

Bridging the Gap

Objects, JSON, and AJAX

Week 5

Hello, JavaScript Objects!

- Object oriented programming in JavaScript is NOT like Java
- For now, simple collection of variables relating to a particular entity
- Playing Cards Example
 - Most basic approach uses only variables
 - Can't use arrays for suits and ranks – No connection!
 - Card Object (rank and suit)
- Can initialize empty object and set values later
- Can make array of cards

Generalizations

- Create empty object = {};
- Add variables using .notation = “some value”;
 - Can be any type – Not typesafe remember?
- Define object literals = {“name” : “My Object”, “someProp” : “prop”}
- Null means no (object)

JSON – JavaScript Object Notation

- JavaScript object literal in string form
- Can parse as object - `myObj = JSON.parse('{ "some" : "json" })'`
- Can stringify object to JSON – `JSON.stringify(myObj)`

AJAX – Asynchronous JavaScript And XML

- Commonly used XML before JSON became popular
- Basic idea is sending and receiving data without reloading the page
- Gmail example

Accessing an external json file

- Create basic skeleton application (i.e. public/index.html, public/javascripts/app.js)
 - Ignore css for now
- Create directory inside public called cards
- Create file *aceOfSpades.json*
- Populate with

```
{  
  "rank" : "ace",  
  "suit" : "spade"  
}
```

Accessing an external json file

- Need to circumvent browser security restrictions (XSS)
 - Cannot access local files from JavaScript
 - Must be served
 - Mac
 - `open -a Google\ Chrome --args --allow-file-access-from-files`
 - Windows
 - `chrome.exe --allow-file-access-from-files`
 - Get json and add to page as `<p>card.rank + " of " + card.suit</p>`
- ```
$.getJSON("cards/aceOfSpades.json", function (card) {
 console.log(card);
})
```

# JSON Arrays

- Create hand.json

```
[
 {"suit" : "spades", "rank" : "3"},
 {"suit" : "hearts", "rank" : "ace"},
 {"suit" : "diamonds", "rank" : "10"},
 {"suit" : "clubs", "rank" : "7"},
]
```

- Use `getJSON` and `forEach` to create a `<ul>` with each `<li>` containing one card's information



# Getting Images from Flickr

- Create new skeleton application in Flickr directory
- Preview JSON response from API
  - `api.flickr.com/services/feeds/photos_public.gne?tags=dogs&format=json`
- getJSON from API
- forEach to add `<img>` to page
- Now make them fade in