

## [4] JSON

- JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax
- We use it when dealing with data exchange between a client and a server or when storing data in a structured format.
- JSON is a text-based data interchange format, so it can represent a limited set of data types: objects, arrays, strings, numbers, booleans, and null. Functions and undefined values cannot be serialized to JSON. When working with JSON, ensure that your data adheres to these constraints.

### JSON string into JS object

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

// Parsing JSON into a JavaScript object
const person = JSON.parse(jsonString);

console.log(person.name); // "John"
console.log(person.age); // 30
console.log(person.city); // "New York"
```

### JS object into JSON string

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

// Serializing a JavaScript object to a JSON string
const jsonString = JSON.stringify(person);

console.log(jsonString); // '{"name":"John","age":30,"city":"New York"}'
```

### Customize the serialization process

```
const person = {
  name: "John",
  age: 30,
```

```
email: "john@example.com"
};

// Serialize while omitting the 'email' property
const jsonString = JSON.stringify(person, (key, value) => {
  if (key === "email") return undefined;
  return value;
});
console.log(jsonString); // '{"name":"John","age":30,"city":"New York"}'
```

## Handling Errors

```
const invalidJsonString = '{"name": "John", "age": 30,}'; // Invalid JSON

try {
  const person = JSON.parse(invalidJsonString);
  console.log(person);
} catch (error) {
  console.error("Error parsing JSON:", error.message);
}
```

## Space Parameter for pretty-printing the JSON string.

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

// Adding indentation for a more readable JSON string
const jsonString = JSON.stringify(person, null, 2);

console.log(jsonString);
/* Output:
{
  "name": "John",
  "age": 30,
  "city": "New York"
}
*/
```

## JSON can represent arrays of objects

```
const colors = ["red", "green", "blue"];

const jsonColors = JSON.stringify(colors);
```

```
const parsedColors = JSON.parse(jsonColors);  
  
console.log(parsedColors); // ["red", "green", "blue"]
```