JAVASCRIPT: OBJECTS AND JSON

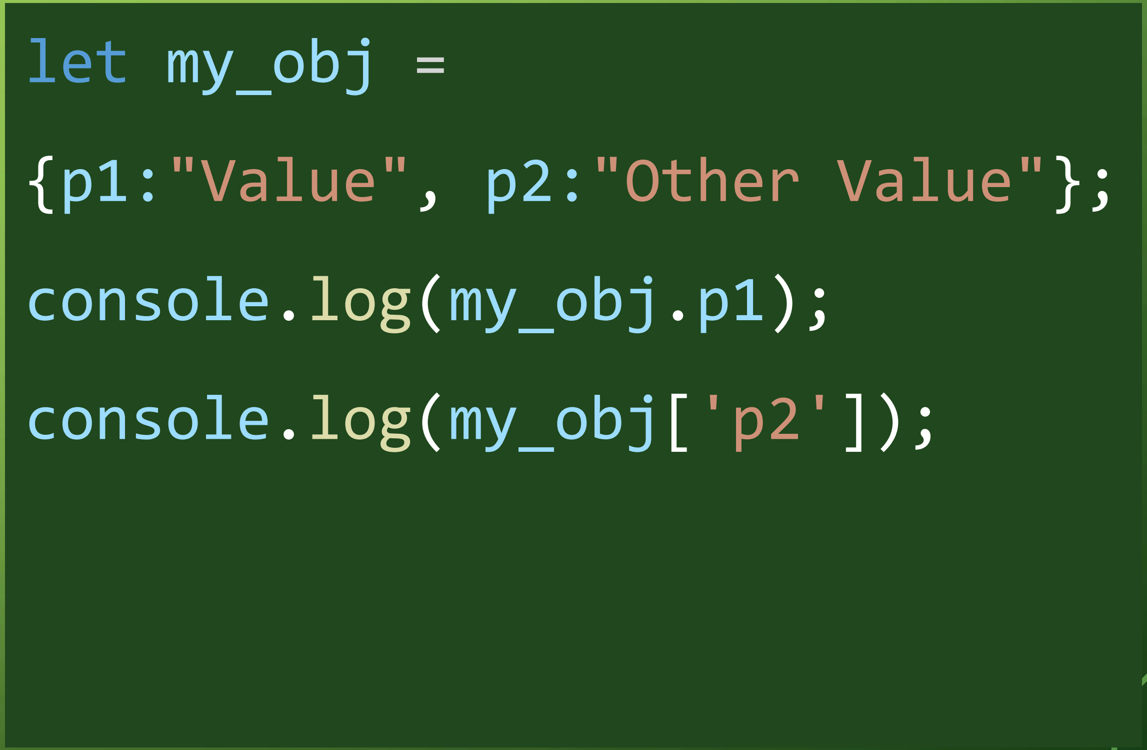
DR. JASON BOWEY

OUTLINE

- Review: JavaScript Objects
- Review: JavaScript Arrays
- JavaScript Object Notation (JSON)
- Exercise: Populating Select Menu
- Exercise: Populating Page Elements

REVIEW – JAVASCRIPT OBJECTS

- Objects in JavaScript are essentially key-value pairs
- Basically, dictionaries in Python
- Access elements with ‘.’ or [ ]



```
let my_obj =
{p1:"Value", p2:"Other Value"};
console.log(my_obj.p1);
console.log(my_obj['p2']);
```

REVIEW – JAVASCRIPT ARRAYS

- Store data sequentially
- Basically, Python Lists
- Access values by index (starting at 0)
- Arrays can contain a mix of data types

```
let my_array = [1, "50", "Twelve"];
console.log(my_array[0]);
```

PROBLEM

- So far in this course we covered some basic web technologies and problems:
  - HTML, CSS, JavaScript
  - Tables, Forms, Layout
- We haven't discussed how to populate our pages with actual **content**
- Start by creating a form with a drop down menu:
  - What is the problem when the number of items scales very large?

PROBLEM 2

- Imagine an online shop with many products for sale
- How does the page know which products are available?
- How are the products displayed?

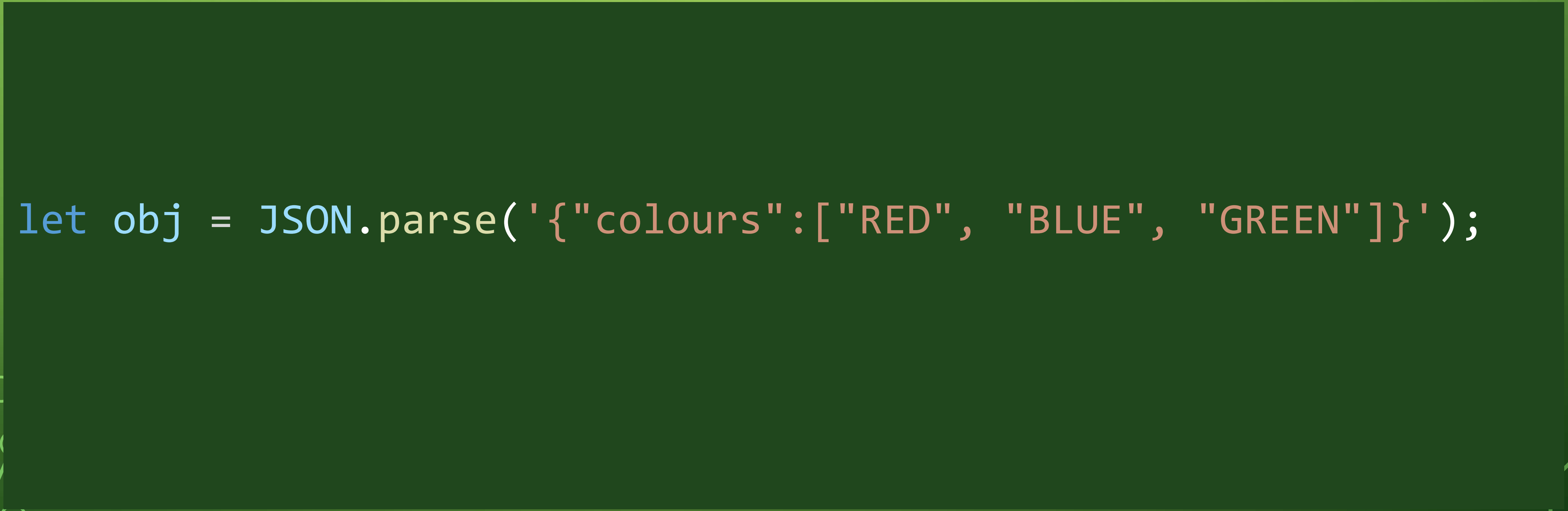
SOLUTIONS?

- Does it make sense to hardcode these value in .html files?
- Should the values be hardcoded in the JavaScript code?
- Is there another option?

JAVASCRIPT OBJECT NOTATION (JSON)

- It is common to retrieve data from a server (e.g., from a database)
- It is also common to want to send data to a server
- JSON is a string format that closely resembles JavaScript objects
- Able to convert JS objects to a JSON string and send the string to the server
- Able to ask the server for data and receive it as a JSON string
- Convert the JSON string to a JS object and use as needed

CONVERTING JSON TO JS OBJECT



```
let obj = JSON.parse('{"colours":["RED", "BLUE", "GREEN"]}');
```

CONVERTING JS OBJECT TO JSON

```
let json_string = JSON.stringify(obj);
```

FORMATTING A JSON STRING

```
{
    "name":"Jason",
    "job":"Lecturer",
    "courses":["CMPT 270", "CMPT 281", "CMPT 400"]
}
```

READING JSON FROM A FILE

```
function async myFunction() {
    const response = await fetch("./colours.json");
    const colours = await response.json();
}
```

EXERCISE 1: POPULATING A SELECT MENU

- As discussed earlier, a select menu with hundreds of choices shouldn't be hardcoded into your .html
- Instead, the options can be externalized to a separate .json file (or a .json string could be retrieved from the server)
- Use JS to read the contents of the json file, and populate the select menu with the contents

EXERCISE 2: POPULATING CONTENTS OF A SHOP

- A more complicated example, externalize the items as a list of objects
- Use JS to read the .json file and add html to display all of the items

NEXT CLASS