Web Development

Day 10: JavaScript, again...

Objects, JSON, APIs, Testing, The DOM & jQuery

Maydm

Icebreaker!

Maydm

Today's Schedule

Morning:

- Objects
- Objects vs Arrays
- JSON vs Objects

Afternoon:

- JavaScript with HTML
- The Document Object Model
- jQuery
- Project: Quote Generator

Maydm

JavaScript Review

- Comments
- Variables
- Data Types
- Operating on Variables
- Template Literals
- Booleans
- Functions
- If Statements
- Arrays
- Loops
- Objects
- JSON

Maydm

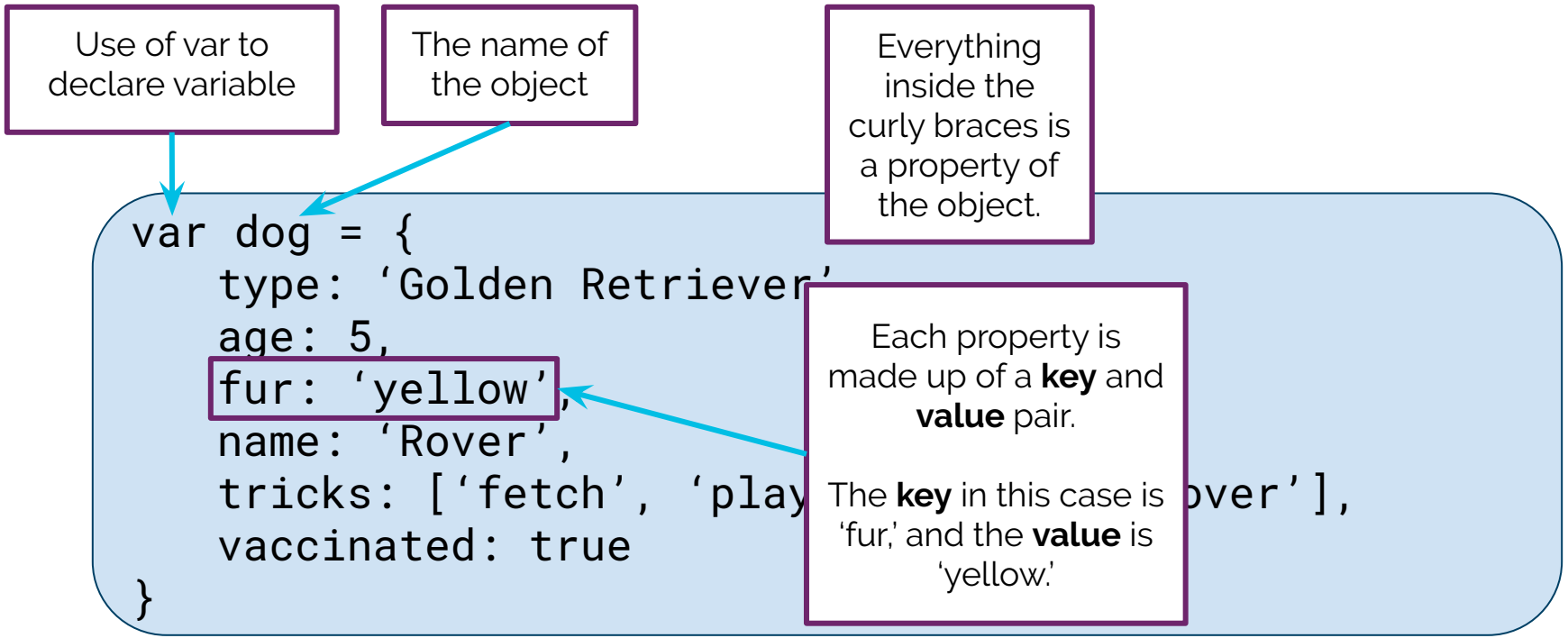
Objects

Sometimes you need to store data that's related but isn't the same as a list. An **object** is a collection of properties.

Think about a dog. A dog has the following properties:

- Type of dog
- Age
- Color of fur
- Owner
- Known tricks
- Vaccines

Anatomy of an Object



Maydm

Storing Data in Objects

All data types can be stored as **values** inside an object. Notice we have 3 strings, a number, an array, and a Boolean stored as **values**.

```
var dog = {
    type: 'Golden Retriever',
    age: 5,
    fur: 'yellow',
    name: 'Rover',
    tricks: ['fetch', 'play dead', 'roll over'],
    vaccinated: true
}
```

Maydm

Accessing the Data in Objects

Data (values) inside objects can be accessed using the **keys**. In this example, the keys include type, age, fur, owner, tricks, and vaccinated.

```
var dog = {
    type: 'Golden Retriever',
    age: 5,
    fur: 'yellow',
    name: 'Rover',
    tricks: ['fetch', 'play dead', 'roll over'],
    vaccinated: true
}
```

Maydm

Accessing the Data in Objects

Values can be accessed using the **variable name**, **a period, and a key**. This is called **dot notation**.

```
var dog = {...}

dog.type; // returns ‘Golden Retriever’
dog.age; // returns 5
dog.tricks; // returns [‘fetch’, ‘play dead’, ‘roll over’]
```

Maydm

Accessing the Data in Objects

Values can also be accessed using **bracket notation**, using the **variable name, the key, and brackets.**

```
var dog = {...}

dog['type'];     // returns 'Golden Retriever'
dog['age'];      // returns 5
dog['tricks'];   // returns ['fetch', 'play dead',
'roll over']
```

Maydm

Accessing the Data in Objects

If you want to access an array that's stored inside an object, start with the variable and property, then add an index!

```
var dog = {...}

dog['tricks'];       // returns ['fetch', ...]
dog['tricks'][0];    // returns 'fetch'
dog.tricks[0];       // returns 'fetch'
```

Maydm

Adding Data to Objects

Add properties to an existing object by using dot or bracket notation by giving a **key** and setting it equal to the **value**.

```
var dog = {...}

// These do the same thing.
dog.owner = “Jane”;
dog[‘owner’] = “Jane”;
```

Maydm

Overwriting Data in Objects

You can also overwrite a property's value by reassigning it using the same bracket or dot notation.

```
var dog = {...}

dog.name = “Fido”;
dog[‘name’] = “Fido”;
```

Maydm

Deleting Data in Objects

You can also delete a property from an object using the delete keyword and the key.

```
var dog = {...}

delete dog.tricks;
delete dog['tricks'];
```

Maydm

Practice with Objects

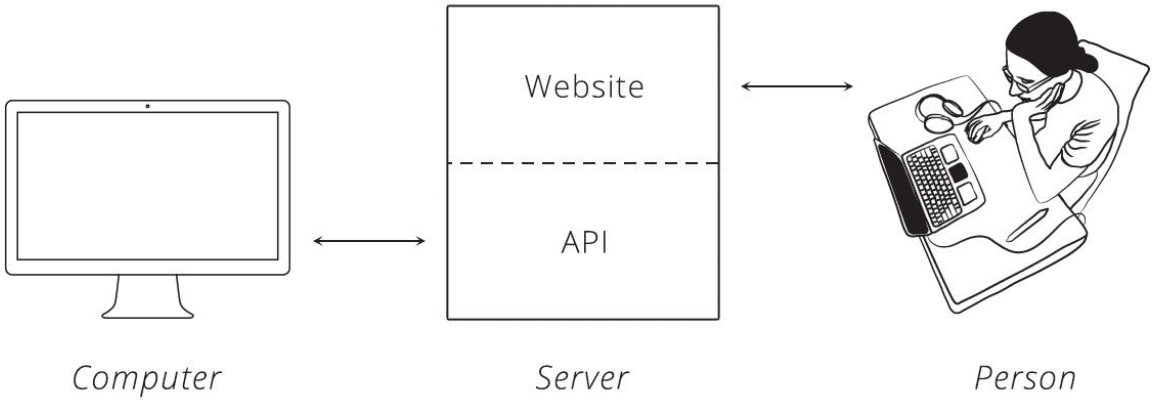
Work through the Day 9 “Objects” exercises on JS Bin.

Arrays vs Objects

When would you want an object instead of an array?

APIs

Application Programming Interfaces are software that allows two programs to talk to one another. On the internet, APIs are what allow a server to return data to a user's web browser.



Maydm

JSON

Many web APIs return data in **JSON** format: JavaScript Object Notation. JSON looks like JavaScript but it is a separate language used to deliver data in a reliable format.

Javascript:

```
var student = {
    firstName: “Joe”,
    lastName: “Smith”
}
```

JSON:

```
{
    “firstName”: “Joe”,
    “lastName”: “Smith”
}
```

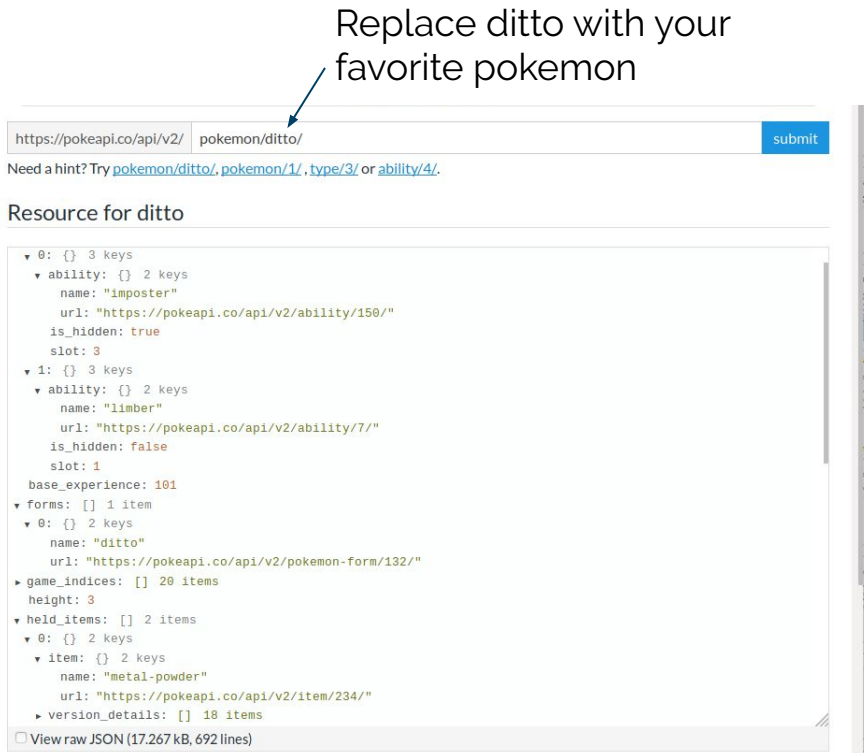
Maydm

Pokemon API

Check out how JSON and APIs work by playing with the Poke API.

Visit https://pokeapi.co

Replace “ditto” with your favorite Pokemon to see their data in JSON format



Maydm

JavaScript Review

- Comments
- Variables
- Data Types
- Operating on Variables
- Template Literals
- Booleans
- Functions
- If Statements
- Arrays
- Loops
- Objects
- JSON
- The DOM

Maydm

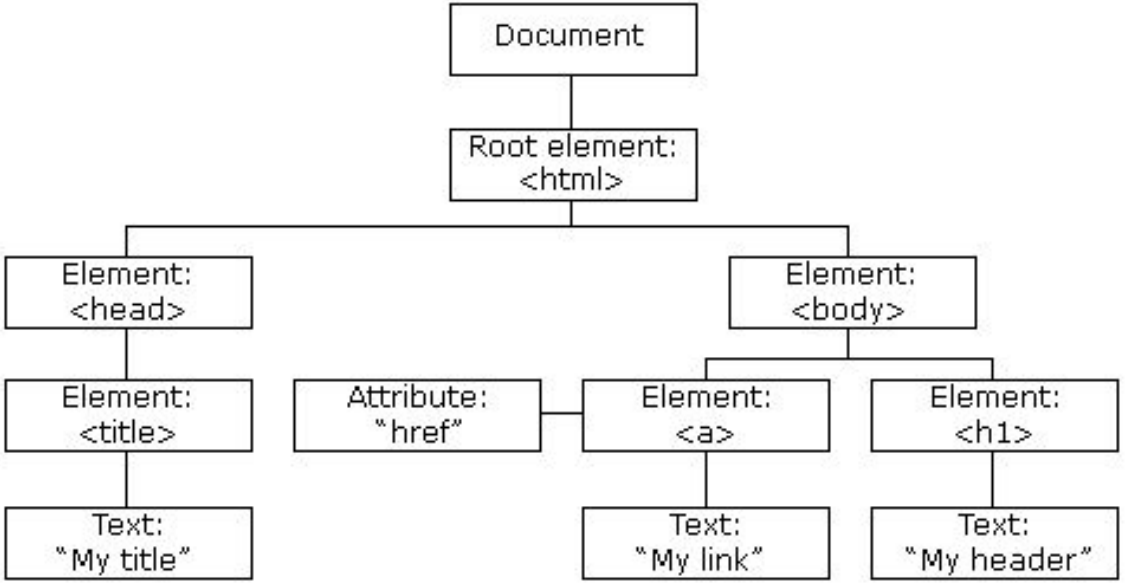
Using JS with HTML

JavaScript works with HTML using the **Document Object Model**.

The DOM defines how a document (the website, in this case) is structured and accessed.

The DOM changes with each web page, based on the HTML of the page.

The Document Object Model



Maydm

DOM Methods

You've already seen one DOM method regularly in the JS projects you've been working on:

```
document.getElementById(element);
```

Remember, methods are built-in functions that belong to specific objects and can only be used by those objects. This method can only be used by the **document** object.

Commonly Used DOM Methods

- document.getElementById(id)
- document.getElementsByTagNam e(name)
- document.createElement(name)
- parentNode.appendChild(node)
- element.innerHTML()
- element.setAttribute()
- element.getAttribute()
- element.addEventListener()
- element.insertAdjacentHTML()
- window.onload()
- window.scrollTo()
- console.log()

Maydm

Selecting a DOM element by ID

You've seen the getElementById DOM method before in some of the projects. Because IDs are unique on a page, it is a very useful tool for manipulating HTML.

```
var testDiv = document.getElementById('test');
testDiv.innerHTML = 'TESTING!'
```

Maydm

Selecting a DOM element by ID

The two chunks of code below do the same thing, though generally the second is considered better code. Why do you think that is?

```
document.getElementById('test').innerHTML = 'TESTING!'

// is the same as

var testDiv = document.getElementById('test');
testDiv.innerHTML = 'TESTING!'
```

Maydm

Selecting a DOM element by HTML Tag

Another way to select elements is by HTML tag with the DOM method **getElementsByTagName**, which takes the tag as an argument. This returns an array of elements, which can be accessed through the indices.

```
var allDivs = document.getElementsByTagName("div");

allDivs[0].innerHTML = ‘This is the first div!’;
allDivs[1].innerHTML = ‘This is the second div!’;
```

Maydm

Selecting a DOM element by CSS Selector

Using the DOM method querySelector, you can find an element with any selector that works in CSS, including class, ID or tag. Selectors can also be combined to find specific tag types with classes or IDs. This returns the first element that matches the selector.

```
var example = document.querySelector('.example');
// selects the first element with the class "example"

var avengers = document.querySelector('ul#avengers');
// selects the Unordered List with the ID "avengers"
```

Maydm

DOM Method: Changing HTML

Once an element is selected, you can change the HTML with the **innerHTML** method. This will replace any other HTML that exists in the element.

```
var newList = document.getElementById('list');

list.innerHTML = '<ul><li>Item 1</li><li>Item
2</li></ul>';
```

Maydm

DOM Method: Changing Text

If you're only changing the text of an element, use the **innerText** method.

```
var test = document.getElementById('test');

test.innerText = 'TESTING!';
```

Maydm

DOM Method: Changing Style

You can change the style of an element with style and the attribute you want to change. Note: If the attribute would normally use a hyphen, it uses camel case in JavaScript.

```
var test = document.getElementById('test');

test.style.fontSize = '20px';
```

Maydm

DOM Method: HTML Attributes

HTML elements often have attributes, like the src attribute for image tags. Using the getAttribute and setAttribute methods, you can see if an attribute exists, what it is, or change it to something else.

```
var catPhoto = document.getElementById('cat');

catPhoto.setAttribute('src',
'https://knowyourmeme.com/photos/406282-grumpy-cat');
```

Maydm

DOM Method: Adding elements

The DOM method insertAdjacentHTML adds HTML to a specific point in the DOM. The HTML can be added in the following places:

- `'beforebegin'`: Before the `element` itself.
- `'afterbegin'`: Just inside the `element`, before its first child.
- `'beforeend'`: Just inside the `element`, after its last child.
- `'afterend'`: After the `element` itself.

Maydm

DOM Method: Adding elements

```
var element = document.getElementById('someDiv');

element.insertAdjacentHTML('beforebegin', '<div>This
will be added before the selected element</div>');

element.insertAdjacentHTML('beforeend', '<p>This will
be added after the last child element of the div but
before the end of the div</p>');
```

Maydm

Running JS on HTML Page

Using buttons is a common way to execute JS functions, and it is easy to implement using the “onclick” attribute. You can also add any necessary arguments to the function, just like you would using a JS file.

```
<button onclick="myFunction()">
    Run a function
</button>
```

Maydm

JavaScript Libraries

Just like with CSS, there are shortcuts for using JavaScript with **libraries**.

A **JavaScript library** is a set of functions that work as shortcuts to plain JavaScript methods.

There are 100s of JS libraries, though be careful before you start using one! It might not be maintained anymore, or it might be out of date. Many libraries that were popular a few years ago are now considered obsolete because of advances in “vanilla” JavaScript.

jQuery

jQuery is a JavaScript library that makes HTML manipulation easier than with “vanilla” JS.

jQuery can be used in the same file as vanilla JS!

Since it is a library, it must be included in the HTML page in order to use its methods. Add a link to the jQuery library just as you would your own JS file.

Important: Because the browser reads line by line, include the jQuery library before any scripts that use jQuery or the code won’t work!

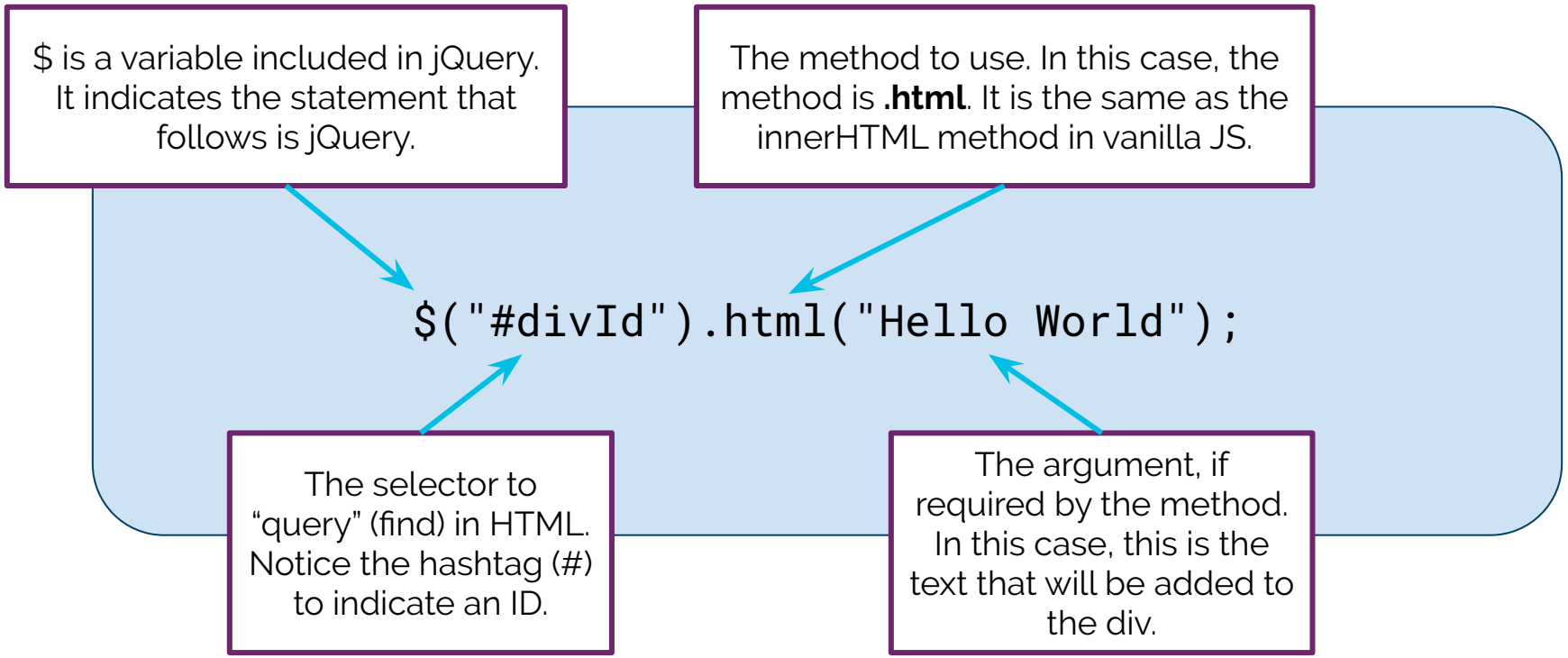
Using jQuery

```
document.getElementById.innerHTML = “Hello World”;

// The “vanilla” JS above is the same as jQuery below

$("#divId").html("Hello World");
```

Anatomy of a jQuery Statement



Maydm

Project: Random Quotation Generator

Open the “Random Quotation Generator” starter code and fork the pen.

Just as before, we’ll read through the code together, starting with the HTML.

You can use either vanilla JS or jQuery for this project. If you use jQuery, remember to include a link to the library on your HTML page.

Reflection

Write in your journal about how you feel or what you learned today.

Prompts:

- How are objects and arrays similar? How are they different?
- Why would you want to use a JavaScript library instead of vanilla JS?
- How are methods like functions?