

CODE EXERCISE

10 JavaScript JSON Exercises

CODING EXERCISES TEST YOUR SKILLS



10 JavaScript JSON Exercises	1
Exercise 1: Parsing JSON	2
Exercise 2: Stringifying JavaScript Object	2
Exercise 3: Nested JSON Parsing	3
Exercise 4: Modifying Parsed JSON	3
Exercise 5: Array of Objects in JSON	4
Exercise 6: Filtering Data from JSON Array	4
Exercise 7: Adding Objects to JSON Array	5
Exercise 8: JSON from APIs	5
Exercise 9: Handling Missing Data in JSON	6
Exercise 10: Complex JSON Structure	6

10 JavaScript JSON Exercises

🌟 Exciting Resource Alert! 🌟 Dive into the world of JavaScript JSON with our latest collection of 10 hands-on coding exercises!

Whether you're a beginner or looking to sharpen your skills, these exercises cover everything from parsing and stringifying JSON to complex data manipulation and fetching data from APIs. With detailed explanations and complete code snippets, you'll master the art of handling JSON in your web projects in no time.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

#JavaScript #JSON #WebDevelopment #CodingExercises #Programming
#DataHandling #TechLearning

Engage with the post, share your thoughts, or even showcase your projects using these exercises. Let's grow together in our coding journey!

Exercise 1: Parsing JSON

Objective: Learn to parse a JSON string into a JavaScript object.

```
const jsonString = '{"name":"John", "age":30, "city":"New York"}';
```

Task: Parse the JSON string and display John's age.

```
const obj = JSON.parse(jsonString);  
console.log(obj.age); // Expected output: 30
```

Explanation: This exercise demonstrates the use of `JSON.parse()` to convert a JSON string into a JavaScript object, allowing for easy access to its properties.

Exercise 2: Stringifying JavaScript Object

Objective: Convert a JavaScript object into a JSON string.

```
const user = {name: "Jane", age: 25, city: "Chicago"};
```

Task: Convert the object into a JSON string.

```
const jsonString = JSON.stringify(user);  
console.log(jsonString); // Expected output:  
'{"name":"Jane","age":25,"city":"Chicago"}'
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Explanation: Teaches how to use `JSON.stringify()` to create a JSON string from a JavaScript object, making it suitable for storage or transmission.

Exercise 3: Nested JSON Parsing

Objective: Parse a nested JSON string.

```
const jsonString = '{"name":"Mike", "age":35, "address":{"street":"5th Avenue", "city":"New York"}}';
```

Task: Access and print the city from the address.

```
const user = JSON.parse(jsonString);  
console.log(user.address.city); // Expected output: New York
```

Explanation: Focuses on accessing nested properties in a JavaScript object parsed from a JSON string.

Exercise 4: Modifying Parsed JSON

Objective: Modify an object obtained from parsing JSON and convert it back to a JSON string.

```
const jsonString = '{"name":"Lucy", "age":28, "city":"Los Angeles"}';
```

Task: Increase Lucy's age by 1, then convert the object back to a JSON string.

```
const user = JSON.parse(jsonString);  
user.age += 1;  
const updatedJsonString = JSON.stringify(user);
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
console.log(updatedJsonString); // Expected output:
```

```
'{"name":"Lucy","age":29,"city":"Los Angeles"}
```

Explanation: Demonstrates modifying a property of a parsed JSON object and then stringifying the modified object back into JSON.

Exercise 5: Array of Objects in JSON

Objective: Work with an array of objects in JSON format.

```
const jsonString = '[{"name":"Tom", "age":30}, {"name":"Jerry", "age":3}]';
```

Task: Parse the JSON and log each name and age.

```
const users = JSON.parse(jsonString);
```

```
users.forEach(user => console.log(` ${user.name} is ${user.age} years old.`));
```

Explanation: This exercise showcases how to parse JSON containing an array of objects and iterate over it.

Exercise 6: Filtering Data from JSON Array

Objective: Filter objects based on a condition from a parsed JSON array.

```
const jsonString = '[{"name":"Anna", "age":22}, {"name":"Bob", "age":30}, {"name":"Carol", "age":25}]';
```

Task: Find and log people older than 24.

```
const users = JSON.parse(jsonString);
```

```
const filteredUsers = users.filter(user => user.age > 24);
```

```
console.log(filteredUsers);
```

Explanation: Teaches filtering data in an array parsed from JSON, using array methods like `filter()`.

Exercise 7: Adding Objects to JSON Array

Objective: Add a new object to an array of objects in JSON format.

```
const jsonString = '[{"name":"Dave", "age":20}, {"name":"Eve", "age":30}]';
```

Task: Add a new person to the array and convert it back to JSON.

```
const users = JSON.parse(jsonString);  
users.push({"name":"Frank", "age":28});  
const updatedJsonString = JSON.stringify(users);  
console.log(updatedJsonString);
```

Explanation: Shows how to add a new item to an array parsed from JSON and then stringify the updated array.

Exercise 8: JSON from APIs

Objective: Fetch JSON data from an API and log a specific property.

```
// This is a hypothetical example; the URL is not real  
fetch('https://api.example.com/data')  
  .then(response => response.json())  
  .then(data => console.log(data.title));
```

Task: Log the title property from the fetched JSON data.

Explanation: Introduces fetching JSON data from an API, parsing it, and accessing a property, simulating real-world application data handling.

Exercise 9: Handling Missing Data in JSON

Objective: Safely access a property that may not exist in a parsed JSON object.

```
const jsonString = '{"name":"George", "age":40}';
```

Task: Attempt to log the city property, providing a default value if it doesn't exist.

```
const user = JSON.parse(jsonString);  
console.log(user.city || "City not provided");
```

Explanation: Covers error handling by providing a fallback for missing data, a common scenario when dealing with JSON.

Exercise 10: Complex JSON Structure

Objective: Navigate a complex JSON structure.

```
const jsonString = '{"company":"Tech  
Corp","employees":[{"name":"Henry","department":"R&D"}, {"name":"Izzy","department":"Marketing"}]}';
```

Task: Log the name and department of each employee.

```
const company = JSON.parse(jsonString);  
company.employees.forEach(employee => console.log(`${employee.name} works  
in ${employee.department}.`));
```

Explanation: Focuses on parsing complex JSON structures and accessing nested arrays and objects, simulating real-world data scenarios.

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>