

In JavaScript, JSON (JavaScript Object Notation) and JavaScript objects may seem similar at first glance, but there are key differences between them. Here's an example that demonstrates these differences:

```
```\javascript
// JavaScript object
var jsObject = {
 name: "John",
 age: 30,
 city: "New York"
};

// JSON data
var jsonData = '{"name": "John", "age": 30, "city": "New York"}';

// Accessing properties of JavaScript object
console.log(jsObject.name); // Outputs: "John"
console.log(jsObject.age); // Outputs: 30

// Accessing properties of JSON data
// JSON needs to be parsed to convert it into a JavaScript object
var parsedData = JSON.parse(jsonData);
console.log(parsedData.name); // Outputs: "John"
console.log(parsedData.age); // Outputs: 30

// Converting JavaScript object to JSON
var convertedToJson = JSON.stringify(jsObject);
console.log(convertedToJson); // Outputs: '{"name":"John","age":30,"city":"New York"}'

// Checking types
console.log(typeof jsObject); // Outputs: "object"
console.log(typeof parsedData); // Outputs: "object"
console.log(typeof jsonData); // Outputs: "string"
```\
```

In this example:

1. `jsObject` is a JavaScript object.
2. `jsonData` is a string representing JSON data.
3. We access properties directly on the JavaScript object (`jsObject.name`) but need to parse JSON data (`parsedData.name`) before accessing properties.
4. We convert the JavaScript object to JSON using `JSON.stringify()`.
5. We use `typeof` operator to check the types, which shows that `jsObject` and `parsedData` are both of type "object", while `jsonData` is of type "string".

The main differences are:

- JavaScript objects are native to JavaScript and can be directly manipulated, accessed, and iterated over.
- JSON is a string representation of data and needs to be parsed into a JavaScript object before it can be accessed or manipulated.
- JSON is purely a data format, while JavaScript objects can contain methods and are part of the JavaScript language itself.