CODE EXERCISE
JSON: JAVASCRIPT OBJECT NOTATION

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data interchange format. It is based on a subset of the JavaScript Programming Language and is used for representing simple structured data. JSON is easy for humans to read and write and easy for machines to parse and generate.

JSON is often used to transmit data between a server and a web application, as an alternative to XML. It is represented as key-value pairs and values can be of various types like strings, numbers, booleans, arrays, and objects. JSON objects are surrounded by curly braces {} and consist of key-value pairs, where keys are strings and values can be of any type. JSON arrays are surrounded by square brackets [] and contain a list of values, which can be of any type.

JSON is also commonly used as a format for data storage, configuration files, and as a language-independent data exchange format. It is simple, widely supported, and flexible, making it a preferred choice for developers.

Example 1: A JSON object representing a person

```
{
    "name": "John Doe",
    "age": 30,
    "address": {
        "street": "123 Main St",
        "city": "Anytown",
        "state": "NY",
        "zip": "12345"
    },
    "phoneNumbers": [
        "555-555-5555",
        "555-555-5556"
    ],
    "isEmployed": true
}
```

This JSON object represents a person, with the following properties:

name: a string representing the person's name
age: an integer representing the person's age
address: an object containing the person's address information

JSON: JAVASCRIPT OBJECT NOTATION

phoneNumbers: an array of strings representing the person's phone numbers
isEmployed: a boolean representing whether or not the person is currently employed.

LAB 1

1) Create a sample JSON file containing information about the NY Subway system:

```
{
    "lines": {
        "1": {
            "name": "1",
            "color": "red",
            "stations": ["Van Cortlandt Park-242 St", "238th St", "231st St", "Marble
Hill-225 St"]
        },
        "2": {
            "name": "2",
            "color": "red",
            "stations": ["Wakefield-241 St", "Nereid Ave", "219th St", "Gun Hill Rd"]
        }
    }
}
```

And save it in the same folder as your JavaScript file, let's call it nyc-subway.json

2) In your JavaScript file, use the fetch() method to retrieve the data from the JSON file.

```
fetch('nyc-subway.json')
.then(response => response.json())
.then(data => {
    // your code to handle the data here
})
.catch(err => {
    console.log(err);
});
```

3) Once the data is retrieved, you can access the data and do whatever you want with it. For example, you can loop through the subway lines and stations and display them on the page.

JSON: JAVASCRIPT OBJECT NOTATION

```
fetch('nyc-subway.json')
.then(response => response.json())
.then(data => {
    for(line in data.lines) {
        let subwayLine = data.lines[line];
        console.log("Line : " + subwayLine.name);
        console.log("Color : " + subwayLine.color);
        console.log("Stations : " + subwayLine.stations);
    }
})
.catch(err => {
    console.log(err);
});
```

4) You can also create a function that accepts a subway line name and returns the list of stations on that line.

```
function getStations(lineName) {
    fetch('nyc-subway.json')
    .then(response => response.json())
    .then(data => {
        for(line in data.lines) {
            let subwayLine = data.lines[line];
            if(subwayLine.name === lineName) {
                console.log(subwayLine.stations);
                return subwayLine.stations;
            }
        }
    })
    .catch(err => {
        console.log(err);
    });
}
console.log(getStations("1"));
```

5) Like before make sure that the file path to the json file is correct and accessible. And that's it! Now you can use the data from the external JSON file to get information about the NY Subway system and use it in your JavaScript code.

CODE EXERCISE
JSON: JAVASCRIPT OBJECT NOTATION

LAB 2

Save the following as movies.json and use for this exercise:

```json
[
  {
    "title": "The Shawshank Redemption",
    "director": "Frank Darabont",
    "release_date": "1994-09-23"
  },
  {
    "title": "The Godfather",
    "director": "Francis Ford Coppola",
    "release_date": "1972-03-24"
  },
  {
    "title": "The Godfather: Part II",
    "director": "Francis Ford Coppola",
    "release_date": "1974-12-20"
  },
  {
    "title": "The Dark Knight",
    "director": "Christopher Nolan",
    "release_date": "2008-07-16"
  },
  {
    "title": "12 Angry Men",
    "director": "Sidney Lumet",
    "release_date": "1957-04-10"
  }
]
```

Create a JavaScript program that retrieves information about movies from an external JSON file and displays the information on a web page.

- Use the fetch() method to retrieve the data from the JSON file.

- The information should be displayed in a format of your choice, for example, a list of movies and their respective details such as title, director, and release date, or a function that accepts a movie title and returns its details.

CODE EXERCISE
JSON: JAVASCRIPT OBJECT NOTATION

- A search bar should be added that allows users to search for movies by title

Deliverable:

- A single HTML file called movie.html

- A single JavaScript file called movie.js

- A single JSON file called movies.json

- A readme file that explains the usage of your program.