

JSON

A quick look at this one object

Introduction to JSON

- JSON (JavaScript Object Notation) is a lightweight data-interchange format widely used for data storage and communication.
- It is fundamental to understanding a big part of web development.

What is JSON?

JSON is a text-based format that represents structured data using key-value pairs. It is based on JavaScript object syntax but can be used with many programming languages.

- Example: `{"name": "John", "age": 25, "city": "New York"}`

JSON Structure

JSON consists of objects and arrays. Objects are enclosed in curly braces {} and contain key-value pairs. Arrays are enclosed in square brackets [] and can contain multiple values.

- Example: {"name": "John", "age": 25, "hobbies": ["reading", "playing guitar"]}

Common Data Types in JSON

JSON supports various data types, including strings, numbers, booleans, null, objects, and arrays.

- Example: `{"name": "John", "age": 25, "isStudent": true, "marks": null}`

Generating JSON

JSON can be generated dynamically in code by creating objects and arrays, and then serializing them into a JSON string.

```
const person = { name: "John", age: 25 };  
const jsonStr = JSON.stringify(person);
```

Parsing JSON

JSON data can be parsed in code to extract values and manipulate the data.

```
const jsonStr = '{"name": "John", "age": 25}';  
const person = JSON.parse(jsonStr);  
console.log(person.name); // Output: John
```

Practical Uses of JSON

JSON is widely used for data exchange between client and server, storing configuration data, and representing structured data in APIs and databases.

- Example: API responses in JSON format, configuration files in JSON, database storage of structured data in JSON columns.

Conclusion

- Recap: JSON is a lightweight, text-based data-interchange format.
- Key Takeaway: Understanding JSON is essential for working with data in modern web development, as it provides a flexible and standardized way to represent structured data.