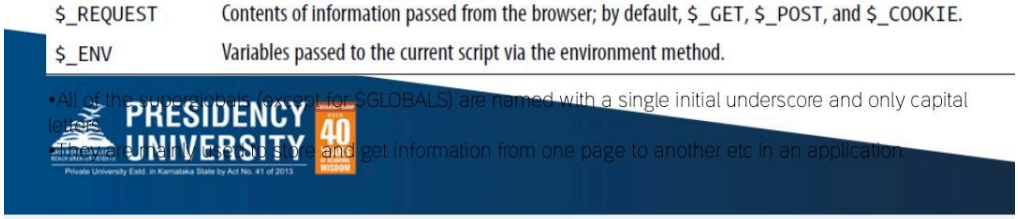
Example 3-18. Allowed and disallowed static variable declarations

```
<?php
  static $int = 0;          // Allowed
  static $int = 1+2;        // Disallowed (will produce a Parse error)
  static $int = sqrt(144); // Disallowed
?>
```

Superglobal variables

Starting with PHP 4.1.0, several predefined variables are available. These are known as *superglobal variables, which means that they are provided by the PHP environment* but are global within the program, accessible absolutely everywhere.

Superglobal name	Contents
$GLOBALS	All variables that are currently defined in the global scope of the script. The variable names are the keys of the array.
$_SERVER	Information such as headers, paths, and script locations. The entries in this array are created by the web server, and there is no guarantee that every web server will provide any or all of these.
$_GET	Variables passed to the current script via the HTTP Get method.
$_POST	Variables passed to the current script via the HTTP Post method.
$_FILES	Items uploaded to the current script via the HTTP Post method.
$_COOKIE	Variables passed to the current script via HTTP cookies.
$_SESSION	Session variables available to the current script.
$_REQUEST	Contents of information passed from the browser; by default, $_GET, $_POST, and $_COOKIE.
$_ENV	Variables passed to the current script via the environment method.



- All of the superglobals (except for $GLOBALS) are named with a single initial underscore and only capital letters.
- Superglobals are used to store and get information from one page to another etc in an application.

PHP

What is PHP?

- PHP stands for **P**HP: **H**ypertext **P**reprocessor
- PHP is a server-side scripting language, like ASP
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is an open source software
- PHP is free to download and use

What is a PHP File?

- PHP files can contain text, HTML tags and scripts
- PHP files are returned to the browser as plain HTML
- PHP files have a file extension of ".php", ".php3", or ".phtml"

Why PHP?

- PHP runs on different platforms (Windows, Linux, Unix, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP is FREE to download from the official PHP resource: www.php.net
- PHP is easy to learn and runs efficiently on the server side

Where to Start?

To get access to a web server with PHP support, you can:

- Install Apache (or IIS) on your own server, install PHP, and MySQL
- Or find a web hosting plan with PHP and MySQL support

PHP Syntax

PHP (recursive acronym for *PHP: Hypertext Preprocessor*) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

A PHP script is executed on the server, and the plain HTML result is sent back to the browser.

Basic PHP Syntax

A PHP script can be placed anywhere in the document.A PHP script starts with **<?php** and ends with **?>**:

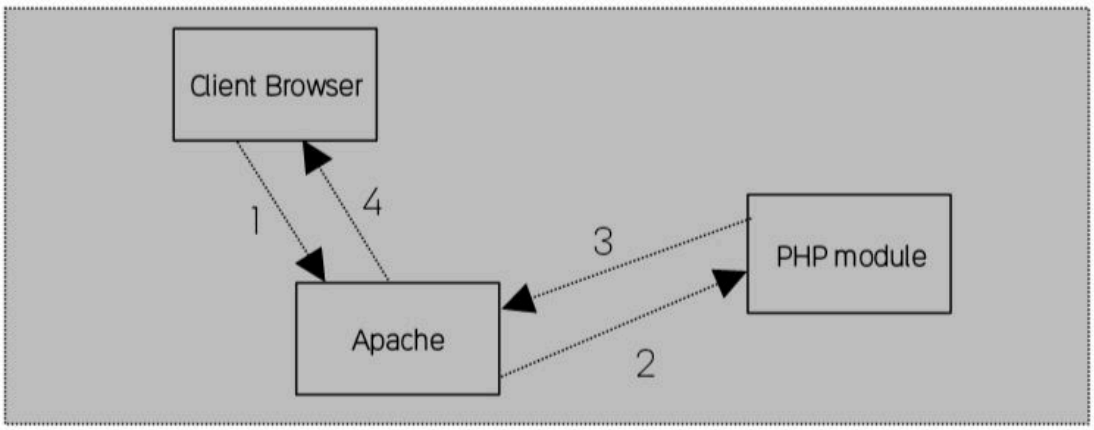
```
<?php
// PHP code goes here
?>
```

The default file extension for PHP files is ".php".
A PHP file normally contains HTML tags, and some PHP scripting code.
Example:

```
<?php
echo "Hello world";
```

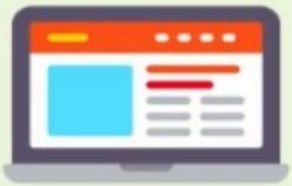
Working Principle of PHP

How PHP generates HTML/JS Web pages



1: Client from browser send HTTP request (with POST/GET variables)
2: Apache recognizes that a PHP script is requested and sends the request to PHP module
3: PHP interpreter executes PHP script, collects script output and sends it back
4: Apache replies to client using the PHP script output as HTML output

- PHP is a server side scripting language embedded in XHTML. It is an alternative to CGI, ASP, JSP
- The PHP processor(module) works in two modes.
  - If the PHP processor finds XHTML tags in the PHP script then the code is simply copied to the output file.
  - When the processor finds the PHP code in the script then that code is interpreted and the output is copied to the output file
- If you click for view the source on the web browser, you can never see the PHP script because the output of the PHP script is send directly to the browser but you can see the XHTML tags.



VS

CLIENT SIDE

- Frontend
- Collects user input
- Client side scripts mostly deal with visual and user input aspects
- Scripts may be restricted to run in a sandbox

SERVER SIDE

- Backend
- Processes user input
- Server side scripts mostly deal with transactions and complex computations
- Processes are transparent to the users

CLIENT SIDE SCRIPTING	SERVER SIDE SCRIPTING
A technique used in web development that involves using scripts that runs on the client machine's browser	A technique used in web development that involves using scripts on the web server to produce a response that is customized for each client's request to the website
Executed in the client side or the web browser	Executed in the back end or the web server
PHP, Python, Java, Ruby, ASP.NET are used	HTML, CSS, and JavaScript are used
Provides more security for the data	Provides less security for the data

Scripting Language

- A new style of programming language different from system programming languages
- Designed as glue language or system integration language
- A single statement can execute huge number of machine instructions
- Scripting languages are interpreted(rather than compiled)
- Are normally 'typeless'
  - Build complex algorithms and data structures..
- Can create dynamic web pages
  - Change based on user input

Advantages of Scripting Languages

- Easy to learn
- Requires minimum programming languages or experience to develop the web pages
- Allows simple creation and editing in variety of text editors
- Develop dynamic and interactive web pages.
- Validating the content

Types of Scripting Languages

- Server-side Scripting Language
  - Can use huge resources of the server
  - Complete all processing in the server and send plain pages to the client
  - Reduces client-side computation overhead
  - Run on web servers
  - Example: - ASP, JSP, Servlets, PHP

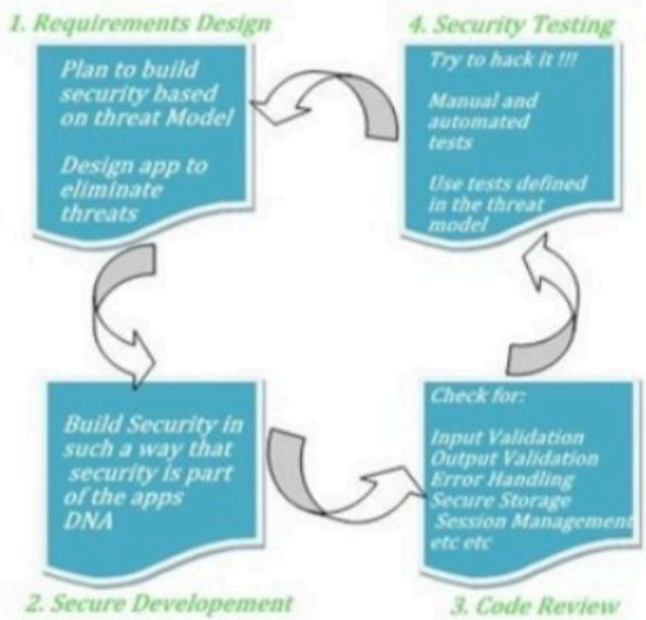
Web 1.0 vs Web 2.0

Web 1.0	Web 2.0
The mostly read only Web	The widely read -write web
45million global user(1996).	1 billion + global user(2006)
Focused on companies	Focused on communities
Home pages	Blogs
Owning content	Sharing content
HTML,portals	XML,RSS
Web forms.	Web Application
Netscape	Google
Page views	Cost per click

Web 2.0

Sad Facts

- Same old Keyword based search.
- Web application are still rigid
- Each Website have its own data and it is not sharing it.
- Computers can not understand any thing
- Web 2.0 is Social change. The technical part has not change much.

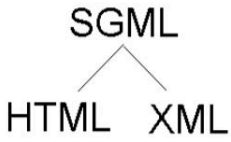


What is XML?

- *e*X*tensible* M*arkup* L*anguage*
- Markup language for documents containing structured information
- Defined by four specifications:
  - XML, the Extensible Markup Language
  - XLL, the Extensible Linking Language
  - XSL, the Extensible Style Language
  - XUA, the XML User Agent

XML....

- Based on Standard Generalized Markup Language (SGML)
- Version 1.0 introduced by World Wide Web Consortium (W3C) in 1998
- Bridge for data exchange on the Web



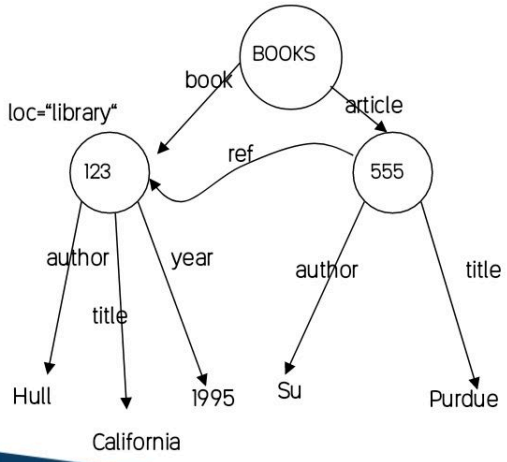
PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
40 YEARS

Comparisons

- XML
  - Extensible set of tags
  - Content orientated
  - Standard Data infrastructure
  - Allows multiple output forms
- HTML
  - Fixed set of tags
  - Presentation oriented
  - No data validation capabilities
  - Single presentation

XML Data Model: Example

```
<BOOKS>
<book id="123" loc="library">
  <author>Hull</author>
  <title>California</title>
  <year> 1995 </year>
</book>
<article id="555" ref="123">
  <author>Su</author>
  <title> Purdue</title>
</article>
</BOOKS>
```



Authoring XML Documents (cont'd)

- Authoring guidelines:
  - All elements must have an end tag.
  - All elements must be cleanly nested (overlapping elements are not allowed).
  - All attribute values must be enclosed in quotation marks.
  - Each document must have a unique first element, the root node.

Authoring XML Data Islands

- A data island is an XML document that exists within an HTML page.
- The <XML> element marks the beginning of the data island, and its ID attribute provides a name that you can use to reference the data island.

Document Type Definitions (DTD)

- An XML document may have an optional DTD.
- DTD serves as grammar for the underlying XML document, and it is part of XML language.
- DTDs are somewhat unsatisfactory, but no consensus exists so far beyond the basic DTDs.
- DTD has the form:

  <!DOCTYPE name [markupdeclaration]>

DTD (cont'd)

- Consider an XML document:

```
<db><person><name>Alan</name>
                <age>42</age>
                <email>agb@usa.net </email>
        </person>
        <person>.....</person>
        .....
</db>
```


DTD (cont'd)

- DTD for it might be:

```
<!DOCTYPE db [
    <!ELEMENT db (person*)>
    <!ELEMENT person (name, age, email)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT age (#PCDATA)>
    <!ELEMENT email (#PCDATA)>
]>
```

What is a *mediator*?

- A complex software component that integrates and transforms data from one or several sources using a declarative specification
- Two main contexts:
  - Data conversion: converts data between two different models
    - e.g. by translating data from a relational database into XML
  - Data integration: integrates data from different sources into a common view

Converting Relational Database to XML

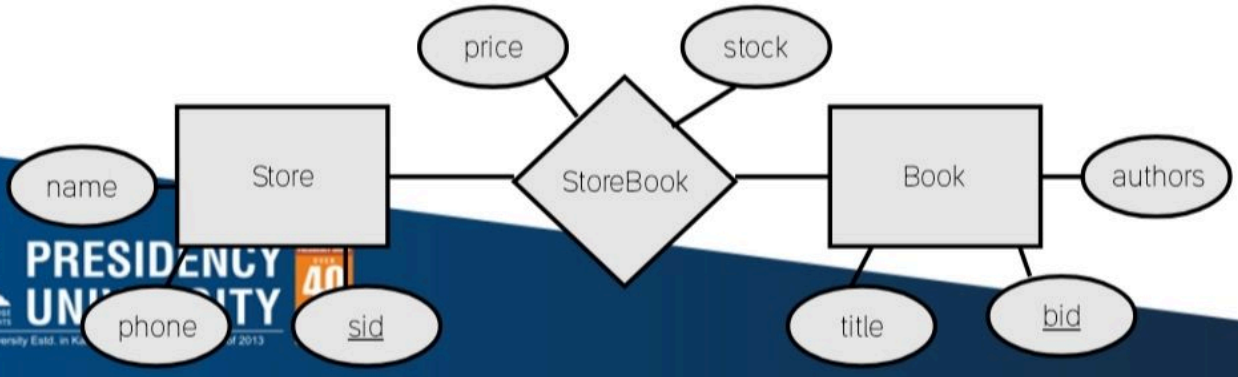
Example: Export the following data into XML and group books by store

- Relational Database:

  Store (sid, name, phone)

  Book (bid, title, authors)

  StoreBook (sid , bid, price, stock)



XML

- Structure of XML Data
- XML Document Schema
- Querying and Transformation
- Application Program Interfaces to XML
- Storage of XML Data
- XML Applications

Introduction

- XML: Extensible Markup Language
- Defined by the WWW Consortium (W3C)
- Derived from SGML (Standard Generalized Markup Language), but simpler to use than SGML
- Documents have tags giving extra information about sections of the document
  - E.g. <title> XML </title> <slide> Introduction ...</slide>
- **Extensible**, unlike HTML
  - Users can add new tags, and *separately* specify how the tag should be handled for display

PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
40 YEARS

XML Introduction (Cont.)

- The ability to specify new tags, and to create nested tag structures make XML a great way to exchange **data**, not just documents.
  - Much of the use of XML has been in data exchange applications, not as a replacement for HTML
- Tags make data (relatively) self-documenting
  - E.g.

```
<university>
    <department>
      <dept_name> Comp. Sci. </dept_name>
      <building> Taylor </building>
      <budget> 100000 </budget>
    </department>
    <course>
       <course_id> CS-101 </course_id>
       <title> Intro. to Computer Science </title>
       <dept_name> Comp. Sci </dept_name>
       <credits> 4 </credits>
    </course>
</university>
```

XML Document Schema

- Database schemas constrain what information can be stored, and the data types of stored values
- XML documents are not required to have an associated schema
- However, schemas are very important for XML data exchange
  - Otherwise, a site cannot automatically interpret data received from another site
- Two mechanisms for specifying XML schema
  - Document Type Definition (DTD)
    - Widely used
  - XML Schema
    - Newer, increasing use

Document Type Definition (DTD)

- The type of an XML document can be specified using a DTD
- DTD constraints structure of XML data
  - What elements can occur
  - What attributes can/must an element have
  - What subelements can/must occur inside each element, and how many times.
- DTD does not constrain data types
  - All values represented as strings in XML
- DTD syntax
  - <!ELEMENT element (subelements-specification) >
  - <!ATTLIST element (attributes) >

Element Specification in DTD

- Subelements can be specified as
  - names of elements, or
  - #PCDATA (parsed character data), i.e., character strings
  - EMPTY (no subelements) or ANY (anything can be a subelement)
- Example

```
<! ELEMENT department (dept_name  building, budget)>
<! ELEMENT dept_name (#PCDATA)>
<! ELEMENT budget (#PCDATA)>
```

- Subelement specification may have regular expressions

```
<!ELEMENT university ( ( department | course | instructor | teaches )•)>
```

  - Notation:
    - "|" - alternatives
    - "+" - 1 or more occurrences
    - "*" - 0 or more occurrences

University DTD

```
<!DOCTYPE  university [
  <!ELEMENT university ( (department|course|instructor|teaches)+)>
  <!ELEMENT department ( dept name, building, budget)>
  <!ELEMENT course ( course id, title, dept name, credits)>
  <!ELEMENT instructor (IID, name, dept name, salary)>
  <!ELEMENT teaches (IID, course id)>
  <!ELEMENT dept name( #PCDATA )>
  <!ELEMENT building( #PCDATA )>
  <!ELEMENT budget( #PCDATA )>
  <!ELEMENT course id ( #PCDATA )>
  <!ELEMENT title ( #PCDATA )>
  <!ELEMENT credits( #PCDATA )>
  <!ELEMENT IID( #PCDATA )>
  <!ELEMENT name( #PCDATA )>
  <!ELEMENT salary( #PCDATA )>
]>
```

IDs and IDREFs

- An element can have at most one attribute of type ID
- The ID attribute value of each element in an XML document must be distinct
  - Thus the ID attribute value is an object identifier
- An attribute of type IDREF must contain the ID value of an element in the same document
- An attribute of type IDREFS contains a set of (0 or more) ID values. Each ID value must contain the ID value of an element in the same document

University DTD with Attributes

- University DTD with ID and IDREF attribute types.

```
<!DOCTYPE university-3 [
    <!ELEMENT university ( (department|course|instructor)+)>
    <!ELEMENT department ( building, budget )>
    <!ATTLIST department
        dept_name ID #REQUIRED >
    <!ELEMENT course (title, credits )>
    <!ATTLIST course
        course_id ID #REQUIRED
        dept_name IDREF #REQUIRED
        instructors IDREFS #IMPLIED >
    <!ELEMENT instructor ( name, salary )>
    <!ATTLIST instructor
        IID ID #REQUIRED
        dept_name IDREF #REQUIRED >
    . . . declarations for title, credits, building,
        budget, name and salary . . .
]>
```

XSLT

- A **stylesheet** stores formatting options for a document, usually separately from document
  - E.g. an HTML style sheet may specify font colors and sizes for headings, etc.
- The **XML Stylesheet Language (XSL)** was originally designed for generating HTML from XML
- XSLT is a general-purpose transformation language
  - Can translate XML to XML, and XML to HTML
- XSLT transformations are expressed using rules called **templates**
  - Templates combine selection using XPath with construction of results

Limitations of DTDs

- No typing of text elements and attributes
  - All values are strings, no integers, reals, etc.
- Difficult to specify unordered sets of subelements
  - Order is usually irrelevant in databases (unlike in the document-layout environment from which XML evolved)
  - (A | B)* allows specification of an unordered set, but
    - Cannot ensure that each of A and B occurs only once
- IDs and IDREFs are untyped
  - The *instructors* attribute of an course may contain a reference to another course, which is meaningless
    - *instructors* attribute should ideally be constrained to refer to instructor elements

XML Schema

- XML Schema is a more sophisticated schema language which addresses the drawbacks of DTDs. Supports
  - Typing of values
    - E.g. integer, string, etc
    - Also, constraints on min/max values
  - User-defined, comlex types
  - Many more features, including
    - uniqueness and foreign key constraints, inheritance
- XML Schema is itself specified in XML syntax, unlike DTDs
  - More-standard representation, but verbose
- XML Scheme is integrated with namespaces
- BUT: XML Schema is significantly more complicated than DTDs.

What is jQuery?

- jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. (jQuery.com)

Why learn jQuery?

- Write less, do more:
  - *$("p.neat").addClass("ohmy").show("slow");*
- Performance
- Plugins
- It's standard
- ... and fun!

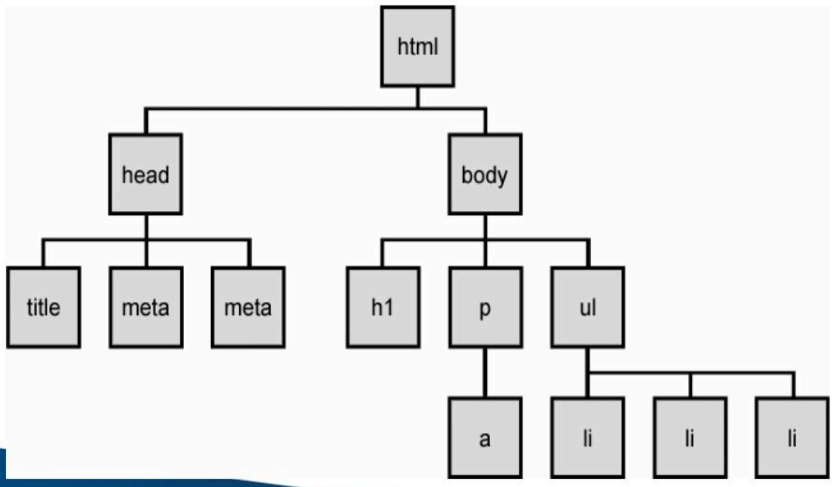
That little snippet loops through all <p> elements with the class "neat" and then adds the class "ohmy" to it, whilst slowly showing the paragraph in an animated effect. No browser checks, no loop code, no complex animation functions, just one line of code!

Example: Show/Hide Button

Aspects of the DOM and jQuery

- **Identification:** how do I obtain a reference to the node that I want.
- **Traversal:** how do I move around the DOM tree.
- **Node Manipulation:** how do I get or set aspects of a DOM node.
- **Tree Manipulation:** how do I change the structure of the page.

The DOM tree



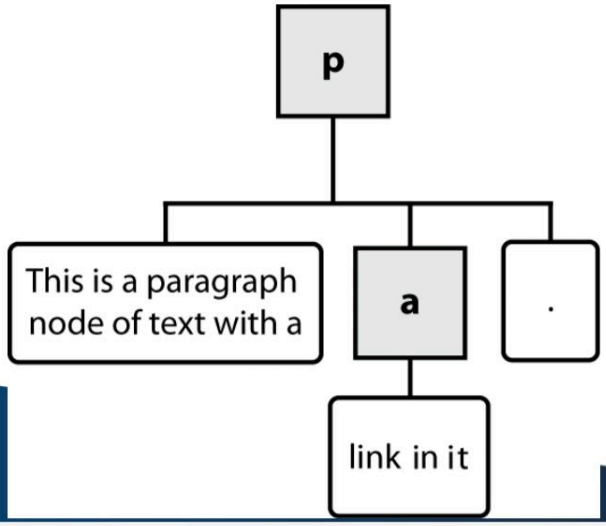
PRESIDENCY UNIVERSITY
Private University Estd. in Karnataka State by Act No. 41 of 2013
40 YEARS

Selecting groups of DOM objects

name	description
getElementById	returns array of descendents with the given tag, such as "div"
getElementsByTagName	returns array of descendents with the given tag, such as "div"
getElementsByName	returns array of descendents with the given name attribute (mostly useful for accessing form controls)
querySelector *	returns the first element that would be matched by the given CSS selector string
querySelectorAll *	returns an array of all elements that would be matched by the given CSS selector string

Types of DOM nodes

```
<p>
        This is a paragraph of text with a
        <a href="/path/page.html">link in it</a>.
</p>
```



Traversing the DOM tree

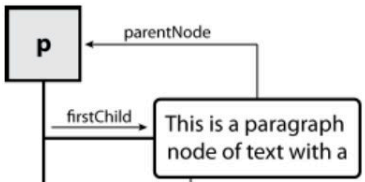
name(s)	description
firstChild, lastChild	start/end of this node's list of children
childNodes	array of all this node's children
nextSibling, previousSibling	neighboring nodes with the same parent
parentNode	the element that contains this node

- [w3schools' DOM node properties](#)
- [quirksmode's compatibility information](#) (IE6 sucks)

DOM tree traversal example

```
<p id="foo">This is a paragraph of text with a
<a href="/path/to/another/page.html">link</a>.</p>
```

HTML



What is AngularJS

MVC Javascript Framework by Google for Rich Web Application Development

Why AngularJS

"Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views".

- Structure, Quality and Organization
- Lightweight (< 36KB compressed and minified)
- Free
- Separation of concern
- Modularity
- Extensibility & Maintainability
- Reusable Components

" HTML? Build UI Declaratively! CSS? Animations! JavaScript? Use it the plain old way!"

jQuery

- Allows for DOM Manipulation
- Does not provide structure to your code
- Does not allow for two way binding

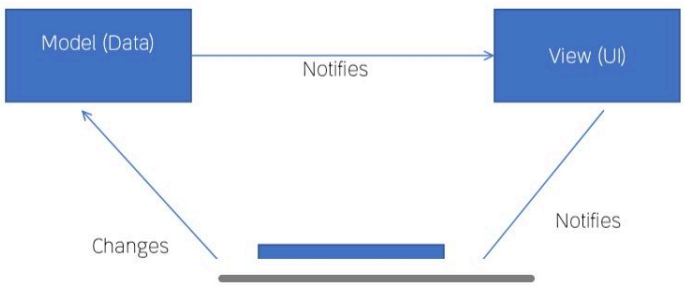
Features of AngularJS

- Two-way Data Binding – Model as single source of truth
- Directives – Extend HTML
- MVC
- Dependency Injection
- Testing
- Deep Linking (Map URL to route Definition)
- Server-Side Communication

Data Binding

```
<html ng-app>
<head>
  <script src='angular.js'></script>
</head>
<body>
  <input ng-model='user.name'>
  <div ng-show='user.name'>Hi {{user.name}}</div>
</body>
</html>
```


MVC



Expressions

Expressions allow you to execute some computation in order to return a desired value.

- {{ 1 + 1 }}
- {{ 946757880 | date }}
- {{ user.name }}

you shouldn't use expressions to implement any higher-level logic.

Directives

Directives are markers (such as attributes, tags, and class names) that tell AngularJS to attach a given behaviour to a DOM element (or transform it, replace it, etc.)

Some angular directives

- The ng-app - Bootstrapping your app and defining its scope.
- The ng-controller - defines which controller will be in charge of your view.
- The ng-repeat - Allows for looping through collections