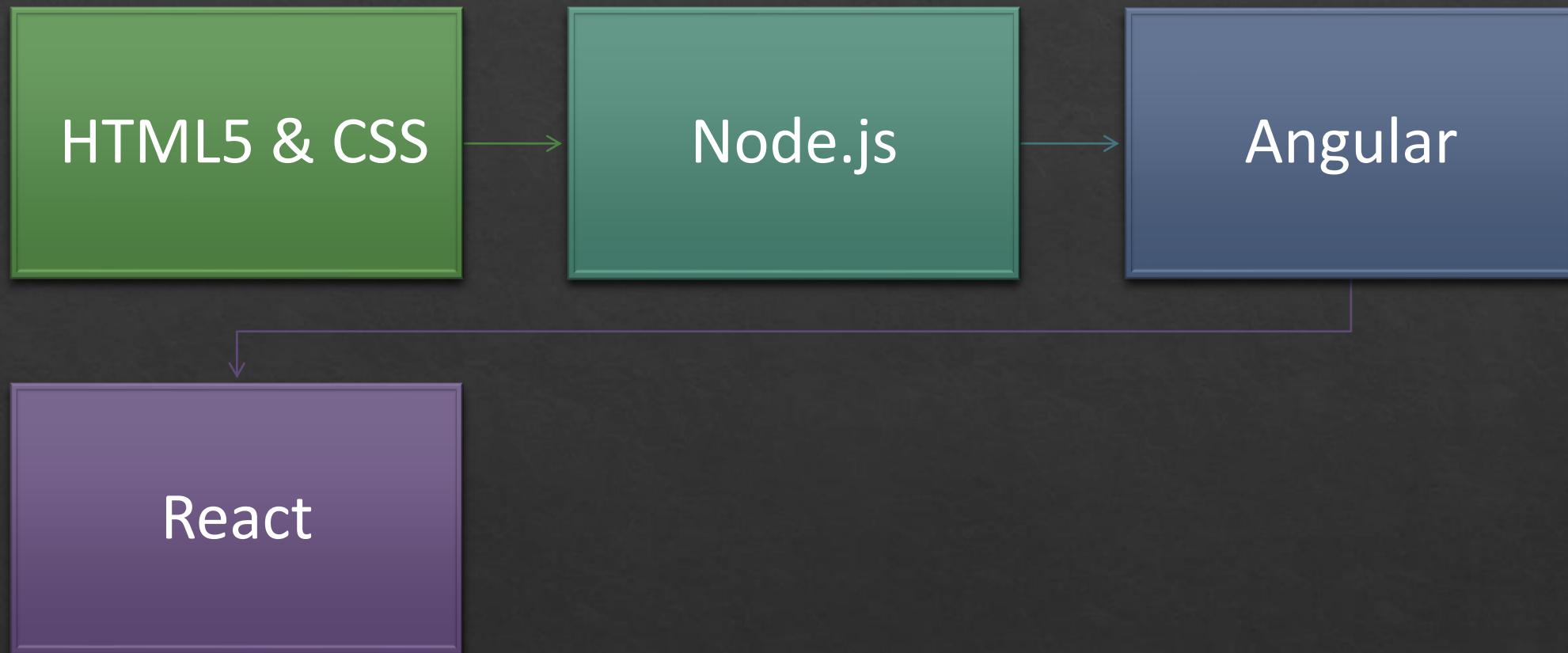


Web Technologies

ANIL JOSEPH

Agenda



Anil Joseph

Introduction

- ❖ Over 20 years of experience in the industry
- ❖ Technologies
 - ❖ C, C++
 - ❖ Java, Enterprise Java
 - ❖ .NET & .NET Core
 - ❖ **UI Technologies: React, Angular, jQuery, ExtJs**
 - ❖ Mobile: Native Android, React Native
- ❖ Worked on numerous projects
- ❖ Conducting trainings for corporates (700+)

Software

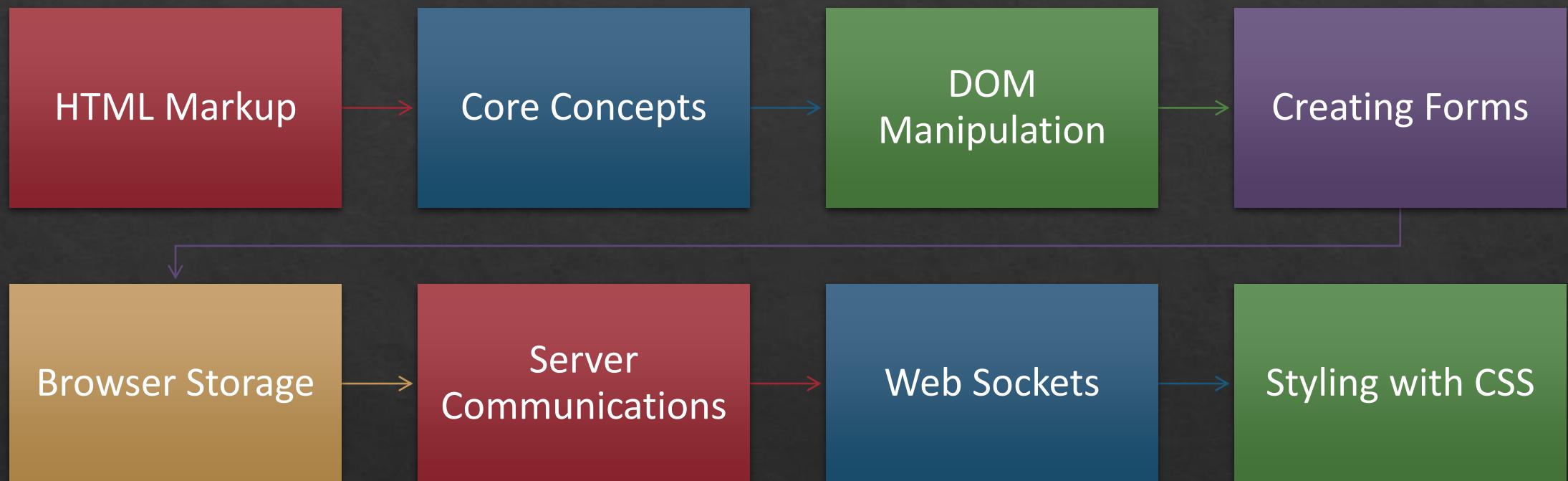
Visual Studio Code

Node.js

Browsers

HTML5

HTML Agenda



What is HTML 5

- ❖ HTML 5 comprises of
 - ❖ Markup
 - ❖ JavaScript API's
- ❖ Most HTML5 features are supported by the modern browsers.

HTML5 Features



More descriptive
markup(Semantic)

New tags introduced that are
more descriptive



Better Interface and
Applications

Easier to create more richer and
interactive applications.



Multimedia with less
reliance on plug-ins

No requirement of using Flash
or Silverlight for video, audio
and vector graphics.



Backward Compatible



JavaScript libraries available for common
functionalities like Forms, Sockets, Local Storage etc.

HTML Markup

- ❖ html
- ❖ head
- ❖ body
- ❖ div(division)
- ❖ p (paragraph)
- ❖ form
- ❖ input
- ❖ select, option
- ❖ textarea
- ❖ button
- ❖ label
- ❖ h1,h2,h3,h4,h5,h6
- ❖ img
- ❖ table
- ❖ thead
- ❖ tbody
- ❖ th, tr, td
- ❖ ol, ul(lists)
- ❖ li (list item)

Semantic Markup

- ❖ header
 - ❖ For an introduction of a document or section, could include navigation
- ❖ footer
 - ❖ For a footer of a document or section, could include the name of the author, the date of the document, contact information, or copyright information
- ❖ nav
 - ❖ Defines the navigation
- ❖ main
 - ❖ The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.
- ❖ section
 - ❖ For a section in a document. Such as chapters, headers, footers, or any other sections of the document
- ❖ article
 - ❖ Specifies independent, self-contained content, could be a news-article, blog post, forum post, or other articles which can be distributed independently from the rest of the site

Semantic Markup

- ❖ **aside**
 - ❖ For content aside from the content it is placed in. The aside content should be related to the surrounding content
- ❖ **canvas**
 - ❖ For making graphics with a script
- ❖ **figure**
 - ❖ For grouping a section of stand-alone content, could be a video
- ❖ **figcaption**
 - ❖ The caption of the figure section
- ❖ **time**
 - ❖ For defining a time or a date, or both

Semantic Markup

- ❖ audio
 - ❖ For multimedia content, sounds, music or other audio streams
- ❖ video
 - ❖ For video content, such as a movie clip or other video streams
- ❖ datalist
 - ❖ A list of options for input values
- ❖ mark
 - ❖ For text that should be highlighted
- ❖ details & summary
 - ❖ represents a disclosure widget
- ❖ progress
 - ❖ represents the completion progress of a task
- ❖ meter
 - ❖ Displaying a scalar measurement within a known range

Getting Started

- ❖ HTML 5 pages can be viewed on any browser
- ❖ However, features of HTML 5 are implemented by only the latest browsers
- ❖ The doc type for HTML 5
 - ❖ <!DOCTYPE html>
 - ❖ Much simpler than earlier versions

Javascript API's

- ❖ DOM Manipulation
- ❖ Forms
- ❖ Geolocation
- ❖ Web Storage
- ❖ Web Workers
- ❖ Media Capture
- ❖ Web Notifications
- ❖ Offline Web Applications
- ❖ Server Sent Events
- ❖ Web Sockets
- ❖ XMLHttpRequest
- ❖ Canvas 2D Context & WebGL

Form Field Types

- ❖ text
- ❖ password
- ❖ email
- ❖ url
- ❖ range
- ❖ number
- ❖ date
- ❖ datetime
- ❖ color
- ❖ tel
- ❖ search

Methods and Properties

- ❖ boolean checkValidity();
 - ❖ Validates the control against constraints.
 - ❖ If any constraints are not satisfied, then the user-agent fires the ***invalid*** event and the method returns false.
 - ❖ If all constraints are satisfied does nothing, the method returns true
- ❖ readonly attribute ValidityState validity;
 - ❖ Returns the validity state of the control
- ❖ readonly attribute DOMString validationMessage;
 - ❖ Return a suitably localized message that the user agent would show the user if this were the only form control with a validity constraint problem.
- ❖ void setCustomValidity(in DOMString error);
 - ❖ set the ***custom validity error*** message to the value of the given message argument.

Validity States

- ◊ **valueMissing**
 - ◊ Returns true if a required element is empty.
- ◊ **typeMismatch**
 - ◊ Applies to all the new type attributes. For example, if an email value is incorrect, this attribute returns true.
- ◊ **patternMismatch**
 - ◊ Returns true when an element contains the pattern attribute and doesn't conform to the regular expression.
- ◊ **tooLong**
 - ◊ When any element surpasses its max length property this attribute will return true.
- ◊ **rangeUnderflow** and **rangeOverflow**
 - ◊ If an element's min or max attributes are above or below the specified values, this attribute will return true.
- ◊ **stepMismatch**
 - ◊ When an element with the step attribute doesn't conform to the required step value, this attribute returns true.
- ◊ **valid**
 - ◊ If any of the values listed above return true, this attribute returns false to indicate the field is invalid. Otherwise, if all the conditions are met, it will return true.

Additional Form Support

- ❖ **autofocus**
 - ❖ Support for placing the focus on a specific form element.
- ❖ **required**
 - ❖ Support for enabling the input field as required.
- ❖ **placeholder**
 - ❖ Support for placing a placeholder text in a form field
- ❖ **pattern**
 - ❖ Define a regular expression for validation
- ❖ **contenteditable**
 - ❖ Support for inline content editing

Browser Storage

- ❖ HTML 5 introduces a new mechanism for storing data on the client
 - ❖ Web Storage
 - ❖ Easy to use, powerful and reasonably secure.
 - ❖ Other mechanisms
 - ❖ Web SQL Databases
 - ❖ Indexed Databases

Web Storage

- ❖ Allows us to store information as key-value pairs on the client-side
- ❖ Unlike cookies the information stored does not get exchanged to the server.
- ❖ Browsers restricts every domain to store up to 5mb to 10mb of data on the client.
- ❖ Currently supported by most browsers.
- ❖ Web Storage gives us 2 scopes to store information
 - ❖ localStorage: A key/value pair, tied to the domain and persists across browser sessions.
 - ❖ sessionStorage: A key/value pair tied to the domain and erased when the browser session ends.
- ❖ The API's are implemented as javascript functions using the
 - ❖ window.localStorage object
 - ❖ window.sessionStorage object

Web Storage Methods and events

- ❖ Methods

- ❖ `setItem(key, value)`
- ❖ `getItem(key)`
- ❖ `removeItem(key)`
- ❖ `clear()`

- ❖ Events

- ❖ `storage`

Indexed Databases

- ❖ The Indexed Database API is a web browser standard interface for a local database of records, holding simple values and hierarchical objects.
- ❖ It facilitates high performance searches on the data using indexes.
- ❖ Support is included on Firefox 4.0, Chrome (version 11 onwards), IE 10.
- ❖ Like normal databases, we can create a multiple databases.
- ❖ Databases hold information in an ***ObjectStore***.
- ❖ Each record in the ObjectStore consists of a keyPath and a corresponding value(object).
- ❖ ObjectStores can also include indexes for faster retrieval of records and can store large amount of objects.

Indexed Databases API

- ❖ IndexedDB has two API modes
 - ❖ synchronous
 - ❖ asynchronous
- ❖ The API is exposed through the ***window.indexedDB*** object
- ❖ Accessing the indexedDB object
 - ❖ `var indexedDB = window.indexedDB || window.webkitIndexedDB || window.mozIndexedDB || window.msIndexedDB;`

Opening the database

- ❖ `var request = indexedDB.open(name, versionNo); request.onsuccess = function(evt) {...};
request.onerror = function(evt) {...}; request.onupgradeneeded = function(evt) {...};`
- ❖ The open method returns a ***IDBRequest*** object.
- ❖ The IDBRequest object has a property called ***result*** of type ***IDBDatabase***.

Creating an Object Store

- ❖ The ***IDBDatabase*** type has a method called `createObjectStore`.

- ❖ Example

- ❖

```
var objectStore =    request.result.createObjectStore("users",
  "id",autoIncrement: true});
```



```
{ keyPath:
```
 - ❖ **objectStore.createIndex("name", "name", {unique: false});**

ObjectStore Methods

- ❖ `add(value, optional key)`
 - ❖ Add a new object to the store
- ❖ `delete(key)`
 - ❖ Deletes an object from the store
- ❖ `get(key)`
 - ❖ Retrieves an object from the store
- ❖ `openCursor (optional range, optional direction)`
 - ❖ Returns a list of records
- ❖ Working with object store requires a transaction.
- ❖ The transaction is started from the ***IDBDatabase*** type.

AJAX

- ❖ Asynchronous JavaScript and XML.
- ❖ Ajax is methodology to asynchronously invoke a server endpoint using the HTTP(s) protocol from a browser application using JavaScript. The data exchange is in XML format.
- ❖ Lately the preferred data format is JSON.
- ❖ All browsers provide a JavaScript object called XMLHttpRequest which is the core object for AJAX.
- ❖ There are many high-level libraries built over XMLHttpRequest
- ❖ Fetch API is one such library which is part of HTML5

Web Sockets

- ❖ HTML5 Web Sockets defines a bi-directional, full-duplex communication channel that operates through a single TCP socket over the **web**
- ❖ It provides efficient, low-latency and low cost connection between **web** client and server, based on IETF **WebSocket** protocol
- ❖ Enables building **scalable, real-time** applications

Web Sockets

- ❖ Creating a socket

- ❖ `var wsUrl = 'ws://localhost:8888/DummyPath';`
 - ❖ `var websocket = new WebSocket(wsUrl);`

- ❖ Registering events

- ❖ `websocket.onopen = function (evt) { onOpen(evt) }; websocket.onclose = function (evt) { onClose(evt) };`
 - `websocket.onmessage = function (evt) { onMessage(evt) }; websocket.onerror = function (evt) { onError(evt) };`

- ❖ Sending Data

- ❖ `websocket.send("HTML5 WebSocket!");`

Geolocation

- ❖ Geolocation is the mechanism of finding out where you are in the world and (optionally) sharing that information with people you trust.
- ❖ Ways to locate
 - ❖ Global Positioning System(GPS)
 - ❖ IP Address
 - ❖ GSM/CDMA Cell Id's
 - ❖ WiFi and Bluetooth MAC Address
 - ❖ User Input

W3C Geolocation API

- ❖ A high level interface to get the geolocation on a device that is enabled.
- ❖ Currently supported by almost all new browsers.

Geolocation API

- ❖ The whole implementation of geolocation is in a single object called **window.navigator.geolocation**.
- ❖ Methods
 - ❖ `getCurrentPosition(successCallback, errorCallback, [options]).`
 - ❖ `watchPosition(successCallback, errorCallback, [options])`
 - ❖ `clearWatch(watchId)`
- ❖ Parameters
 - ❖ `successCallback`: invoked if the call was success.
 - ❖ `errorCallback`: invoked on an exception
 - ❖ `options`: a instance of `PositionOptions`.

PositionOptions properties

- ❖ enableHighAccuracy:Boolean
 - ❖ Flags the API to get as close to the exact location of the device
- ❖ maximumAge: Integer
 - ❖ Signals to the API that it will accept cached position with the age no greater than the time specified in milliseconds
- ❖ Timeout: Integer
 - ❖ The timeout in milliseconds

The Position Object

- ❖ The position object holding all geolocation information is returned on a successful callback.
- ❖ Properties
 - ❖ coords: Coordinates
 - ❖ Contains geographical coordinates
 - ❖ timestamp:
 - ❖ The time when the Position was obtained.

Coordinates Objects Property

- ❖ Latitude
 - ❖ measured in decimal degrees
- ❖ Longitude
 - ❖ measured in decimal degrees
- ❖ Altitude
- ❖ Accuracy
 - ❖ Specified in meters
- ❖ Heading
 - ❖ The direction of travelling
- ❖ Speed
 - ❖ The speed of travelling

PositionError

- ❖ This object holds the error information.
- ❖ Properties
 - ❖ Code: Integer
 - ❖ PERMISSION_DENIED(1)
 - ❖ POSITION_UNAVAILABLE(2)
 - ❖ TIMEOUT(3)
 - ❖ Message: String
 - ❖ Detailed message.

Web Workers

- ❖ Web Workers allow us to do background processing on the client side.
- ❖ JavaScript is a single threaded language, hence if a task takes too long to execute we force the user to wait.
- ❖ Web workers allow us to write concurrent programs.

Web Workers Scenarios

- ❖ Prefetching and/or caching data for later use
- ❖ Spell checker
- ❖ Analyzing video or audio data
- ❖ Background I/O or polling of webservices
- ❖ Processing large arrays or huge JSON responses
- ❖ Updating many rows of a local web database

Web Workers

- ❖ Workers do not have access to
 - ❖ The DOM
 - ❖ The window object
 - ❖ The document object
 - ❖ The parent object
- ❖ They can access
 - ❖ The location object
 - ❖ The navigator object
 - ❖ XMLHttpRequest
 - ❖ Timeout methods

Web Workers

- ❖ Create a new worker
 - ❖ `var worker = new Worker("task.js");`
- ❖ Send Message to a worker
 - ❖ `worker.postMessage("hello");`
- ❖ The worker script receives the message by handling the `onmessage` event.
- ❖ The worker can send back data by invoking
 - ❖ `self.postMessage("result")`
- ❖ The main script handles the `onmessage` event of the worker to receive the message.
- ❖ Stop a worker
 - ❖ `Worker.terminate();`
 - ❖ `self.close();`

Worker Events

- ❖ Events on the calling script
 - ❖ onmessage
 - ❖ onerror
- ❖ Events on the worker script
 - ❖ onmessage

Shared Web Workers

- ❖ Web workers are dedicated to the creator(the script that loads it).
- ❖ Shared web workers allow any number of scripts to communicate with a single worker.
- ❖ Shared web workers adhere to the same rules as their dedicated counterparts.
- ❖ In addition, page scripts may only communicate with shared web workers from the same origin/domain

Shared Web Workers

- ❖ Creating a Shared Web Worker
 - ❖ `var worker = new SharedWorker("jsworker.js");`
- ❖ Unlike dedicated web workers, you must communicate via a ‘port’ object.
 - ❖ `worker.port.start();`
- ❖ When the script starts the port, it connects to the loaded worker, in the worker the `onconnect` event is raised.
- ❖ Sending a message.
 - ❖ `worker.port.postMessage("hello");`
- ❖ When the worker receives the message the `onmessage` event is fired for that connection(`port`).
- ❖ Responses from the worker are received in the `onmessage` event of the port in the script.

Media Support

- ❖ HTML 5 introduces new markup to embed media on a page without the need of plugins.
- ❖ HTML Tags
 - ❖ <audio src="music.mp3"></audio>
 - ❖ <video src="movie.mp4"></video>

Supported Video Codec

- ❖ H.264
 - ❖ A high quality codec standardized in 2003 created by the MPEG group.
 - ❖ Supported by IE9, Safari 4, Chrome 3, IOS
- ❖ Theora
 - ❖ A royalty free codec
 - ❖ Supported in FF3.5, Chrome4, Opera 10
- ❖ VP8
 - ❖ Codec by google that's royalty free and similar quality as H.264
 - ❖ Supported by FF4, Chrome5, Opera

Supported Audio Codecs

- ❖ Advanced Audio Coding
 - ❖ The audio format that Apple uses with high quality and similar to H.264
 - ❖ It's not a free codec
 - ❖ Supported by Safari, Chrome , IOS
- ❖ MP3
 - ❖ Patented format.
 - ❖ Supported by IE9, Safari, Chrome, IOS
- ❖ Vorbis(OOG)
 - ❖ Open source, royalty free
 - ❖ Supported by FF, Chrome, Opera

Containers

- ❖ Containers are metadata files that identifies and interleaves audio or video.
- ❖ They wrap the audio and video streams
- ❖ Supported Containers
 - ❖ OOG with Theora video and Vorbis audio which works on FF, Chrome and Opera
 - ❖ MP4 with H.264 video and AAC audio which works on Safari and Chrome as well as in Flash and iOS devices
 - ❖ WebM using VP8 video and vorbis audio which works on FF, Chrome, Opera and Flash.

Attributes

- ❖ Controls
 - ❖ To display the controls
- ❖ Poster
 - ❖ To display a poster
- ❖ Autoplay
 - ❖ To auto play the video
- ❖ Loop
 - ❖ To loop

Video Methods & Properties

- ❖ play()
- ❖ pause()
- ❖ currentTime
- ❖ duration
- ❖ volume
- ❖ mute
- ❖ paused
- ❖ ended

Media Events

- ❖ oncanplay
- ❖ onplay
- ❖ onplaying
- ❖ onpause
- ❖ onended
- ❖ onvolumechange
- ❖ onerror

Limitations of HTML 5 Media

- ❖ No way to manage rights. Flash becomes a viable solution for such situations
- ❖ We have to encode in multiple formats, which could be expensive.

CSS

CSS 3.0

- ❖ Concepts
- ❖ Selectors
- ❖ Responsive Design
- ❖ Media Queries
- ❖ Box and Flex Model
- ❖ Bootstrap

Cascading Style Sheets

” ”

CSS is a stylesheet language used to describe the presentation of a document written in HTML.



CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

Selectors

- ◊ *
 - ◊ Targets every single element on a page
- ◊ #x
 - ◊ Targets the elements with the id="x"
- ◊ x
 - ◊ Targets the elements of type "x"
- ◊ .x
 - ◊ This is a class selector. More reusable

Selectors

❖ **x y**

- ❖ The descendant selector
- ❖ Selects the element “y” which is a descendant of “x”

❖ **x > y**

- ❖ Selects the element “y” which is direct child of “x”

❖ **x + y**

- ❖ Selects the element “y” which immediately follows (sibling) “x”

Selectors

- ❖ `x[attr]`
 - ❖ Attribute selector
 - ❖ Selects all elements “x” which have a the attribute “attr” defined.
- ❖ `x[attr=“foo”]`
 - ❖ Selects all elements “x” which have a the attribute “*attr*” ***equal to “foo”***.
- ❖ `x[attr*=“foo”] (CSS 3.0)`
 - ❖ Selects all elements “x” which have a the attribute “*attr*” ***contains “foo”***.
- ❖ `x[attr^=“foo”] (CSS 3.0)`
 - ❖ Selects all elements “x” which have a the attribute “*attr*” ***starts with “foo”***.

Selectors

- ❖ `x[attr $="foo"] (CSS 3.0)`
 - ❖ Selects all elements “x” which have a the attribute “*attr*” **ends with “foo”**.
- ❖ `x[attr ~="foo"] (CSS 3.0)`
 - ❖ Selects all elements “x” which have a the attribute “*attr*” **containing value “foo” that’s space seperated.**
- ❖ `x:checked (CSS 3.0)`
 - ❖ Targets an element that is checked(checkbox, radiobutton).
- ❖ `x:not(#id)`
 - ❖ Negation pseudo class
 - ❖ Selects all elements “x” which does not have the id equal to “id”

CSS 3

- ❖ x:first-child
 - ❖ Selects the first child of “x”
- ❖ x:last-child
 - ❖ Selects the first child of “x”
- ❖ x:nth-child(3)
 - ❖ Selects the 3rd child of type “x”
- ❖ x:nth-child(odd)
 - ❖ Selects the 3rd child of type “x”
- ❖ x:nth-child(even)
 - ❖ Selects the 3rd child of type “x”

CSS Animations

- ❖ Animations in CSS are defined by *keyframes*.
- ❖ The *keyframes* contains the rules for the animation.
- ❖ Animation rules are applied to the properties of an element.
- ❖ Example

```
@keyframes testAnimation
{
    0% { opacity: 0.0; }
    50% { opacity: 0.5; }
    100% { opacity: 1.0; }
}
```

CSS Animations

- ❖ Applying an animation

```
.myClass:hover{  
    animation: myanim;  
    animation-duration: 5s;  
    animation-iteration-count: 2;  
}
```

CSS Animations-Prefixes

- ❖ @-keyframes
 - ❖ The standard define by W3C
- ❖ @-webkit-keyframes
 - ❖ The webkit implementations like chrome, safari
- ❖ @-moz-keyframes
 - ❖ Mozilla implementaion like firefox
- ❖ @-o-keyframes
 - ❖ Opera implementation
- ❖ @-ms-keyframes
 - ❖ Microsoft implementation

CSS Animations

- ❖ Fill-Mode
 - ❖ none
 - ❖ backwards
 - ❖ forwards
 - ❖ Both
- ❖ Direction
 - ❖ none
 - ❖ reverse
 - ❖ alternate
 - ❖ alternate-reverse

CSS Transform Functions

- ❖ `translate(x,y)`
 - ❖ Defines a 2D translation
- ❖ `translate3d(x,y,z)`
 - ❖ Defines a 3D translation
- ❖ `translateX(x)`
 - ❖ Defines a translation, using only the value for the X-axis
- ❖ `translateY(y)`
 - ❖ Defines a translation, using only the value for the Y-axis
- ❖ `translateZ(z)`
 - ❖ Defines a 3D translation, using only the value for the Z-axis
- ❖ `scale(x,y)`
 - ❖ Defines a 2D scale transformation
- ❖ `scale3d(x,y,z)`
 - ❖ Defines a 3D scale transformation
- ❖ `scaleX(x)`
 - ❖ Defines a scale transformation by giving a value for the X-axis

CSS Transform Functions

- ❖ `rotate(angle)`
 - ❖ Defines a 2D rotation, the angle is specified in the parameter
- ❖ `rotate3d(x,y,z,angle)`
 - ❖ Defines a 3D rotation
- ❖ `rotateX(angle)`
 - ❖ Defines a 3D rotation along the X-axis
- ❖ `rotateY(angle)`
 - ❖ Defines a 3D rotation along the Y-axis
- ❖ `rotateZ(angle)`
 - ❖ Defines a 3D rotation along the Z-axis
- ❖ `skew(x-angle,y-angle)`
 - ❖ Defines a 2D skew transformation along the X- and the Y-axis
- ❖ `skewX(angle)`
 - ❖ Defines a 2D skew transformation along the X-axis
- ❖ `skewY(angle)`
 - ❖ Defines a 2D skew transformation along the Y-axis

CSS 3.0 Media Queries

- ❖ Media Queries is a module allowing content rendering to adapt to conditions such as screen resolution.
- ❖ It became a W3C recommended standard in June 2012.
- ❖ A media query consists
 - ❖ *media type*
 - ❖ one or more expressions

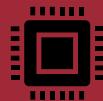
Media types

- ❖ screen
- ❖ print
- ❖ handheld
- ❖ projection
- ❖ tv
- ❖ all

Media Expressions

- ❖ min-width
- ❖ min-height
- ❖ max-width
- ❖ max-height
- ❖ device-width
- ❖ device-height
- ❖ orientation
- ❖ aspect-ratio
- ❖ device-aspect-ratio
- ❖ color

CSS Box Model



All HTML elements can be considered as boxes.



In CSS, the term "box model" is used when referring to design and layout.



The CSS box model is essentially a box that wraps around HTML elements



It consists of:

Margins
Borders
Padding
actual content.

CSS 3.0 Flexbox



In this flexbox model, the children of a box are laid out either horizontally or vertically.



Unused space can be assigned to a particular child or distributed among the children by assignment of flex to the children that should expand.



Nesting of boxes (horizontal inside vertical, or vertical inside horizontal) can be used to build layouts in two dimensions.

Flexbox styles

display

- values : display or inline-display

flex-direction

- row (default): left to right
- row-reverse: right to left
- column: same as row but top to bottom
- column-reverse

flex-wrap

- nowrap (default): single-line / left to right
- wrap: multi-line / left to right
- wrap-reverse: multi-line / right to left

flex-flow

- This is a shorthand flex-direction and flex-wrap properties
- flex-flow: <'flex-direction'> | | <'flex-wrap'>

Flexbox styles

justify-content

- **flex-start** (default): items are packed toward the start line
- **flex-end**: items are packed toward to end line
- **center**: items are centered along the line
- **space-between**: items are evenly distributed in the line; first item is on the start line, last item on the end line
- **space-around**

order(child)

- By default, flex items are laid out in the source order. However, the `order` property controls the order in which they appear in the flex container.

Bootstrap

- ❖ A CSS framework for building responsive, mobile-first sites.
- ❖ Works on all browsers(Bootstrap 3)
 - ❖ Bootstrap 4 support for IE from IE 10
- ❖ Provides CSS and HTML design templates
- ❖ Provides reusable components
- ❖ Supports LESS and SASS
- ❖ Customizable

CSS Animation editors

- ❖ CSS3 Please
 - ❖ <http://css3please.com/>
- ❖ CSS 3 generator
 - ❖ <http://css3generator.com/>
- ❖ Ceasar
 - ❖ <http://matthewlein.com/ceaser/>
- ❖ Cubic –Beizer
 - ❖ <http://cubic-bezier.com>

Anil Joseph

anil.jos@gmail.com

WhatsApp : +91 98331 69784

Thank You