

**From Web Documents
to
Web Applications**

Noah Mendelsohn
Tufts University
Email: noah@cs.tufts.edu
Web: <http://www.cs.tufts.edu/~noah>

Goals

- **Documents...why do they matter?**
- **Designing good document formats**
- **Explore the differences between static documents and “live” applications**
- **Learn the *Rule of Least Power***
- **Learn how (what used to be called) AJAX applications:**
 - Increased the power of the Web
 - Stretched the architecture of the traditional document-oriented Web

Documents and Data Formats

Document Data formats: various approaches

■ **Documents vs data:**

- *Data* tends to be fields, often labeled, often unordered (e.g. studentId:123)
- *Documents* tend to treat order as significant (this paragraph comes *before* that one...often intended for human readers)
- *Smart documents*: include data fields in document (e.g. insurance policy template with space for policy holder name, amount of policy, etc.)

■ **Approaches**

- Ad-hoc: just do whatever works
- Meta-formats: common conventions used for lots of purposes
 - Examples: plain text, CSV (comma-separated variables), XML JSON
 - Reason: common tools usable on all the formats (e.g. Python libraries, spreadsheet import)
- Self-describing: fields labeled *in each document*
Implicit: specification tells you what's where

Sharing Documents, Data, & Applications

Sharing documents, data & applications

- **Since early days of computing: all have been shared**
- **Often, we send *documents* through a network (e.g. Web pages)**
- **Programs & data/documents on:**
 - Punched cards, paper tape, magnetic tape
 - Floppy disks, CD ROMS, thumb drives, etc.
- **Networking early days:**
 - Sharing via FTP and e-mail
- **Model**
 - Copy the data/documents
 - *Manually install the programs*

What makes applications different?

- **Documents don't *do* anything**
- **Programs can:**
 - Perform arbitrary computations for you
 - Alter your machine
 - Run for a long time, with multiple interactions
 - Masquerade as other programs (spoofing)
 - Do much more (open network connections, etc.)!
- **The Web:**
 - Allowed instantaneous “activation” of documents
 - Now doing the same for applications

What makes applications different?

- **Documents don't *do* anything**
- **Programs can:**
 - Perform arbitrary computations for you
 - Alter your machine
 - Run for a long time, with multiple interactions
 - Masquerade as other programs (spoofing)
 - Do much more (open network connections, etc.)!
- **The Web:**
 - Allowed instantaneous “activation” of documents
 - Eventually did the same for applications
 - ***A new model: applications built from smart documents***

Encoding logic and semantics

Declarative code sets out the *desired result* not details of the necessary computation (e.g. how to compute where the 1st character goes when centering something)

- **Declarative markup** (e.g. `<boldface>some text</boldface>`)

- **Constraint based** (a powerful kind of declarative coding):

- `<center>some text</center>`
- `<p style="margin-left:20%"> some text</p>`

- **Functional:** answer is the result of some computation

- **Imperative**

- E.g. To center text:
 - 1) Compute width
 - 2) Compute position
 - 3) Position cursor
 - 4) Write the text.

Semantics depend on context...

- * Center *within* `<body>` or `<div>`
- * Margin from left of *current* border and width 20% of containing `<body>` or `<div>`

- **HTML and CSS are *declarative*, Javascript is *imperative***

JavaScript & Web Pages

Javascript

- **A general purpose programming language**
- ***Runs on Web pages***
- **Turing complete – this is a big deal!**
 - Check out the amazing: <http://bellard.org/jslinux/>
 - *Remember, these are running entirely in a Web page in your browser!*
- **Proving the power of JavaScript**
 - The JavaScript is emulating a CPU and controller chip *at the instruction level*
 - A Linux is booted on that!
- **Also:**
 - https://archive.org/details/win3_stock and [https://archive.org/details/mac MacOS 7.0.1 compilation](https://archive.org/details/mac_MacOS_7.0.1_compilation)
<https://www.pcjs.org/devices/pc/machine/5160/cga/256kb/win101/>

See tech notes at: <http://bellard.org/jslinux/tech.html>

JavaScript is a *scripting language*

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Demo #1</title>
</head>
<body>
<h1>JavaScript Demo</h1>
<p>Try moving your mouse over the word below:</p>
<p id="hellopara" style=:font-size:large"
  onmouseover='this.style.color="red";'
  onmouseout='this.style.color="black";'>Hello!</p>
</body>
</html>
```

Even a single statement can be used in some cases

The ability to write single statement programs tends to be a distinguishing characteristic of scripting languages.

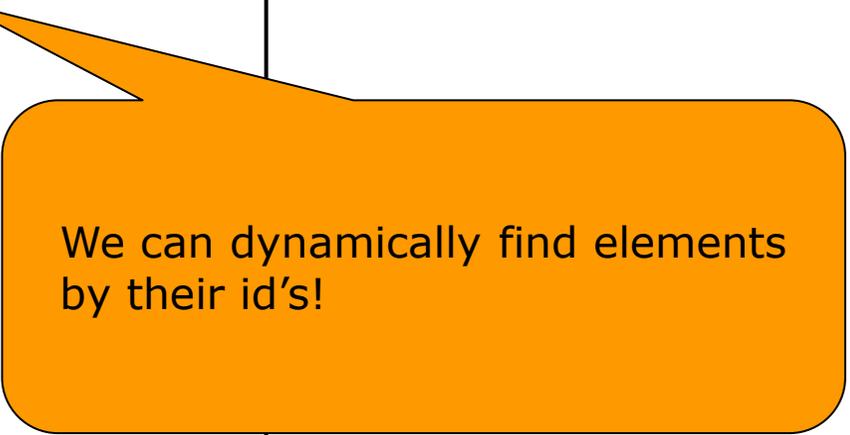
JavaScript is *integrated with the Web page*

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Demo #1</title>
</head>
<body>
<h1>JavaScript Demo</h1>
<p>Try moving your mouse over the word below:</p>
<p id="hellopara" style=:font-size:large"
  onmouseover='this.style.color="red";'
  onmouseout='this.style.color="black";'>Hello!</p>
</body>
</html>
```

What is "this"?

JavaScript is *integrated with the Web page*

```
<html>
<head>
<script type="text/javascript">
function makeColor(elem,color) {
    elem.style.color=color;
}
function loadfunction() {
    helloPara = document.getElementById('hellopara');
    helloPara.onmouseover = function() {
        makeColor(this,"red");
    };
    helloPara.onmouseout = function() {
        makeColor(this,"black");
    };
}
</script>
<title>JavaScript Demo #2</title>
</head>
<body onload='loadfunction()' >
<h1>JavaScript Demo #2</h1>
<p>Try moving your mouse over the word below:</p>
<p id="hellopara" style=:font-size:large">Hello!</p>
</body>
</html>
```



We can dynamically find elements by their id's!

Uses for JavaScript on the Web

- **Making Web pages smarter**

- Form field input checking
- Dynamic highlighting & selection
- *In this case, we still have a document*

- **Using HTML as an application *container***

- The user interacts with an application, not a document
- Example: the Linux system we just saw
- GMail, Yahoo Mail

- **AJAX**

- Building smart documents *and* applications by *integrating content from multiple sources dynamically at the client*
- The original mashup: <http://www.housingmaps.com/> (no longer active ☹)

The original housingmaps mashup – the first Ajax app!

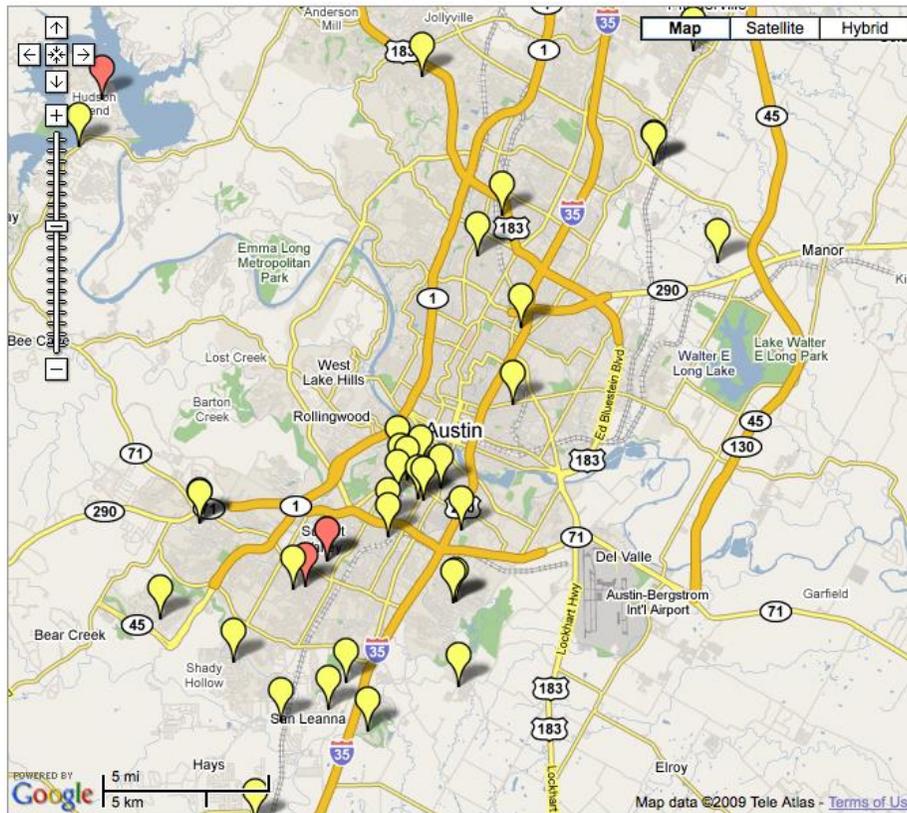
[For Rent](#) [For Sale](#) [Rooms](#) [Sublets](#)

Powered by [craigslist](#) and [Google Maps](#)

(this site is in no way affiliated with craigslist or Google)

[About / Feedback](#)

City: Price: [Show Filters](#) ^{New} [Refresh](#) [Link](#)



pics	price	description	city	date
	\$245K	Deer Creek Ranch Rare Find in Dsisd over 2000 sq ft -	Dripping Spr	1/05
	\$176K	11.76 Acre Land near Hamilton Pool in Dripping Springs -	Dripping Spr	1/05
	\$183K	4/2.5 Zero down & closing.Others Round Rock Leander, Pflugerville -	Kyle	1/05
	\$225K	Amazing Wood floors on 1st level . So many upgrades, must see -	Cedar Park	1/04
	\$213K	Upgraded Model home @ price of a Base model. -	Cedar Park	1/04
	\$183K	Just Reduced--Beautiful 3 bedroom 2.5 bath in Olympic Heights -	Kingsland	1/04
	\$275K	See All The Extras On This Home-open hs today! -	Austin	1/04
	\$190K	2br / 2ba Condo Villas on Travis w/ Lake Access -	Austin	1/04
	\$250K	Hot Area, Central East Austin -	Austin	1/04
	\$279K	17.50 Acres with 4/2 home and barn with wash rack. -	Georgetown	1/04
	\$189K	Open House Today! 6512 Alum Rock Cove -	Austin	1/04
	\$272K	Great Landscaping Accents This Two Bed Home! -	Georgetown	1/04
	\$150K	Great investment opportunity in Harris Branch -	Austin	1/04
	\$166K	3 Bdmv 2.5 Bath - Price reduced -	Pflugerville	1/04
	\$154K	3 Bedrooms - 1.5 Baths - Mis# 8417183 -	Austin	1/04
	\$235K	3 Bed, 2 Bath In Desirable Round Rock! -	Round Rock	1/04
	\$212K	Enjoy the Good Life in this Sun City Beauty! -	Georgetown	1/04
	\$255K	Enjoy Exquisite Style for a Great Price! -	Georgetown	1/04
	\$285K	2 Bed, 2 Bath Loaded with Upgrades! -	Georgetown	1/04
	\$289K	Stunning 2 Bedroom Home in Move-In Condition! -	Georgetown	1/04



Paul Rademacher, 32

The man who opened up the map

Google

There's a magical moment when an unfamiliar piece of information--say, an address or an image--produces a flash of recognition, when we suddenly know where to place it. That happened to Paul Rademacher in April 2005, when he fired up his hacked version of the brand-new Google Maps site. To ease his housing hunt, Rademacher had deciphered, then modified, the JavaScript behind Google's application, creating a version that retrieved data from two different sources: Google and craigslist, the popular classified site. The result was a hybrid page that displayed Google's familiar map and scattered across it icons indicating houses for rent around San Francisco.

Rademacher's new picture of the world--or at least of selected cities--took the Web by storm. Even Google employees wrote on a company Web page that his site, housingmaps.com, "blew our minds right off our shoulders." Thousands of people realized that Google's maps were a giant canvas on which they could doodle, taking the locations of crime scenes, favorite restaurants, or cheap gas stations and creating online tableaux for all to see. But more than that, Rademacher had shown a way to combine data and tools from completely different websites to create something new. One blogger called it a "mashup," a word DJs use to describe the mixture of vocal and instrumental tracks from different songs, and the term stuck.

From : <http://www2.technologyreview.com/tr35/profile.aspx?TRID=437>

Uses for JavaScript on the Web

- **Making Web pages smarter**

- Form field input checking
- Dynamic highlighting & selection
- *In this case, we still have a document*

- **Using HTML as an application *container***

- The user interacts with an application, not a document
- Example: the Linux system we just saw
- GMail, Yahoo Mail

- **AJAX**

- Building smart documents *and* applications by *integrating content from multiple sources dynamically at the client*
- The original mashup: <http://www.housingmaps.com/> (no longer active ☹)

- **HTML Games and Animations**

- E.g. Euro-soccer HTML5 game <http://play.famobi.com/euro-soccer-sprint>

Summary: Integrating JavaScript with Web Pages

- **Web applications are HTML pages w/JavaScript smarts**
- **HTML serves as the *display list* for the application**
 - JavaScript can manipulate the HTML tree & CSS at anytime
 - *Page is dynamically re-rendered*
- **History footnote:**
 - 1997 or so: Netscape had invented JavaScript (which has nothing to do with Java, but was named to sound like it did...)
 - Microsoft was behind in the “browser wars”, but...
 - ...Microsoft had in an internal product a rendering engine that can incrementally re-render a page when parts of a document change
 - They repurposed that engine to make Internet Explorer’s dynamic HTML rendering better than Netscape’s
 - Result: Internet Explorer became the leading browser (until Firefox, based on a reworked Netscape engine, became an attractive open-source alternative)

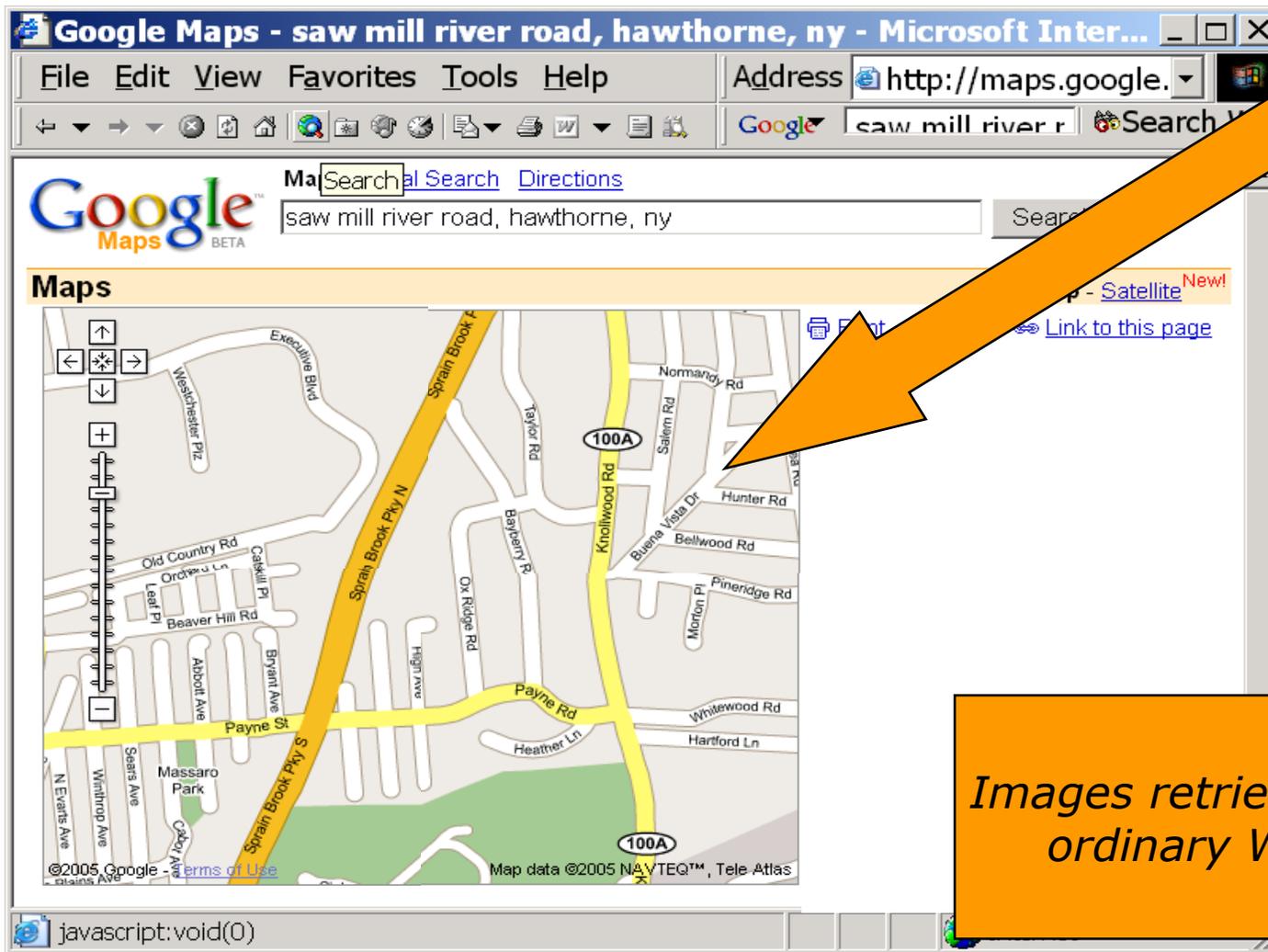
Mashups 101

The original mashup



Google Maps + craigslist = www.housingmaps.com

How Google Maps Work



Images retrieved in segments using ordinary Web HTTP Requests

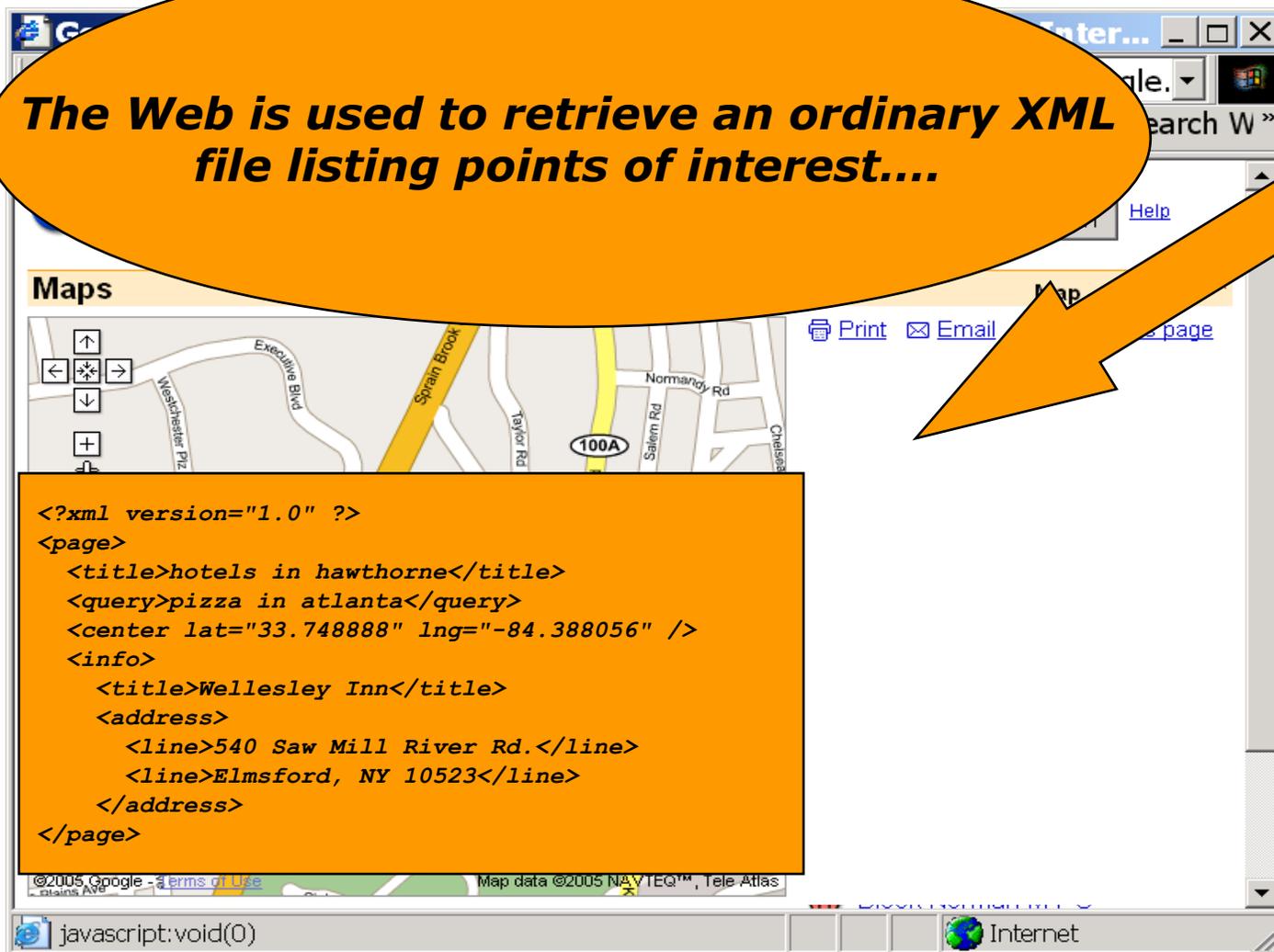
How Google Maps Work



**JavaScript at client tracks mouse and moves images for smooth panning...
asynchronously requests new image tiles in background**

How Google Maps Work

The Web is used to retrieve an ordinary XML file listing points of interest....



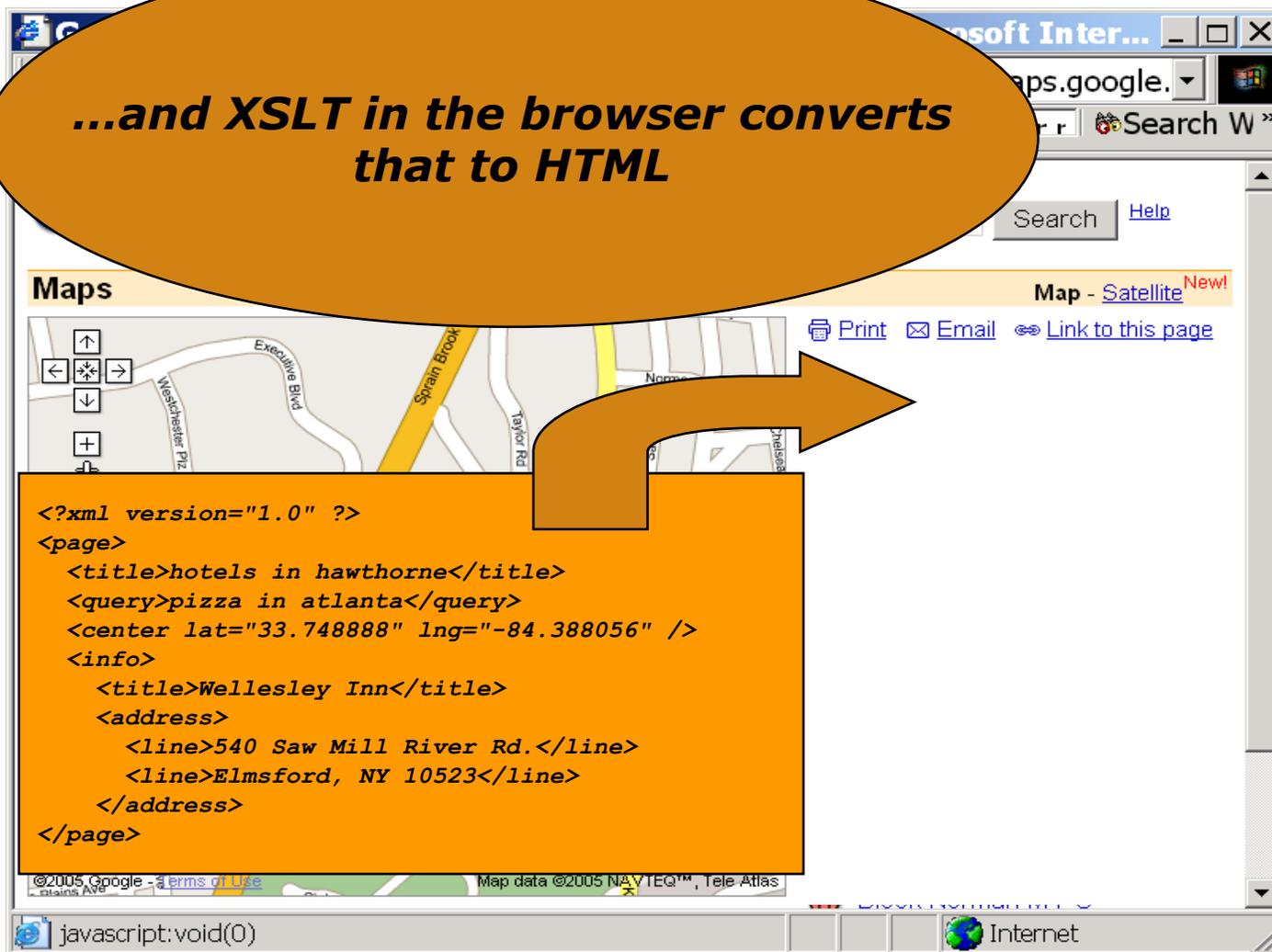
The image shows a screenshot of a web browser window displaying Google Maps. A large orange oval highlights the text: "The Web is used to retrieve an ordinary XML file listing points of interest....". A large orange arrow points from this text towards the XML data snippet shown in a separate orange box. The XML snippet is as follows:

```
<?xml version="1.0" ?>
<page>
  <title>hotels in hawthorne</title>
  <query>pizza in atlanta</query>
  <center lat="33.748888" lng="-84.388056" />
  <info>
    <title>Wellesley Inn</title>
    <address>
      <line>540 Saw Mill River Rd.</line>
      <line>Elmsford, NY 10523</line>
    </address>
  </page>
```

The browser window also shows a map of a street area with labels like "Executive Blvd", "Sprain Brook", "Taylor Rd", "Normandy Rd", and "100A". The browser's address bar shows "javascript:void(0)" and the status bar shows "Internet".

How Google Maps Work

...and XSLT in the browser converts that to HTML



Microsoft Inter...
aps.google.
Search W
Search Help
Map - Satellite ^{New!}
Print Email Link to this page

```
<?xml version="1.0" ?>  
<page>  
  <title>hotels in hawthorne</title>  
  <query>pizza in atlanta</query>  
  <center lat="33.748888" lng="-84.388056" />  
  <info>  
    <title>Wellesley Inn</title>  
    <address>  
      <line>540 Saw Mill River Rd.</line>  
      <line>Elmsford, NY 10523</line>  
    </address>  
  </page>
```

©2005 Google - Terms of Use Map data ©2005 NAVTEQ™, Tele Atlas
javascript:void(0) Internet

How Google Maps Work

...and XSLT in the browser converts that to HTML

```
<?xml version="1.0" ?>
<page>
  <title>hotels in hawthorne</title>
  <query>pizza in atlanta</query>
  <center lat="33.748888" lng="-84.388056" />
  <info>
    <title>Wellesley Inn</title>
    <address>
      <line>540 Saw Mill River Rd.</line>
      <line>Elmsford, NY 10523</line>
    </address>
  </page>
```

Microsoft Inter...
aps.google.
Search W
Search Help
Maps Map - Satellite ^{New!}
Print Email Link to this page
Tino's Italian Steak House
769-7236 - 0.3 mi SE
Polytechnic University
(914) 347-6940 - 1.2 mi SW
C MT Pleasant Taxi
(914) 769-2306 - 0.1 mi N
D New Town Taxi MT Pleasant
Taxi
(914) 747-0999 - 0.1 mi N
E Wellesley Inn
(914) 592-3300 - 2.7 mi SW
F All Westchester Saw Mill
Multiplex Cinemas
(914) 747-2333 - 1.1 mi SW
G Sabatino's Coal Brick Oven &
Cafe
(914) 773-0108 - 0.9 mi SW
H Block Norman M PC
©2005 Google - Terms of Use Map data ©2005 NAVTEQ™, Tele-Atlas
javascript:void(0) Internet

How the Mashup works

- **Javascript opens a connection to Craigslist**
- **Craiglist information converted to Google-Maps compatible form**
- **Information is “injected” into Google Maps client side runtime.**

How Google Maps Work

...now we've got real estate listings instead of restaurants

Microsoft Inter...
aps.google...
Search W...
Search Help

Maps Map - Satellite ^{New!}

Print Email Link to this page

description

Must see - beautifll 3 bedroom - renovated Broo

1 bedroom coop for rent in full service building Brc

Fabulous Apt. for Rent Broo

3 bedroom 2 full bath duplex Bay:

3 bedroom 1 huge bathroom Little

2/3 bedroom apt in pvt.house Ossi

Bay Ridge-1BR W/Private Deck- By Owner Broo

Cozy 2 BR House, acre, Fireplace for rent Holbi

Brand New Three Bedroom, Heat Included, Bedroom Balcony South

Beautiful 3BR Lrm. Eik. 2nd Fl. Full ...

```
<?xml version="1.0" ?>
<page>
  <title>Housing from Craigsilst</title>
  <info>
    <title>Cute apartment</title>
    <address>
      <line>101 5th Avenue.</line>
      <line>New York, NY 10001</line>
    </address>
  </page>
```

©2005 Google - Terms of Use Map data ©2005 NAVTEQ™, Tele-Atlas

javascript:void(0) Internet

How the Mashup works

- **Javascript opens a connection to Craigslist**
- **Craiglist information converted to Google-Maps compatible form**
- **Information is “injected” into Google Maps client side runtime.**

Challenge
What if Everything on the Web was
Javascript?

The Rule of Least Power

- Javascript is powerful, but...you can't extract much information from JavaScript source
- If all Web content were JavaScript...we wouldn't have Google!
- Where possible, use declarative languages
 - HTML
 - CSS
 - Etc..

Read TAG Finding "Rule of Least Power"

<http://www.w3.org/2001/tag/doc/leastPower.html>

Architecture Challenges: Naming

How do we name application states

- URI fragment IDs name elements in an HTML document
- What to do when the document hosts an application?
- This proved a subtle problem...read the TAG finding
- What would you want to link in that [soccer game](#)?
- One useful idiom: if your application is serving documents, give each one a URI
 - Maps, Office Documents, E-mail
- Remember: keep naming orthogonal to access control

Read TAG Finding “Identifying Application State”

<http://www.w3.org/2001/tag/doc/IdentifyingApplicationState>

Architecture Challenges: Security

Mobile Code is Powerful but Dangerous

- **Advantages**

- Can do “anything” – build powerful applications
- Supports very fluid interactivity
- Zero install: we’re always running the latest version of Google Maps, Yahoo! Mail, etc.

- **Dangers:**

- Can do “anything” – at least if security controls fail
- Hard to check what a Turing-complete program is going to do: virus detection unreliable
- Be especially suspicious of code that manipulates other code

***Read Ken Thompson’s Turing Award Lecture
“Reflections on Trusting Trust”***

<http://cm.bell-labs.com/who/ken/trust.html>

Summary

Summary

- **With JavaScript, the Web evolved**
- **Now: documents + applications**
- **Least Power: don't use powerful languages unnecessarily**
- **Model applications as documents where practical**
- **Think hard about naming things with URIs**
- **Mobile code: powerful but dangerous!**
- **Is the Web really working the way Tim Berners-Lee planned it? (We'll discuss that next time)**