

MODULE THREE - OBJECTS

PROTOTYPES & INHERITANCE & JSON



TODAY'S SCHEDULE

1. Cool Things
2. Review Constructor Functions
3. Prototypes & Inheritance
4. A Classier Way ...
5. Do You Know JSON?
6. Recap, Weekly Tasks, Next Week

LEARNING OBJECTIVES

1. Explore and demonstrate understanding of object prototypes, the prototype chain, using the prototype property and inheritance
2. Explore and use ECMAScript 2015 class syntax
3. Employ the most common functions and methods used with current client-side JavaScript techniques;
4. Optimize code for increased functionality, performance, readability, and reusability

COOL THINGS!

```
3 require File.expand_path("../config/environment", __FILE__)
4 # Prevent database truncation if the environment is production
5 abort("The Rails environment is running in production mode!")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'copybara/rspec'
10 require 'copybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with framework :rspec
17     with library :rails
18   end
19 end
20
21 # Add additional requires below this line. Note that require is relative to the
22 # spec/support/ and its subdirectories. Files starting with "rspec" will be
23 # run as spec files by default. Files starting with "rails" will be run
24 # in _spec.rb will both be required as rails files, and one will be
25 # run twice. It is recommended that you use the "rails" prefix for
26 # end with _spec.rb. You can configure the require paths by adding
27 # option on the command line as follows:
28 # rails spec --require=spec_helper --require=spec_helper2
29
30 No results found for 'mongoid'
31
32 mongoid
33
34 buffer
```

RESOURCES, LINKS TUTORIALS AND OTHER COOL THINGS...

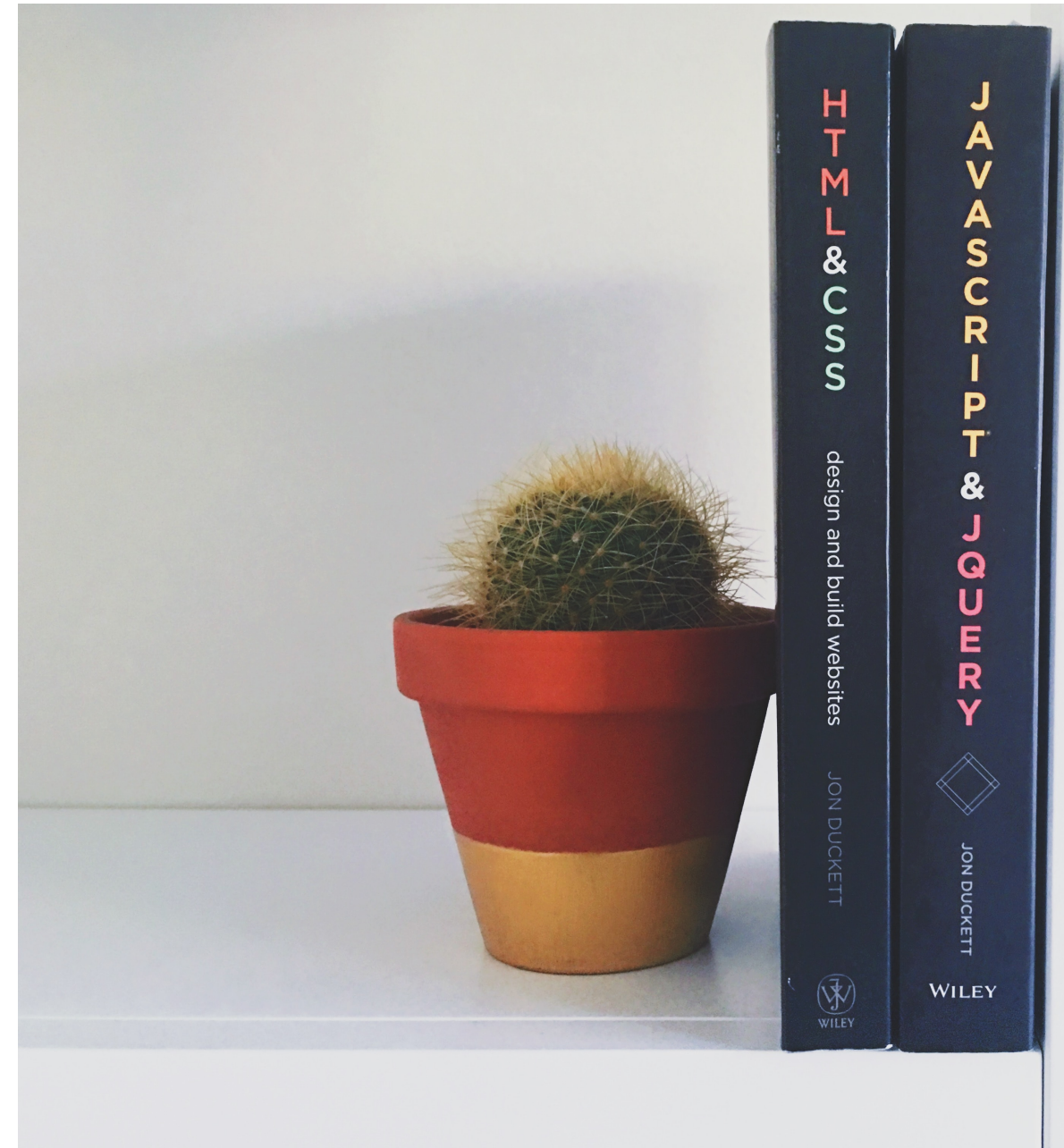
- ▶ <https://www.freecodecamp.org/news/is-vanilla-javascript-worth-learning-absolutely-c2c67140ac34/>
- ▶ <https://medium.com/better-programming/9-projects-to-inspire-front-end-developers-in-2020-a404545f6369>
- ▶ <https://learnvanillajs.com/>
- ▶ <https://dzone.com/articles/hiring-my-cat-as-a-software-developer>

LET'S REVIEW OOJS

```
3 require File.expand_path("../config/environment", __FILE__)
4 # Prevent database truncation if the environment is production
5 abort("The Rails environment is running in production mode!")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with.test_framework :rspec
17     with.library :rails
18   end
19 end
20
21 # Add additional requires below this line. See the README for more info.
22
23 # Requires supporting ruby files with custom matchers and matchers
24 # spec/support/ and its subdirectories. Files starting with "rspec"
25 # run as spec files by default. Files starting with "helper"
26 # in _spec.rb will both be required and run as spec files.
27 # run twice. It is recommended to use "spec_helper" for
28 # end with _spec.rb. You can configure the require helper
29 # option on the command line using --rspec-require-helper
30
31 No results found for 'mongoid'
```


!IMPORTANT OOP CONCEPTS

- ▶ **Abstraction** : creating a simple model of a more complex thing
- ▶ **Encapsulation** : our object becomes a container or capsule for properties and methods
- ▶ **Instantiation**: creating an object instance from a class
- ▶ **Polymorphism**: the ability for multiple objects to implement the same functionality



OOP + JS = OOJS


```
function Kitty(first, last, age, gender, interests) {
  this.name = {
    first : first,
    last : last
  };
  this.age = age;
  this.gender = gender;
  this.interests = interests;
  this.bio = function() {
    alert(this.name.first + ' ' + this.name.last + ' is ' + this.age + ' years old. He likes ' + this.interests[0] + ' and '
      + this.interests[1] + '.');
  };
  this.greeting = function() {
    alert('Hi! I\'m ' + this.name.first + '.');
  };
}
```

```
let bart = new Kitty('Bart', 'Meowser', 16, 'male', ['catnip', 'chasing strings', 'snoozing'] );  
let stevie = new Kitty('Stevie', 'Nicks', 9, 'female', ['strings', 'sleeping', 'eating']); |
```


WHAT'S GOING ON HERE?

- ▶ constructor functions help us to create object templates
- ▶ we can instantiate new object instances by calling the constructor function, using the new keyword & passing through the required arguments

CONSTRUCTOR FUNCTIONS

The **constructor function** allows us to create a class. It's pretty much like a standard function, except we aren't creating an empty object or returning it.

CONSTRUCTOR FUNCTIONS

Constructors help to give our code **order**. We can create constructors in one place, then **create object instances** when we need them.

COOL, BUT...

- one **small drawback** is that every time we call our constructor function, we are defining `greeting()` every time
- Don't worry, we'll improve on this when we talk about **prototypes and inheritance**.

PROTOTYPES & INHERITANCE

```
3 require File.expand_path("../support/spec_helper.rb", __FILE__)
4 # Prevent database truncation if the environment is production
5 abort("The Rails environment is running in production mode!")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with :rspec, :framework
17     with :lib, :rails
18   end
19 end
20
21 # Add additional requires below this line. Examples of supported
22 # Requires supporting ruby files with support/ and its subdirectories.
23 # spec/support/ and its subdirectories. Files matching spec/support/*_spec.rb
24 # will be added as spec files by default. This can be configured using
25 # run as spec files by default. This can be configured using
26 # run as spec files by default. This can be configured using
27 # run as spec files by default. This can be configured using
28 # run as spec files by default. This can be configured using
29 # run as spec files by default. This can be configured using
30 # run as spec files by default. This can be configured using
```

No results found for 'mongoid'

mongoid

buffer

A PROTOTYPE-BASED LANGUAGE

- ▶ JavaScript is often described as a
prototype-based language
- ▶ prototypes are the mechanism by which
objects inherit features from one another in
JS

HOW DOES THIS WORK?

- ▶ objects have a **prototype object** (acts a template object that it inherits methods and properties from)
- ▶ an object's prototype object may also have a prototype object, creating a **prototype chain**

PROTOTYPAL INHERITANCE

- ▶ a link is made between the object instance and its prototype, which is derived from the **prototype property on the constructor**
- ▶ the properties and methods are found by **walking up the chain of prototypes**

PROTOTYPAL INHERITANCE



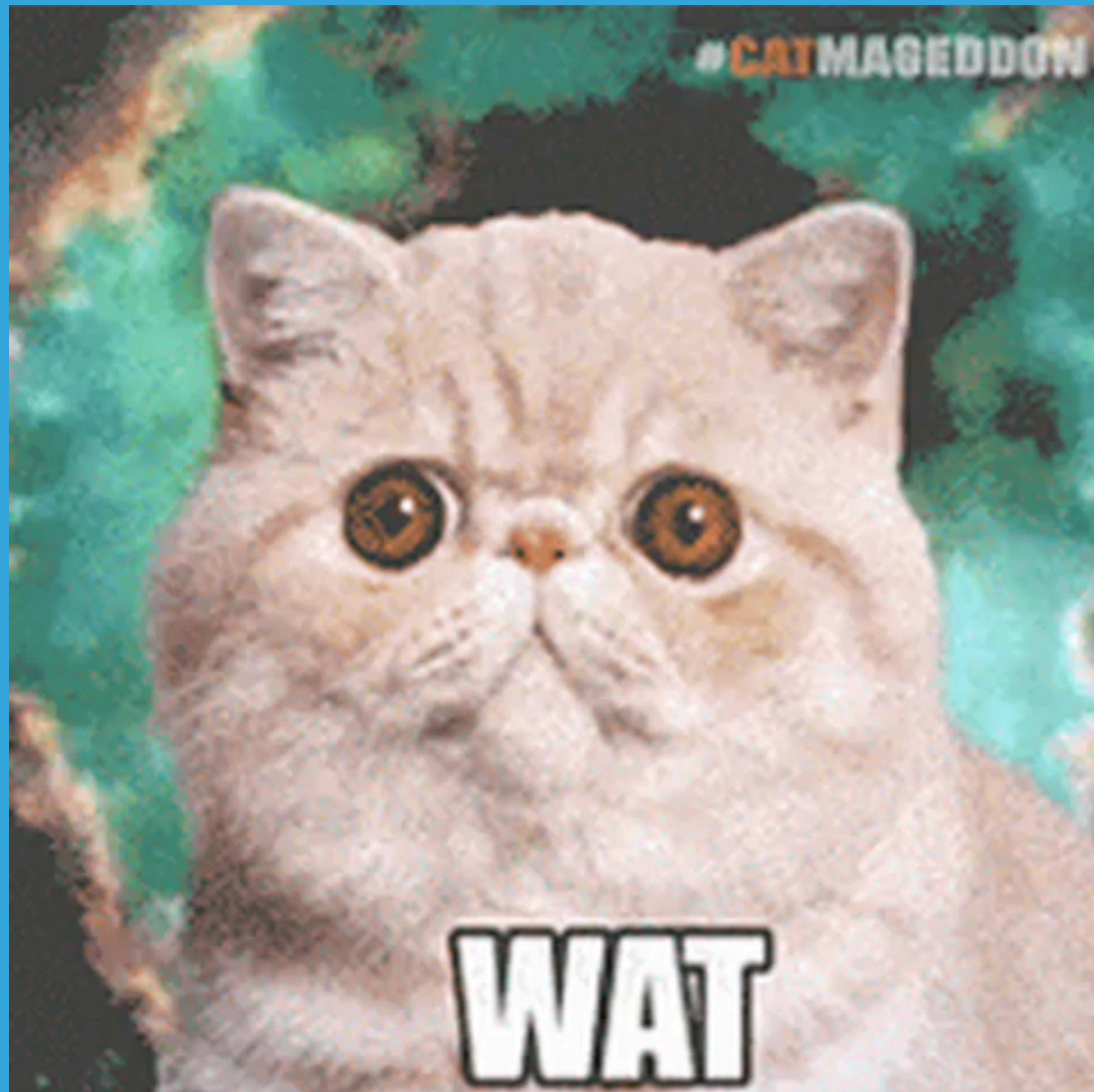
SOURCE : https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object_prototypes

THE PROTOTYPE CHAIN

- ▶ When it comes to object prototypes, properties and methods are defined on the prototype property of the Object's constructor function, **not the object instances themselves.**
- ▶ Methods and properties are **not** copied from one object to another in the prototype chain.

MODIFYING PROTOTYPES

- ▶ We can **modify the prototype property** on our constructor function
- ▶ Good idea to **add our methods to the prototype property object** of our constructor function instead of inside the constructor function itself



Weekly Learning >

Week 9 >

Code Examples >

Week 9 Start > Example One

A CLASSIER WAY

```
3 require File.expand_path("../config/environment", __FILE__)
4 # Prevent database truncation if the environment is production
5 abort("The Rails environment is running in production mode!")
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with framework: :rspec
17     with library: :rails
18   end
19 end
20
21 # Add additional requires below this line. Examples:
22 #
23 # Requires supporting ruby files with custom matchers in spec/support/
24 # spec/support/ and its subdirectories. Files matching spec/support/*_matcher.*
25 # in _spec.rb will both be required by default and added to the list of
26 # files to run twice. It is recommended that you not use this feature
27 # end with _spec.rb. You can configure the behavior with the
28 # option on the command line. For example:
29 #
30 # mongoid
31 #
32 # buffer
```


JAVASCRIPT GOT CLASSY!

- ▶ ECMAScript 2015 introduced **classes** to JavaScript
- ▶ **cleaner and easier syntax**, same old prototypal inheritance
- ▶ **subclasses, getters and setters**

Weekly Learning >

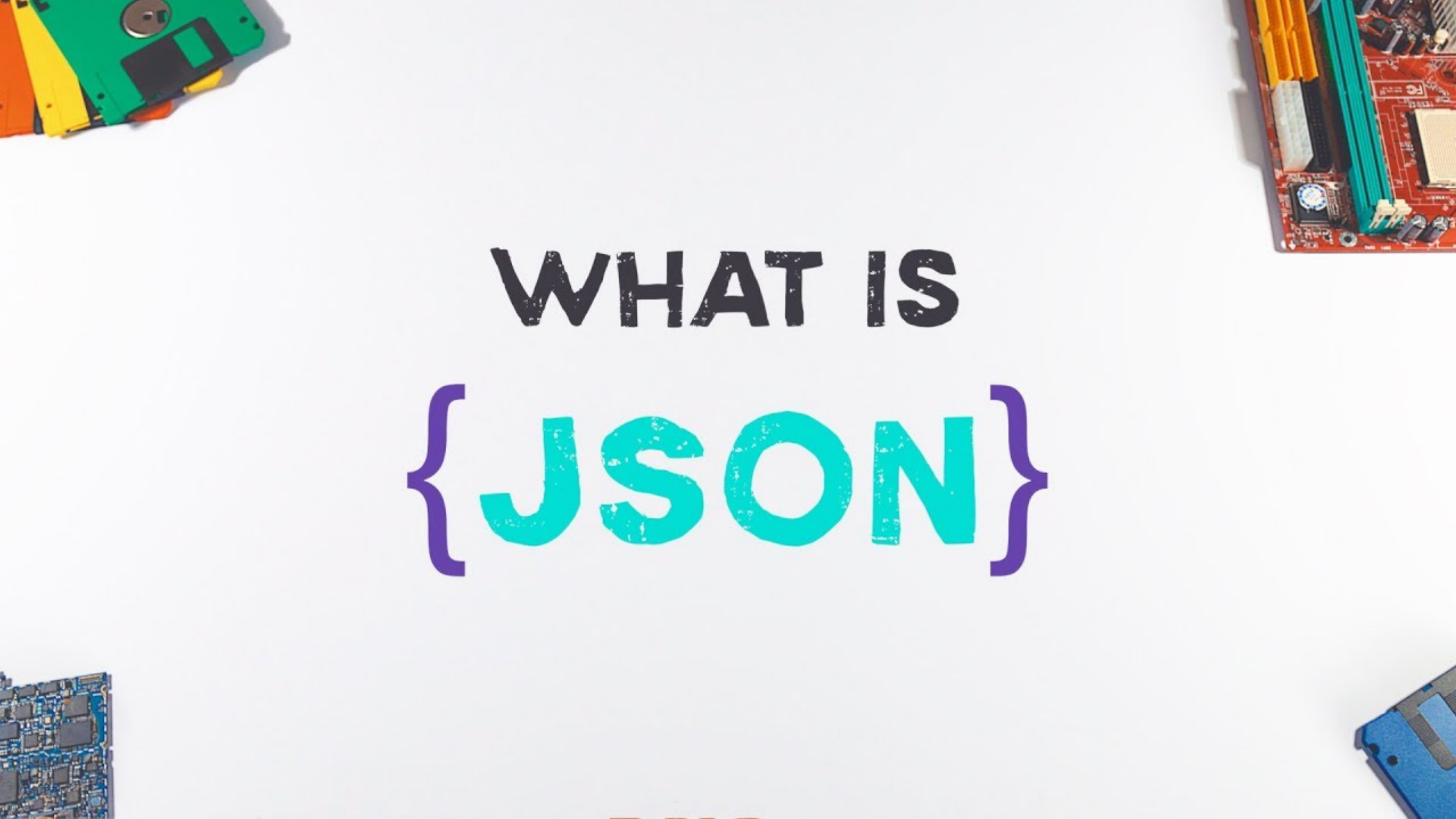
Week 9 >

Code Examples >

Week 9 Start > Example Two



DO YOU KNOW JSON?



WHAT IS {JSON}

Weekly Learning >

Week 9 >

Code Examples >

Week 9 Start > Example

Three

RECAP, THIS WEEK, NEXT WEEK

OOP

- ▶ we use objects to **model** real world things in our programs/ provide easy access to **functionality**
- ▶ objects contain related data and code which **represent info or functionality**

OOJS

- ▶ We can create define object templates using a **constructor function** (like a class in JS) or we can use **class syntax**
- ▶ Either way **inheritance is prototype-based**

PROTOTYPES & INHERITANCE

- ▶ prototypes and inheritance are complex, but they provide a lot of power and flexibility
- ▶ likely only use for larger projects
- ▶ don't have too many levels of inheritance and keep track of where you define your methods and properties

JSON

- ▶ stands for JavaScript Object Notation
- ▶ separate from JavaScript
- ▶ text-based data format following JavaScript object syntax
- ▶ used for transmitting data in web applications

NEXT WEEK – APIs!!!!!!!

THANKS TO (SOURCES):

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object_prototypes

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Inheritance>

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON#:~:text=JavaScript%20Object%20Notation%20\(JSON\)%20is,page%2C%20or%20vice%20versa\).](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON#:~:text=JavaScript%20Object%20Notation%20(JSON)%20is,page%2C%20or%20vice%20versa).)