

CC Lab: Week 03

Object-oriented programming, JSON, AJAX,
pulling simple data

Instructor: Paweena Prachanronarong

Before we get started...

Do you guys know about lynda.com?

<https://www.lynda.com/libproxy.newschool.edu/Login/>

There's also [CodeAcademy](#)

HOMEWORK

DOWNLOAD FILES

<http://lovepaween.com/CCLab.zip>

Objects !

- A container to hold both data and behavior i.e. properties (a variable that belongs to an object) and functions
- The syntax for accessing the property (values associated with an object) of an object is:
`objectName.propertyName`
- To call a method (actions that can be performed on objects):
`objectName.methodName()`
- Uses `.syntax` because data is now inside the object
- Objects' properties and methods let us work with more important data

We know objects, remember?

Almost everything in JavaScript is an object!

- window
- document
- time
- arrays
- functions

We also know some built-in methods!

- `.write()` (helps write any given variable on the browser)
- `.getElementById()` (accesses the first element with the specified id)
- `.getElementsByTagName()` (accesses all elements with the specified tagname)

Creating objects

Syntax in JavaScript:

```
objectName.propertyName
```

```
objectName.methodName()
```

Longhand example:

```
var meal1 = new object();
```

```
meal1.appetizer = 'caprese salad';
```

```
meal1.main = 'grilled salmon';
```

```
meal1.dessert = 'strawberry shortcake';
```


Creating objects

Shorthand example:

```
var meal1 = {appetizer: 'caprese salad',  
             main: 'grilled salmon',  
             dessert: 'strawberry shortcake'  
};
```

**NOTE: Don't need "" on numbers or booleans

Objects in JavaScript:

Open File: 01-objects.html

```
//1. create objects
var meal1 = {  appetizer: 'caprese salad',
               main: 'grilled salmon',
               dessert: 'strawberry shortcake'};

var meal2 = {  appetizer: 'tom kha gai',
               main: 'pad thai',
               dessert: 'mango and sticky rice'};

var meal3 = {  appetizer: 'edamame',
               main: 'california roll',
               dessert: 'green tea ice cream'};

//2. define function
function menuInfo(){
  console.log(this.main);
  //use "this" because it refers to the current object
};

//3. give objects some methods, methods will call the function!
meal1.madeUpMethod = menuInfo;
//will call the function menuInfo associated with meal1

//4. call the object
meal1.madeUpMethod();
```

JSON

JavaScript **O**bject **N**otation

JSON: JavaScript Object Notation

- Text format for sharing data
- Uses JavaScript syntax
- Platform and language independent
- like XML, but faster (eXtensible Markup Language)
- Frequently used with AJAX (Asynchronous JavaScript and XML)

JSON strings

```
{ "japaneseRestaurants" : "Ennju",  
  "mexicanRestaurants" : "Dos Toros",  
  "burgerJoints" : "Shake Shack"  
}
```

- Note the " " on both Keys and Values
- No spaces in Keys
- JSON cannot be a function; can be a char, bool, string, number, array

Can also have lists, created with brackets

```
var info = //this line is JS, the rest is JSON
{"japaneseRestaurants" : "Ennju",
 "mexicanRestaurants" : "Dos Toros",
 "burgerJoints" : ["Shake Shack",
                   "McDonald's",
                   "Goodburger"]}
};
console.log(info.burgerJoints[1]);
```

What do you think the console will print?

JSON objects and arrays:

Open File: 02-json-objects.html

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Some restaurants in the hood</h2>
5
6     <p>
7       Japanese Restaurants: <span id="jjapanese"></span><br />
8       Mexican Restaurants: <span id="jmexican"></span><br />
9       Burgers: <span id="jburgers"></span><br />
10    </p>
11
12    <script>
13      var JSONObject= {
14        "japanese":"Ennju",
15        "mexican":"Dos Toros",
16        "burgers":[ "Shake Shack",
17                   "McDonald's",
18                   "Goodburger"]
19      };
20      document.getElementById("jjapanese").innerHTML=JSONObject.japanese
21      document.getElementById("jmexican").innerHTML=JSONObject.mexican
22      document.getElementById("jburgers").innerHTML=JSONObject.burgers[1]
23    </script>
24
25  </body>
26 </html>
```


AJAX

Asynchronous **J**avaScript **A**nd **X**ML

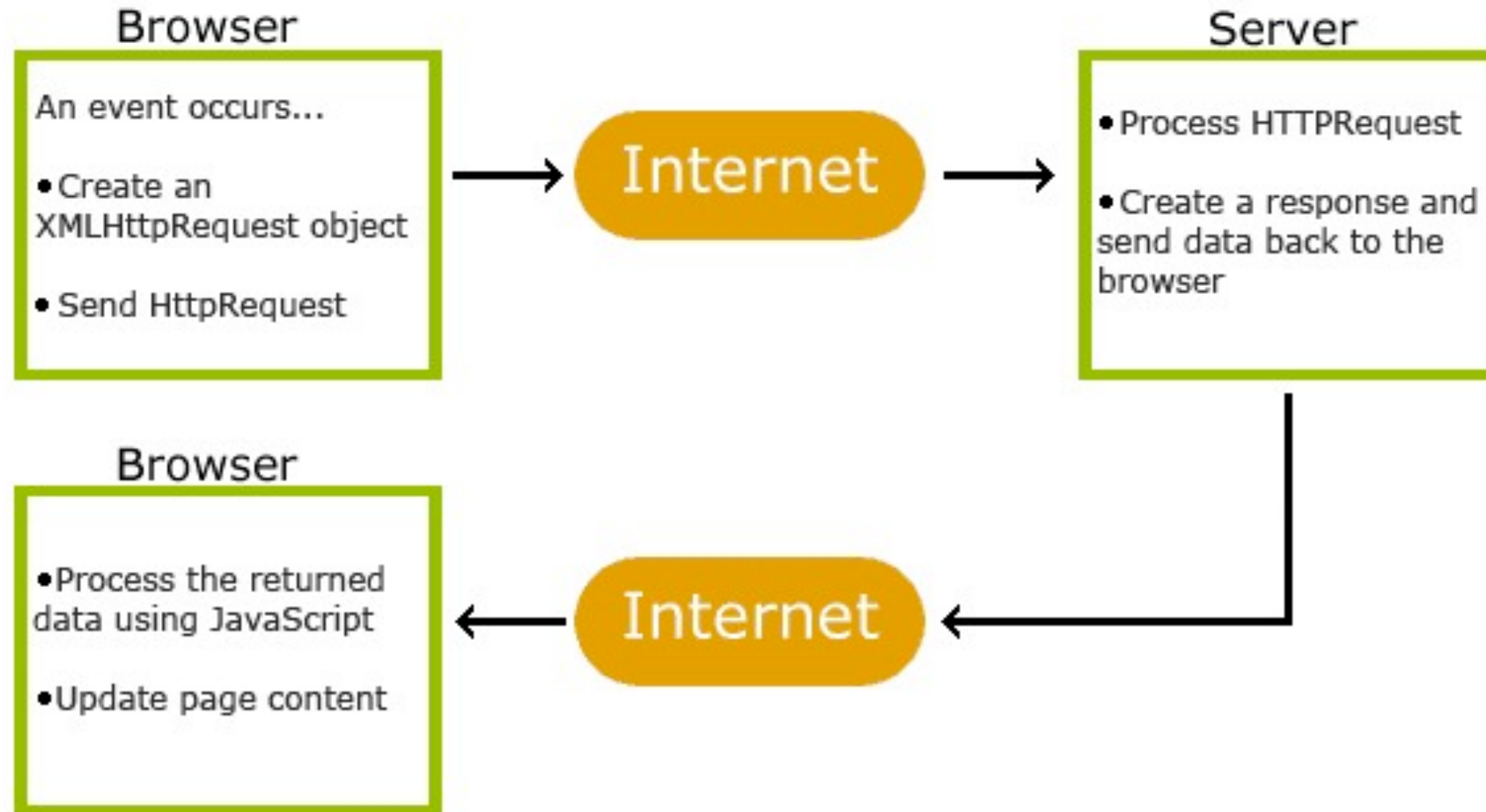
AJAX: Asynchronous JavaScript and XML

- AJAX can selectively access documents on a server and modify a part of a page displayed by the browser, and update it without the need to reload the whole document with all images, menus, etc.
- Asynchronous refers to events that are happening in the background independently of the main application flows
- AJAX is the communication between the server and client (browser) – you create a request and then let the server pass a response back
- Return in JSON format

Examples of AJAX

- Auto-complete like at google.com
- Social bookmarking sites like reddit
- Google Maps

How AJAX works



http://www.w3schools.com/ajax/ajax_intro.asp

**BRINGING 'EM
TOGETHER**

If you have a server, put the files on your server

AJAX requires a server, but allows you to get info without
visiting other page

Open the "data.json" file

What is going on here?

<http://www.atlantasushibar.com/files/attach/images/7322/335/007/4fb68a286b42bd8a60978cc0d765db6c.jpg>

```
1 {
2   "sushi": [
3     {
4       "japaneseName": "sake",
5       "englishName": "salmon",
6       "image": "01-sake.jpg",
7       "tastes": [
8         {
9           "taste1": "umami",
10          "taste2": "yummeRz"
11        }
12      ]
13    },
14    {
15      "japaneseName": "tobiko",
16      "englishName": "flying-fish roe",
17      "image": "02-tobiko.jpg",
18      "tastes": [
19        {
20          "taste1": "salty",
21          "taste2": "strong rice flavor",
22          "taste3": "seaweed",
23          "taste4": "fishy"
24        }
25      ]
26    },
27    {
28      "japaneseName": "inari",
29      "englishName": "fried tofu",
30      "image": "03-inari.jpg",
31      "tastes": [
32        {
33          "taste1": "vinegary",
34          "taste2": "light"
35        }
36      ]
37    }
38  ]
39 }
40 }
41 }
```

Open the
"mySushi.html"
file

```
13 //AJAX REQUIRES A SERVER BUT ALLOWS YOU TO GET INFO WITHOUT VISITING OTHER PAGE!!
14 //AJAX method loads in JSON FILE
15 $.ajax({//takes a number of parameters
16     type: 'GET',
17     url: 'data.json',
18     dataType: 'json',
19     //data: data,
20     success: jsonParser //success is if it loads
21 });//end of ajax
22 });//end of document ready
23
24 //now define a function
25 function jsonParser(json) { //argument of json
26 //name of file and function are arguments
27 //now works through the JSON FILE
28
29 $.each(json.sushi, function(k,v){//specifcying portion of file to loop through
30     var eachImage = 'images/' + v.image;
31     $('#container').append('<img src="' + eachImage + '>');
32
33     var eachEnglishName = v.englishName;
34     $('#theenglishname').append(eachEnglishName + '<br>');
35
36     var eachTaste = v.tastes[0];
37     $.each(eachTaste, function(k,v) {
38         });
39     });
40     console.log(json.sushi[1].tastes[0].taste1);
41
42 }
```


Side-by-side!

```
1  {
2    "sushi": [
3      {
4        "japaneseName": "sake",
5        "englishName": "salmon",
6        "image": "01-sake.jpg",
7        "tastes": [
8          {
9            "taste1": "umami",
10           "taste2": "yummeRz"
11          }
12        ]
13      },
14      {
15        "japaneseName": "tobiko",
16        "englishName": "flying-fish roe",
17        "image": "02-tobiko.jpg",
18        "tastes": [
19          {
20            "taste1": "salty",
21            "taste2": "strong rice flavor",
22            "taste3": "seaweedy",
23            "taste4": "fishy"
24          }
25        ]
26      },
27      {
28        "japaneseName": "inari",
29        "englishName": "fried tofu",
30        "image": "03-inari.jpg",
31        "tastes": [
32          {
33            "taste1": "vinegary",
34            "taste2": "light"
35          }
36        ]
37      }
38    ]
39  }
40 }
41 }
```

```
13 //AJAX REQUIRES A SERVER BUT ALLOWS YOU TO GET INFO WITHOUT VISITING OTHER PAGE!!
14 //AJAX method loads in JSON FILE
15 $.ajax({//takes a number of parameters
16   type: 'GET',
17   url: 'data.json',
18   dataType: 'json',
19   //data: data,
20   success: jsonParser //success is if it loads
21 }); //end of ajax
22 }); //end of document ready
23
24 //now define a function
25 function jsonParser(json) { //argument of json
26 //name of file and function are arguments
27 //now works through the JSON FILE
28
29 $.each(json.sushi, function(k,v){//specifcying portion of file to loop through
30   var eachImage = 'images/' + v.image;
31   $('#container').append('<img src="' + eachImage + '>');
32
33   var eachEnglishName = v.englishName;
34   $('#theenglishname').append(eachEnglishName + '<br>');
35
36   var eachTaste = v.tastes[0];
37   $.each(eachTaste, function(k,v) {
38     });
39 });
40 console.log(json.sushi[1].tastes[0].taste1);
41
42 }
```


Check out my site. It's so beautiful!


Chrome File Edit View History Bookmarks Window Help Tue 1:13 PM Paweena...ronarong


Types of sushi - grabbing x


lovepaweena.com/CCLab/mySushi.html

Design For This Cer Getting Started W3Schools Online W MFA D+T Bootcamp http://a.parsons.ed talk2myShirt Design and Technol Other Bookmarks

Mmmmm....Sushiiiiii

 Sake

 Tobiko

 Inari

salmon
flying-fish roe
fried tofu

Elements Resources Network Sources Timeline Profiles Audits Console

```
ok mySushi.html:12
salty mySushi.html:40
GET http://lovepaweena.com/CCLab/is/iquery-1.10.2.min.map 404 (Not Found) is/iquery-1.10.2.min.map:1
> |
```

API

Application **P**rogramming **I**nterface

API: Application Programming Interface

"A means of putting information in and getting information out of your system without having to type it yourself."

<http://www.headshift.com/our-blog/2009/06/03/whats-an-api-and-why-do-i-want/>

"In practice, most often an API is a library that includes specifications for routines, data structures, object classes and variables.

An object API is a prescription of how objects work in a given object-oriented language."

http://en.wikipedia.org/wiki/Application_programming_interface

API: Application Programming Interface

You can use AJAX to parse through information (as JSON objects) found in APIs.

Example of API url:

http://api.wunderground.com/api/Your_Key/conditions/q/CA/San_Francisco.json

SIGN UP FOR A KEY

<http://www.wunderground.com/weather/api/d/docs>

An API for wunderground could return data like this (see as JSON):

```
1  { "response":
2    { "version": "0.1", "termsofService": "http://www.wunderground.com/weather/api/d/terms.html",
3      "features": {
4        "astronomy": 1 }
5    },
6
7    "moon_phase": {
8      "percentIlluminated": "81",
9      "ageOfMoon": "10", "current_time": {
10       "hour": "9", "minute": "56"
11     },
12
13     "sunrise": {
14       "hour": "7",
15       "minute": "01"
16     },
17
18     "sunset": {
19       "hour": "16",
20       "minute": "56"
21     }
22   } //end of moon phase
23 }
```


Open "astronomyAPI.html" and put in your key

```
19 $(document).ready(function(){
20   $.ajax({
21     url: "http://api.wunderground.com/api/Your_Key/astronomy/q/CA/San_Francisco.json",
22     dataType : "jsonp", //jsonp for cross-origin files
23     success : function(parsed_json) {
24       console.log('success!!');
25       //each nested object goes in []
26       var moonIllum = parsed_json['moon_phase']['percentIlluminated'];
27       var moonAge = parsed_json['moon_phase']['ageOfMoon'];
28       var hourNow = parsed_json['moon_phase']['current_time']['hour'];
29       var minuteNow = parsed_json['moon_phase']['current_time']['minute'];
30
31       //var sunset = parsed_json['moon_phase']['sunset']['hour'];
32       console.log("percent illuminate of moon= "+moonIllum+"%");
33       console.log("The moon's age = "+ moonAge);
34       console.log("The current time= "+ hourNow+": "+minuteNow);
35
36       //console.log(parsed_json); //great debugging tool here!
37
38
39       $("#container").append("The age of the moon="+moonAge);
40
41     } //end of success
42   }); //end of ajax
43 }); //end of doc ready
```

Homework: Take it further!

1. If you haven't yet, add a profile photo to the class blog.
2. If you haven't yet, order Arduino kits!
3. Pick an API and do something cool with it.

BONUS: Use the variables you've created to parse through data (like moonAge and hourNow as shown in the above picture) and insert them into the canvas animation from Week 02 to create a data visualization!

4. Post code/screenshot/URL and references to the blog (**can include people too!**).
5. Write about one thing you learned on your own on the blog.

Good API tutorial:

- <http://www.codecademy.com/tracks/apis>

References:

- <http://www.w3schools.com/json/>
- <http://www.w3schools.com/ajax/default.asp>
- <http://www.noupe.com/ajax/how-ajax-works.html>
- <http://jsonlint.com/>