

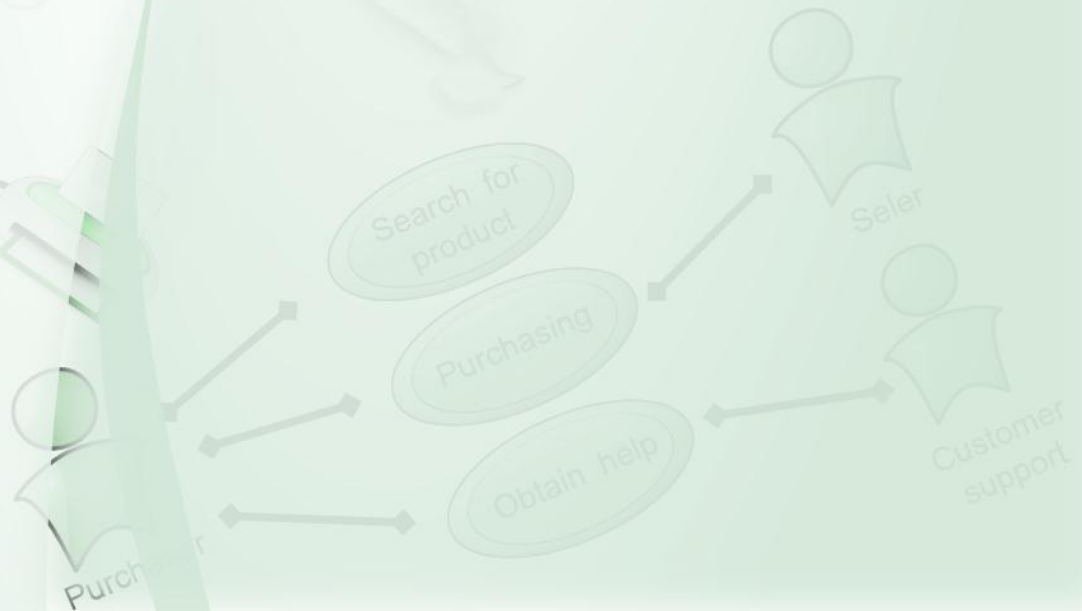
HTML 5

an overview



Outline

- HTML5 background
- HTML tags
- JS APIs
- Real time examples
- Questions



Rough Timeline of Web Technologies

1991 HTML

1994 HTML 2

1996 CSS 1 + JavaScript

1997 HTML 4

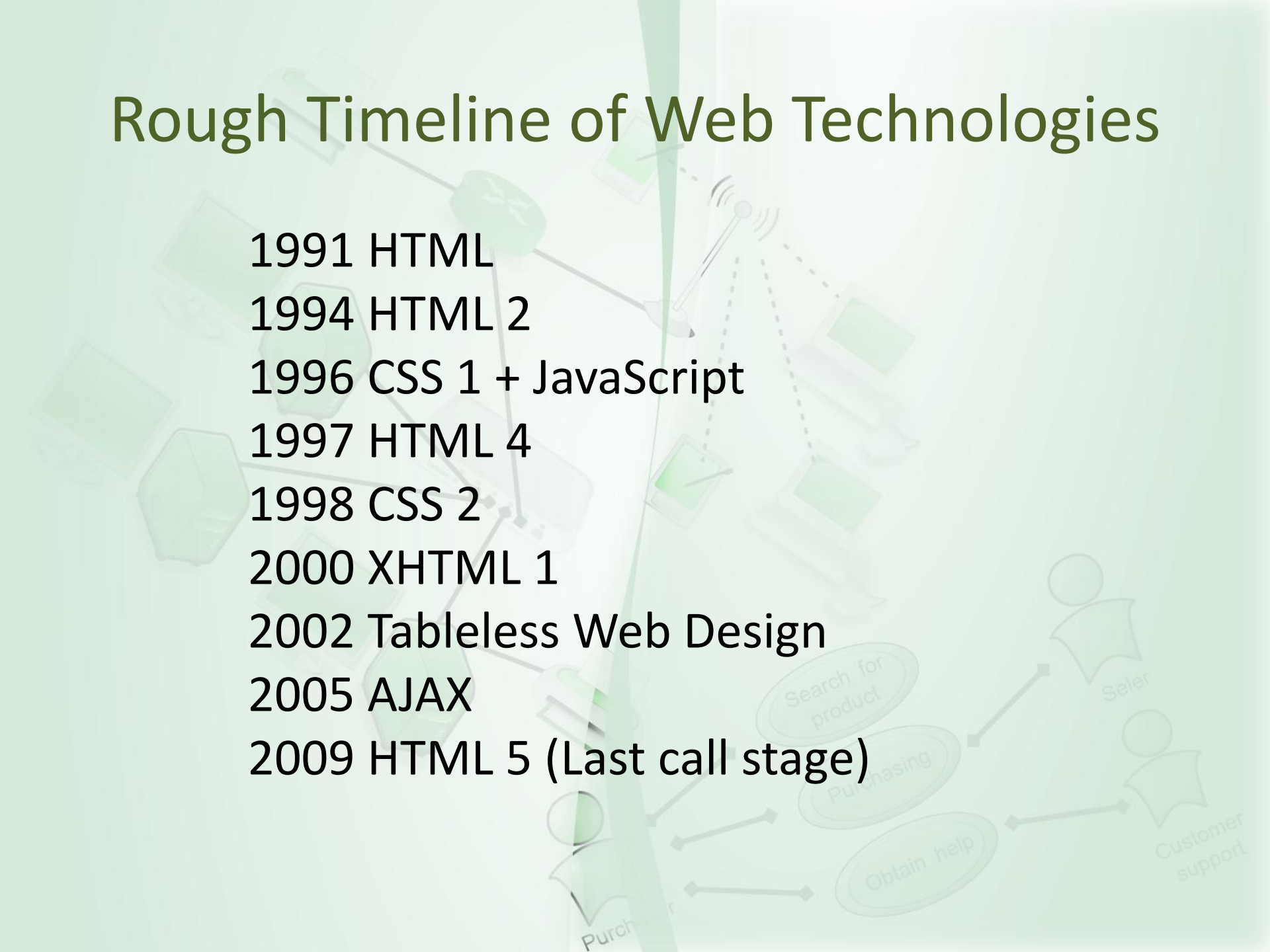
1998 CSS 2

2000 XHTML 1

2002 Tableless Web Design

2005 AJAX

2009 HTML 5 (Last call stage)



HTML5 Background

What is HTML5?

„(...) the 5th major revision of the core language of the World Wide Web: the Hypertext Markup Language (HTML)”

W3C (www.w3.org)

Created to fix several problems and improve interoperability between different user-agents.

HTML5 Background Interoperability

HTML5 will allow for better cross browser compatibility between mobile, desktop, netbook, pda, e-readers and whatever else can display a web page



HTML5 Background standardization

According to W3C timetable, it's estimated that HTML5 will reach W3C Recommendation phase in late 2010.

However now it's still a in Working Draft stage. It is expected to reach Candidate Recommendation phase in 2012.



HTML5 Tags

New Tags

<command>

A button, or a radiobutton, or a checkbox

<details>

For describing details about a document, or parts of a document

<summary>

A caption, or summary, inside the details element

<figure>

For grouping a section of stand-alone content, could be a video

<hgroup>

For a section of headings, using `<h1>` to `<h6>`, where the largest is the main heading of the section, and the others are sub-headings

<figcaption>

The caption of the figure section

<mark>

For text that should be highlighted

<meter>

For a measurement, used only if the maximum and minimum values are known

<progress>

The state of a work in progress

<time>

For defining a time or a date, or both

<wbr>

Word break. For defining a line-break opportunity.

<progress>

The state of a work in progress.

HTML5 Tags

Ruby Tags

<ruby>

For ruby annotation (Chinese notes or characters)

<rt>

For explanation of the ruby annotation

<rp>

What to show browsers that do not support the ruby element

HTML5 Tags

Structure layout

<article>

For external content, like text from a news-article, blog, forum, or any other content from an external source

<aside>

For content aside from the content it is placed in. The aside content should be related to the surrounding content

<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

<header>

For an introduction of a document or section, could include navigation

<section>

For a section in a document. Such as chapters, headers, footers, or any other sections of the document

Mostly those are semantic replacements for `<div>` and ``

HTML5 Tags

Structure layout

`<header>`

`<nav>`

`<article>`

`<section>`

`<aside>`

`<footer>`

HTML5 Tags

Media Tags

<audio>

For multimedia content, sounds, music or other audio streams

<video>

For video content, such as a movie clip or other video streams

<source>

For media resources for media elements, defined inside video or audio elements

<embed>

For embedded content, such as a plug-in

HTML5 Tags

Form Tags

<datalist>

A list of options for input values (combobox)

<keygen>

Generate keys to authenticate users

<output>

For different types of output, such as output written by a script

HTML5 Tags

Input Tags

tel	The input value is of type telephone number
search	The input field is a search field
url	The input value is a URL
email	The input value is one or more email addresses
datetime	The input value is a date and/or time
date	The input value is a date
month	The input value is a month
week	The input value is a week
time	The input value is of type time
datetime-local	The input value is a local date/time
number	The input value is a number
range	The input value is a number in a given range
color	The input value is a hexadecimal color, like #FF8800

HTML5 Tags

Form input types

- **search**
- **tel**
- **url**
- **email**
- **datetime, date, month, week, time, and datetime-local**
- **number**
- **range**
- **color**

With attributes:

- **list** — points to a datalist element containing value suggestions
- **max and min** — constrains valid date/time and number values
- **pattern** — allows a regular expression as a constraint
- **placeholder** — displays a text hint for the field

HTML5

link attributes

- For *a* or *area* elements
- media attribute - Specifies which media the target URL uses
- ping - Gives the URLs to be notified if the user follows the hyperlink
- rel - Describes the relationship between the current document and the destination URI

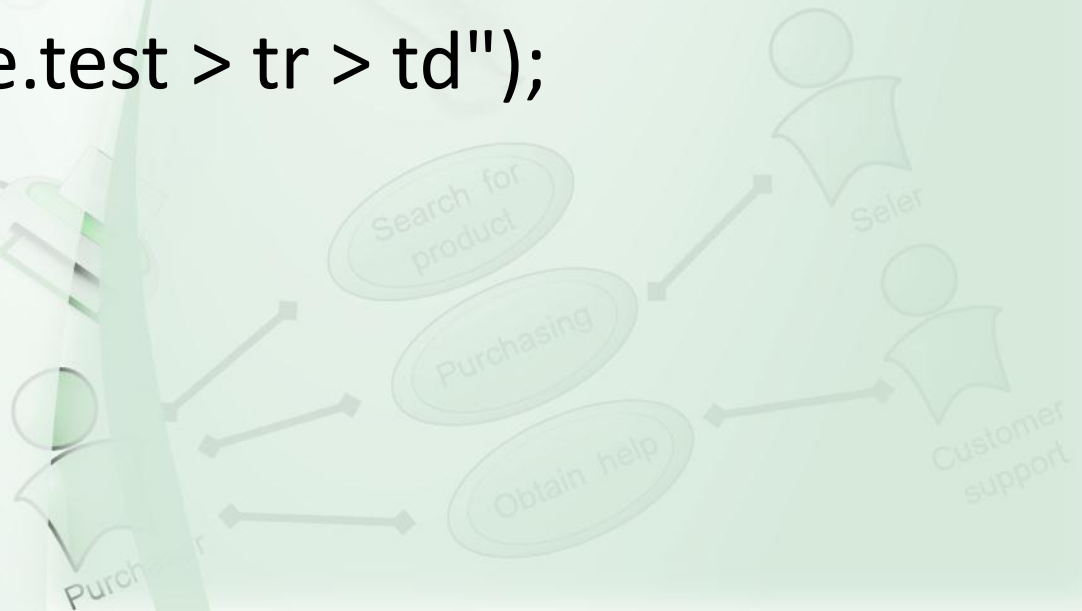
JavaScript API

New Selectors

```
getElementsByClassName('entry');
```

```
querySelectorAll("ul li:nth-child(odd)");
```

```
querySelector("table.test > tr > td");
```

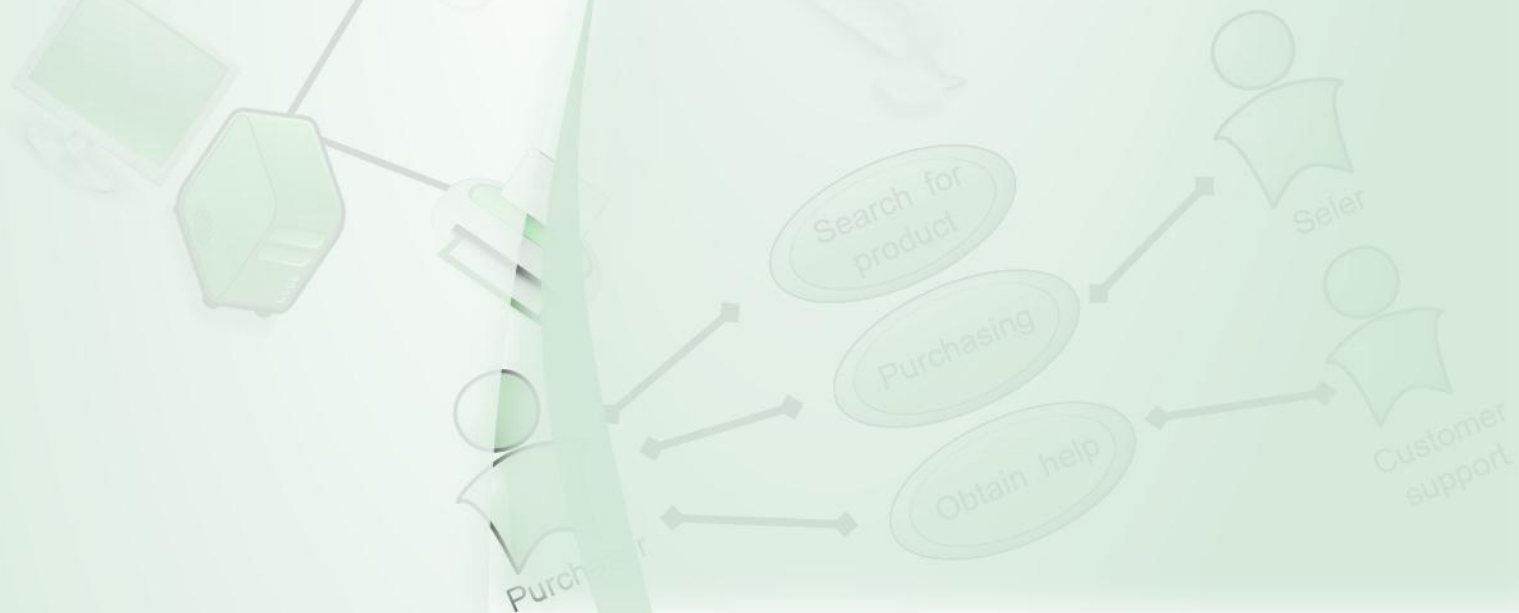


JavaScript API

Web Storage

The `sessionStorage` keeps data in per tab storage.

The `localStorage` is the persistent storage that never expires.



JavaScript API

Web SQL Database Storage

```
var db = window.openDatabase("Database Name", "Database Version");
db.transaction(
    function(tx) {
        tx.executeSql("SELECT * FROM test", [], successCallback, errorCallback);
    }
);
```

- Web SQL Storage is available even if client went offline.



JavaScript API

Offline Application Cache API

- Possibility to create fully-functional offline application
- Manifest: `<html manifest="cache.manifest">`

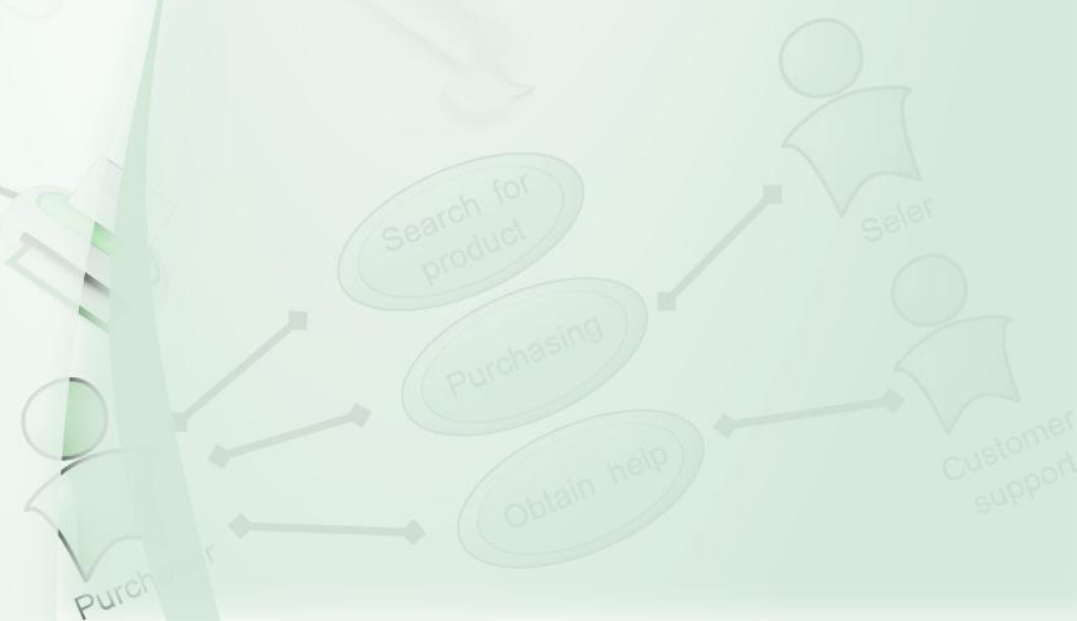
CACHE MANIFEST

CACHE:

index.html
help.html
style/default.css
images/logo.png
images/background.png

NETWORK:

server.cgi



JavaScript API

Web workers

- API for running scripts in the background independently of any user interface scripts
- have a high start-up performance cost, and a high per-instance memory cost
- separate JS processes, execute concurrently, don't block the UI, multicore CPU



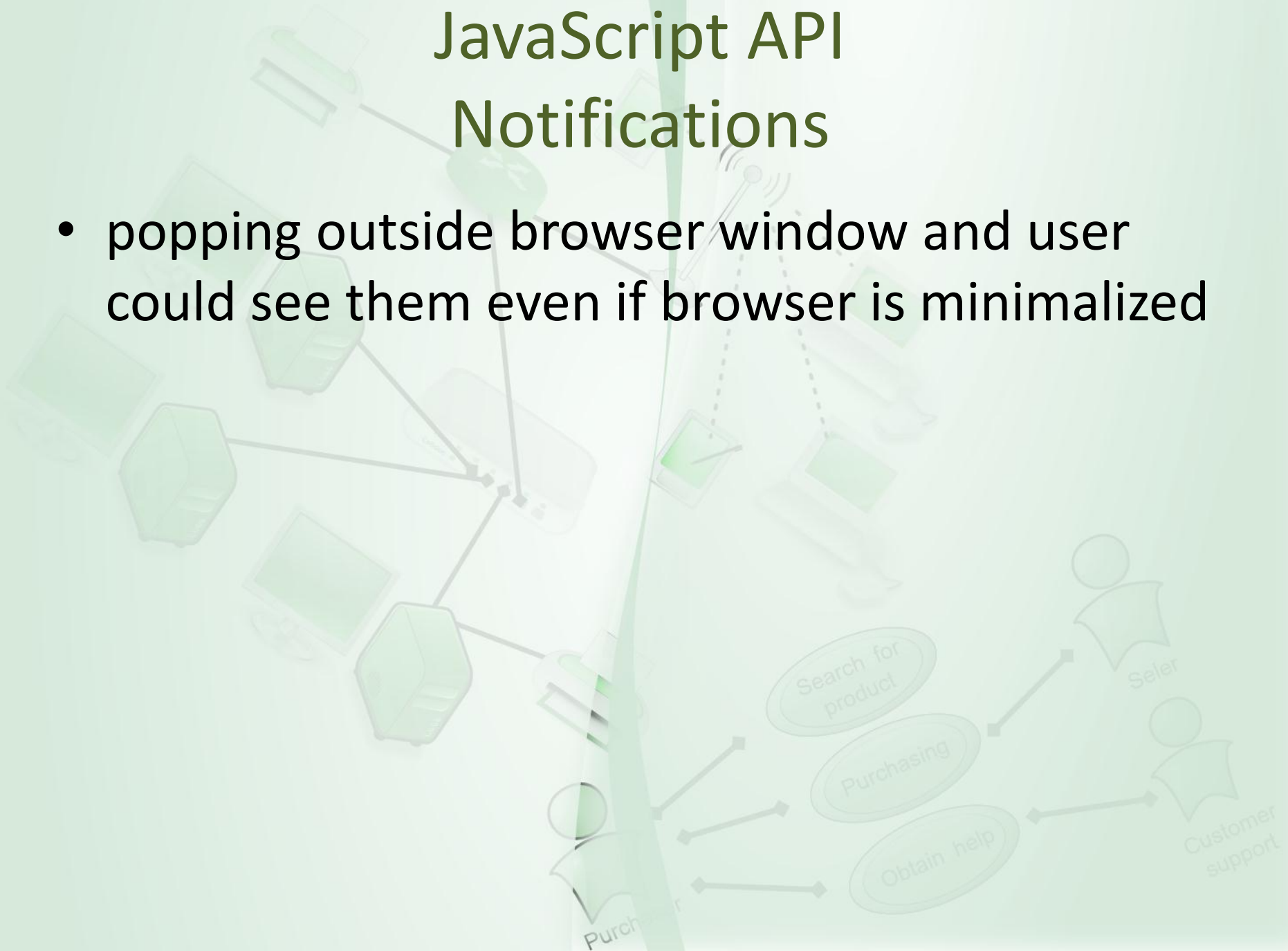
JavaScript API

Web sockets

- “TCP for the Web” - bidirectional communication technology for web applications.
- Once you get a Web Socket connection, you can send data from browser to server by calling a `send()` method, and receive data from server to browser by an `onmessage` event handler

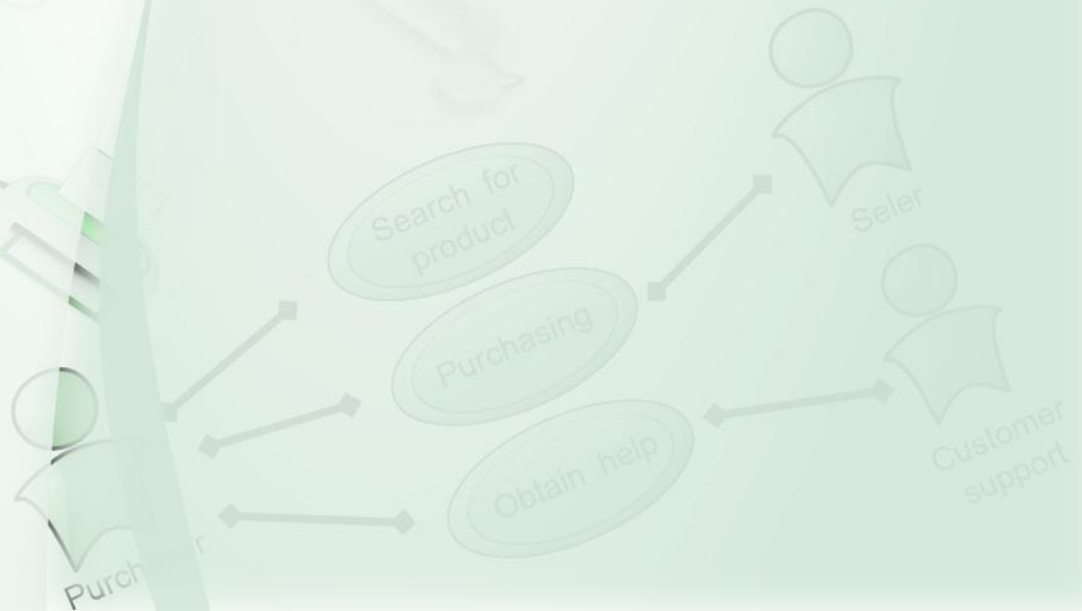
JavaScript API Notifications

- popping outside browser window and user could see them even if browser is minimalized



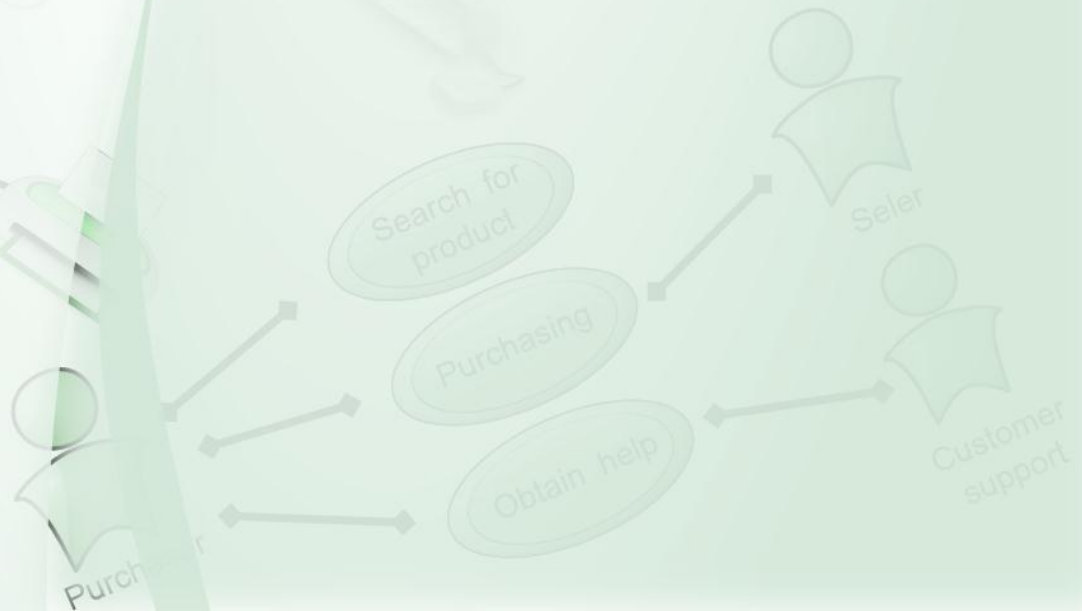
JavaScript API Drag and Drop

- Everyone knows this
- No JS!
- possibility to drag and drop text/images/files from other windows and desktop



JavaScript API Geolocation

- may recognize user's location on frontend as well as backend
- not always accurate



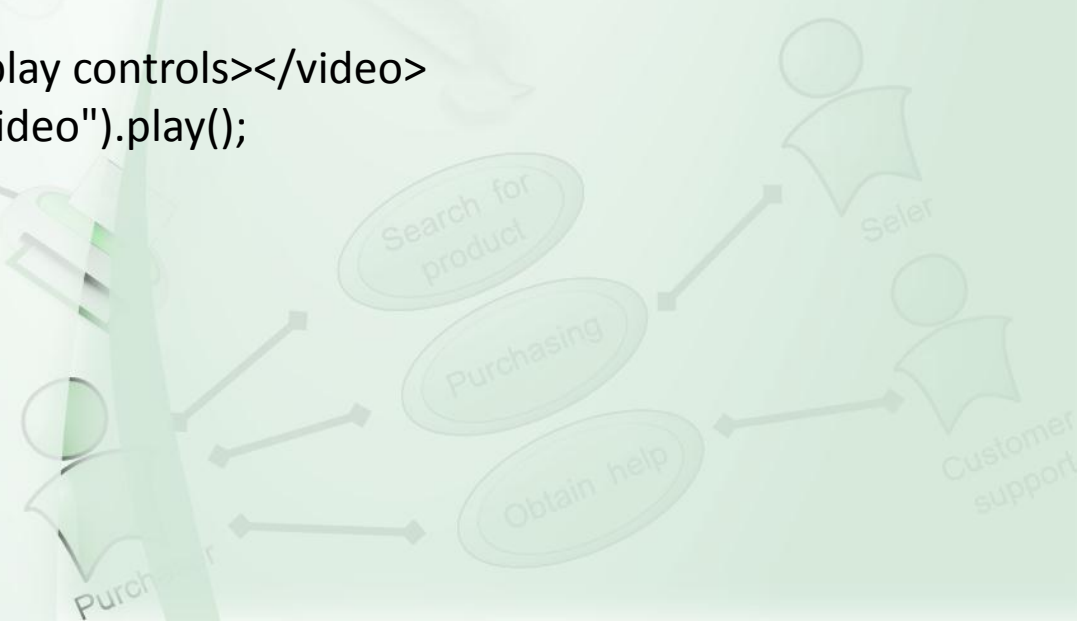
JavaScript API

Audio and Video manipulation

- Allows to super easily manipulate audio and video

```
<audio src="sound.mp3" controls></audio>  
document.getElementById("audio").muted=false;
```

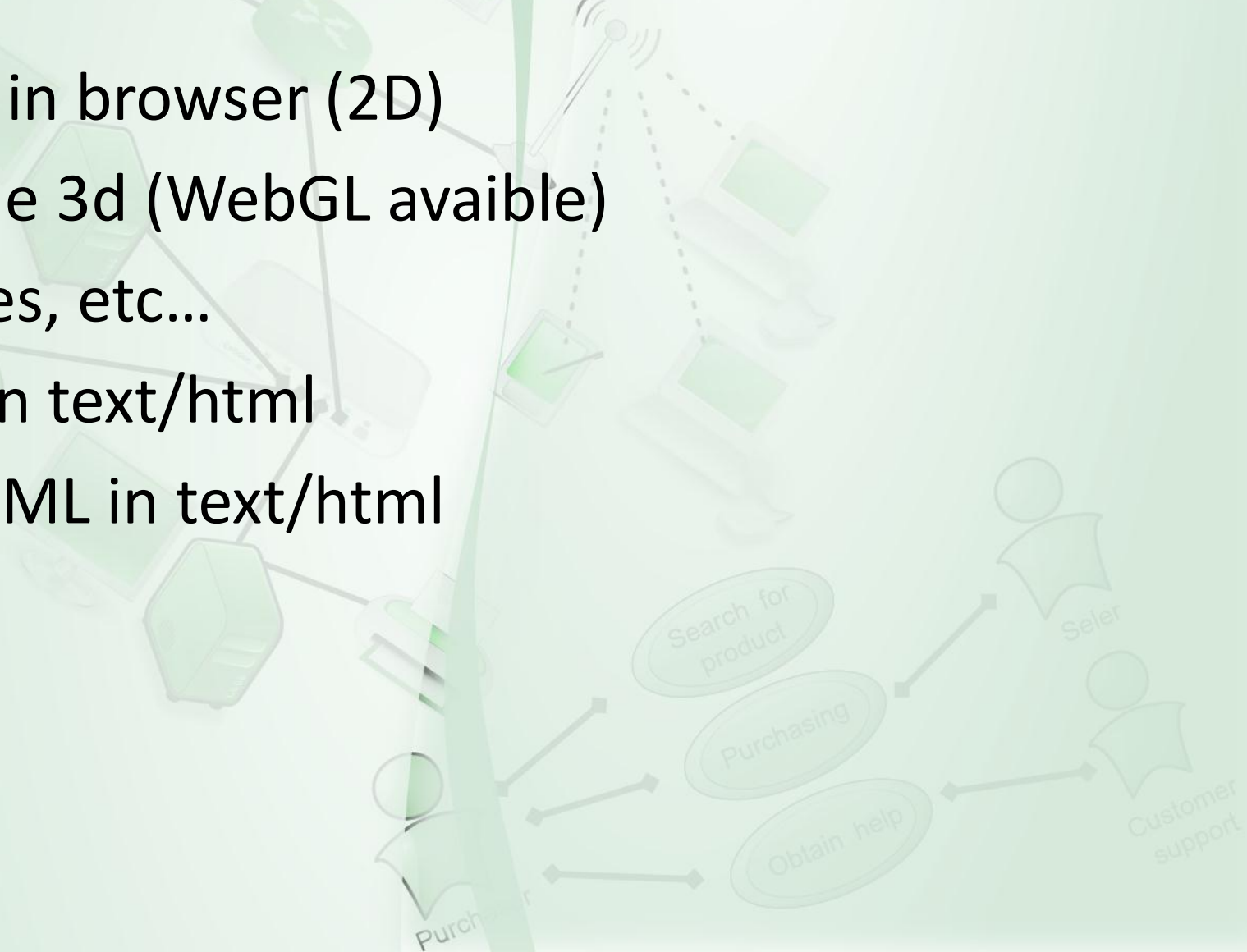
```
<video src='movie.mp4' autoplay controls></video>  
document.getElementById("video").play();
```



JavaScript API

Canvas

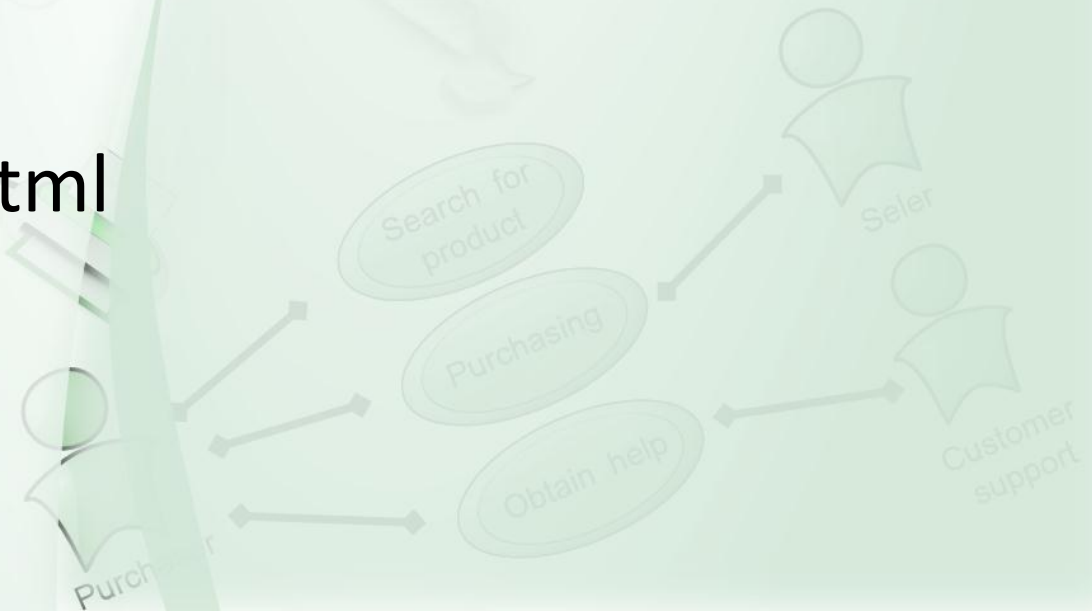
- draw in browser (2D)
- Simple 3d (WebGL available)
- Games, etc...
- SVG in text/html
- MathML in text/html



HTML5

Quake II browser port

- Playable!
- Based on Jake2 (Java Quake 2 port)
- GWT + WebGL + WebSockets
- 30 fps on laptop
- SVG in text/html
- MathML in text/html



HTML5

Error handling

- designed so that old browsers can safely ignore new HTML5 constructs
- specification gives detailed rules for lexing and parsing so different compliant browsers will produce the same result in the case of incorrect syntax



Usefull links

- HTML5 presentation:

1# <http://slides.html5rocks.com>

2# <http://html5apps-io2010.appspot.com>

- Browser support:

<http://html5demos.com/>

- GWT Quake 2 port

<http://code.google.com/p/quake2-gwt-port/>

- Sketchpad (on Canvas)

<http://mugtug.com/sketchpad/>

