INLEDANDE WEBBPROGRAMMERING MED JAVASCRIPT

INTRODUCTION TO WEB PROGRAMING USING JAVASCRIPT

ME152A

L6: 1. WEB FRAMEWORKS AND APIs

2. INTRODUCTION TO HTML5

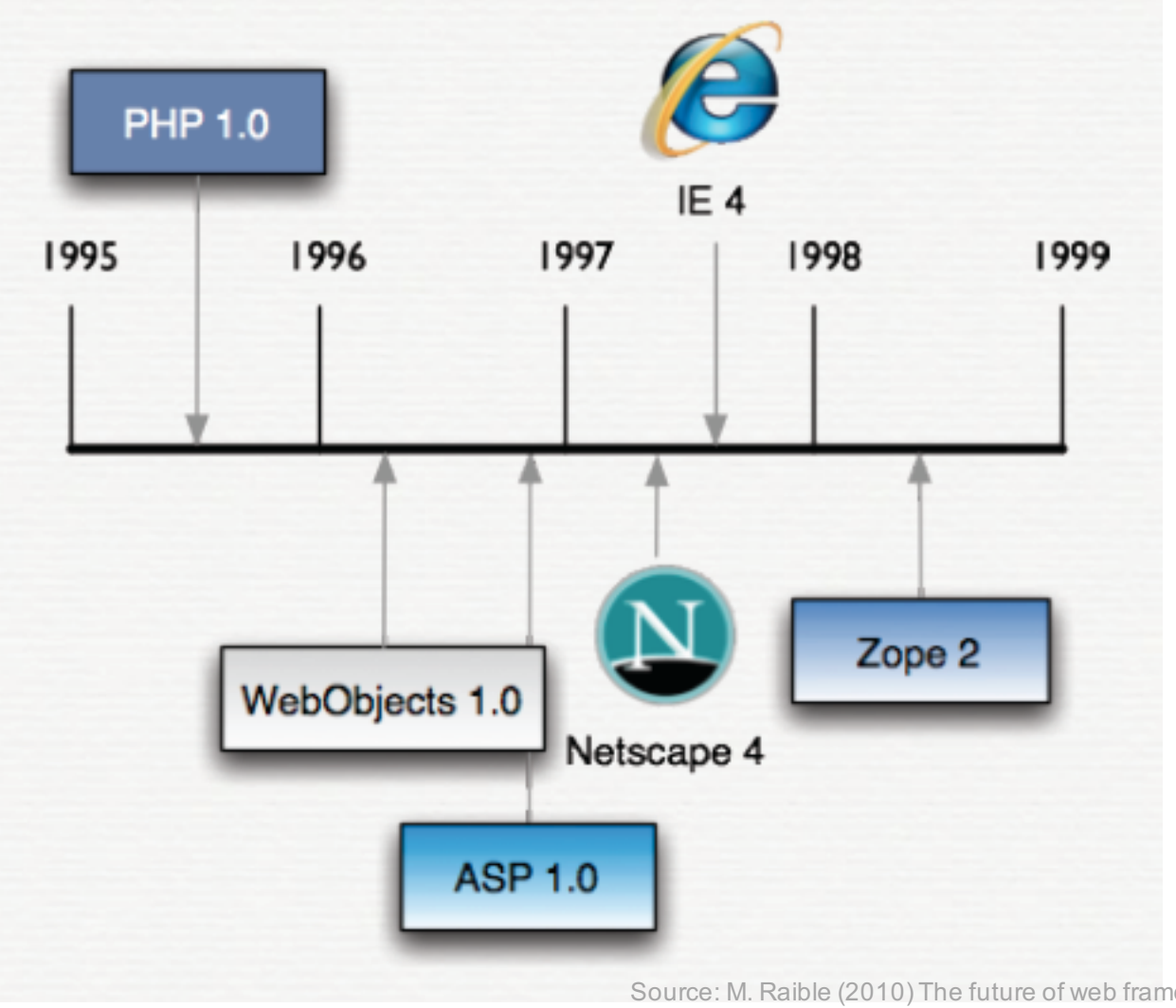
OUTLINE

- Introduction to web frameworks
- History of web frameworks
- Architecture: Model View Controller (MVC)
- Some frameworks
- Web APIs + couple of examples
- Summary

INTRODUCTION: WEB FRAMEWORKS

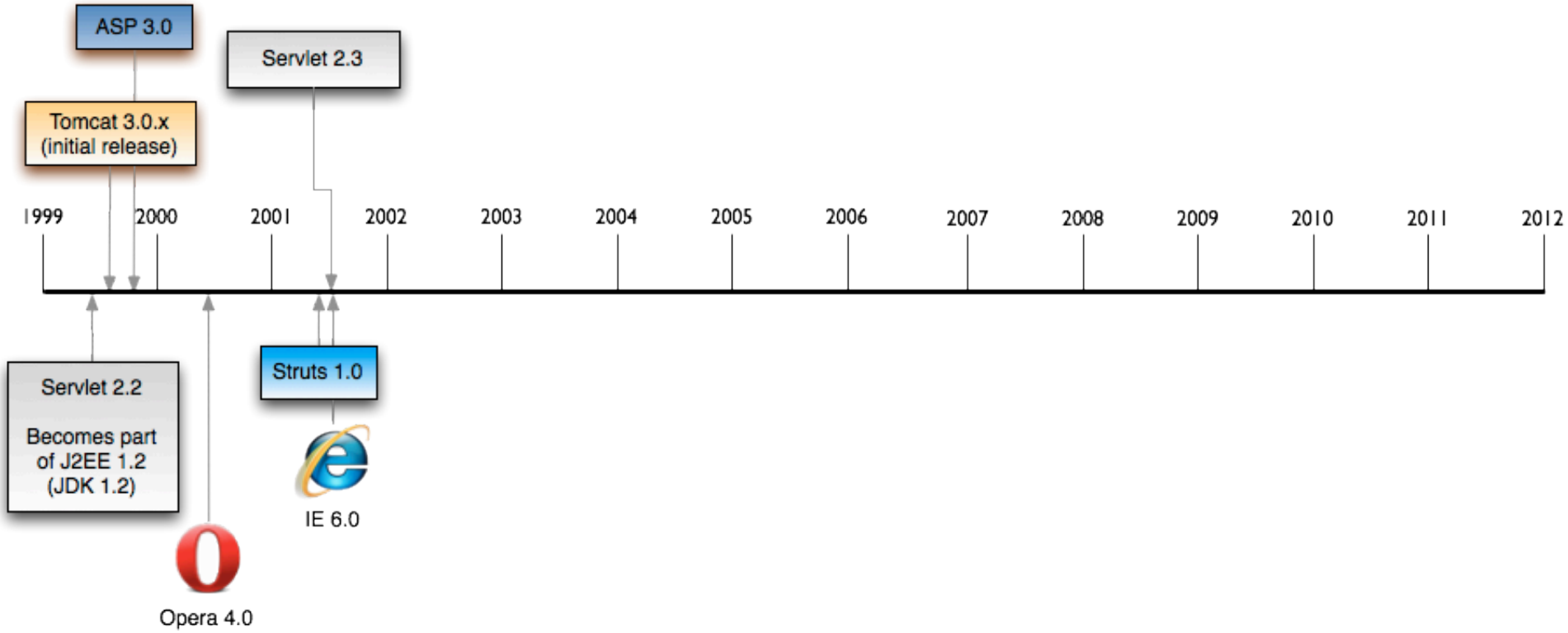
- A web application framework is a software framework that is designed to support the development of Web-based applications and services.
- The framework aims to alleviate the overhead associated with common activities performed in Web development.
- To simplify the definition:
  - Web framework simplifies the development
    - handling sessions
    - data validation, APIs, etc.
  - Frameworks are libraries that help you develop faster

HISTORY OF WEB FRAMEWORKS (1995-1999)



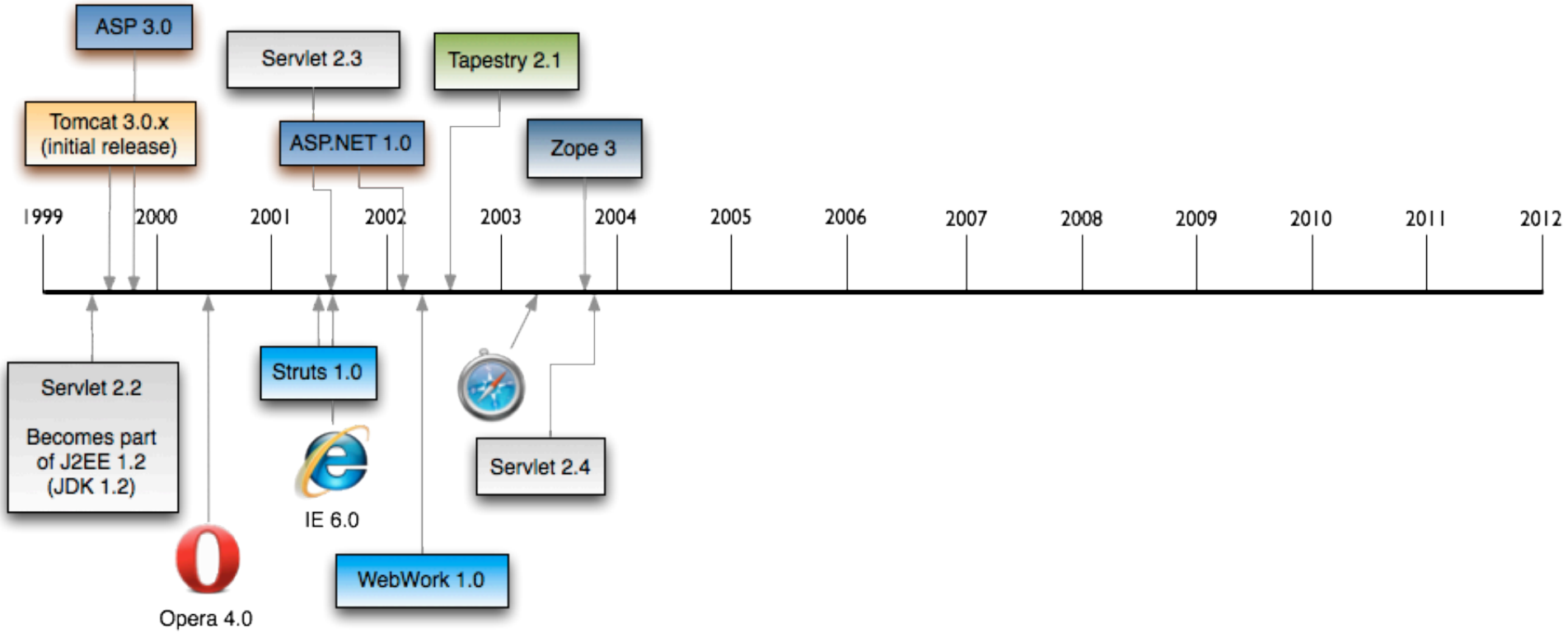
Source: M. Raible (2010) The future of web frameworks

HISTORY OF WEB FRAMEWORKS (1999-2012)



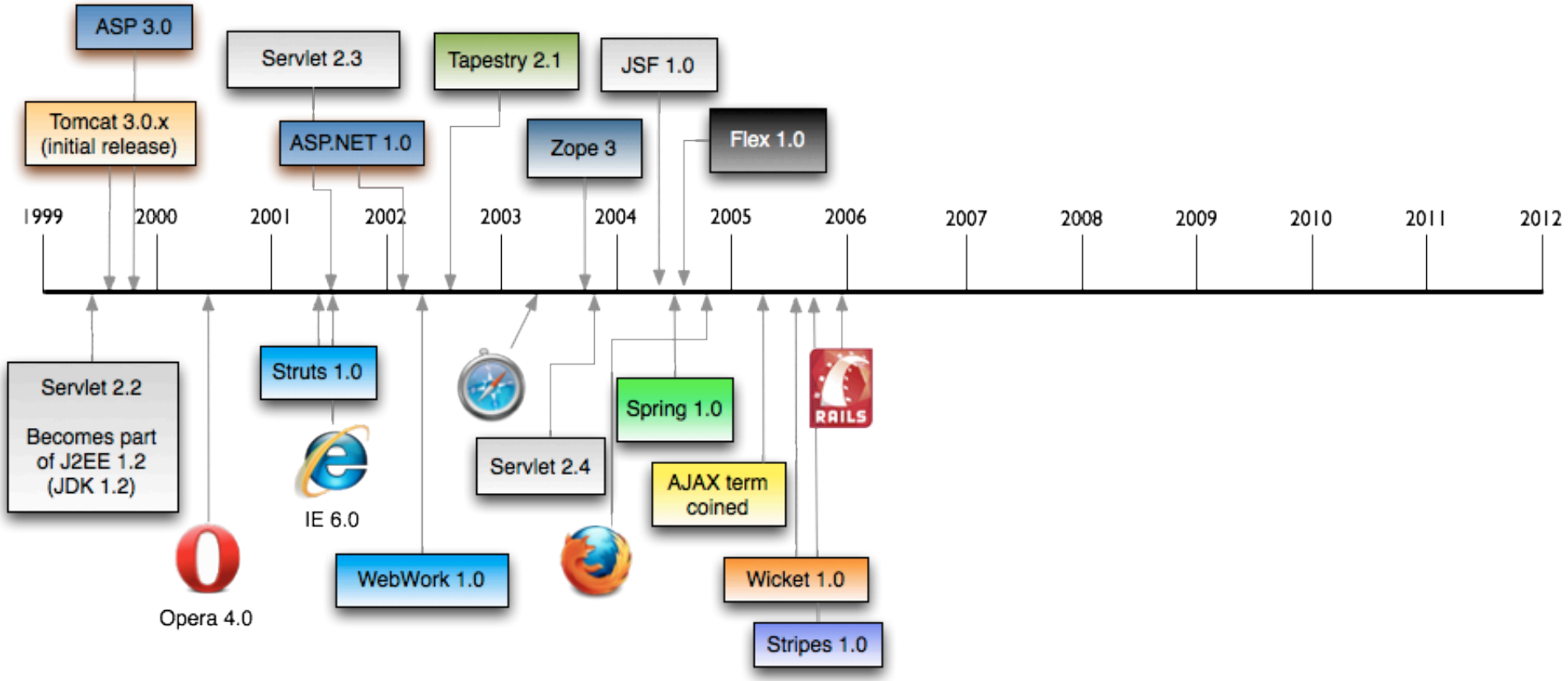
Source: M. Raible (2010) The future of web frameworks

HISTORY OF WEB FRAMEWORKS (1999-2012)



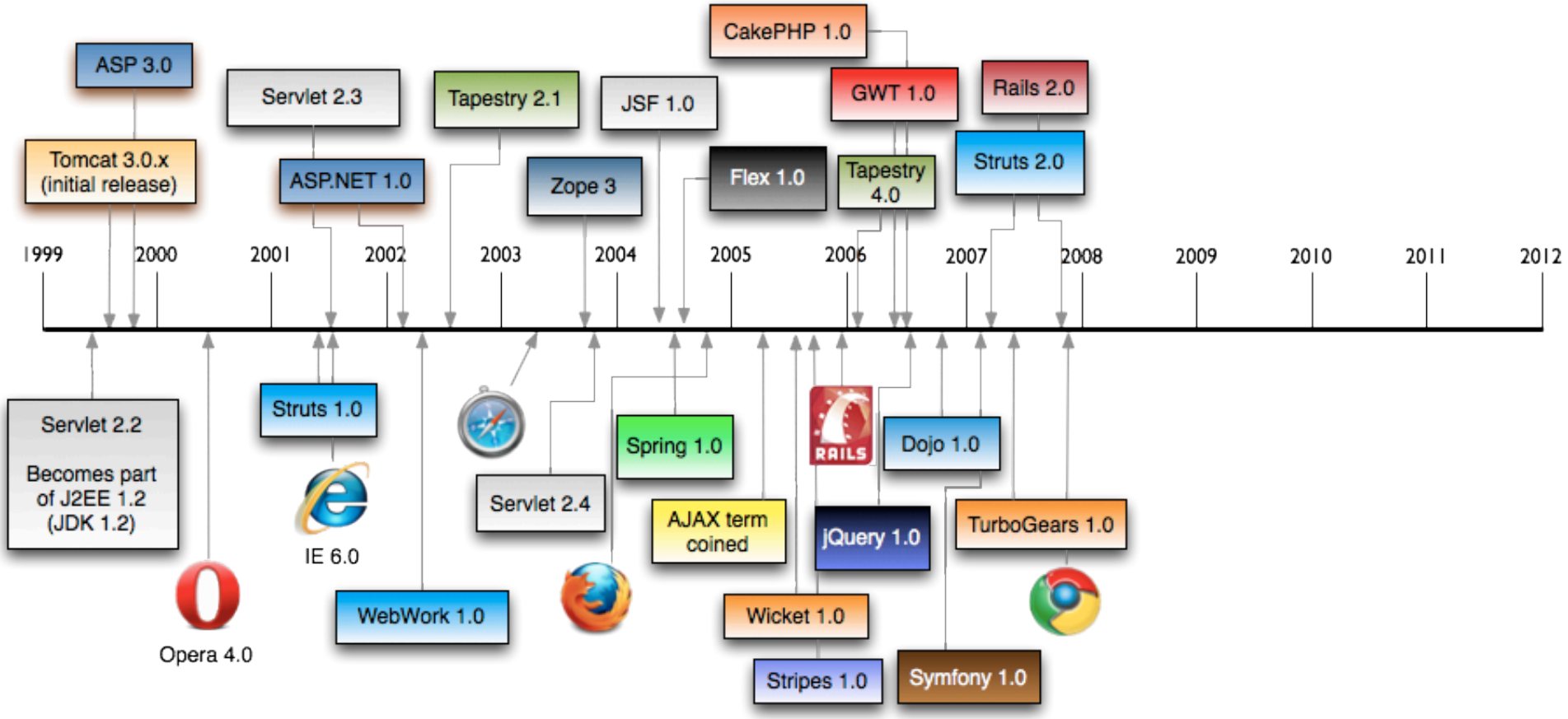
Source: M. Raible (2010) The future of web frameworks

HISTORY OF WEB FRAMEWORKS (1999-2012)

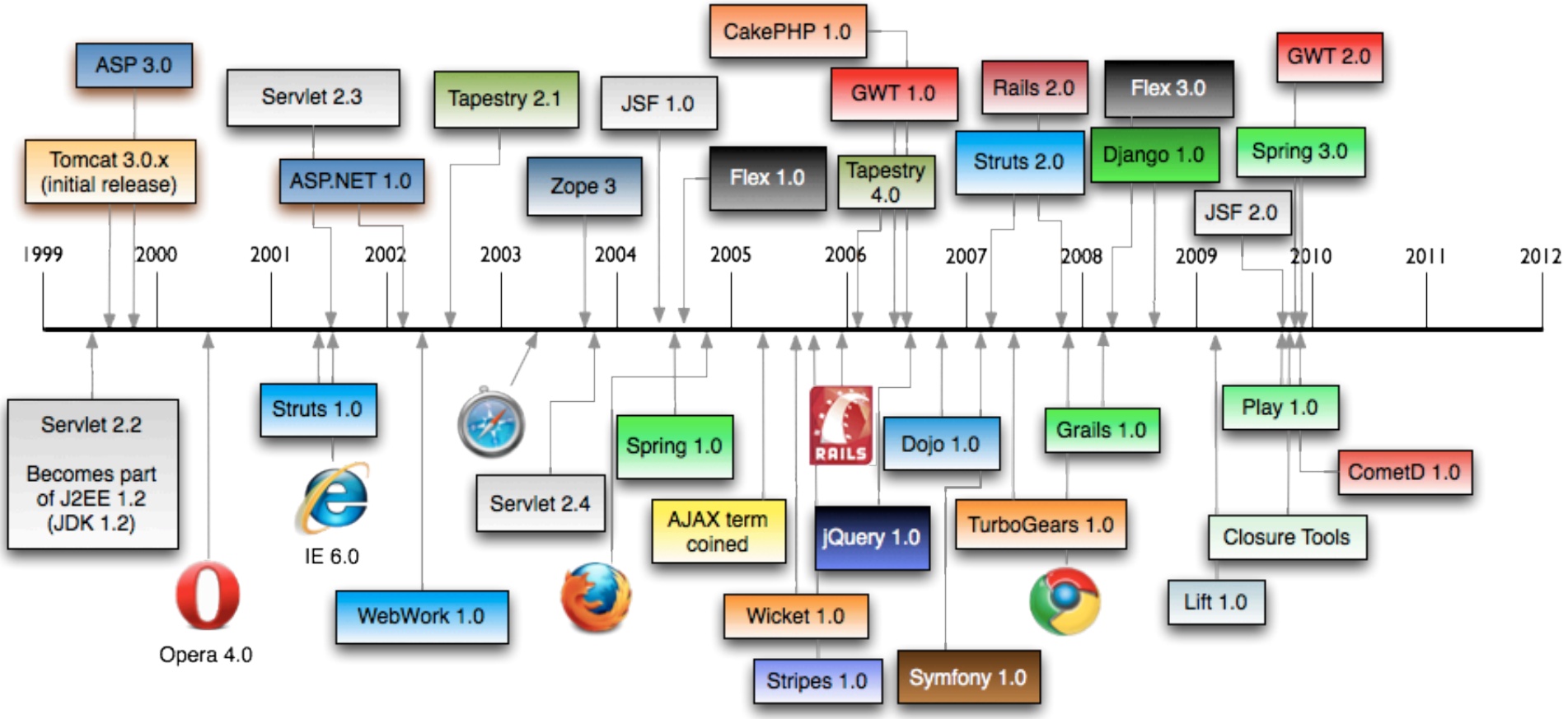


Source: M. Raible (2010) The future of web frameworks

HISTORY OF WEB FRAMEWORKS (1999-2012)



HISTORY OF WEB FRAMEWORKS (1999-2012)



Source: M. Raible (2010) The future of web frameworks

SOME SUSTAINABLE FRAMEWORKS …

- Ruby on Rails is an open source web Framework

- Grails is the premier dynamic language web framework for the JVM

- jQuery

- AngularJS
- …

Stable release of all this frameworks under 2016

ARCHITECTURE

- Model View Controller (MVC)
  - MVC was first described in 1979 by Trygve Reenskaug, then working on Smalltalk at Xerox PARC.

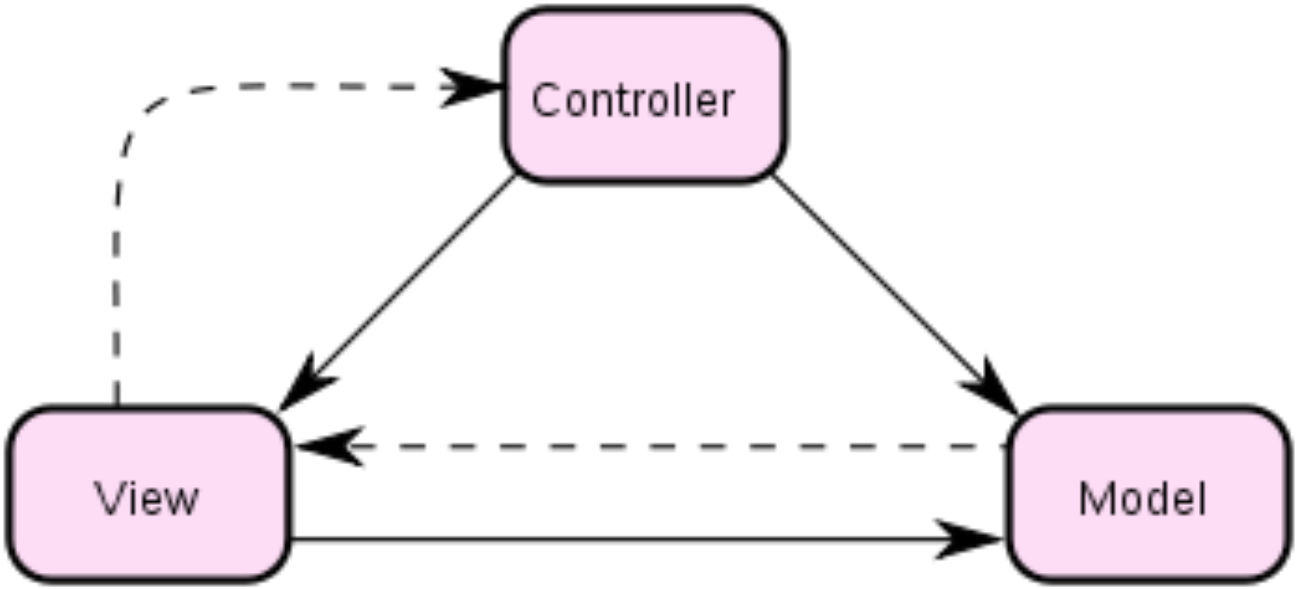
MVC

- Many web application frameworks are based on Model View Controller (MVC)
- Architecture perspective:
  - request-based,
  - component-based,
  - hybrid,
  - meta, and
  - RIA-based

- Content management Systems
  - Higher-layer web application frameworks: e.g. Drupal

MODEL VIEW CONTROLLER (MVC)

- Architectural pattern to separate the data model and business rules from user interface.



APPLICATION PROGRAMMING INTERFACE (API)

- "is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API."
- It serves as an interface between different software programs and facilitates their interaction
- API:
  - General - e.g., APIs bundled in libraries of a programing language,
  - Specific - addressing specific problem for e.g. Java API for XML Web Services,
  - Language dependent - for a particular programing language,
  - Language independent -for several programming languages.

WHAT ABOUT WEB APIs?

- Used in the context of Web Development
- Synonym for web service based on Representational State Transfer (RESTful) services.
- A Web APIs is
  - defined set of Hypertext Transfer Protocol (HTTP) request messages, and
  - response messages in XML or JSON format.
- Web APIs allow the combination of multiple services into new applications known as **mashups**.
- They are used to share content

SHARE CONTENT EASILY WITH WEB APIs

- Publishing APIs allowed to create an open architecture for sharing content between different applications
- For example:
  - Photos: Flickr, Facebook,
  - Content can be dynamically posted: Twitter, Facebook
  - Video content can be embedded: Youtube…
  - User information can be shared: Open Social platform, *is a set of common application programming interfaces (APIs) for web-based social network applications. Initially developed by Google along with MySpace and a number of other social networks.*

HOW TO USE THE API

- Example with Flickr API
- Signup for Flickr API key
  - api_key
- RESTful methods
  - http://api.flickr.com/services/rest/?

  - method=flickr.photos.getRecent&api_key= =**[your api key here]**&per_page=10

HOW TO USE THE API

- Flickr sent the data in XML format (by default)

https://api.flickr.com/services/rest/?method=flickr.photos.getRecent&api_key=a89dbfbfbfcdfe74c9fea26ef16d2d81&per_page=10

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<rsp stat="ok">
  <photos page="1" pages="100" perpage="10" total="1000">
    <photo id="25161730514" owner="123880967@N05" secret="16cd2e91ec" server="1515" farm="2" title="" ispublic="1" isfriend="0" isfamily="0"/>
    <photo id="25165586393" owner="136640337@N04" secret="8a3cddf4f5" server="1720" farm="2" title="" ispublic="1" isfriend="0" isfamily="0"/>
    <photo id="25165587003" owner="8053324@N04" secret="5c57b4f790" server="1650" farm="2" title="carneval_rethymno_2016_119" ispublic="1"
    isfriend="0" isfamily="0"/>
    <photo id="25165587803" owner="8542563@N08" secret="2b1f3586e6" server="1594" farm="2" title="Avila" ispublic="1" isfriend="0" isfamily="0"/>
    <photo id="25491523030" owner="7451276@N08" secret="af23b46483" server="1671" farm="2" title="" ispublic="1" isfriend="0" isfamily="0"/>
    <photo id="25671223852" owner="40434076@N08" secret="0e31b1b7d9" server="1554" farm="2" title="P1610738-Edit-Edit" ispublic="1" isfriend="0"
    isfamily="0"/>
    <photo id="25697122441" owner="29375681@N00" secret="a51d5b7e1f" server="1643" farm="2" title="Some call it work (huh) #mindspace #telaviv
    #Techstars" ispublic="1" isfriend="0" isfamily="0"/>
    <photo id="25766135316" owner="137837759@N07" secret="7e00c184f6" server="1665" farm="2" title="Untitled" ispublic="1" isfriend="0"
    isfamily="0"/>
    <photo id="25766135516" owner="137966163@N03" secret="acff5fff77" server="1695" farm="2" title="A lovely day in Amsterdam. Zoo and Micropia
    museum, both awesome" ispublic="1" isfriend="0" isfamily="0"/>
    <photo id="25792232495" owner="138352935@N05" secret="0a24fb889d" server="1587" farm="2" title="Bathroom renovation in Watford" ispublic="1"
    isfriend="0" isfamily="0"/>
  </photos>
</rsp>
```

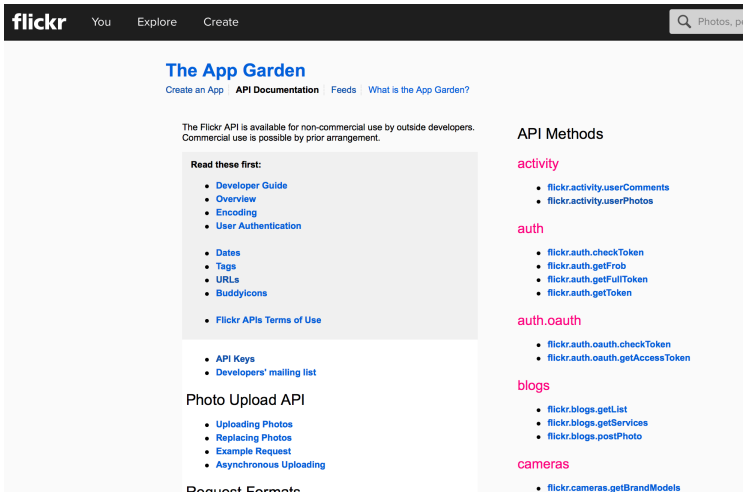

- For retrieving the data in JSON format, add:
  - **&format=json**

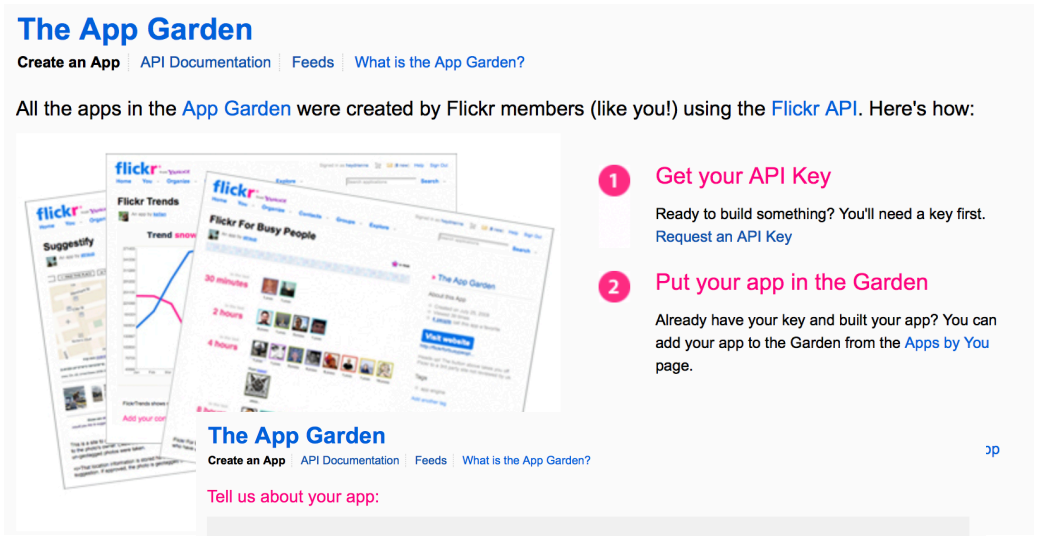
https://api.flickr.com/services/rest/?method=flickr.photos.getRecent&api_key=a89dbfbfbfcdfe74c9fea26ef16d2d81&per_page=10&format=json

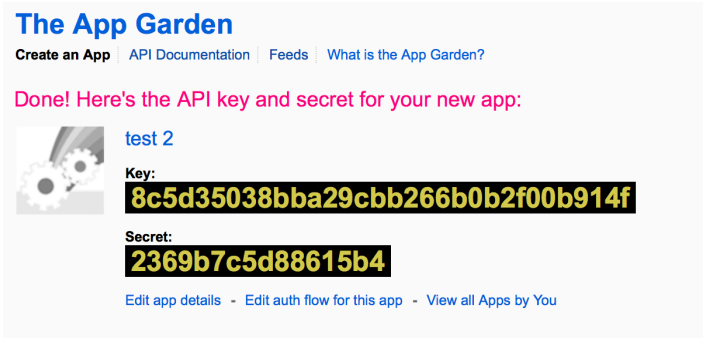

```
jsonFlickrApi({"photos":{"page":1,"pages":100,"perpage":10,"total":1000,"photo":
[{"id":"25165593383","owner":"126056148@N03","secret":"42ca90971b","server":"1486","farm":2,"title":"ABJ_7286.jpg","ispublic":1,"isfriend":0,"isfa
mily":0},{"id":"25165594543","owner":"27099493@N04","secret":"7d8cefcf23","server":"1608","farm":2,"title":"A shot from this weekend. Spring is
here! #spring #knoxville #tennessee #knoxvilletn #thankspunxsutawneyphil #smellingdandelions #dandelion #sunny #sunshine #backyard #dog #furbaby
#dogsofinstagram #pitbull #pitbullsofinstagram","ispublic":1,"isfriend":0,"isfamily":0},
{"id":"25491529900","owner":"136640337@N04","secret":"50f8b8b1db","server":"1548","farm":2,"title":"","ispublic":1,"isfriend":0,"isfamily":0},
{"id":"25491530110","owner":"8053324@N04","secret":"2d320fdffa","server":"1682","farm":2,"title":"carneval_rethymno_2016_120","ispublic":1,"isfrie
nd":0,"isfamily":0},
{"id":"25491530340","owner":"36376325@N06","secret":"b60088c2b5","server":"1582","farm":2,"title":"15","ispublic":1,"isfriend":0,"isfamily":0},
{"id":"25766142706","owner":"39226163@N05","secret":"be6f819bf8","server":"1507","farm":2,"title":"DSC_7628","ispublic":1,"isfriend":0,"isfamily":
0},{"id":"25792238615","owner":"137622888@N06","secret":"8c5a6444e1","server":"1616","farm":2,"title":"Under 16 V
Navan","ispublic":1,"isfriend":0,"isfamily":0},
{"id":"25792239945","owner":"135970890@N03","secret":"0a26a5535a","server":"1456","farm":2,"title":"#unsplush
#photocandyapp","ispublic":1,"isfriend":0,"isfamily":0},
{"id":"25792240085","owner":"60875441@N06","secret":"ced3649962","server":"1663","farm":2,"title":"_DSC4487-Amorphis,
Bakga\u030arden.jpg","ispublic":1,"isfriend":0,"isfamily":0},
{"id":"25792240275","owner":"127784217@N02","secret":"0cc20e0a9f","server":"1506","farm":2,"title":"Printable Any Year Personalized Chalkboard
Inspired Birthday Invitation Digital File","ispublic":1,"isfriend":0,"isfamily":0}]},"stat":"ok"})
```

FLICKR API – SIGN UP

- Apply for API Key: https://www.flickr.com/services/api/







FLICKR API EXAMPLE

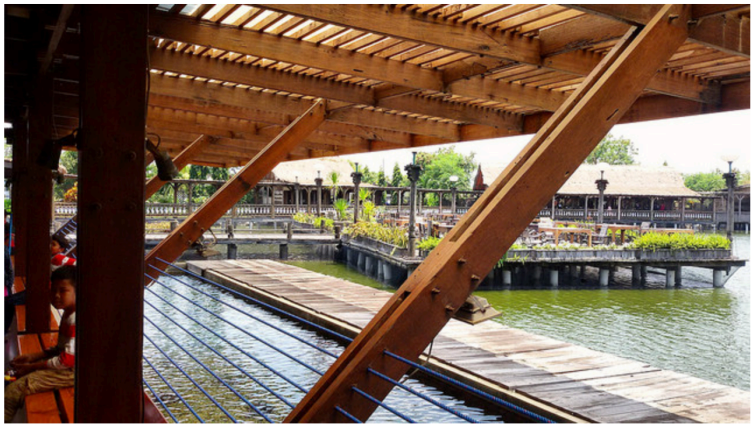
- You can use:
  - JQuery + JSON + flickr API

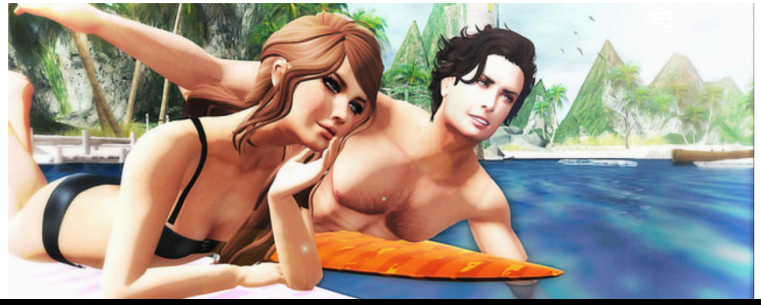
Select size of photos (in pixels) you want them to be displayed

Square [75X75] | Large Square [150X150] | Thumbnail | Small | Medium | Original

Hit This button to fetch photos

Fetch Recent Photos





http://www.w3resource.com/API/flickr/tutorial.php

GOOGLE VISUALIZATION API + MAPS API

- Simple example shows:
  - University locations in the map and the table (that interact with each other)
  - Column charts displaying the number of students/staff
- Creating a mashup using:
  - Google Visualization API
    - Table
    - Charts
    - Maps
  - Charts connected to Google spreadsheets
  - Data table and maps using local data

```
<!--Load the Google AJAX API. -->
<!--Google maps API.-->
<!--Load the Visualization API package.-->
```

SEMI-COMPLETE EXAMPLE

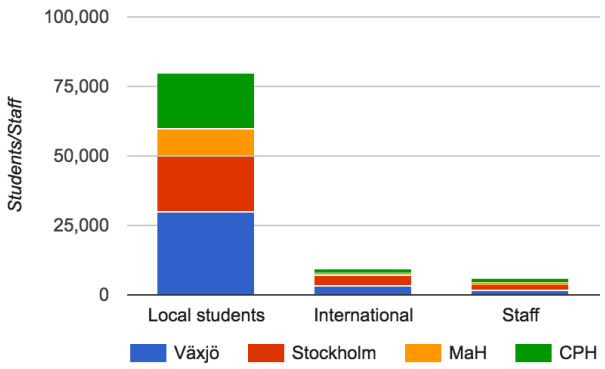
```
<!-- Load the Google AJAX API. -->
      <script type="text/javascript" src="http://www.google.com/jsapi"></script>
<!--Google maps API-->
     <script src="http://maps.google.com/maps?file=api&v=2&api;key=ABCDEF”
type="text/javascript"></script>    <script type="text/javascript">
//Load the Visualization API package.
    google.load('visualization', '1’, {'packages': ['table', 'map', 'corechart']});

//with setOnLoadCallback nothing starts before everything is ready
    google.setOnLoadCallback(initialize);
    function initialize() {
// remote dataset in spreadsheet
   var query = new
google.visualization.Query('https://spreadsheets.google.com/pub?key=0ArHcBZKhLxH-
dC1yUWlQTTRXcUhWX1Y2MGVweFNfWlE');
     query.send(draw);}
//check the error
    function draw(response) {if (response.isError()) {
       alert('Error in query');        }
//get data from spreadsheet
    var uniData = response.getDataTable();
//draw chart
 var chart = new google.visualization.LineChart(document.getElementById('chart_div'));
chart.draw(uniData, {'isStacked': true, 'legend': 'bottom','vAxis': {'title':
'Students/Staff'}});
...
</script>
```

THE SAMPLE

- maps-charts-spreadsheet_universities.html
  - Consuming Google Visualization API and Spreadsheet data
    - Charts
    - Maps
    - Table

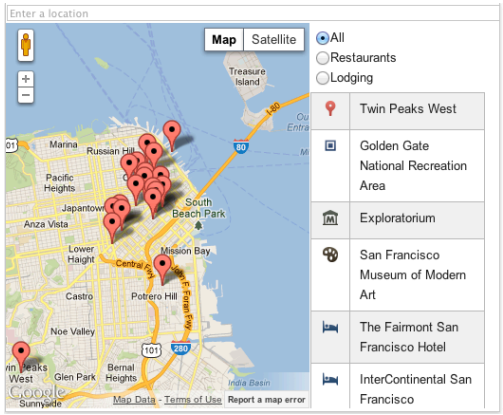
City	University	Campus
Vaxjo, Sweden	Linnaeus University	✓
Stockholm, Sweden	KTH	✗
Malmo, Sweden	MaH	✓
Copenhagen, Denmark	ITU	✗



WHAT YOU CAN DO WITH GOOGLE MAPS API TODAY

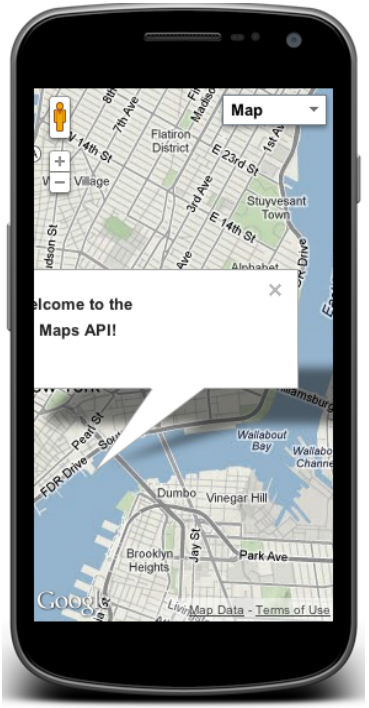
- Build location-based apps
  - Use Google tools and services to build innovative location-based apps.
- Build maps for mobile apps
  - Build high performance apps that work on multiple mobile devices.
- Visualize Geospatial Data
  - Create 3D images with the Earth API, heat maps, and more.
- Customize your maps
  - Create customized maps that highlight your data, imagery, and brand.

BUILD LOCATION-BASED APPS



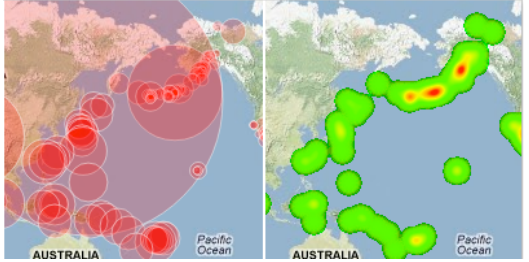
- Connect people with places
  - Google Places API, e.g.: nearby places
  - Google Places Autocomplete API, e.g. suggestions as user types
- Work with a powerful database
  - Google Places API, over 50 million businesses and points of interest
- Get your users from A to B
  - Distance Matrix API, e.g.: find the best driving routes and the time it takes to get to their destination

BUILD MAPS FOR MOBILE APPS



- Build native Android apps
  - Google Maps Android API
- Native iOS SDK
  - Google Maps SDK for iOS
- Cross platform apps
  - With the Google Maps JavaScript API v3, you can create HTML5 applications that can be reused across both desktop and mobile platforms.
- Add Places context to your app
- Embed maps with ease…

VISUALIZE GEOSPATIAL DATA

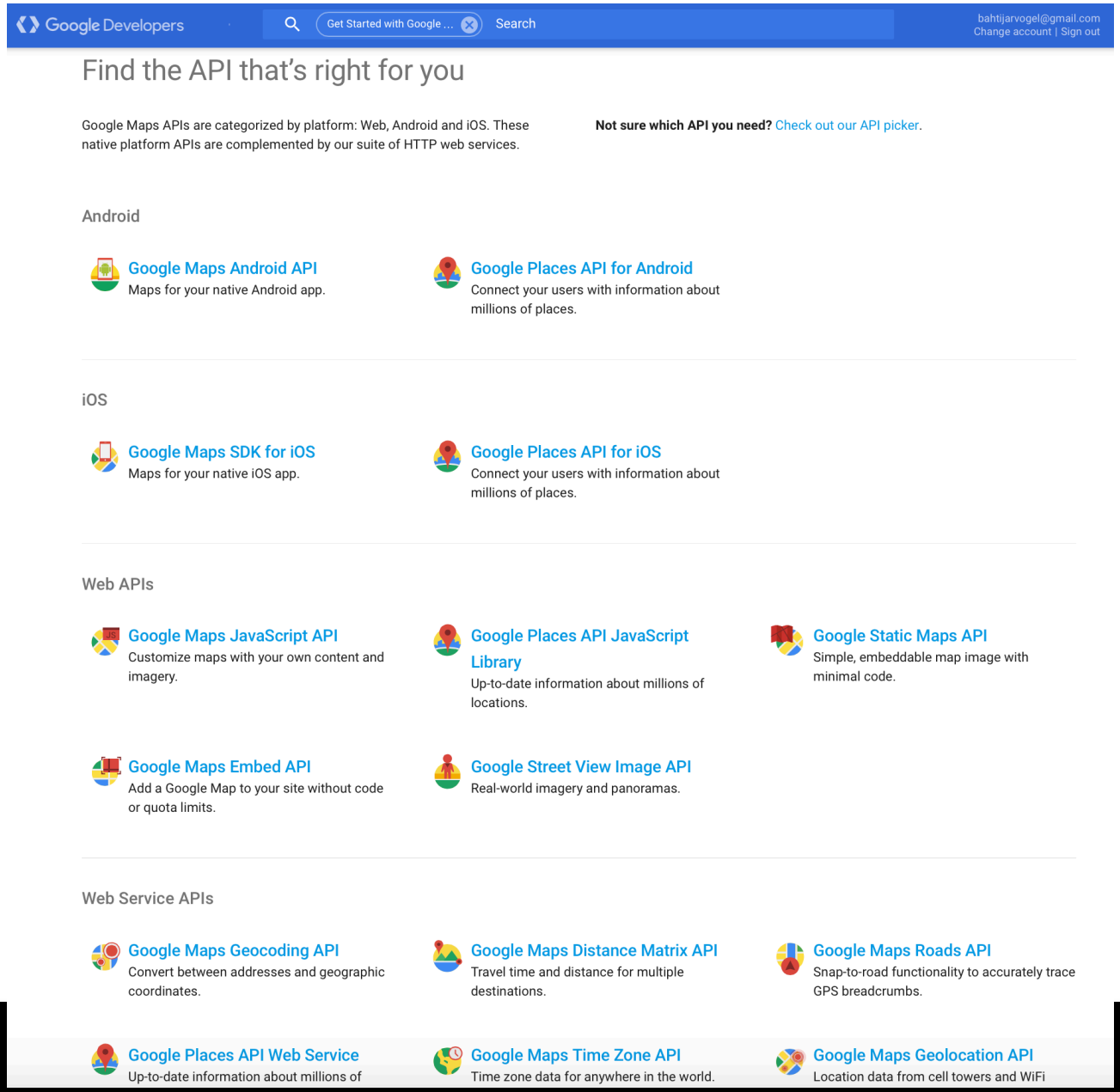


- Create interactive visualizations of your geospatial data.
- Beyond map pins
  - Google Maps JavaScript API v3 provides you with the framework to add a number of unique features and content to a Google Maps interface.
- Visualize your data
  - Google Maps API's visualization options to convey information clearly and beautifully, e.g.: symbols and heatmaps
- Your geospatial data in the cloud
  - Google Maps Engine is a revolutionary geospatial solution that lets you publish your mapping data on our secure, cloud-based mapping platform.
- Add 3D imagery to your apps
  - Google Earth API to add impressive 3D buildings and terrain imagery into your applications, e.g.: View Earth layers, Overlay KML, creative interactive fly-overs

CUSTOMIZE YOUR MAPS

- Customize the look and feel
- With Styled Maps, you can customize the base map layers to fit your needs. Change the base map colors to match your brand, draw attention to your data, or remove unwanted map features.
- Display your own Street View imagery
- you can access our Street View global imagery database or add your own images (such as the inside of a building)
- Create custom routing options
  - Directions API, you can create the most suitable routing options for your users. Implement "drag and drop directions”

GOOGLE MAPS API DOCUMENTATION



NOW SOME INTRODUCTION TO HTML5

OUTLINE

- Introduction: The history of HTML5
- Who is developing HTML5
- A New Vision
- HTML5: Features – APIs
- What's New in HTML5
- Examples
- Summary

NOW SOME INTRODUCTION TO HTML5

1. New Doctype

Still using that pesky, impossible-to-memorize XHTML doctype?

view plain copy to clipboard print ?

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2.     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

If so, why? Switch to the new HTML5 doctype. You'll live longer — as Douglas Quaid might say.

view plain copy to clipboard print ?

```
1. <!DOCTYPE html>
```

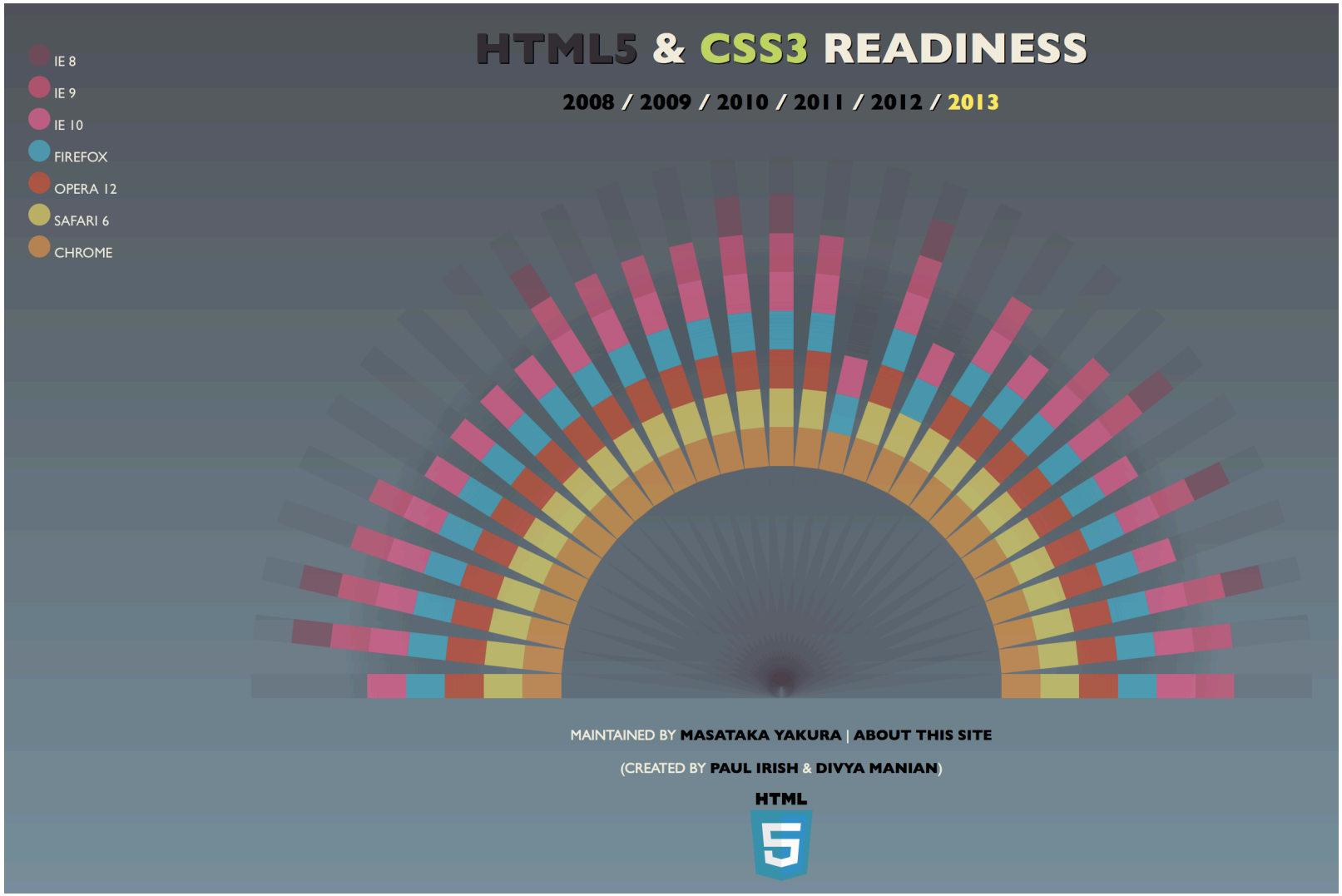
INTRODUCTION: THE HISTORY OF HTML5

- Hyper Text Markup Language (HTML)
- 1993 published as an internet draft.
- ‘90s versions 2.0, 3.2 and 4.0 .
- 1999, version 4.01 W3Consortium
- Web Hypertext Application Working Group (WHATWG) in 2004 – HTML5 specifications.
- W3C involved in 2006, first working draft is published in 2008.

INTRODUCTION: THE HISTORY OF HTML5

- HTML5 brand new, but won't be ready for ten more years
- Candidate recommendation 17 Dec, 2012, and 2022 proposed recommendation
  - http://www.w3.org/2012/12/html5-cr#access
  - http://www.w3.org/TR/2012/CR-html5-20121217/
- HTML5 is rapidly evolving to address real and practical improvements to the web platform.
- Important: browser vendors are actively adding the support

HTML5 & CSS3 READINESS



http://html5readiness.com/

WHO IS DEVELOPING HTML5

- Web Hypertext Application Technology Working Group (WHATWG): Founded in 2004 by individuals working for browser vendors Apple, Mozilla, Google, and Opera, WHATWG develops HTML and APIs for web application development and provides open collaboration of browser vendors and other interested parties.

- World Wide Web Consortium (W3C): The W3C contains the HTML working group that is currently charged with delivering their HTML5 specification.

- Internet Engineering Task Force (IETF): This task force contains the groups responsible for Internet protocols such as HTTP. HTML5 defines a new WebSocket API that relies on a new WebSocket protocol, which is under development in an IETF working group.

A NEW VISION

- Various design principles, spelled out in the WHATWG specification
  - **Compatibility**
    - Google discovered that millions of pages use common ID and Class names for DIV tags: DIV id="header" to markup header content, why not use <header> element
  - **Utility**
    - id="html5" or id=html5 or ID="html5"
    - a step for separation of presentation and content
  - **Interoperability**
    - Native browser ability instead of complex JavaScript code
    - A new, simplified DOCTYPE
    - A new, simplified character set declaration
    - Powerful yet simple HTML5 APIs
  - **Universal access**
    - Accessibility,
    - Media independence,
    - Support for all world languages.

A PLUGIN–FREE PARADIGM

- HTML5 provides native support for many features that used to be possible only with plugins or complex hacks
  - a native drawing API,
  - native video,
  - native sockets, and so on.

- Plugins, of course, present many problems:
  - Plugins cannot always be installed.
  - Plugins can be disabled or blocked (for example, the Apple iPad does not ship with a Flash plugin).
  - Plugins are a separate attack vector.
  - Plugins are difficult to integrate with the rest of an HTML document (because of plugin boundaries, clipping, and transparency issues).

HTML5: FEATURES - APIS

- Canvas (2D and 3D)
- Cross-document messaging
- Geolocation
- Audio and Video
- Forms
- MathML
- Microdata
- Server-Sent events
- Scalable Vector Graphics (SVG)
- WebSocket API and protocol
- Web origin concept
- Web storage
- Indexed database
- Application Cache (Offline Web Apps) Web Workers
- Drag and Drop
- XMLHttpRequest

WHAT'S NEW IN HTML5

- **HTML4 DOCTYPEs:**

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

- **HTML5:**

```
<!DOCTYPE html>
```

- **HTML4** the character set declaration:

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

- Now, with **HTML5** it is:

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

NEW AND DEPRECATED ELEMENTS

- HTML5 introduces many new markup elements

•*HTML5 Content Types*

Content Type	Description
Embedded	Content that imports other resources into the document, for example `audio`, `video`, `canvas`, and `iframe`
Flow	Elements used in the body of documents and applications, for example `form`, `h1`, and `small`
Heading	Section headers, for example `h1`, `h2`, and `hgroup`
Interactive	Content that users interact with, for example `audio` or `video` `controls`, `button`, and `textarea`
Metadata	Elements—commonly found in the `head` section— that set up the presentation or behavior of the rest of the document, for example `script`, `style`, and `title`
Phrasing	Text and text markup elements, for example `mark`, `kbd`, `sub`, and `sup`
Sectioning	Elements that define sections in the document, for example `article`, `aside`, and `title`

SEMANTIC MARKUP

- HTML5 defines a new semantic markup to describe an element's content.
- No immediate benefits

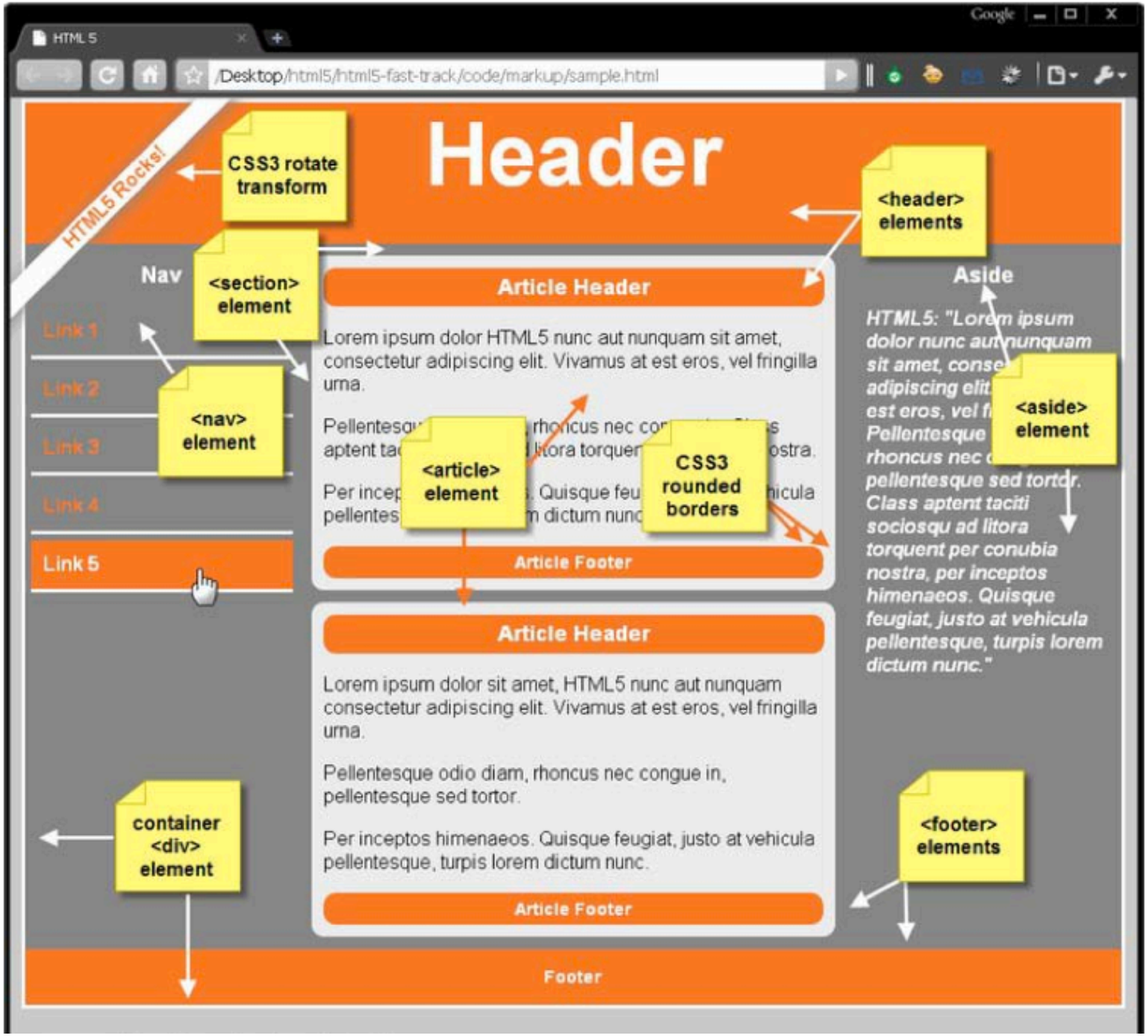
- *New sectioning HTML5 elements*

Sectioning Element	Description
header	Header content (for a page or a section of the page)
footer	Footer content (for a page or a section of the page)
section	A section in a web page
article	Independent article content
aside	Related content or pull quotes
nav	Navigational aids

AN EXAMPLE OF HTML5 PAGE

- Check the code file under examples/intro (sample.html, html5.css)
- New DOCTYPE,
- Character set, and
- Semantic markup elements
- Cascading Style Sheets 3 (CSS3)
    - CSS3 features:
        - rounded corners (border-radius) and
        - rotate transformations (transform: rotate();).
        - CSS3 is still under development and is modularized with subspecifications: *transformation, animation,* and *transition* are all areas that are in separate subspecifications.

        - To display rounded corners, gradients, shadows, and transformations, it is currently necessary to use prefixes such as -moz- (for Mozilla), o- (for Opera), -webkit- (for WebKit- based browsers such as Safari and Chrome), and -ms- (for Internet Explorer) in your declarations.


HTML 5
Google
/Desktop/html5/html5-fast-track/code/markup/sample.html
HTML5 Rocks!
CSS3 rotate transform
Header
<header> elements
Nav
<section> element
Article Header
Aside
Link 1
Link 2
Link 3
Link 4
Link 5
<nav> element
Lorem ipsum dolor HTML5 nunc aut nunquam sit amet, consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna.
<article> element
CSS3 rounded borders
Article Footer
HTML5: "Lorem ipsum dolor nunc aut nunquam sit amet, conse adipiscing elit est eros, vel f Pellentesque rhoncus nec c pellentesque sed tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Quisque feugiat, justo at vehicula pellentesque, turpis lorem dictum nunc."
<aside> element
Article Header
Lorem ipsum dolor sit amet, HTML5 nunc aut nunquam consectetur adipiscing elit. Vivamus at est eros, vel fringilla urna.
Pellentesque odio diam, rhoncus nec congue in, pellentesque sed tortor.
Per inceptos himenaeos. Quisque feugiat, justo at vehicula pellentesque, turpis lorem dictum nunc.
Article Footer
container element
<footer> elements
Footer

SIMPLIFYING SELECTION USING THE SELECTORS API

- *Previous JavaScript Methods to Find Elements*

Function Description		Example
`getElementById()`	Returns the element with the specified `id` attribute value	`<div id="foo">` `getElementById("foo");`
`getElementsByName()`	Returns all elements whose name attribute has the specified value	`<input type="text" name="foo">` `getElementsByName("foo");`
`getElementsByTagName()`	Return all elements whose tag name matches the specified value	`<input type="text">` `getElementsByTagName("input");`

- *New QuerySelector Methods*

Function Description		Example	Result
`querySelector()`	Return the first element in the page which matches the specified selector rules(s)	`document.querySelector("input.error");`	Return the first input field with a style class of "error"
`querySelectorAll()`	Returns all elements which match the specified rule or rules	`document.querySelectorAll("#results td");`	Return any table cells inside the element with `id` results

EXAMPLE OF USING THE SELECTOR API

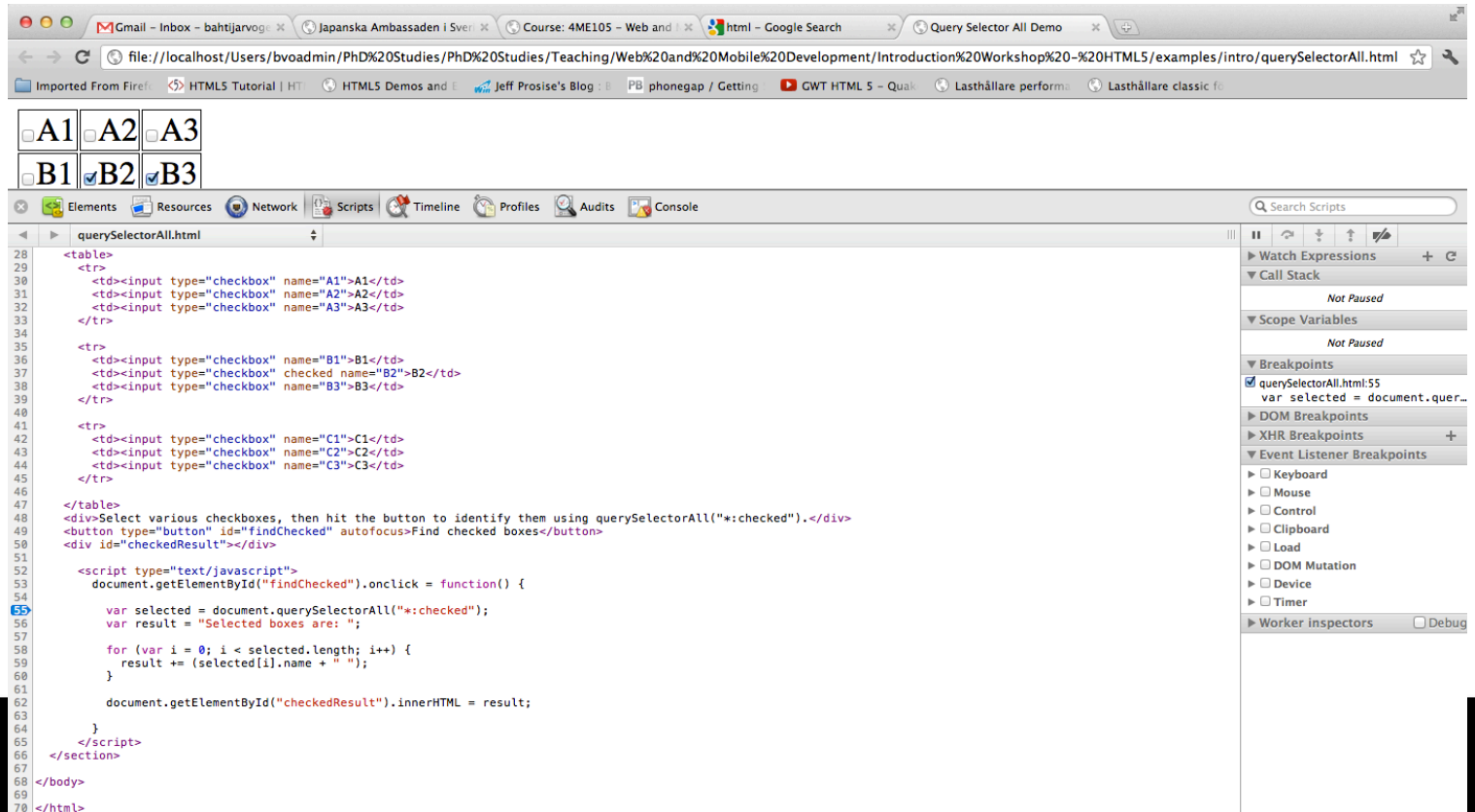
•Check the code file under examples/intro (querySelector.html; querySelectorAll.html)

- Selector API makes it easy to select sections of the document
- Find whichever cell of a table currently had the mouse hovering over it

- querySelector() - the first element that matches either rule is selected
- querySelectorAll() - any element matching any of the listed rules is returned,
- multiple rules are comma- separated.

JAVASCRIPT LOGGING AND DEBUGGING

- Not technically a feature of HTML5, JavaScript logging and in-browser debugging tools have been improved greatly over the past few years.
- Firefox add-on, Firebug.
- Built-in development tools:
  - Safari's Web Inspector, Google's Chrome Developer Tools, Internet Explorer's Developer Tools, and Opera's Dragonfly.




```
<table>
  <tr>
    <td><input type="checkbox" name="A1">A1</td>
    <td><input type="checkbox" name="A2">A2</td>
    <td><input type="checkbox" name="A3">A3</td>
  </tr>

  <tr>
    <td><input type="checkbox" name="B1">B1</td>
    <td><input type="checkbox" checked name="B2">B2</td>
    <td><input type="checkbox" name="B3">B3</td>
  </tr>

  <tr>
    <td><input type="checkbox" name="C1">C1</td>
    <td><input type="checkbox" name="C2">C2</td>
    <td><input type="checkbox" name="C3">C3</td>
  </tr>

</table>
<div>Select various checkboxes, then hit the button to identify them using querySelectorAll("*:checked").</div>
<button type="button" id="findChecked" autofocus>Find checked boxes</button>
<div id="checkedResult"></div>

  <script type="text/javascript">
    document.getElementById("findChecked").onclick = function() {

      var selected = document.querySelectorAll("*:checked");
      var result = "Selected boxes are: ";

      for (var i = 0; i < selected.length; i++) {
        result += (selected[i].name + " ");
      }

      document.getElementById("checkedResult").innerHTML = result;

    }
  </script>
</section>

</body>

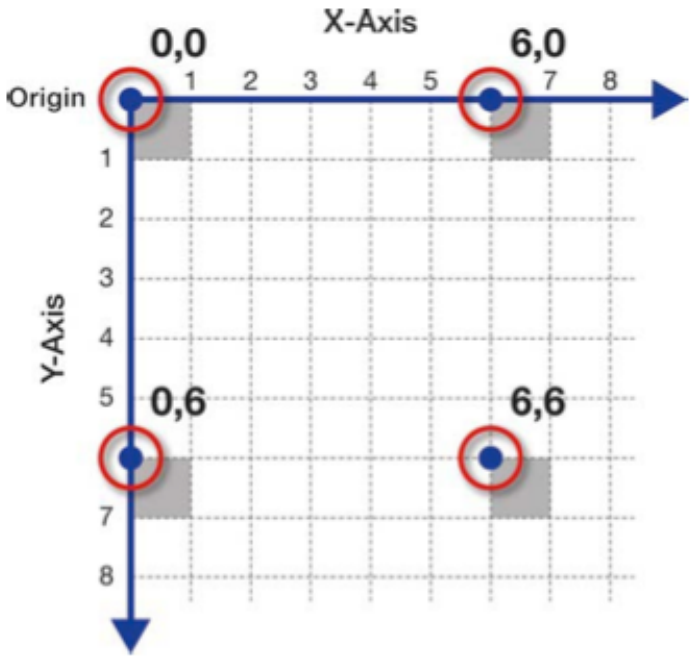
</html>
```

USING THE CANVAS API

- HISTORY: The canvas concept was originally introduced by Apple to be used in Mac OS X WebKit to create dashboard widgets.
- Enables you to dynamically generate and render graphics, charts, images, and animation.
- Before the arrival of canvas, you could only use drawing APIs in a browser through plug-ins such as Adobe plug-ins for Flash and Scalable Vector Graphics (SVG), Vector Markup Language
- Canvas provides simple two-dimensional drawing API
- <canvas></canvas>
- it creates a rectangular area on the page, 300 pixels wide and 150 pixels high by default.
- Use JavaScript to manipulate it
- You can add graphics, lines, and text to it; you can draw on it; and you can even add advanced animations to it.

USING THE CANVAS API

- Canvas Coordinates
- *x and y coordinates on a canvas*



- CSS can be applied to the canvas element itself to add borders, padding, margins, etc.
- Lately, all browser vendors now provide support for HTML5 Canvas

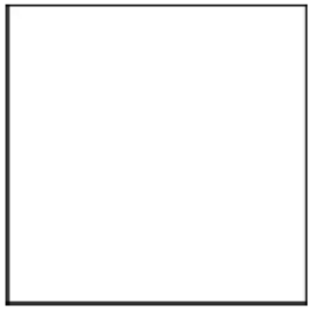
Checking for Browser Support

```
try {
        document.createElement("canvas").getContext("2d");
        document.getElementById("support").innerHTML =
        "HTML5 Canvas is supported in your browser.";
        } catch (e) {
          document.getElementById("support").innerHTML = "HTML5
        Canvas is not supported in your browser.";
    }
```

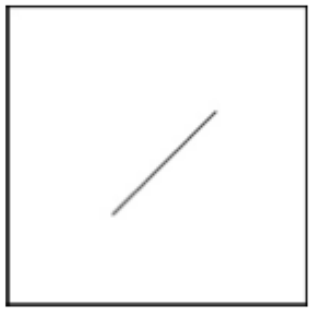
ADDING A CANVAS TO A PAGE

- Check the code file under examples/intro/canvas (trails-diagonal.html;)
- *The Canvas Element*
  <canvas height="200" width="200"></canvas>
- *Canvas Element with a Solid Border*
- <canvas id="diagonal" style="border: 1px solid;" width="200" height="200"> </canvas>

A simple HTML5 canvas element on an HTML page



- *Drawing a diagonal line on a canvas*



Creating a Diagonal Line on a Canvas

```
function drawDiagonal() {
        // Get the canvas element and its drawing context
        var canvas = document.getElementById('diagonal');
        var context = canvas.getContext('2d');

        // Create a path in absolute coordinates
        context.beginPath();
        context.moveTo(70, 140);
        context.lineTo(140, 70);

        // Stroke the line onto the canvas
        context.stroke();
    }
    window.addEventListener("load", drawDiagonal, true);
```

HTML5: AUDIO AND VIDEO

- HTML5: Use audio and video without plugins while providing a common, integrated, and scriptable API.
- Audio and Video Restrictions
- Browser Support for Audio and Video: which browsers support the new audio and video elements is not enough; you also need to know which codecs are supported.

Browser	Codec and Container Support
Chrome	Ogg (Theora and Vorbis) WebM (VP8 and Vorbis) MPEG 4 (H.264 and AAC)
Firefox	Ogg (Theora and Vorbis) WebM (VP8 and Vorbis)
Internet Explorer	MPEG 4 (H.264 and AAC)
Opera	Ogg (Theora and Vorbis) WebM (VP8 and Vorbis)
Safari	MPEG 4 (H.264 and AAC)

Audio and Video Codec and Container Support

HTML5: USING THE AUDIO AND VIDEO API

- HTML5: Use audio and video without plugins while providing a common, integrated, and scriptable API.
- Being part of the native browser environment.
- Checking for Browser Support:

```
var hasVideo =
!!(document.createElement('video').canPlayType);
```

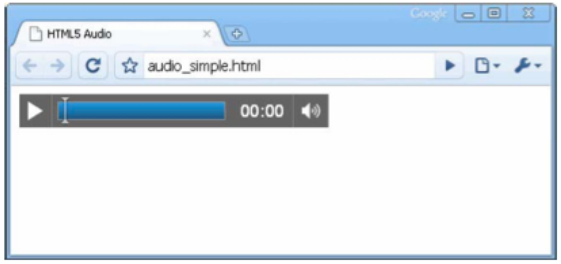
- If you want to display text message for nonsupporting browsers:

```
<video src="video.webm" controls> Your browser
does not support HTML5 video. </video>
```

AUDIO EXAMPLE

•Check the code file under examples/intro/av (audio_simple.html; audio_multisource.html; audio_no_control.html)

- The Basics: Declaring Your Media Element




```
<!DOCTYPE html>
<html>
  <title>HTML5 Audio </title>
  <audio controls src="johann_sebastian_bach_air.ogg">
            An audio clip from Johann Sebastian Bach.
  </audio>
</html>
```

- An Audio Element with Multiple Source Elements

```
<audio controls>
    <source src="johann_sebastian_bach_air.ogg">
    <source src="johann_sebastian_bach_air.mp3">
    An audio clip from Johann Sebastian Bach.
</audio>
```

- Using the Autoplay Attribute

```
<audio autoplay> </audio>
```

WORKING WITH VIDEO

- The HTML5 video element is very similar to the audio element, but with a few extra attributes thrown in.

Additional Video Attributes

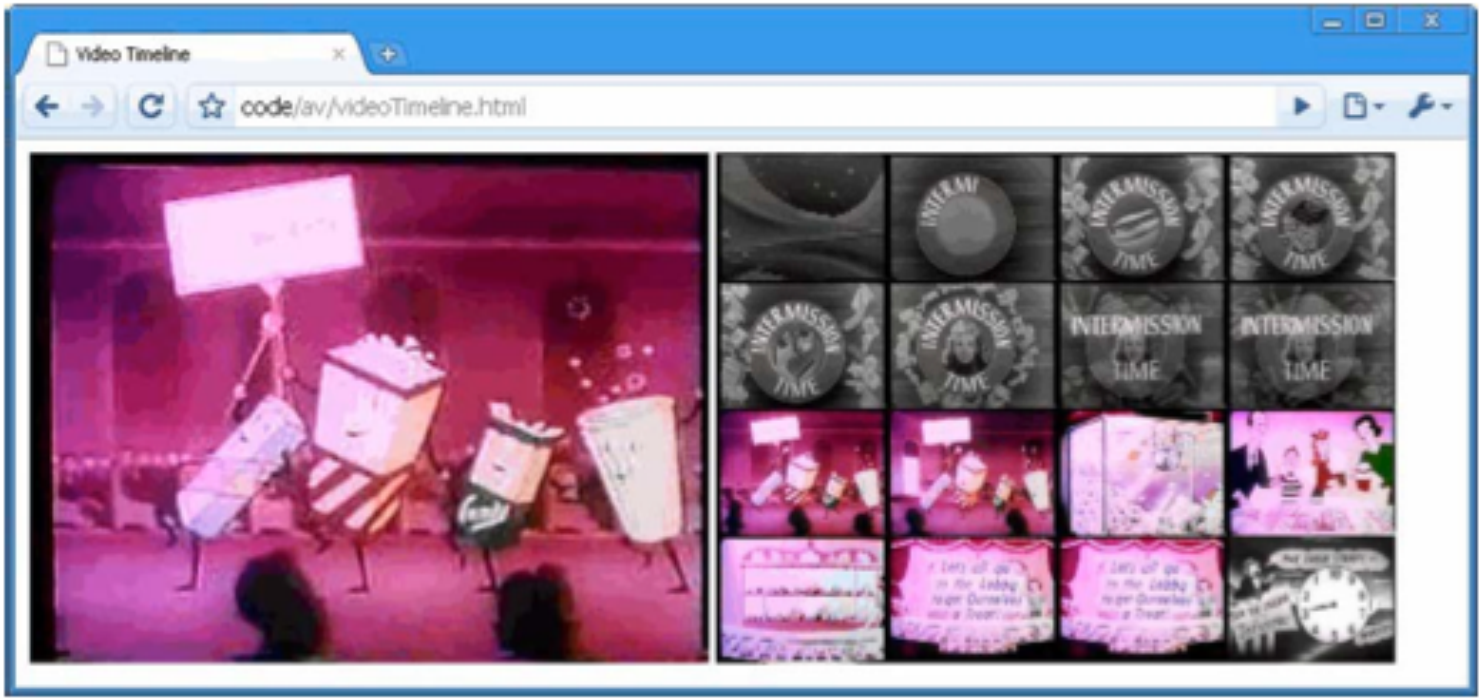
Attribute	Value
`poster`	The URL of an image file used to represent the video content before it has loaded. Think "movie poster." This attribute can be read or altered to change the poster.
`width, height`	Read or set the visual display size. This may cause centering, letterboxing, or pillaring if the set width does not match the size of the video itself.
`videoWidth, videoHeight`	Return the intrinsic or natural width and height of the video. They cannot be set.

- Another key element of video is HTML5 Canvas

CREATING A VIDEO TIMELINE BROWSER

•Check the code file under examples/intro/av (videoTimeline.html;)

- Example shows how a video element can have its frames grabbed and displayed in a dynamic canvas.



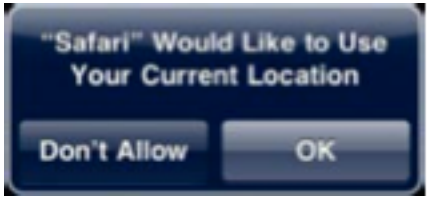
MOUSEOVER VIDEO PLAYBACK

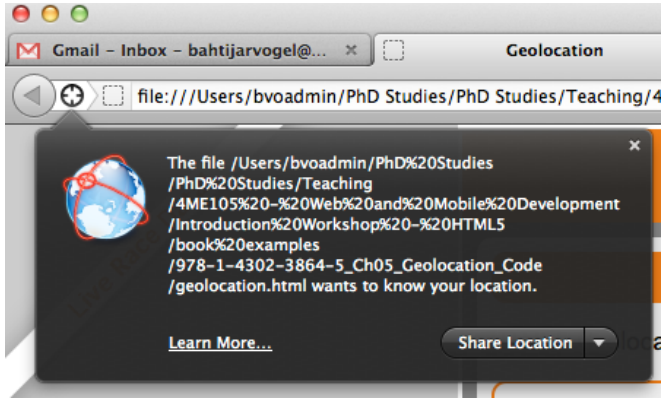
•Check the code file under examples/intro/av (mouseoverVideo.html;)

- Based on the mouse movement over the video trigger the play and pause routines.

USING THE GEOLOCATION API: TRIGGERING THE PRIVACY PROTECTION MECHANISM

- Using the Geolocation API, you can request users to share their location
- You request a position and, if the user agrees, the browser returns location information.
- Triggering the Privacy Protection Mechanism





Checking for Browser Support

```
function loadDemo() {
        if(navigator.geolocation) {
        document.getElementById("support").innerHTML = "Geolocation supported.";
        } else {
        document.getElementById("support").innerHTML = "Geolocation is not supported
in                          your browser.";
        }
}
```

USING THE GEOLOCATION API: A POSITION OBJECT

- A position object:
  - contain coordinates—as the attribute coords—and a timestamp for when the location data was gathered.
- The first three attributes:
  - latitude
  - longitude
  - accuracy
- Other attributes:
  - altitude—the height of the user's location, in meters
  - altitudeAccuracy—once again in meters, or null if no altitude is provided
  - heading—direction of travel, in degrees relative to true north
  - speed—ground speed in meters per second

USING THE GEOLOCATION API: UPDATELOCATION()

- *One-Shot Position Request*

```
void getCurrentPosition(in PositionCallback successCallback,
                        in optional PositionErrorCallback errorCallback,
                        in optional PositionOptions options);
```

The updateLocation() Function

```
function updateLocation(position) {
        var latitude = position.coords.latitude;
        var longitude = position.coords.longitude;
        var accuracy = position.coords.accuracy;
        var timestamp = position.timestamp;
        document.getElementById("latitude").innerHTML = latitude;
        document.getElementById("longitude").innerHTML = longitude;
        document.getElementById(“accuracy”).innerHTML = accuracy
        document.getElementById("timestamp").innerHTML = timestamp;•
}
```

USING THE GEOLOCATION API: ERROR HANDLER

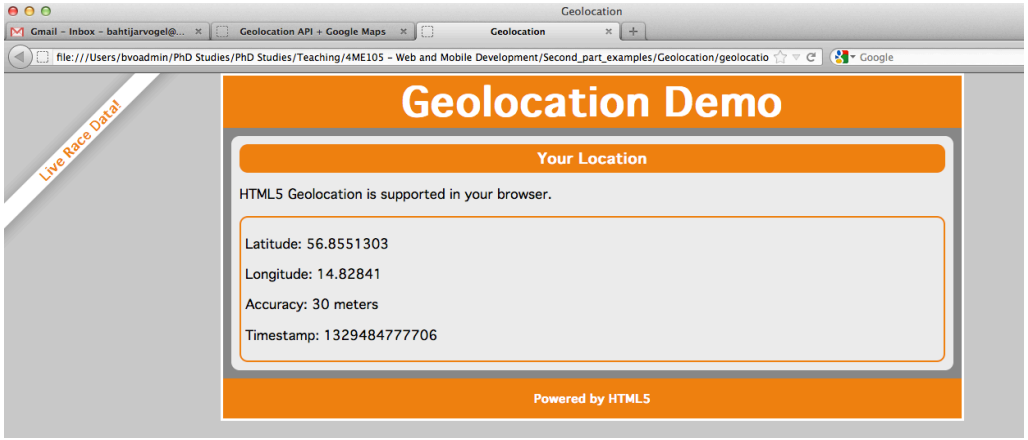
Using an Error Handler

```
function handleLocationError(error) {
        switch(error.code){
        case 0:
          updateStatus("There was an error while retrieving your location: " +
                                    error.message);
        break;
        case 1:
        updateStatus("The user prevented this page from retrieving a location.");
        break;
        case 2:
        updateStatus("The browser was unable to determine your location: " +
                                   error.message);
        break;
        case 3:
        updateStatus("The browser timed out before retrieving the location.");
        break;
        }
}
```

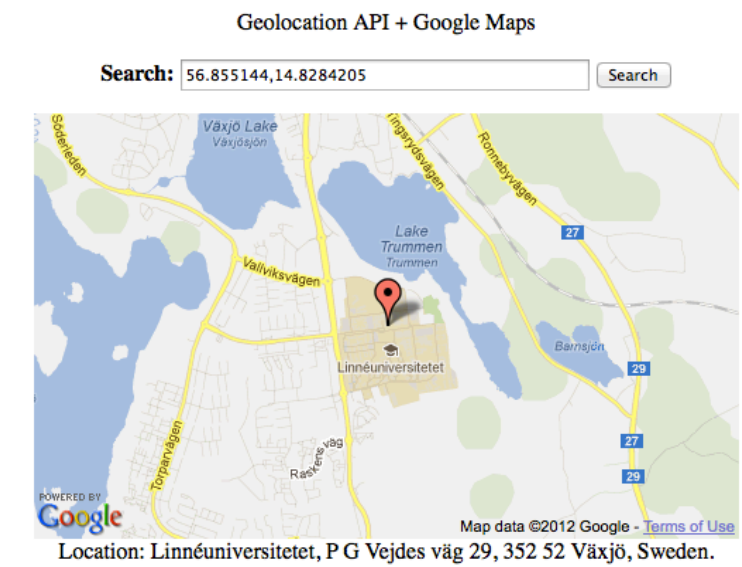
GEOLOCATION EXAMPLE APPLICATION IN ACTION

- Check the code file under examples/Geolocation (geolocation.html, geo-html5.css; geolocation_maps.html)

Geolocation API Demo



Geolocation API + Google Maps



Location: Linnéuniversitetet, P G Vejdes väg 29, 352 52 Växjö, Sweden.

OVERVIEW OF HTML5 FORMS

- Forms should still be encapsulated in a <form> element where the basic submission attributes are set.
- Forms still send the values of the controls to the server when the user or the application programmer submits the page.
- All of the familiar form controls—text fields, radio buttons, check boxes, and so on—are still present and working as before (albeit with some new features).
- Form controls are still fully scriptable for those who wish to write their own modifiers and handlers.
- Functional Forms
  - HTML5 Forms has instead focused on evolving the existing, simple HTML Forms to encompass more types of controls and address the practical limitations that web developers face today.
    - New input types
    - New functions and attributes

EVOLVING INPUT TYPES

- Browser support for HTML5 Forms is growing, but still limited.
- HTML5 markup list maintained at the W3C, new and changed elements : http://dev.w3.org/html5/markup/

New HTML5 Form Elements Appearing in Browsers

Type	Purpose
`tel`	Telephone number
`email`	Email address text field
`url`	Web location URL
`search`	Term to supply to a search engine. For example, the search bar atop a browser.
`range`	Numeric selector within a range of values, typically visualized as a slider
`number`	A field containing a numeric value only

EXAMPLES

<input type="text">

<input type="range" min="18" max="120">

<label for="age">Age</label>
<input id="age" type="range" min="18" max="120"
value="18" onchange="ageDisplay.value=value">
<output id="ageDisplay">18</output>

<input type=”email">

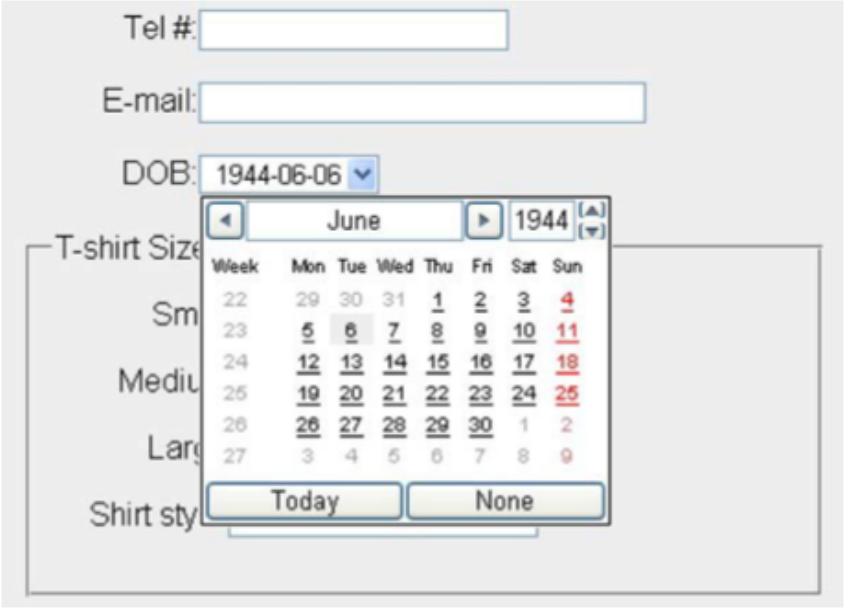
<progress></progress>

<progress value=”30” max=”100”></progress>

HTML5 FORM ELEMENTS

Type	Purpose
`color`	Color selector, which could be represented by a wheel or swatch picker
`datetime`	Full date and time display, including a time zone, as shown in Figure 8-3
`datetime-local`	Date and time display, with no setting or indication for time zones
`time`	Time indicator and selector, with no time zone information
`date`	Selector for calendar date
`week`	Selector for a week within a given year
`month`	Selector for a month within a given year

Display for an input of type datetime (in some browsers only)



ATTRIBUTES

The placeholder attribute

```
<label>Runner: <input name="name" placeholder="First and last name"></label>
```

The autocomplete attribute

<input type="text" name="creditcard" autocomplete="off”>

The autofocus attribute

<input type="search" name="criteria" autofocus>

The spellcheck attribute

<textarea id=”myTextArea” spellcheck=”true”>

The list Attribute and the datalist Element

<datalist id="contactList">

<option value="x@example.com" label="Racer X">

<option value="peter@example.com" label="Peter">

</datalist>

<input type="email" id="contacts" list="contactList">

The min, max and step Attributes

<input id="confidence" name="level" type="range" min="0" max="100" step="5" value="0">

The required Attribute

<input type="text" id="firstname" name="first" required>

VALIDITYSTATE

HTML5 does introduce eight handy ways to enforce correctness on form control entry.

The ValidityState can be accessed from any form control in a browser that supports HTML5 Form validation

Eight possible validity constraints:

valueMissing

```
<input type="text" name="myText" required>
```

typeMismatch

```
<input type="email" name="myEmail">
```

patternMismatch

```
<input type="number" name="creditcardnumber" pattern="[0-9]{16}" title="A credit card number is 16 digits with no spaces or dashes">
```

tooLong

```
<input type="text" name="limitedText" maxLength="140">
```

rangeUnderflow and **rangeOverflow**

```
<input type="range" name="ageCheck" min="18">
<input type="range" name="kidAgeCheck" max="12”>
```

stepMismatch

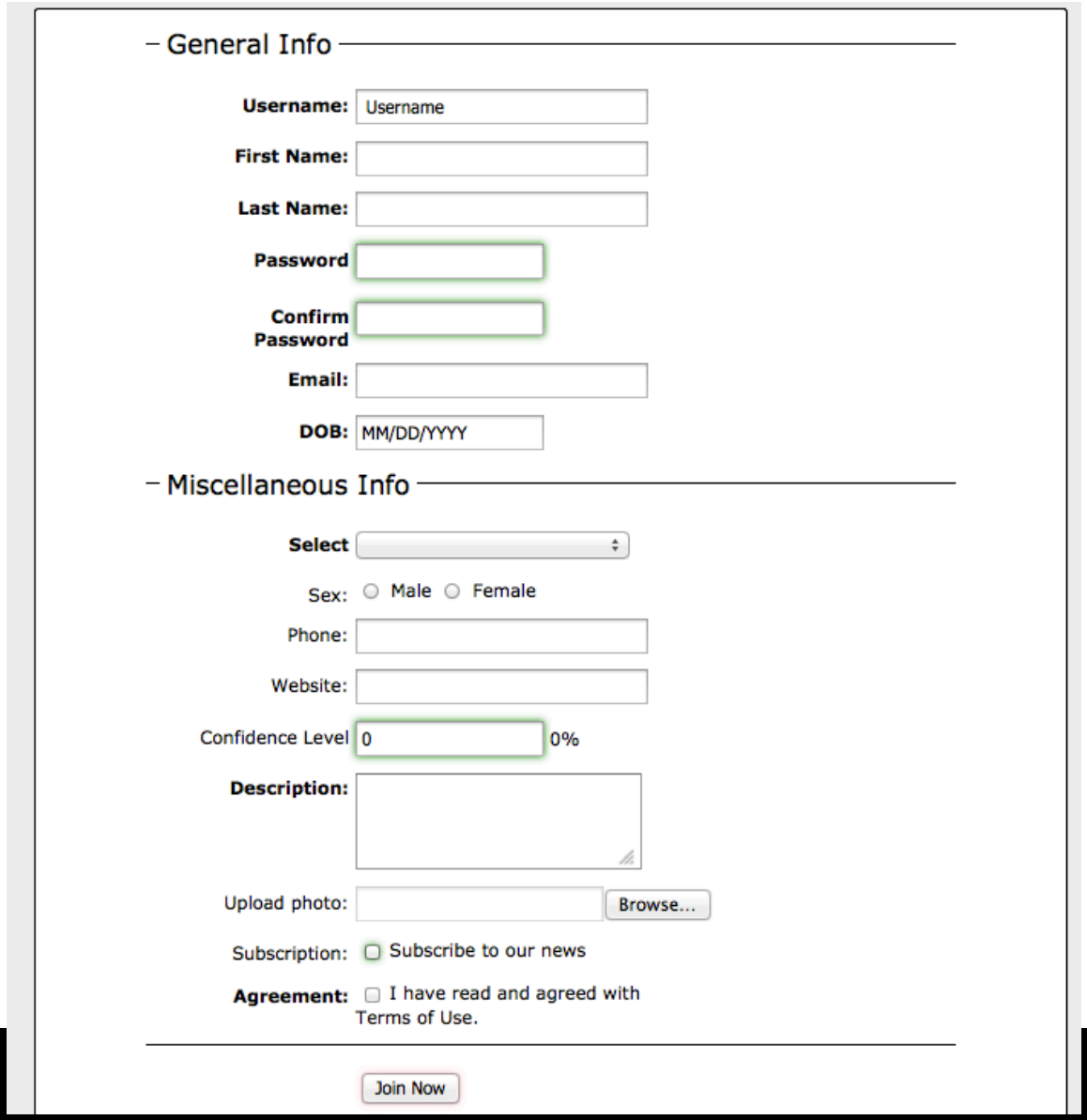
```
<input type="range" name="confidenceLevel" min="0" max="100" step="5">
```

customError

```
passwordConfirmationField.setCustomValidity("Password values do not match.");
```

BUILDING AN APPLICATION WITH HTML5 FORMS

- Check the code file under examples/Forms (index.html, main.js, main.css)

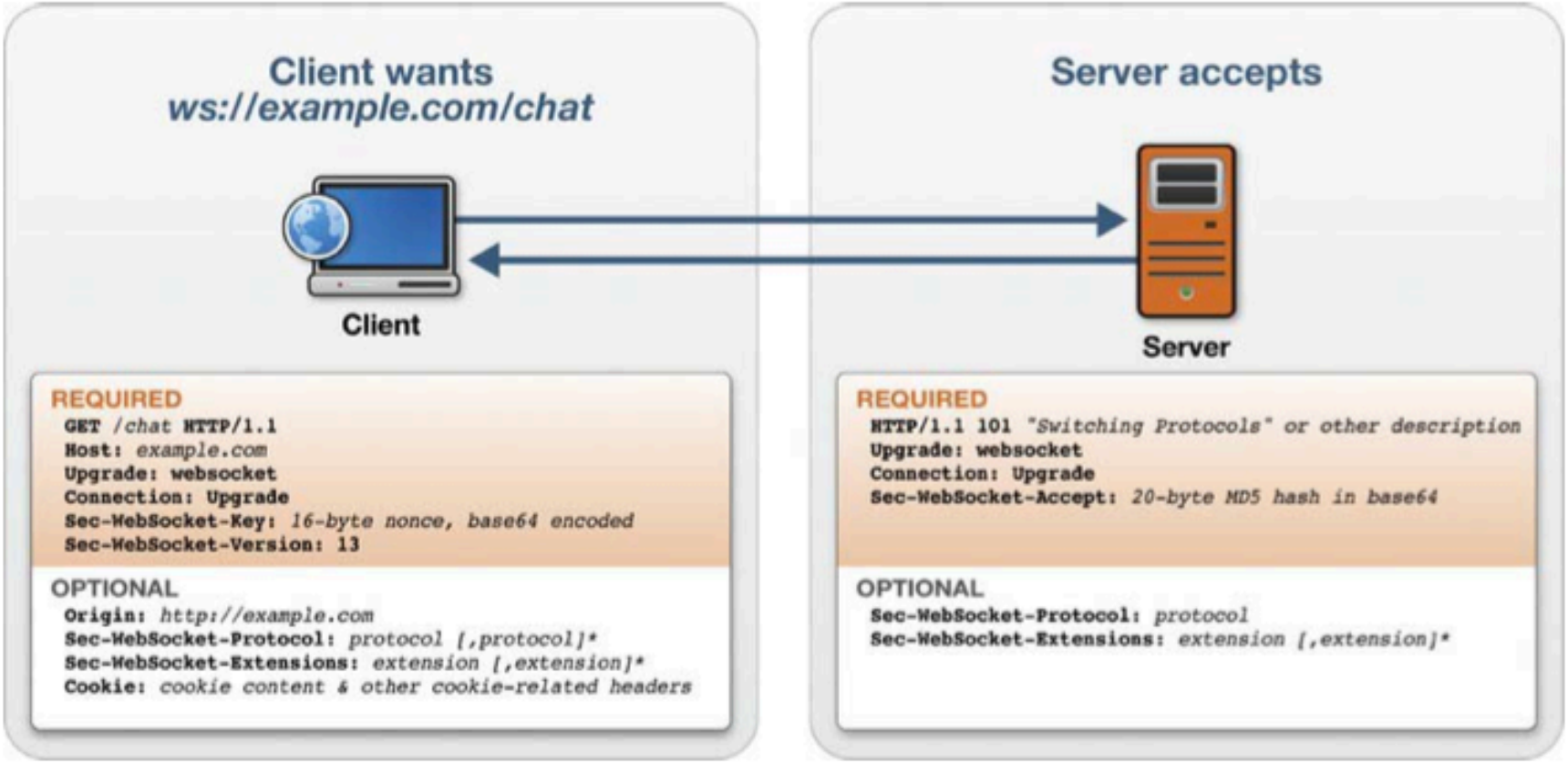


OVERVIEW OF WEBSOCKET

- The websocket provides connection support for faster and more effective two way communication between browsers and servers.
- Connection is made through a TCP socket, thus without sending HTTP headers by reducing the size of data in every call.
- Persistent connection:
  - Allow servers to keep clients updated: **we don't have to call server for updates.**
  - **Server automatically sends information about the current condition.**
- Allows the construction of real-time applications in scalable platforms:
  - Multiplayer video games,
  - Chat rooms, etc.
- The JavaScript API is simple to use with few methods and events are included to open and close the connection and send and listen for messages.
- However, no server supports the new protocol by default.
  - Need to install our own webscoket server in order to establish communication between the browser and server.

THE WEBSOCKET HANDSHAKE

- To establish a WebSocket connection, the client and server upgrade from the HTTP protocol to the WebSocket protocol during their initial handshake



WEBSOCKET SERVERS

- There are lots of WebSocket server implementations out there already and even more under development. The following are just a few of the existing WebSocket servers:
  - Kaazing WebSocket Gateway—a Java-based WebSocket Gateway
  - mod_pywebsocket—a Python-based extension for the Apache HTTP Server
  - Netty—a Java network framework which includes WebSocket support
  - **node.js**—a server-side JavaScript framework on which multiple WebSocket servers have been written
- Kaazing's WebSocket Gateway includes full client-side WebSocket emulation support for browsers without native implementation of WebSocket, which allows you to code against the WebSocket API today and have your code work in all browsers.
- We use simple WebSocket server written in Python located under samples
- To connect a remote host, just create a new WebSocket instance, providing the new object with a URL. ws:// and wss:// prefix indicates a WebSocket and a secure WebSocket connection.
- ```
  myWebSocket.onopen = function(evt) { alert("Connection open ...");};
  myWebSocket.onmessage = function(evt) { alert( "Received Message:" + evt.data); };
  myWebSocket.onclose = function(evt) { alert("Connection closed.");};
  ```
  ```

USING THE WEBSOCKET API

Checking for Browser Support

```
function loadDemo() {
  if (window.WebSocket) {
    document.getElementById("support").innerHTML = "HTML5 WebSocket is supported in your
                        browser.";
  } else {
     document.getElementById("support").innerHTML = "HTML5 WebSocket is not supported in
                                                    your browser.";
  }
}
```

Creating a WebSocket object and Connecting to a WebSocket Server

```
url = "ws://localhost:8080/echo";
w = new WebSocket(url);
```

Listing the protocols

```
w = new WebSocket(url, protocol);
//or you can list several protocols
w = new WebSocket(url, ["proto1", "proto2"]);
```

When the socket opens, its protocol property will contain the protocol that the server chose.

```
onopen = function(e) {
       // determine which protocol the server selected
      log(e.target.protocol)
}
```

USING THE WEBSOCKET API: ADDING EVENT LISTENERS

- WebSocket programming follows an asynchronous programming model; once you have an open socket, you simply wait for events.
- You add callback functions to the WebSocket object to listen for events

onopen,onmessage,onclose,and onerror are called when the events are dispatched.

```
w.onopen = function() {
   log("open");
   w.send("thank you for accepting this websocket request");
}
w.onmessage = function(e) {
    log(e.data);
}
w.onclose = function(e) {
   log("closed");
}
 w.onerror = function(e) {
    log(“error”);
}
```

message handler

```
w.binaryType = "arraybuffer";
w.onmessage = function(e) {
  // data can now be either a string or an ArrayBuffer
  log(e.data);
}
```

USING THE API: SENDING MESSAGES WEBSOCKET

- While the socket is open, you can use the send function to send messages.

Bidirectional browser communication made simple

```
document.getElementById("sendButton").onclick = function() {
 w.send(document.getElementById("inputMessage").value);
 }
```

Before calling send(), measure how much data is backed up in the outgoing buffer.
bufferedAmount attribute represents the number of bytes.

```
document.getElementById("sendButton").onclick = function() {
  if (w.bufferedAmount < bufferThreshold) {
    w.send(document.getElementById("inputMessage").value);
  }
}
```

In addition to strings, WebSocket can send binary data.
The WebSocket API supports sending Blob and ArrayBuffer instances as binary data.

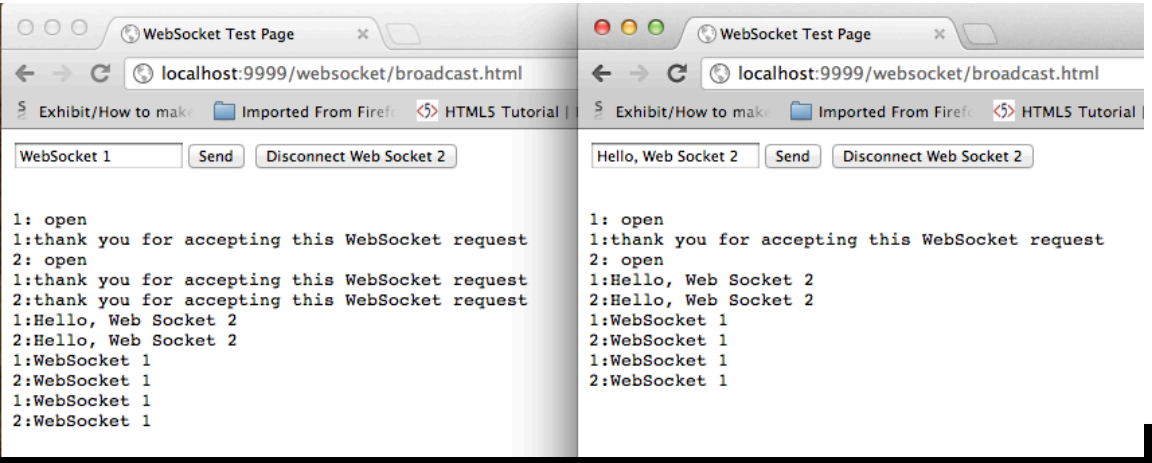
```
var a = new Uint8Array([8,6,7,5,3,0,9]);
w.send(a.buffer);
```

USING WEBSOCKET API

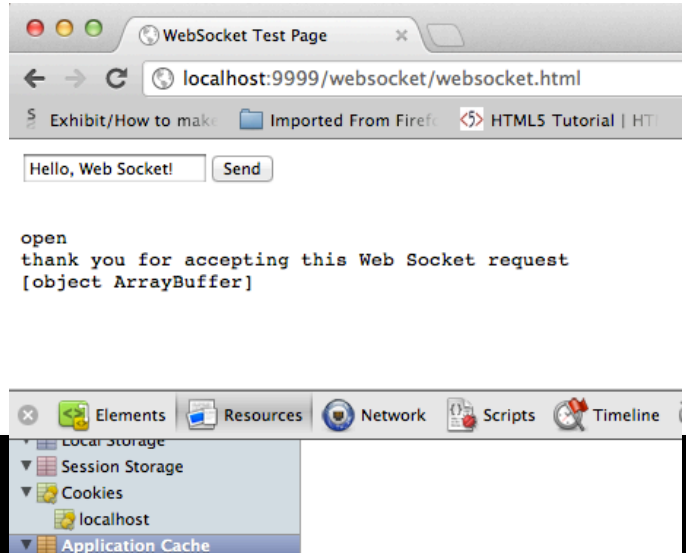
•Check the code file under examples/websocket (websocket,html, broadcast.html and tracker.html)

- Running the WebSocket Page, open the command prompt:
- python -m SimpleHTTPServer 9999
- To run the Python WebSocket echo server accepting connections at ws://localhost:8000/echo, navigate to the folder that contains the file, and issue the following command: python websocket.py
- We have also included a *broadcast* server that accepts connections at ws://localhost:8080/broadcast: python broadcast.py

broadcast.html in action in two browsers



websocket.html in action



WEB STORAGE

- One of the necessary features for any application is the possibility of storing data and making data available when it is needed.
- Cookies were used to store small pieces of information on the client side, but with the limitation to short strings and useful in some specific circumstances.
- The Web Storage API is an improvement on cookies. It lets us store data in the users hard drive and use it later as a desktop application do.
- The storage process is provided in two particular situations:
  1. when information has to be available only during a session, and
  2. when information has to be preserved as long as is determined by the user.
- The API is divided in two parts: sessionStorage and localStorage

Checking for Web Storage Support

```
function checkStorageSupport() {
  //sessionStorage
  if (window.sessionStorage) {
    alert('This browser supports sessionStorage');
  } else {
    alert('This browser does NOT support sessionStorage');
  }
  //localStorage
  if (window.localStorage) {
    alert('This browser supports localStorage');
  } else {
    alert('This browser does NOT support localStorage');
  }
}
```

WEB STORAGE

- Setting and Retrieving Values.

Setting a value can easily be done in a single statement - setItem function

```
sessionStorage.setItem('myFirstKey', 'myFirstValue');
```

Check the code file under examples/ storage (storage.html)

To retrieve the value - getItem function

```
alert(sessionStorage.getItem('myFirstKey'));
```

simpler way to access the storage objects in your code

```
sessionStorage['myFirstKey'] = 'myFirstValue';
//Similarly, the value retrieval call can be rewritten as:
alert(sessionStorage.myFirstKey);
```

Differences Between sessionStorage and localStorage

sessionStorage	localStorage
Values persist only as long as the window or tab in which they were stored (survives a browser refresh; not a browser restart).	Values persist beyond window and browser lifetimes.
Values are only visible within the window or tab that created them.	Values are shared across every window or tab running at the same origin.

WEB STORAGE: WEB SQL DATABASE & THE INDEXED DATABASE API

- One of the proposals, Web SQL Database, has been implemented in Safari, Chrome, and Opera.
- Web SQL Database allows applications access to SQLite through an asynchronous JavaScript interface.
- Although it will not be part of the common Web platform nor the eventual recommended database API for HTML5 applications, the SQL API can be useful when targeting a specific platform such as mobile Safari.
- A second proposal for browser database storage gained prominence in 2010. The Indexed Database API is supported by Microsoft and Mozilla and is seen as a counter to the Web SQL Database.

Browser Support for HTML5 Web SQL Database

Browser	Details
Chrome	Supported in version 3.0 and greater
Firefox Not	supported
Internet Explorer	Not supported
Opera	Supported in version 10.5 and greater
Safari	Supported in version 3.2 and greater

Browser Support for the Indexed Database API

Browser	Details
Chrome	Supported in current versions
Firefox	Supported in current versions
Internet Explorer	Supported in version 10+
Opera	Not currently supported
Safari	Not currently supported

Check the code file under examples/storage (sql.html)

REFLECTION

- Web Frameworks
- APIs and examples
- History and the main concepts behind HTML5
- Examples
  - Simple HTML5 page
  - Selector API
  - Canvas API
  - Audio and Video API
  - Geolocation API
  - WebSocket API
  - Forms API
  - Storage APIs

LINKS AND REFERENCES

- P. Lubbers, B. Albers, and F. Salim, "*Pro HTML5 Programming: use HTML5 to create cutting-edge web applications*", apress, second Edition, 2011.
- HTML5 & CSS3 Readiness: http://html5readiness.com/
- The HTML5 browser test: http://html5test.com/
- List of features and browser support: http://www.caniuse.com/
- Modernizr: http://www.modernizr.com/
- http://en.wikipedia.org/wiki/Early_world_maps
- https://developers.google.com/maps

THANK YOU

QUESTIONS?

Literature:

Haverbeke, M. (2014). *Eloquent JavaScript: A Modern Introduction to Programming*. No Starch Press.