

# APIS



# WHAT IS AN API?

API stands for Application Program Interface.

It's the mechanism that dispenses structured data to webpages but it's hard to make sense of.

Think of the Internet as a series of buildings...some big, others small.



Each  
building  
has a door





And doors  
require  
keys...

# KEY IDEA

API's are doorways into different website's data. For most sites, you'll need a key to get inside (often just called API keys).

# HOW TO ACCESS AN API

- 1) Super Simple: you just need a url and make an API call
- 2) Simple: you'll have to get an API key to perform a transaction with an API.
- 3) Moderate: If the API gets/puts data into an account, you will likely need account information as well.
- 4) Tough: You may have to enter a username/password using OAuth protocol.

Once you unlock the door, you don't know what will be inside (or how many rooms there are).



# WHICH ROOM?

Continuing this metaphor, API “rooms” are called endpoints. Each API has different endpoints that return different data once you transact the key/user/OAuth sequence.

# **READ THE DOCS!**

Every API is unique - you will have to explore its documentation to figure out which endpoint you want to use.

# APIS

This is a huge public catalogue of APIs (but not exhaustive, there are thousands more):

<https://github.com/toddmotto/public-apis>

# API RESPONSES

Lets take a look at what an API gives us back

SWAPI

# JS OBJECTS

API endpoints will almost always return JSON in the form of an object.

```
{  
  "data": "car",  
  "make": "ford",  
  "model": "focus"  
  "details": {  
    "color" : "blue",  
    "mileage" : "54019"  
  }  
}
```

# OBJECT LITERAL

You'll want to turn the response from the API into a variable. This gets called an object literal.

```
var myDataObject = {  
  "data": "car",  
  "make": "ford",  
  "model": "focus"  
  "details": {  
    "color" : "blue",  
    "mileage" : "54019"  
  }  
};
```

The object consists of key - value pairs. The “key” being the property, and the “value” being, well, the value

# OBJECT REFERENCE

You can then access parts of the object with a convenient syntax:

```
var myDataObject = {  
  "data": "car",  
  "make": "ford",  
  "model": "focus"  
  "details": {  
    "color" : "blue",  
    "mileage" : "54019"  
  }  
};
```

```
myDataObject.make; // ford  
myDataObject.details.color // blue
```

**API**

**Object**

ajax

Asynchronous

http method

Endpoint

Success

Error



A large, translucent jellyfish with a glowing yellow-green bell and long, flowing tentacles, swimming gracefully against a dark blue background.

# AJAX

Asynchronous Javascript and XML

# MAKING AN API CALL

We're going to keep using jQuery  
and one of it's most useful  
parts: `.ajax();`

# AJAX

Today, most endpoints return JSON (JavaScript Object Notation), but we still refer to these types of operations as AJAX

# ASYNCHRONOUS

Asynchronous means not existing or occurring at the same time.

تيرن ستون

TURNSTONE

Artisan

پیتزا

أرتيزان

Pizzas

Think of your request as  
ordering a pizza

First you place your order

تيرن ستون

THE

TURNSTONE

When waiting for your pizza,  
you don't have to sit by the door



A photograph of two dogs playing with a stick in a grassy field. A brown dog on the left and a black and white dog on the right are both holding a long, thin stick in their mouths, pulling it between them. They appear to be in motion, possibly running or playing tug-of-war. The background is a soft-focus green field.

You have other things to do in  
the meantime!



But once it comes, you know  
what to do!

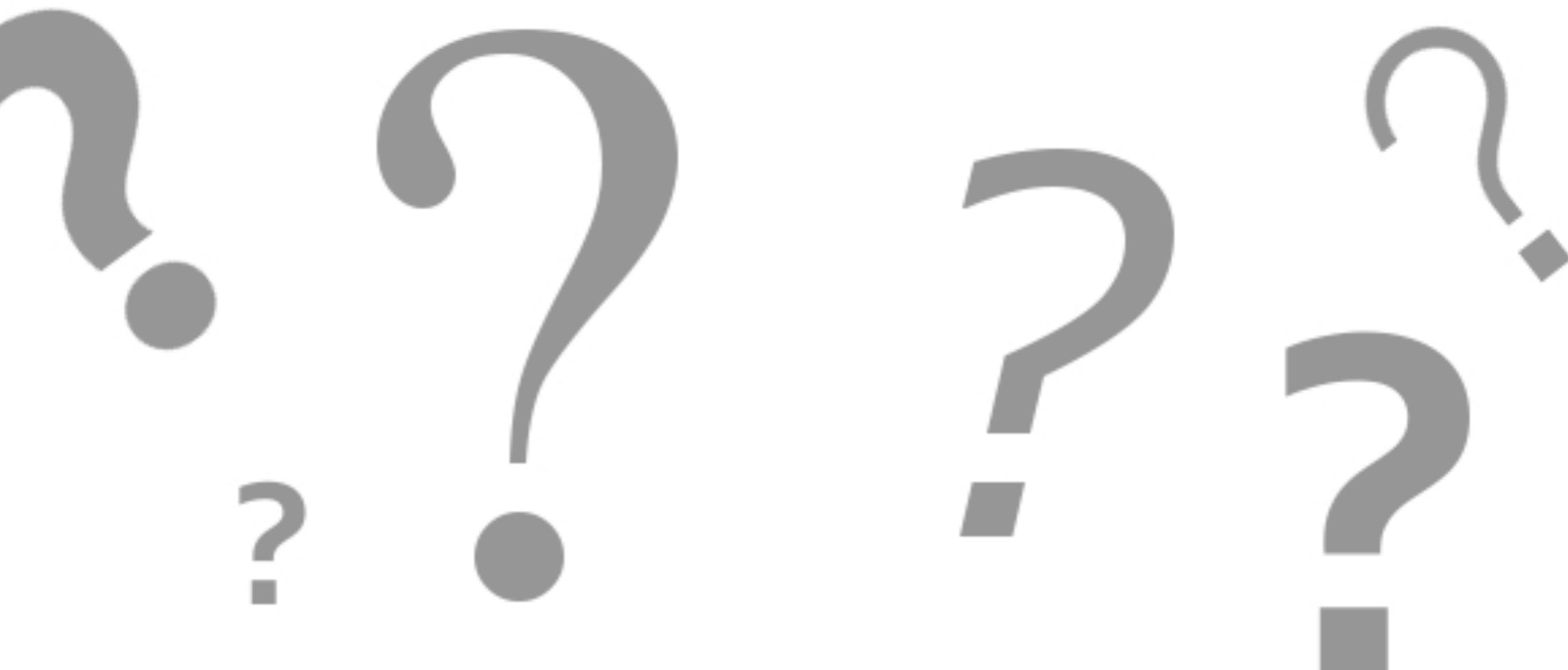
# AJAX

With ajax your request is happening outside of the normal timing of the code. Once it completes, it will run the actions we tell it to

# .AJAX() - BASIC

```
$ .ajax( {  
    method: 'GET',  
    url: 'http://linktoapi.com',  
    success: function (results) {  
        // do something  
    }  
    error: function (results) {  
        // do something else  
    }  
} );
```

# WHAT DOES IT MEAN?



# .AJAX() - BASIC

```
$ .ajax( );
```

The \$ sign means we are using  
jQuery and the . after it means we  
are calling a function. In this case it  
is called ‘ajax’

# .AJAX() - BASIC

```
$ .ajax( {  
    method: 'GET'  
} );
```

In the function we put an object. The first key value pair is telling us what the http method is, or what type of request we are making.

# .AJAX() - BASIC

```
$ .ajax( {  
    method: 'GET' ,  
    url: 'http://linktoapi.com'  
} );
```

The second key - value pair is telling us what endpoint to make the request to, or where the request should go.

# .AJAX() - BASIC

```
$ .ajax( {  
    method: 'GET',  
    url: 'http://linktoapi.com',  
    success: function(results){  
        // do something  
    }  
} );
```

Next we have a function that will run when the results are successfully returned.

# .AJAX() - BASIC

```
$ .ajax( {  
    method: 'GET',  
    url: 'http://linktoapi.com',  
    success: function(results){  
        // do something  
    },  
    error: function(error){  
        // do something else  
    }  
});
```

The last will run if the request is unsuccessful

**API**

**Object**

**ajax**

**Asynchronous**

**http method**

**Endpoint**

**Success**

**Error**



# CODEALONG

## Assignment 1

# **YOUR TURN**

## Assignment 2