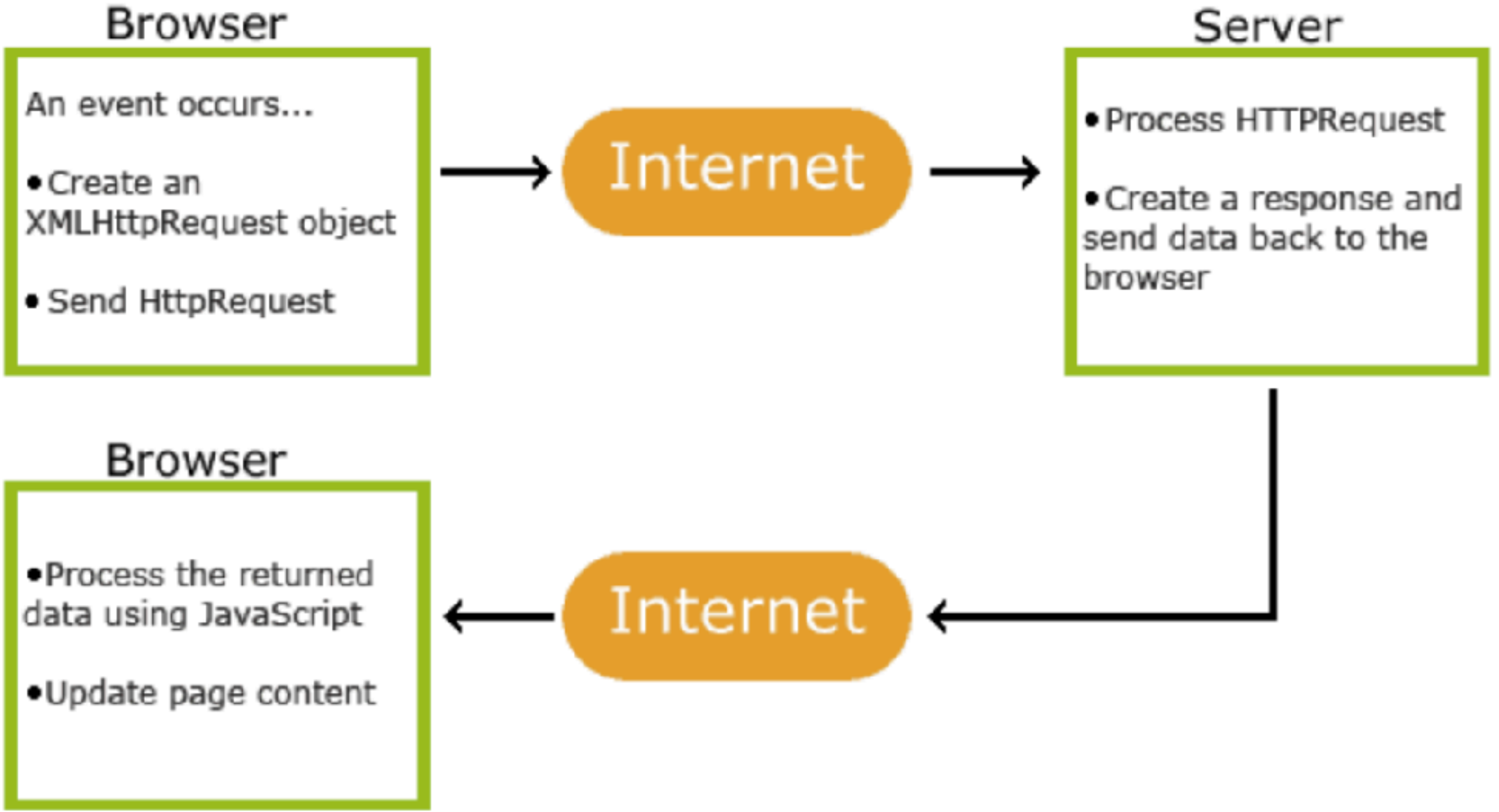
HOW TO PULL IN DATA

AJAX

WHAT	Asynchronous JavaScript And XML
HOW	Requests data from a server, uses JS to display data


Browser
An event occurs...
• Create an XMLHttpRequest object
• Send HttpRequest
Internet
Server
• Process HTTPRequest
• Create a response and send data back to the browser
Internet
Browser
• Process the returned data using JavaScript
• Update page content

AJAX

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

HOW DO THEY WORK TOGETHER?

- With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server and load the external data directly into the selected HTML elements of your web page!

WHAT ABOUT JQUERY AND AJAX?

HOW DO THEY WORK TOGETHER?

- With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server and load the external data directly into the selected HTML elements of your web page!

AJAX AND JQUERY

```
$.ajax({
        type: "POST, PUT, GET, or DELETE",
        url: "your endpoint here",
        data: { //Key value pairs of data here },
        success: function(data, textStatus, jqXHR) { },
        error: function(jqXHR, textStatus, errorThrown) { }
});
```

IN WHAT FORMAT IS THE DATA RECEIVED OR SENT?

ANSWER: JSON
WHAT IS JSON: A data type comprised of key-value pairs

JSON

```
var userInfo = {
        firstname: "Mel",
        lastname: "Serra",
        role: "Instructor",
        pets: [
                {
                        name: "Fido",
                        type: "Dog",
                        favorite_toy: "Tennis ball",
                        age: 7
                },
                {
                        name: "Kitty",
                        type: "Cat",
                        favorite_toy: "Feather on a stick",
                        age: 3
                }
        ],
        favorite_number: 10
};
```

API

API stands for "Application Program Interface", and technically applies to all of software design. However, since the explosion of information technology, the term now commonly refers to web URLs that can be accessed for raw data.

APIs publish data for public use. As third-party software developers, we can access an organization's API and use their data within our own applications.

WHY DO WE CARE?

Because why recreate data when we don't have to? Think about past projects or ideas that would be easier if you could pull in data already gathered elsewhere..

WHAT IS SERIALIZED DATA?

All data sent via HTTP are strings. Unfortunately, what we really want to pass between web applications is *structured data*, as in: native arrays and hashes. Thus, native data structures can be *serialized* into a string representation of the data. This string can be transmitted, and then parsed back into data by another web agent.

WHAT IS SERIALIZED DATA?

There are two major serialized data formats:

JSON stands for "JavaScript Object Notation", and has become a universal standard for serializing native data structures for transmission. It is light-weight, easy to read, and quick to parse.

```json
{
  "users": [
    {"name": "Bob", "id": 23},
    {"name": "Tim", "id": 72}
  ]
}
```

> Remember, JSON is a serialized format. While it may look like an object, it needs to be parsed so we can interact w
it as a true Javascript object.

WHAT IS SERIALIZED DATA?

XML stands for "eXtensible Markup Language", and is the granddaddy of serialized data formats (itself based on HTML). XML is fat, ugly, and cumbersome to parse. However, it remains a major format due to its legacy usage across the web. You'll probably always favor using a JSON API, if available.

```
<users>
  <user id="23">
    <name><![CDATA[Bob]]></name>
  </user>
  <user id="72">
    <name><![CDATA[Tim]]></name>
  </user>
</users>
```

WHERE DO WE FIND APIS?

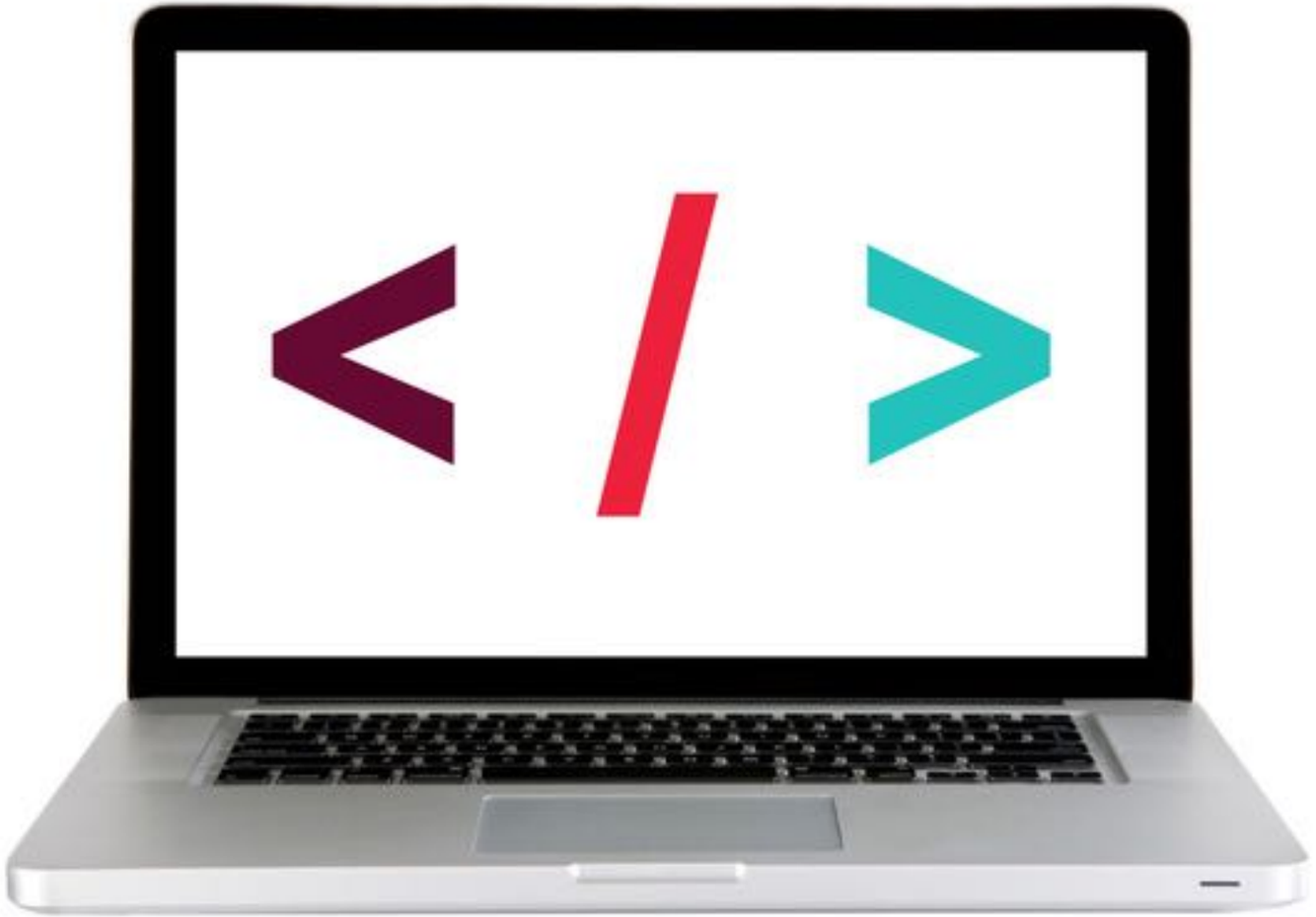
- APIs are published everywhere. Chances are good that most major content sources you follow online publish their data in some type of serialized format. Heck, [even Marvel publishes an API](http://developer.marvel.com/documentation/getting_started). Look around for a "Developers" section on major websites, or ask Google.
- Okay... you can also try the [Programmable Web API Directory](http://www.programmableweb.com/apis/directory) or the [Public APIs Directory](http://www.publicapis.com/).

WHAT IS AN API KEY?

- While the majority of APIs are free to use, many of them require an API "key" that identifies the developer requesting data access. This is done to regulate usage and prevent abuse. Some APIs also rate-limit developers, meaning they have caps on the free data allowed during a given time period.

- It is very important that you not push your API keys to a public Github repo.

LET'S TAKE A CLOSER LOOK



ACTIVITY

KEY OBJECTIVE

COME UP WITH AN ANALOGY

TIMING

5-10 MINUTES

ACTIVITY

FINAL PROJECTS!!!

PRESENTATIONS

GUIDELINES

Walk us through your site!

- Tell us about the features
- Describe the functionality
- How did you accomplish what you accomplished?
- What is the ultimate goal of the site?
- Is your design responsive? Show us!

Talk about your process

- What was tricky?
- What are you especially excited about being able to accomplish?
- Are there any known bugs or features you'd like to add in the future?

EXIT TICKETS