

# HTML 300 Class 06

Working with JSON

#### Schedule

- 1. Homework Review
- 2. Intro to JSON
- 3. Javascript -> HTML
- 4. HTML -> Javascript
- 5. Practice

### **Homework Example**

https://github.com/tatsuroaoki/florida-energy/commits/master

Look at that commit history! It's going to be so easy to go back and find changes as necessary.

### **Homework Example**

https://github.com/maryfranceshull/florida\_energy\_redesign/blob/master/styles/index.scss

Sometimes you've got to break the rules, or leave something not quite finished. Comments help explain the reasoning to future devs, making their work simpler.

### Intro to JSON

### Setup

Create a barebones HTML file. Add an <h1> and a <script> tags in the body.

#### Intro

Javascript Object Notation is a language independent format used to transmit data objects as attribute-value pairs.

It's very common format for asynchronous browser–server communication.

#### Context

JSON is used both to represent objects in Javascript programs, and to communicate information between web APIs and Javascript programs.

Very similar usage, except that object representation allows functions as values.

We'll be focusing on how it will be used with APIs.

### Example

```
var house = {
   "exterior": "wood siding",
   "roof": "spanish tile",
   "bedrooms": 2,
   "bathrooms": 2,
   "sqft": 1200,
   "hasYard": true
}
```

#### {attribute name : value}

Attributes are strings.

Values can be:

- Objects { ... }
- Arrays [ ... ]
- strings
- numbers
- boolean values
- null

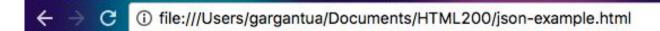
#### Example

```
var house = {
  exterior: "wood siding",
  roof: "spanish tile",
  bedrooms: 2,
  bathrooms: 2,
  sqft: 1200,
  hasYard: true
```

Because attributes are always strings, **Javascript syntax** allows you to omit the quotes around them.

Add this to <script> in your file.

#### **Chrome JS Console**



#### Let's Play House

```
Elements Console Sources Network Timeline Profiles Application Security Audits Adblock Plus

or top ▼ □ Preserve log

house

or Object {exterior: "wood siding", roof: "spanish tile", bedrooms: 2, bathrooms: 2, sqft: 1200...}
```

## **Getting Attributes**

Dot notation

house.exterior

house.hasYard

## **Setting Attributes**

Dot notation, set to new value

house.exterior = "brick"

house.hasYard = false

#### **Creating Attributes**

Dot notation, set attributes that hadn't previously been part of the object

house.garage = false

house.heatingType = "forced air"

```
house
bathrooms: 2
    bedrooms: 2
    exterior: "wood siding"
    garage: false
    hasYard: true
    heatingType: "forced air"
     roof: "spanish tile"
     sqft: 1200
```

#### HTML <-> JSON

Communication can go both ways:

We can dynamically build HTML from JSON.

We can build new JSON objects from HTML form input.

# Javascript -> HTML

### Loop Through Object Attributes

In this context, "attribute" and "key" mean the same thing.

```
for (var key in house) {
  console.log("Key: ", key, "Value: ", house[key])
}
```

## **Build HTML from Object Attributes**

- 1. Loop through the attributes, as seen on previous slide
- 2. For each attribute, create a tag with the key and value nicely formatted inside. Then append the to the <body> element.

## **Build HTML from Object Attributes**

Expected output to page:

exterior: wood siding

roof: spanish tile

bedrooms: 2

bathrooms: 2

sqft: 1200

has Yard: true

### **Arrays of Objects**

```
let neighborhood = [
    exterior: "vinyl siding",
    roof: "tile"
    exterior: "wood shake",
    roof: "steel panels"
    exterior: "stucco",
    roof: "asphalt shingles"
```

### **Build HTML From an Array of Objects**

- Loop through the elements of the array
- For each element, do the same thing we did for a single object

### **Build HTML From an Array of Objects**

Context: This is an extremely common task done with data received from a web API.

#### **Build HTML From Objects**

This feels like a good chunk of code to extract into a function, so it can be reused more easily!

# HTML -> Javascript

#### **Create JSON From HTML Form**

- 1. HTML form with fields for desired object attributes
- 2. JS has a globally available array to hold objects
- 3. Form handler function builds new object with form attributes, pushes it into array
- 4. Update HTML as appropriate with new data

#### **Let's Build Houses**



What fields will our form need? What types of inputs are most appropriate?

Let's code it!

### **Practice**

### **A New Feature For Our Site**

We'll build a slideshow of all the houses in the neighborhood.







#### **Slideshow**

#### Feature description:

- Page displays information about one house at a time (incl. image from new attr we'll add)
- Arrows display to the left and right of the house information
- Clicking an arrow changes which house is displayed, moving through the neighborhood

#### The Plan...

Let's talk through our plan for how to build this feature. Then, we may build on our own or together.

#### More...

The MDN info page for objects:

https://developer.mozilla.org/en-US/docs/Web/JavaS cript/Guide/Working with Objects

Find Creative Commons licensed images:

https://creativecommons.org/use-remix/

### Homework

## **Reading Documentation**

In preparation for next week's lesson, read documentation for Google Charts and write up the information requested.

Finish any remaining HTML & CSS work necessary for redesign project!

# Nice One!

