



# HTML5 with JavaScript APIs

By Vijay Shivakumar



# Requirements ...

## IDEs

Aptana Studio 3.0

## Browsers

Chrome, Firefox, Opera, IE (latest versions)

## Web Servers

Tomcat or IIS or WAMP



# About you...

Designers

Developers

Content Writers

Business Analysts

Prior Knowledge on HTML or HTML5



# Vijay Shivakumar

Designer | Developer | Trainer

Training on web and Adobe products from past 12 years



**CERTIFIED EXPERT**  
Flex® with AIR

By Vijay Shivakumar

<http://learnjs.in>



# What I don not claim...

- To be associated with any of these technologies
- To teach you each and every thing about HTML 5
- That what ever I teach will never change
- That you wont have to learn anything on your own



# Introduction to HTML 5

What is HTML 5 ?

New emerging web, mobile... standard

Why do we need it ?

Increasing user demands for enhanced experience.

Who is behind working for it

WHATWG | Web Hypertext Application Technology Working Group

Apple | Mozilla | Opera

Google, Adobe and many more contributing now.



# What XHTML2.0 would be.

<http://w3.org/TR/2005/WD-xhtml2-20050527/>



# What HTML5 is

<http://whatwg.org>

Web Hypertext Application Technology Working Group

=====

Backward Compatible

Utility

Promote Usage of HTML5





# General Changes

## # Support for existing contents

existing html xhtml pages should get similar results as html5

deal with broken markups

e.g. `<li> item 1`

no corresponding closing tag

`<li> item 2`

badly nested elements

e.g. `<b> a <i> b </b> c </i>`

## # Graceful degrade

New elements to have fallback option

e.g. `<canvas>fallback</canvas>`

## # Use existing user agent specific attributes

## # Supporting widespread practices

e.g `<br>` for `<br/>`

## # Evolution not revolution

it is better to evolve an existing design rather than throwing it away.



# General Changes

## # Utility

Address existing problems

Separation of concerns new meaningful tags

Consistent DOM

## # Promote Usage of HTML5

Well defined behavior across browsers

Avoid complexity

Media independence

Accessibility



# NEW APIs in HTML5



Giving meaning to structure, semantics and appropriateness of tags  
Microdata offer structures for programs (machines).



Making apps start faster and be available without connection  
Offline API, Local Storage, Indexed DB



Accessing the user device which includes. Geolocation API,  
Orientation API (accelerometer), getUserMedia (access camera and mic)



Better communication via Web Sockets and Server pushing data  
Cross domain communication



# NEW APIs in HTML5



Plug-in Free Media



Captivating visuals with SVG, Canvas, WebGL, and CSS3 3D features



Performance Optimization with Web Workers and XMLHttpRequest2



# APIs in HTML5

Header  
Semantics  
Media Tags  
Input Types / Form API  
2D Canvas / 3D canvas  
Geolocation  
Form Validation  
getUserMedia API

Drag and Drop  
Local Storage  
Offline  
CORS  
Web Sockets  
Web Workers  
Microdata  
File API  
History API



# APIs in HTML5

Header

Semantics

Media Tags

Input Types / Form API

2D Canvas / 3D canvas

Geolocation

Form Validation

getUserMedia API

Drag and Drop

Local Storage

Offline

CORS

Web Sockets

Web Workers

Microdata

File API

History API



# APIs in HTML5

Header  
Semantics  
Media Tags  
Input Types / Form API  
2D Canvas / 3D canvas  
Geolocation  
Form Validation  
getUserMedia API

Drag and Drop  
Local Storage  
Offline

CORS  
Web Sockets  
Web Workers

Microdata  
File API  
History API

Post Message  
API



# What we get to learn?

DAY 1

Header  
Semantics  
Media Tags  
Form API  
Canvas 2D

DAY 2

Canvas 2D  
SVG  
Validation  
Geo  
Location  
Storage API

DAY 3

Offline API  
External  
Libraries  
ModernizrJS  
More APIs





# Less Header code



# Header Code in past

HTML 4.01 Strict  
HTML 4.01 Transitional  
HTML 4.01 Frameset  
XHTML 1.0 Strict  
XHTML 1.0 Transitional  
XHTML 1.0 Frameset  
XHTML 1.1

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



# Header Code in Future

`<!DOCTYPE html>`



# Header Code in past

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<html>
```

```
<html lang="en"> (optionally)
```

```
<meta http-equiv="Content-Type"  
      content="text/html; charset=utf-8" />
```

```
<meta charset="utf-8">
```



# Header Code in past

```
<style type="text/css">
```

```
<style>
```

```
<script type="text/javascript">
```

```
<script>
```

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

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



# New Semantics



# DIV for division SPAN for selection

div tags were used to group other tags together



# Meaningful Tags

header  
hgroup  
nav  
main  
section  
article  
aside  
footer

figure  
figcaption  
time  
mark





# New in HTML 5.1

**main :**

shall be used to mark the main content of a web page, excluding footers, headers, navigation blocks, and sidebars. There shall NOT be more than one `<main>` element in a document



# New Meaningful Tags

**header :**

will be on the top of page or content if required.

**hgroup:**

will be used to group h1 to h6 tags together.

**nav:**

will be used to group any navigation elements like anchors and links.

**section:**

will be used to declare contents of the page that is complete and full.

**article:**

will be used to contain the matter / text that is full in itself.



# New Meaningful Tags

## aside :

will be for contents that are either sides of the page that may not be required to understand the section or the contents of the page  
eg. References about the content.

## footer :

will be in the bottom of the page or the content.

## mark :

will be used to highlight the content

## figure :

will be used to group related images together especially the one that needs a caption.

**HEADER**

**NAV**

**MAIN**

**HEADER**

**SECTION**

**ARTICLE**

**ARTICLE**

**ARTICLE**

**HEADER**

**ASIDE**

**FOOTER**



# HTML5 Semantics not supported in your browser ?

HTML5SHIV

<https://github.com/aFarkas/html5shiv>

HTML5 BOILERPLATE

<http://html5boilerplate.com>

Modernizr

<http://modernizr.com>



# How to use HTML5SHIV

Shiv or Shim ?

```
<!--[if lt IE 9]>  
  <script src="script/dist/html5shiv.js"></script>  
<![endif]-->
```



# Form Inputs API

By Vijay Shivakumar

<http://learnjs.in>



# New Input Types

```
<input type="search" />
<input type="color" />
<input type="range" />
<input type="time" />
<input type="date" />
<input type="datetime" />
<input type="week" />
<input type="month" />
<input type="number" />
<input type="datetime-local" />
```

```
<input type="email" />
<input type="tel" />
<input type="url" />
<progress value="0~1"/>
<meter value="0~1"/>
<output value="" id="" />
```





# Normal Keypad in iPhone





# Modified Keypad in iPhone

`<input type= "email" />`



By Vijay Shivakumar

`<input type= "url" />`



<http://learnjs.in>



# Modified Keypad in iPhone

`<input type="number" />`



By Vijay Shivakumar

`<input type="tel" />`



<http://learnjs.in>



# New Attributes on Inputs

autofocus  
placeholder  
required  
autocomplete  
pattern



# Validation API



# Validation API

required  
min  
max  
step  
pattern

attributes for validation



# Media API



# Before Media API

```
<object classid="clsid:d27cdb6e-ae6d-height="344"  
  codebase="http://download.flash/swflash.cab#versio  
  n=6,0,40,0">  
<param name="allowFullScreen" value=""/>  
<param name="allowscriptaccess" value=""/>  
<param name="src" value=""/>  
<embed type="application/x-shockwave-src="link"  
  allowfullscreen=">  
</embed>  
</object>
```





# Before Media API

## Audio | Video

Flash was the most reliable way to play video and audio on the web.

Roughly 99.97% of all desktops have Flash player.

iPhone/iPad does not.

They do support HTML5 `<video>`



# HTML5 Media API

Audio | Video

H.264 : It is the most widely supported format promoted by MPEG LA a patent pool company. But licensing costs browser makers \$5 million a year.

## Support



---

## Does Not Support





# HTML5 Media API

Audio | Video

Ogg : Includes a number of independent open source codec for both audio and video. is patent-free and fully open.

## Support



---

## Does Not Support





# The Goodnews

mozilla

1 Amount 2 Payment 3 Personal

Donate now

\$20 \$10 \$5

\$3 \$ Other amount

☒ One-time ☐ Monthly

Next

Other ways to give: [Bitcoin](#) | [Check](#)

Problems donating? [Visit our FAQ](#) for answers to most common questions.  
Still have problems? [Send us an email](#).

Contributions go to the Mozilla Foundation, a 501(c)(3) organization, to be used to fund research and development of open source software. They are not deductible for tax purposes.

By Vijay Shivakumar

<http://learnjs.in>



# More Goodnews

VP8 : A video compression format by Google in 2010  
launched webM under an irrevocable free patent  
license

WebM is sponsored and supported by over 40 companies  
including mozilla, opera, google, adobe etc...



# HTML5 Media API

## Audio | Video

webM

**Support**



**Does Not Support**





# Video tag attributes (few)

autoplay

loop

played

preload

controls

muted

poster

src



# methods on media API

```
video.canPlayType();  
video.load();  
video.pause();  
video.play();
```





By Vijay Shivakumar

<http://learnjs.in>



# Fallback Options

Flash Player | Infallible, works on all except apple devices

YouTube link | use if the content can be made public

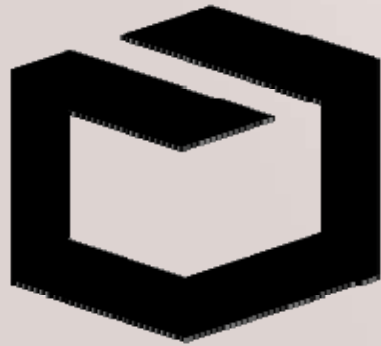
<http://videojs.com>

<http://projekktor.com>

<http://jwplayer.com>

<http://mediaelementjs.com>

Miro video converter (offline and free)



# Canvas API



# Canvas | what is it for...

- Data visualization

<http://raphaeljs.com/> SVG

<http://alteredqualia.com/canvasmol/>

- Animated graphics

<http://www.canvasdemos.com/>

- Web applications

<http://mugtug.com/sketchpad/> | <http://mudcu.be/sketchpad/>



# Canvas | what is it for...?

- Games

<http://www.pirateslovedaisies.com/>

<http://www.google.com/pacman/>

<https://www.google.com/doodles>



# Platforms

## Supported

---



---

**Grrr....**



# Canvas Fundamentals

Dynamic bitmap with JavaScript

Allow drawing into a bitmap area

Think about it as a dynamic PNG

Rectangles, lines, fills, arcs, Bezier curves, etc.

Use Text, Images, Videos and Shapes

Immediate mode : Fire and Forget

It does not remember what you drew last.

It's up to you to maintain your objects tree

This is a black box : content not visible into the DOM

Beware of accessibility issues

Simple API: 45 methods, 21 attributes

By Vijay Shivakumar

<http://learnjs.in>



# Geolocation





# Geolocation API

## Works on

Firefox	IE	Chrome	Safari	Opera	iPhone	Android	Blackberry
3.5	9.0	5	5	10.63	3.2	2.1	6.0

## Sources for Geolocation

IP address / ISP – not very accurate

Wi Fi spots – will give you block and street level accuracy

GPS – will deliver accurate location of the user



# Using Geolocation API

`navigator.geolocation` : will return true if supported on device

---

`getCurrentPosition()`

attempts to get the current location of the user asynchronously

`watchPosition()`

starts monitoring the location of a user at an interval.

`clearWatch()`

stops monitoring the location of a user



# Methods of Geolocation

```
getCurrentPosition( successFunction, failureFunction,  
    {enableHighAccuracy: true, timeout:5000, maximumAge:6000} );  
watchPosition( same as above );
```

---

**enableHighAccuracy:** Is a Boolean setting that allows you to use accurate GPS detection (when available).

**maximumAge:** specifies how recently (in milliseconds) location detection needs to have occurred.

**timeout :** specifies when(inmilliseconds)an attempt to get a user location needs to timeout.



# The Position

## position Object

---

timestamp: returns the time when the location was detected.

coords.latitude: returns the latitude in degrees.

coords.longitude: returns the longitude in degrees.

coords.accuracy: returns how accurate the location is, in meters.

coords.altitude: returns the altitude , if available.

coords.altitudeAccuracy: gives altitude accuracy, in meters, if available.

coords.speed: returns speed (based on previous detected position),  
in meters/second.

coords.heading: returns the angle, in degrees clockwise from true  
north.



# Error Object

## 1 : PERMISSION\_DENIED

the user disallowed sharing his or her location

## 2 : POSITION\_UNAVAILABLE

the position can't be found, the network is down, or GPS is unavailable.

## 3 : TIMEOUT

timeout occurred ,as it took too long to get the user's location.



# Geolocation Fallback

geo.js

<http://code.google.com/p/geo-location-javascript/>



# Offline Browsing API

By Vijay Shivakumar

<http://learnjs.in>



# Offline Browsing API

applicationCache

```
<html manifest="myapp.manifest">
```

Files with extension .manifest and .appcache are common

Can also be a an absolute location of a file on the same domain (crossdomain files wont work)

Set the mime type to support old browsers





# Offline Browsing API

## CACHE MANIFEST

Has 3 sections

CACHE:

NETWORK:

FALLBACK:

These sections can be listed in any order and each section can appear more than once in a single manifest.



# Offline Browsing API

## CACHE MANIFEST

The only required line in the file



# Offline Browsing API

## CACHE:

The default section declares all the files that will be stored for offline usage.

Each file needs to be mentioned in a separate line

Once cached the files will always be fetched from browser cache not from the server.



# Offline Browsing API

## NETWORK:

Shows all the files that need network access to work.

Can take wildcards to represent multiple files and directories. ( \* )

otherwise the network isn't used, even if the user is online



# Offline Browsing API

## FALLBACK:

A list of files that can be used in place of requested files

You can use wildcards ( / ) to create a fallback for any file that you asked if now cached



# Offline Browsing API

#Comment / version 001

Needs to be updated with there is any change in the file



# Storage API



# Types of Storage API

Cookies

Web Storage

Local Storage

Session Storage

Browser Databases ( Indexed DB / Web SQL )





# Support

Firefox	IE	Chrome	Safari	Opera	iPhone	Android	Blackberry
3.0	8.0	3.0	4	10.5	3.0	2.0	6.0



# Properties and Methods

<u>length</u>	Number of stored strings
<u>getItem()</u>	read the value of the key (name)
<u>setItem()</u>	add / modify the value of the key (name)
<u>removeItem()</u>	remove the name and value
<u>clear()</u>	removes all name values of your domain
<u>key()</u>	will return the stored name in that index



# Storage Event

```
addEventListener("storage", callBack)
```

```
window.onstorage = function(){}  
• event properties
```

- **key** : string the named key that was CUD
- **oldValue** : previous value (now overwritten), or null
- **newValue** : new value, or null if an item was removed
- **url** : string the page which called a method that triggered this change



# Storage Error

QUOTA\_EXCEEDED\_ERR

when the app exceeds the allowed storage



# Working With IndexedDB



# File API



# File API

- Until html5 we had to use server side programs to handle files
- HTML5 File API provides ways to access and read local files
- Latest updates <http://www.w3.org/TR/FileAPI/>
- Use `<input type="file" />` or drag n drop



# File API | Features

Has 3 major sections

- file reader

- file writer (not currently implemented)

- file system (not currently implemented)





# File API | File Reader

- Select files to upload on the client side
- Restrict kinds of file from being uploaded
- Generate thumbnails for uploads
- Check the modified date to match on server
- Parse and get detailed file info
- Modify and send to server



# File API | 4 types of file data

## Text Data

- `readAsText()`

## data:// URL

- `readAsDataURL()`

## Binary Data

- `readAsBinaryString()`

## Array Buffer



# History API

By Vijay Shivakumar

<http://learnjs.in>



# History API | Why ?

The url on the browser modifies when making an ajax call

No reference to go back in the async call

History API allows us to make changes to url text

Can not work with local files needs a web server



# History API | How ?

Use the `pushState()` to create a new history  
takes 3 properties

**state** : can be any JSON data

- It is passed back to the `popstate` event handler

**title** : can be any string

- currently unused by major browsers

**url** : can be any string

- that gets displayed in address bar (this wont create links)



# Communication API

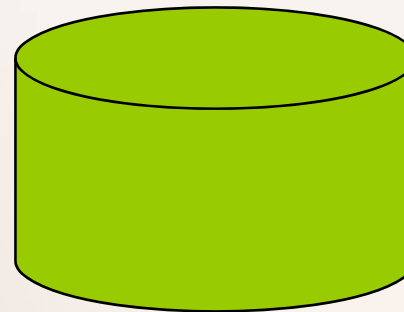


# What is it ?

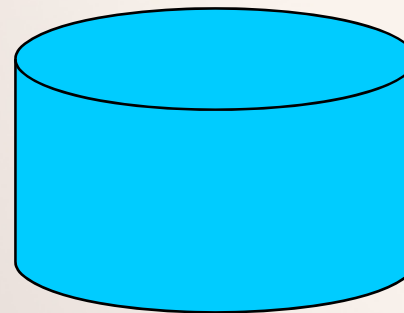
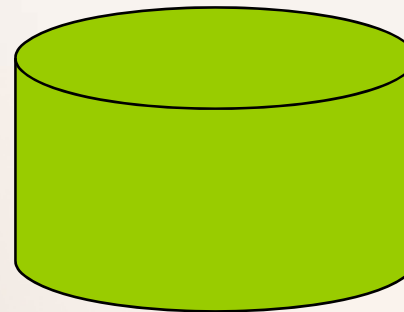
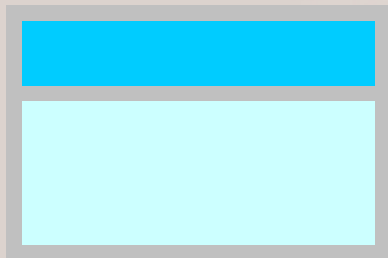
Firefox	IE	Chrome	Safari	Opera	iPhone	Android	Blackberry
3.0	8.0	2	4	9.6	3.0	2.0	6.0

## Cross Document Messaging

Applications from different domains can communicate safely  
Communication between IFrames, and Windows  
Communication is enabled via PostMessage API

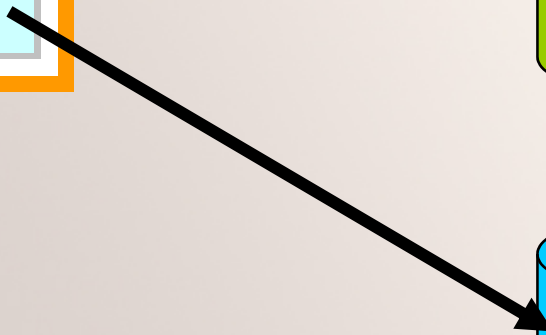
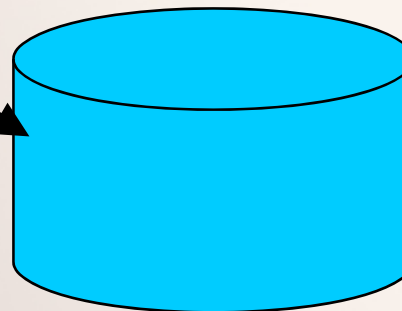
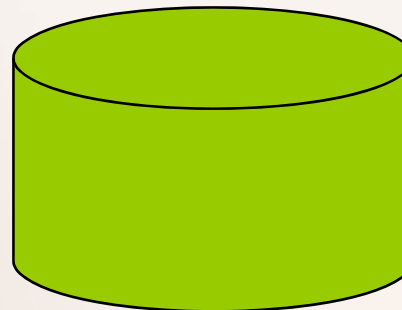
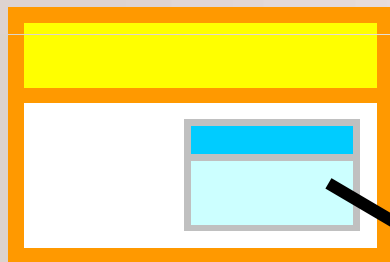


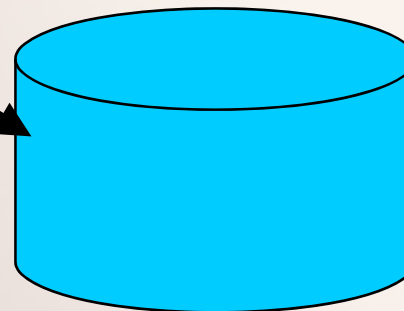
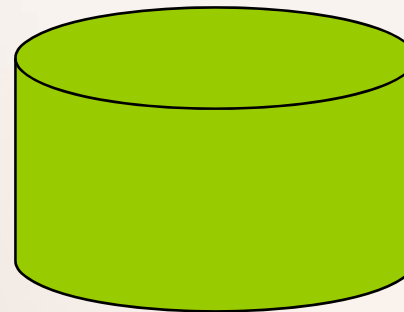
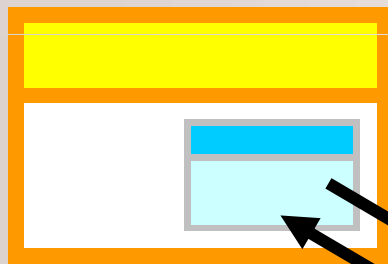


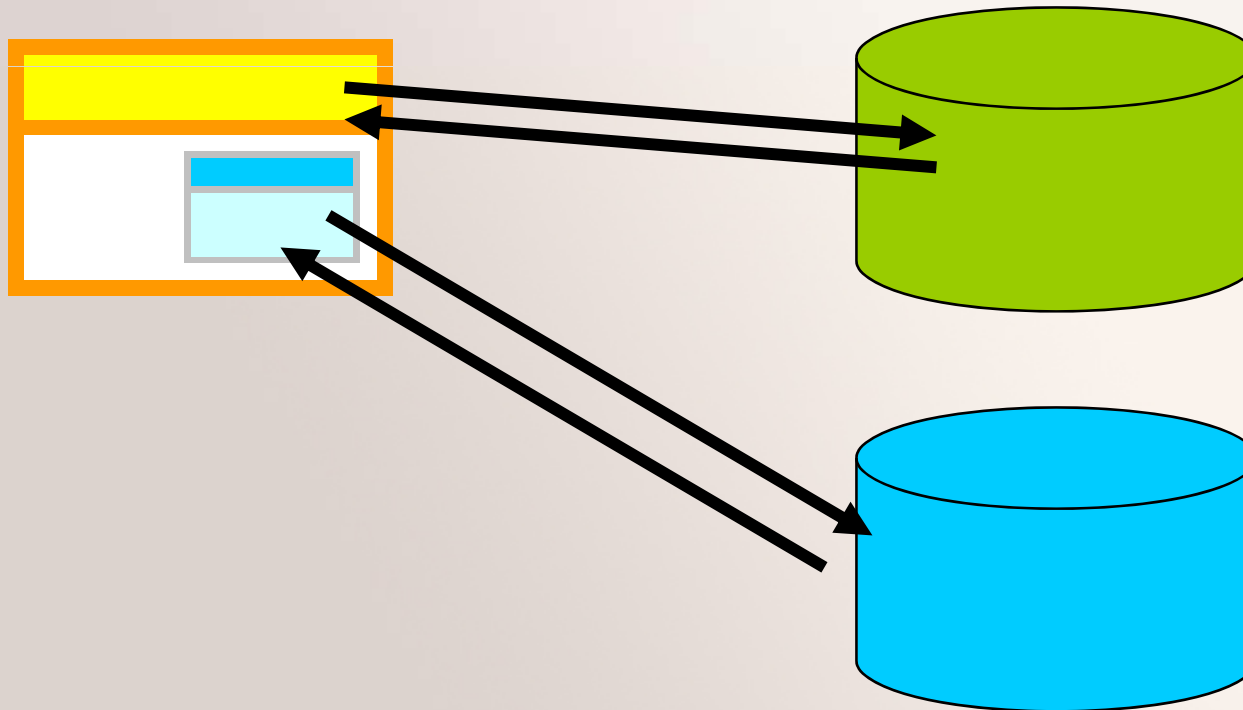


By Vijay Shivakumar

<http://learnjs.in>

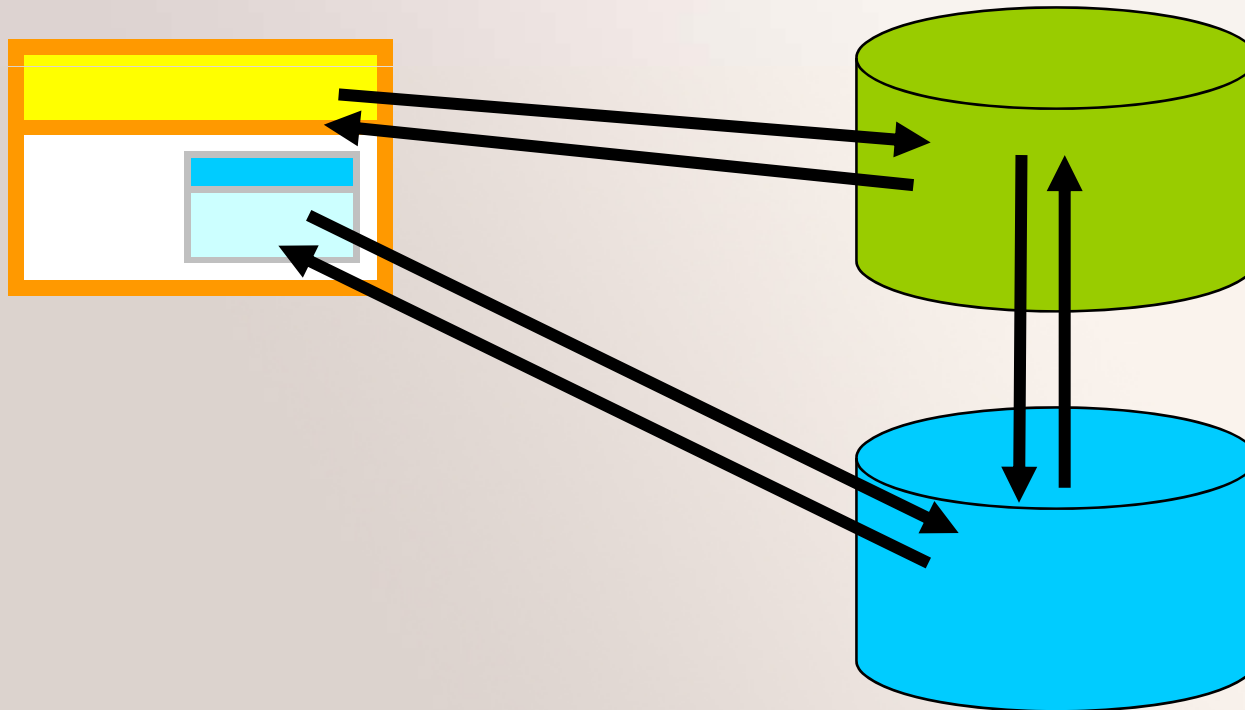






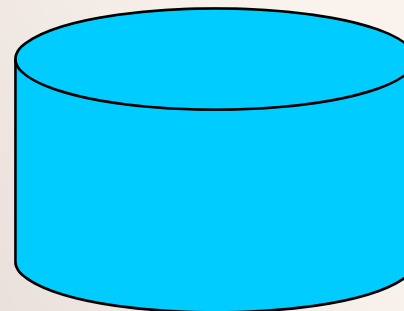
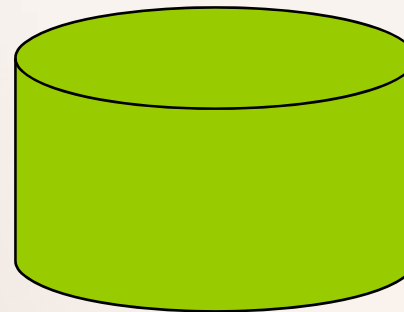
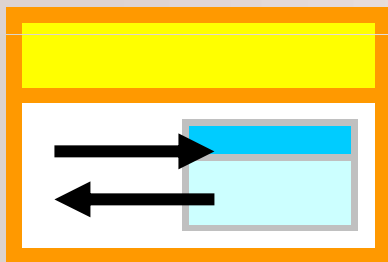
By Vijay Shivakumar

<http://learnjs.in>



By Vijay Shivakumar

<http://learnjs.in>





# Socket API



# Existing scenarios

HTTP a request and response protocol.

Designed to request text files

Poor for real time data on server

– (Chat, Dashboard, Games etc..)

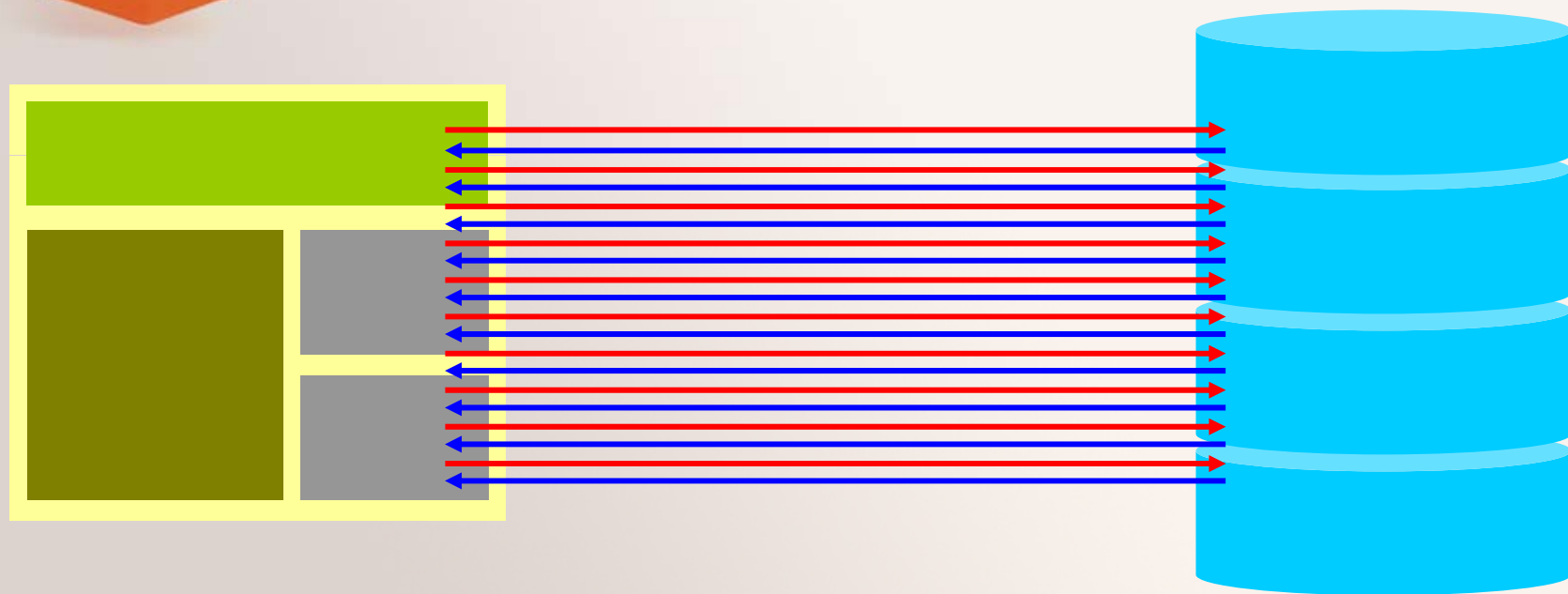
So we adopted

- Recursive Client Request (Polling)
- Server Push





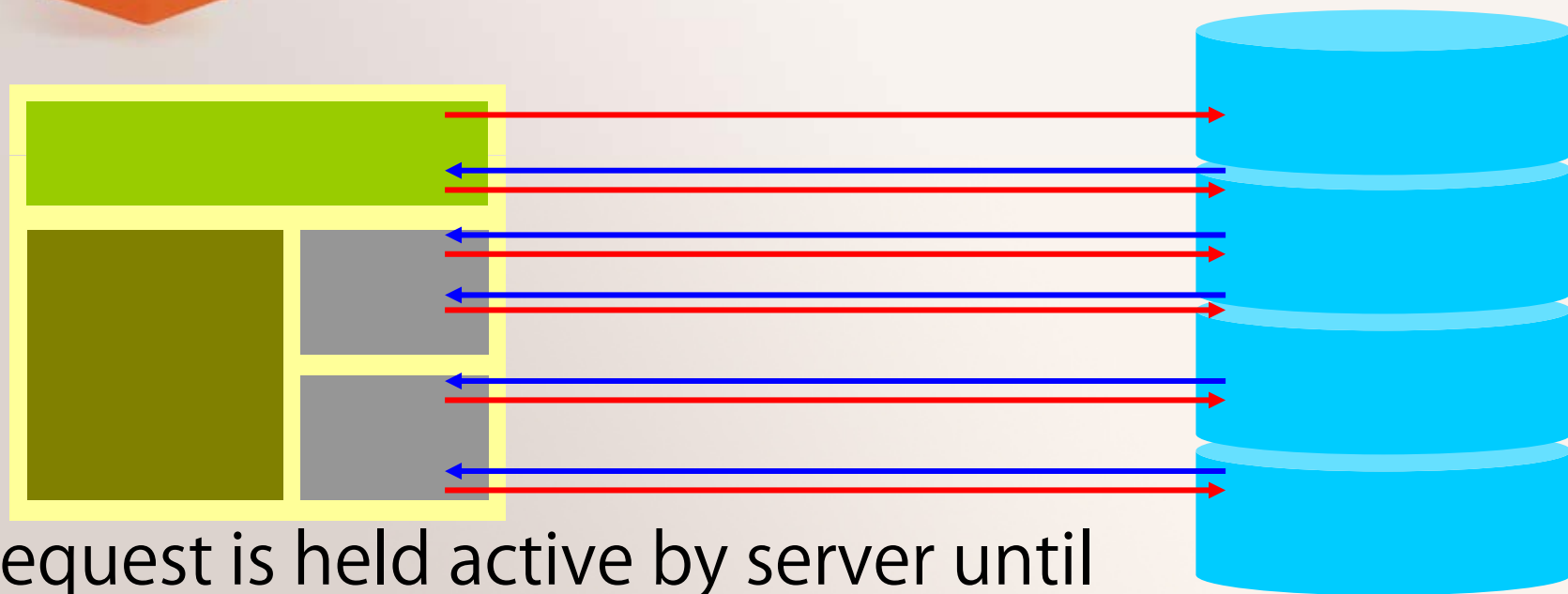
# Polling (Ajax)



Every request has a response even If it is empty.



## Comet / Server Push (Ajax 2)



Request is held active by server until  
there is an update and then responds  
For every response the client will send a new request



# Disadvantages

Large sequence of http requests, more than one a second

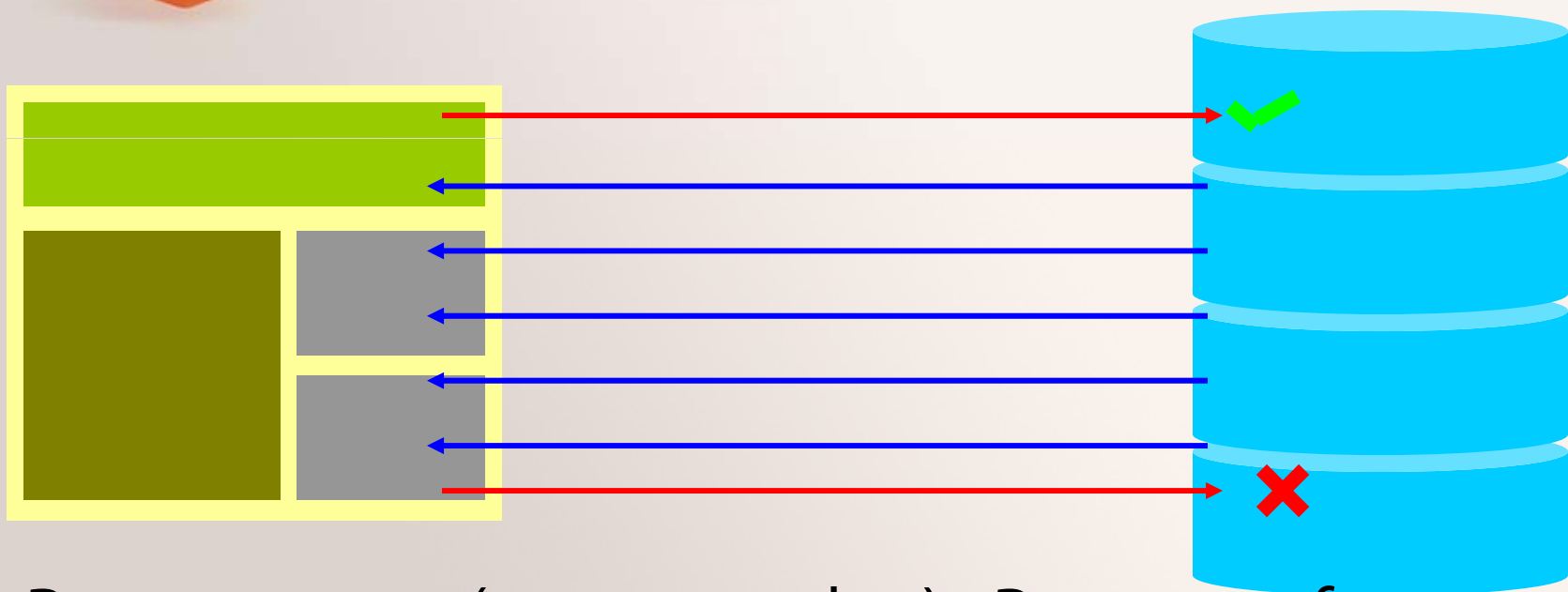
Huge amount of server load as for each request

Overhead of HTTP headers

User authentication



# Socket



Request once (open a socket) , Response for every server updates , Until you choose to close socket



# Socket Methods

```
var ws = new WebSocket("url")  
ws.send("message");  
ws.close(); terminate the socket connection
```

---

```
ws.onopen = openFun;  
ws.onclose = closeFun;  
ws.onmessage = messageFun;  
ws.onerror = errorFun;
```



# Drag N Drop API

By Vijay Shivakumar

<http://learnjs.in>



[vijay.shivu@gmail.com](mailto:vijay.shivu@gmail.com)



# Offline API





# Why do we need it ?

HTML, CSS, and JS stay fairly consistent  
Native browser caching is unreliable  
Caching resources creates faster apps!  
Decent mobile support



# New APIs

Battery Status API

Vibrate API

Tab Focus API / Page

Visibility API

FullScreen API

getUserMedia API

High Resolution Time API

User Timing API

Navigation Timing API

Network Information API

Document Edit API

File Reader / Writer File API

History API

Contacts API



[vijay.shivu@gmail.com](mailto:vijay.shivu@gmail.com)