AJAX

Internet Technology

ITECH

HYPE, MECHANICS, DEMOS

The Hype

In 2006, an Amazon.co.uk search for ‘ajax’
under Books / Computer and Internet returned

52 books

By 2019 the same search (under “Books : Computing & Internet) returned:

+700 books

Making Everything Easier!™

JavaScript® & AJAX FOR DUMMIES®

Learn to:

- Master basic JavaScript as a Web design and application development tool
- Write your own programs
- Use JavaScript with AJAX, XML, and JSON
- Design an interface, animate images, program menus, and manage cookies

Andy Harris
Author of HTML, XHTML, and CSS All-in-One For Dummies

Copyrighted Material

Clean up the speed and appearance of your Web applications with Ajax

Ajax FOR DUMMIES®

Code and usable Ajax applications on companion Web site

A Reference for the Rest of Us!®

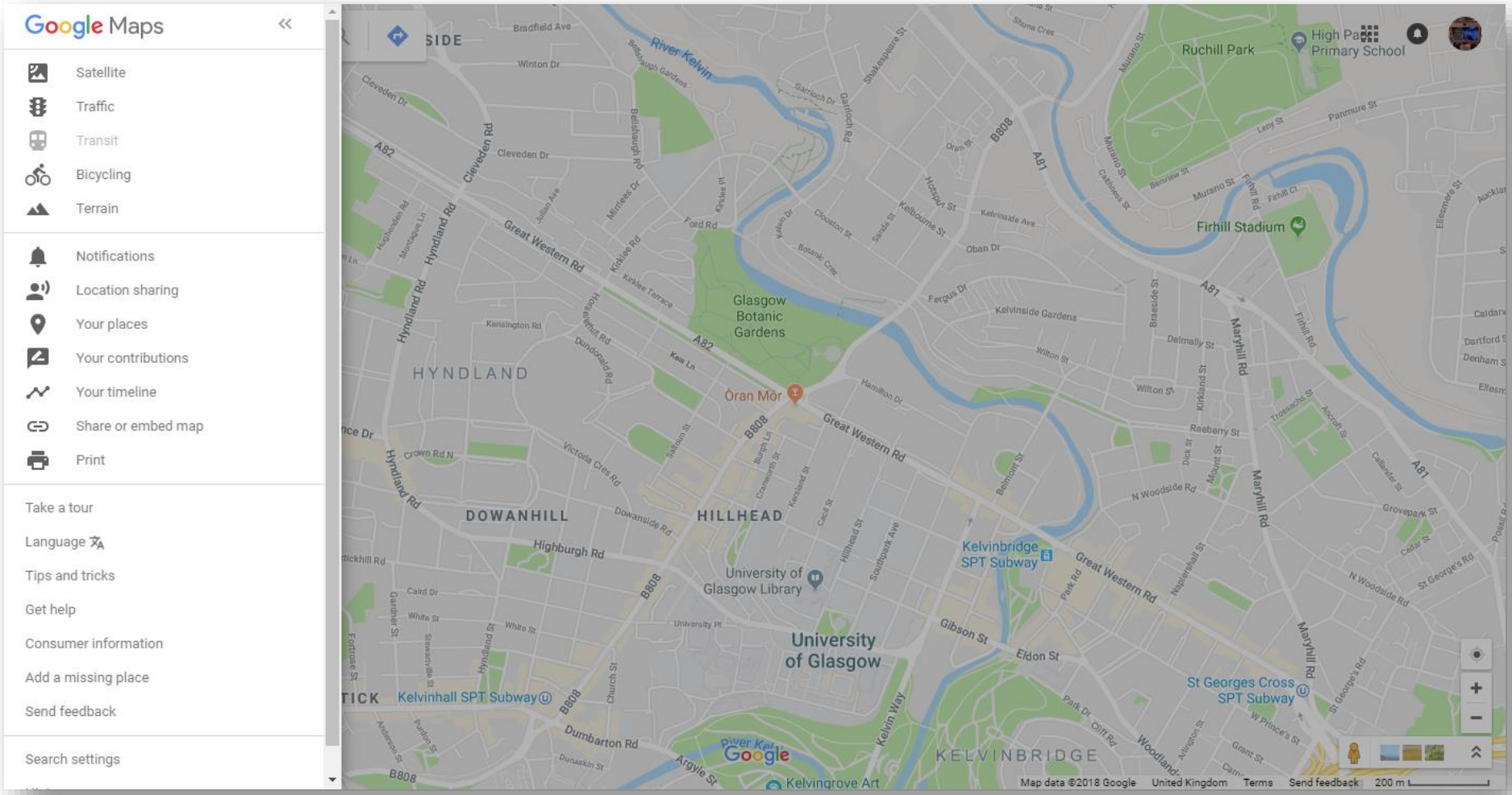
FREE eTips at dummies.com®

Steve Holzner, PhD
Author of Physics For Dummies

Ajax

- A key technology underlying web apps
  - **Asynchronous JavaScript** and **XML**

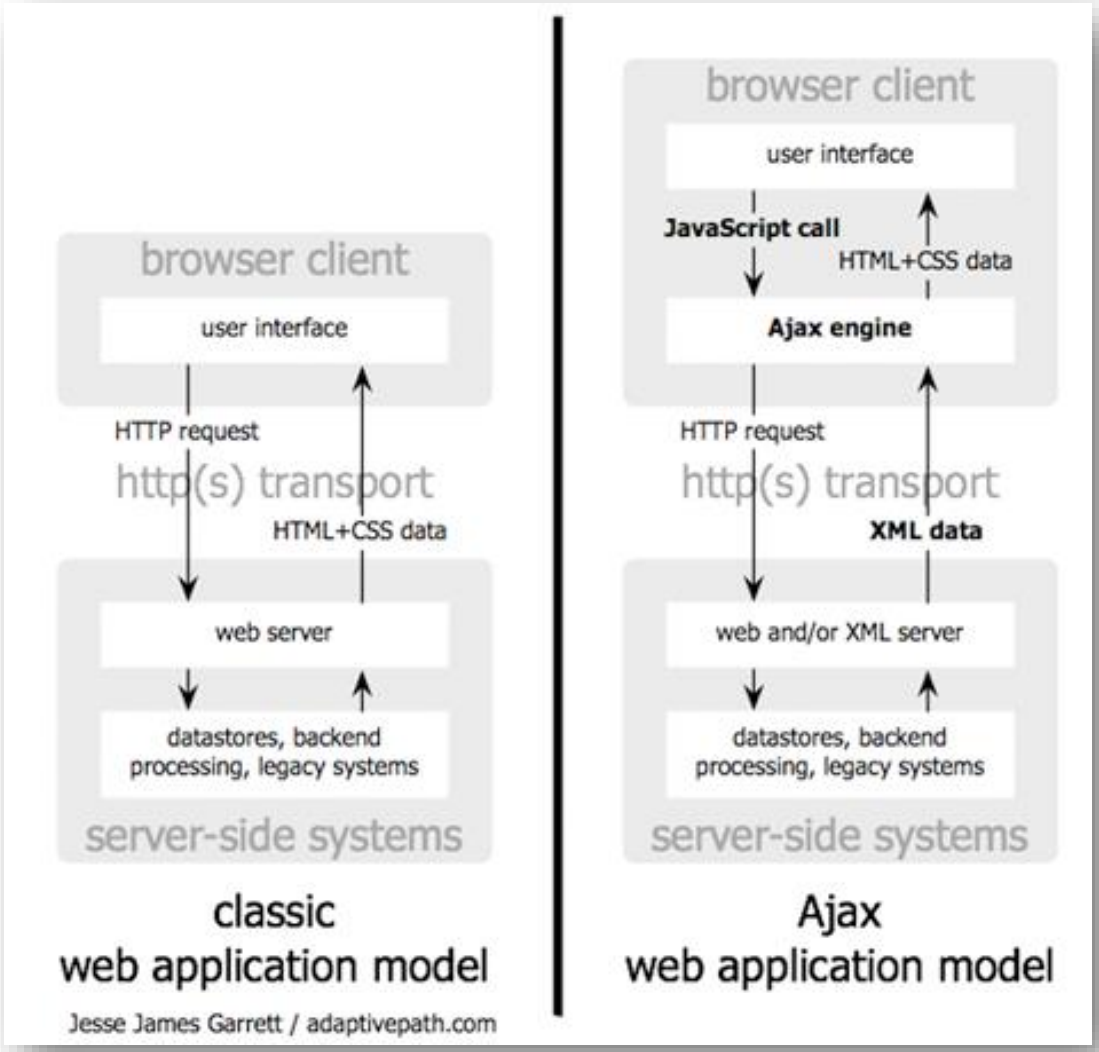
- Critically, all of the components have existed in some form since the late 1990's
  - An example of where browsers introducing non-standard features has been a positive thing


Google Maps
Satellite
Traffic
Transit
Bicycling
Terrain
Notifications
Location sharing
Your places
Your contributions
Your timeline
Share or embed map
Print
Take a tour
Language
Tips and tricks
Get help
Consumer information
Add a missing place
Send feedback
Search settings
River Kelvin
Ruchill Park
High Park Primary School
Firhill Stadium
Glasgow Botanic Gardens
Òran Mòr
HYNDLAND
DOWANHILL
HILLHEAD
KELVINBRIDGE
University of Glasgow
University of Glasgow Library
Kelvinbridge SPT Subway
St Georges Cross SPT Subway
Kelvinhall SPT Subway
Kelvingrove Art
Great Western Rd
Hyndland Rd
Highburgh Rd
Dumbarton Rd
Gibson St
Eldon St
Maryhill Rd
A82
A81
B808
Map data ©2018 Google
United Kingdom
Terms
Send feedback
200 m

Ajax – Origins

- Jesse James Garret coined the term in his 2005 essay:
  - "*Ajax: A New Approach to Web Applications*"
  - http://adaptivepath.org/ideas/ajax-new-approach-web-applications/
- Generated an enormous amount of hype and energy in web development ever since

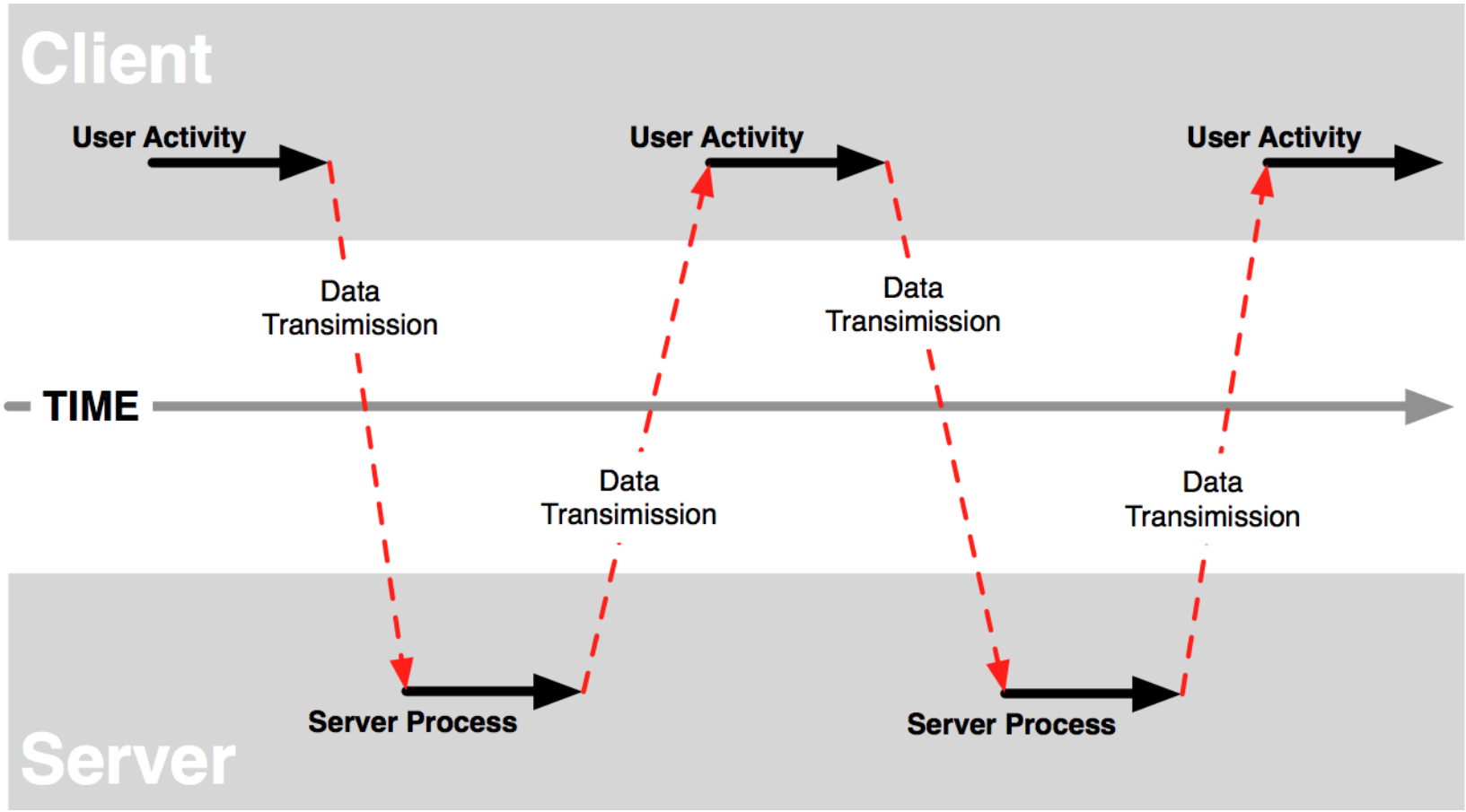
Ajax – Origins



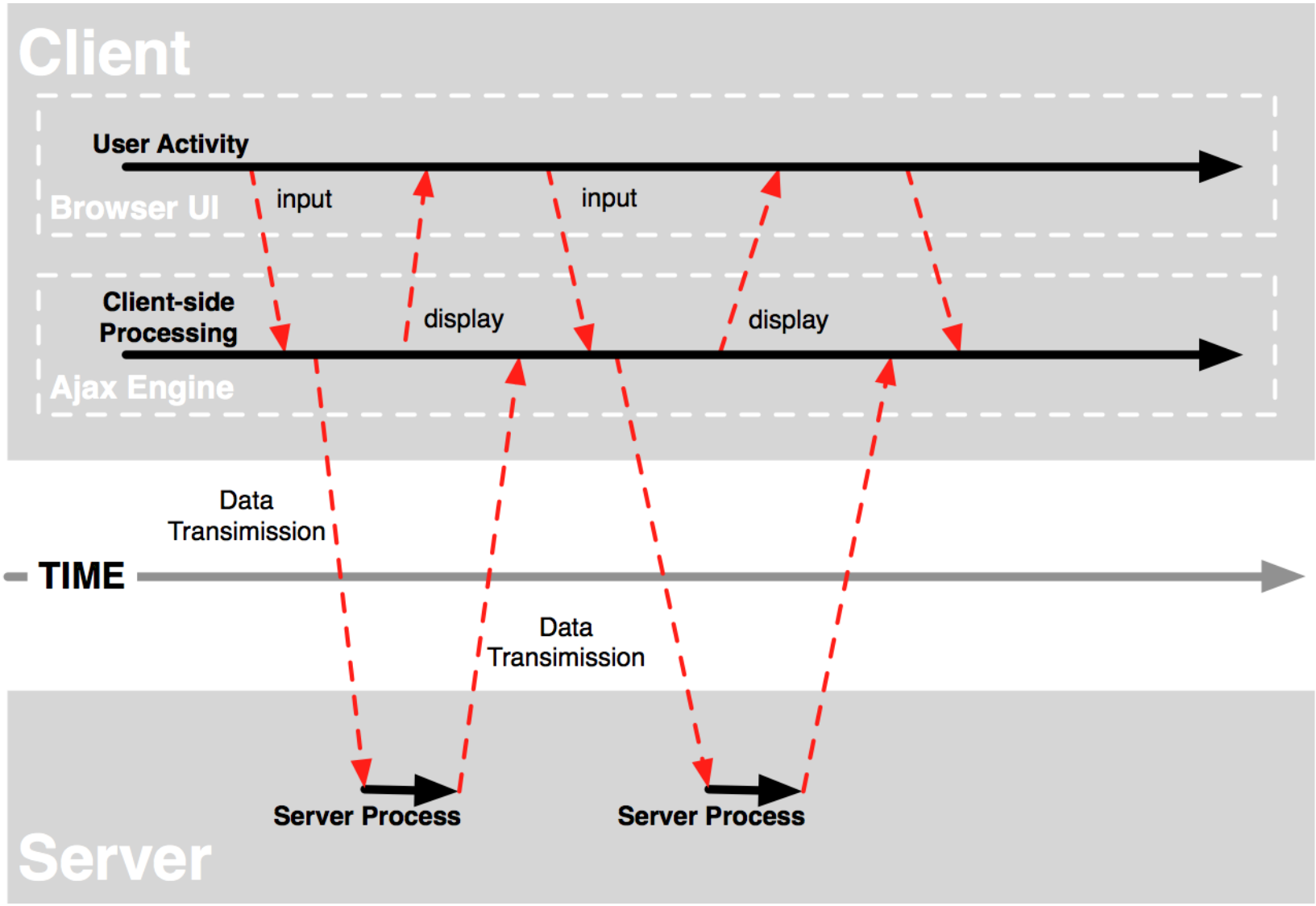
What does Ajax do?

- Ajax **eliminates the need to reload** a web page in order to get new content from the server
- This removes the **start-stop interaction** where a user has to wait for new pages to load.
- An intermediate layer (**Ajax Engine**) is introduced into the communication chain between client and server


The
Mechanics



Traditional Client/Server Synchronous Communication Model



AJAX Client/Server Asynchronous Communication Model

AJAX Components

- JavaScript can manipulate the DOM of a webpage to **create, modify and remove content and style**
- JavaScript event handlers can be attached to **events generated by the user** and browser
- XML can **model data** and we can access it using DOM
- So how does AJAX achieve asynchronous interaction and communication with the server?

XmlHttpRequest Object

- The keystone of AJAX
- Introduced by Microsoft in Internet Explorer 5
- The XHR is an object that is part of the DOM and is built into most modern browsers
- It can communicate with the server by sending HTTP requests (much like normal client/server communication)

XmlHttpRequestObject

- Independent of <form> or <a> elements for generating HTTP GET/POST requests
- It does not block script execution after sending an HTTP request
- As with **content** and **style**, JavaScript can now programmatically manage **HTTP communication**

How AJAX works

1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

https://adaptivepath.org/ideas/ajax-new-approach-web-applications/

XmlHttpRequest Properties

- **readyState** property
  - cycles through several states as it sends an HTTP request to the server, waits whilst the request is processed, and when it receives a response from the server (values 0-4)
- **onreadystatechange** property
  - accepts an EventListener value, specifying the method that the object will invoke whenever the readyState value changes
- **status** property
  - The status property represents the HTTP status code and is of type short (e.g. 200 = OK, 404 = Not Found)

XmlHttpRequest Properties

- **responseXML** property
  - represents the XML response data when the complete HTTP response has been received (when readyState is 4), and when the Content-Type header specifies the MIME (media) type as text/xml, application/xml, or ends in +xml
- **responseText** property
  - contains the text of the HTTP response received by the client
  - XML is not the only method to model data in Ajax applications. A popular alternative is JSON (JavaScript Object Notation)

responseXML

- Simple XML to model an address book entry

```
<Person>
    <firstName>John</firstName>
    <lastName>Smith</lastName>
    <age>25</age>
    <address>
        <streetAddress>21 2nd Street</streetAddress>
        <city>New York</city>
        <state>NY</state>
        <postalCode>10021</postalCode>
    </address>
    <phoneNumber type="home">212 555-1234</phoneNumber>
    <phoneNumber type="fax">646 555-4567</phoneNumber>
    <companyName />
</Person>
```

responseText (JSON)

- Same data as before, but uses fewer characters:

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {                                        // address object
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
    },
    "phoneNumbers": [                                   // array of objects
        { "type": "home", "number": "212 555-1234" },
        { "type": "fax", "number": "646 555-4567" }
    ],
    "companyName": null
}
```

XmlHttpRequest Methods

- **abort()**
  - Cancels the current request
- **open(method, url, async, user, psw)**
  - Specifies the request
    - method: the request type GET or POST
    - url: the file location
    - async: true (asynchronous) or false (synchronous)
    - user: optional user name
    - psw: optional password
- **send()**
  - Sends the request to the server
  - Used for GET requests
- **send(string)**
  - Sends the request to the server
  - Used for POST requests

XmlHttpRequest Methods (cont)

- **getAllResponseHeaders()**
  - Returns all header information of a resource such as length, server-type, content-type, last-modified
- **getResponseHeader()**
  - Returns specific header information such as “Last-Modified”
- **setRequestHeader()**
  - Adds a name/value pair to the header to be sent
  - Can be used with POST data to specify the type of data you want to send with the send() method

AJAX IN ACTION

AJAX Example – HTML header

```
<!DOCTYPE html>
<html>
  <head>
    <title>AJAX example</title>
        <script type="text/javascript">
            function loadDoc() {
                var xhttp = new XMLHttpRequest();
                xhttp.onreadystatechange = function() {
                    if (this.readyState == 4 && this.status == 200) {
                        document.getElementById("demo").
                         innerHTML = this.responseText;
                    }
                };
                xhttp.open("GET",  "ajax_info.txt", true);
                xhttp.send();
            }
        </script>
    </head>
```

AJAX Example – HTML Body

```
    <body>
    <div id="demo">
        <h2>Let AJAX change this text</h2>
        <button type="button" onclick="loadDoc()">Change Content</button>
    </div>
    </body>
</html>
```

- Contents of ajax-info.txt:

```
<h1>AJAX</h1>

AJAX is not a programming language.
<p>
AJAX is a technique for accessing web servers from a web page.
<p>
AJAX stands for Asynchronous JavaScript And XML.
```

Sending a Request

- To send a request to a server, we use the open() and send() methods of the XMLHttpRequest object:

```
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
```

- GET or POST?
  - GET is simpler and faster than POST, and can be used in most cases
  - However, always use POST requests when:
    - A cached file is not an option (update a file or database on the server)
    - Sending a large amount of data to the server (POST has no size limitations)
    - Sending user input (which can contain unknown characters), POST is more robust and secure than GET

Sending a Request (cont)

- If you want to send information with the GET method, add the information to the URL

```
xhttp.open("GET", "demo_get2.asp?
                        fname=Henry&lname=Ford", true);
xhttp.send();
```

- A simple POST request:

```
xhttp.open("POST", "demo_post.asp", true);
xhttp.send();
```

- To POST data like an HTML form, add an HTTP header with setRequestHeader(). Specify the data you want to send in the send() method:

```
xhttp.open("POST", "ajax_test.asp", true);
xhttp.setRequestHeader("Content-type",
                                "application/x-www-form-urlencoded");
xhttp.send("fname=Henry&lname=Ford");
```

Responses from the server

- **responseText**: get the response data as a string

```
document.getElementById("demo").innerHTML = xhttp.responseText;
```

- **responseXML**: get the response data as XML data

```
xmlDoc = xhttp.responseXML;
txt = "";
x = xmlDoc.getElementsByTagName("ARTIST");
for (i = 0; i < x.length; i++) {
      txt += x[i].childNodes[0].nodeValue + "<br>";
}
document.getElementById("demo").innerHTML = txt;
```

```
xhttp.open("GET", "cd_catalog.xml", true);
xhttp.send();
```

Callback functions

- A callback function is a function passed as a parameter to another function
- If you have more than one AJAX task in a website, you should create one function for executing the XMLHttpRequest object, and one callback function for each AJAX task
- The function call should contain the URL and what function to call when the response is ready

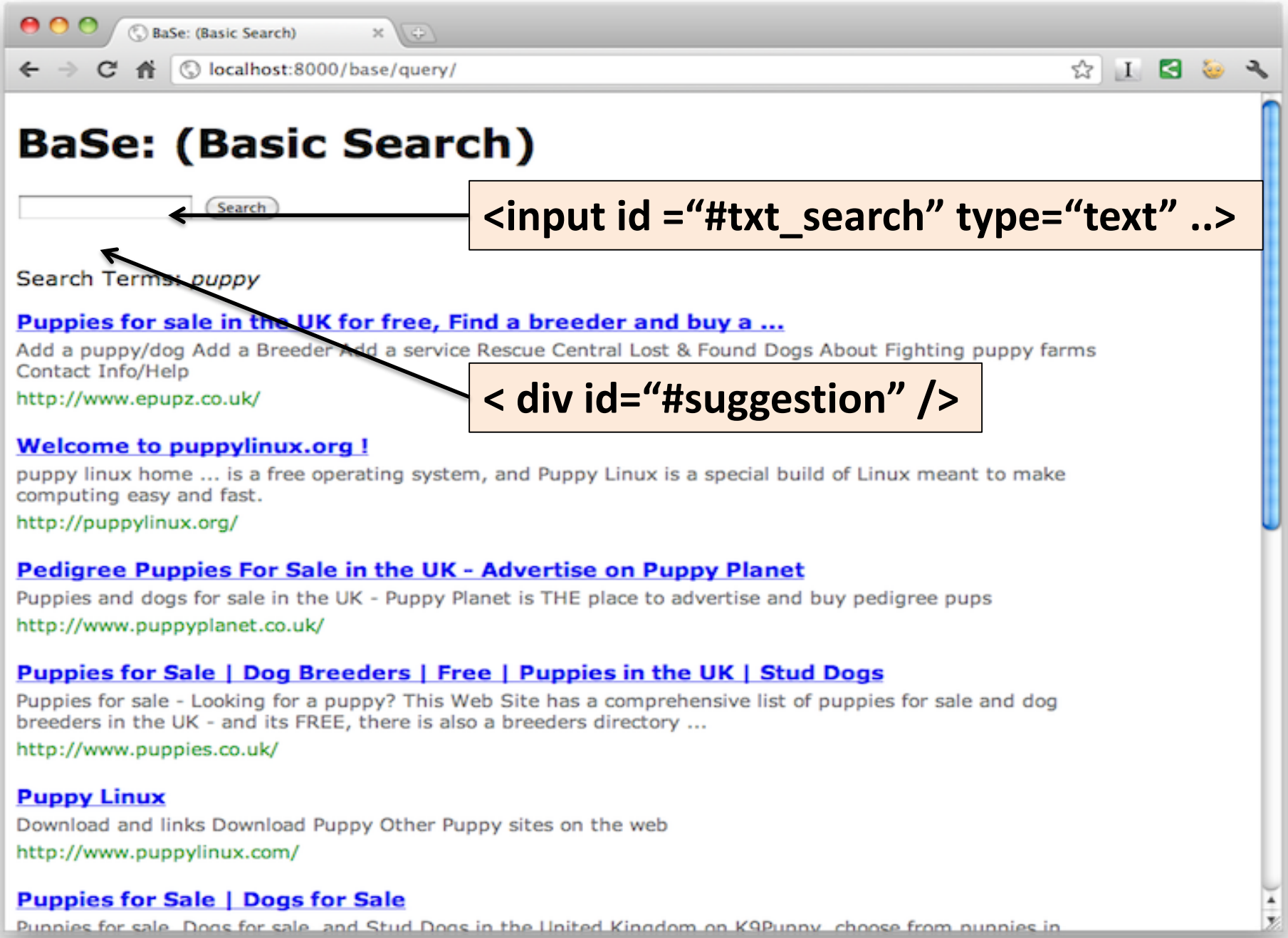
```
loadDoc("url-1", myFunction1);
loadDoc("url-2", myFunction2);

function loadDoc(url, cFunction) {
  var xhttp;
  xhttp=new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      cFunction(this); }}
  xhttp.open("GET", url, true);
  xhttp.send();
}

function myFunction1(xhttp) {
  // action goes here
}
function myFunction2(xhttp) {
  // action goes here
}
```

jQuery AJAX

- Simplifying AJAX in web applications

```
$.ajax({type: ”GET",
        url: ”greeting.php",
        data: ”name=" + name,
        success: function(message) {
          // do something
        }
});
```

SEARCH SUGGESTION EXAMPLE


BaSe: (Basic Search)
localhost:8000/base/query/
BaSe: (Basic Search)
Search
<input id =“#txt_search” type=“text” ..>
Search Terms: puppy
< div id=“#suggestion” />
Puppies for sale in the UK for free, Find a breeder and buy a ...
Add a puppy/dog Add a Breeder Add a service Rescue Central Lost & Found Dogs About Fighting puppy farms Contact Info/Help
http://www.epupz.co.uk/
Welcome to puppylinux.org !
puppy linux home ... is a free operating system, and Puppy Linux is a special build of Linux meant to make computing easy and fast.
http://puppylinux.org/
Pedigree Puppies For Sale in the UK - Advertise on Puppy Planet
Puppies and dogs for sale in the UK - Puppy Planet is THE place to advertise and buy pedigree pups
http://www.puppyplanet.co.uk/
Puppies for Sale | Dog Breeders | Free | Puppies in the UK | Stud Dogs
Puppies for sale - Looking for a puppy? This Web Site has a comprehensive list of puppies for sale and dog breeders in the UK - and its FREE, there is also a breeders directory ...
http://www.puppies.co.uk/
Puppy Linux
Download and links Download Puppy Other Puppy sites on the web
http://www.puppylinux.com/
Puppies for Sale | Dogs for Sale

Views.py

```
def suggest(request):
    """generate search suggestions"""
    # get suggestion so far from post
    search = request.POST['search']
    # search for pattern from list
    suggestion = ""
    suggestion_list = ["puppy dog", "cats hate
dogs", "raining cats and dogs"]
    for s in suggestion_list:
      if s.startswith(search):
        suggestion = s
    # return suggestion
    response = HttpResponse(suggestion)
    return response
```

Jquery Code

```
$(document).ready(function() {
     $("#txt_search").keyup(function() {
          var search = $("#txt_search").val();
          if (search.length > 0) {
               $.ajax({type: "POST",
                    url: "suggest/",
                    data: "search=" + search,
                    success: function(message) {
                      $("#suggest").empty();
                      if (message.length > 0) {
                        message = "Do you mean: " + message + "?";
                        $("#suggest").append(message);
                      }
                    }
            });
          } else {
            // Empty suggestion list
            $("#suggest").empty();
          }
        });
      });
```