ENSF 381
Full Stack Web Development

Lecture 16: JSON

Slides: Ahmad Abdellatif, PhD

Instructor: Novarun Deb, PhD

UNIVERSITY OF CALGARY

What is JSON?

- It is a text-based format that represents structured data in the form of key-value pairs, where data is organized into objects and arrays.

- It is easy for humans to read and write and easy for machines to parse and generate.

- Stands for **J**ava**S**cript **O**bject **N**otation.

JSON usages and advantages

- It is a popular format for representing structured data in web services and APIs.
- Commonly used for data exchange between a server and a web application.
- Lightweight and easy to read and write.
- Supported by many programming languages.

JSON object - example

```
{
  "name": "John Doe",
  "age": 25,
  "isStudent": false,
  "courses": ["Math", "History", "Computer Science"]
}
```

- Information often is structured in name/value pairs.
- Commas are used to separate individual data elements.
- Arrays are encapsulated within square brackets.
- Objects are enclosed within curly braces.
- A JSON array is represented using square brackets [] and can contain a list of values.

Employee information JSON - example

```
{"employee": {
    "id": 101,
    "name": {
      "first": "John",
      "last": "Doe"
    },
    "position": "Software Engineer",
    "skills": ["JavaScript", "Python", "React"],
    "contact": {
      "email": "john.doe@example.com",
      "phone": {
        "mobile": "555-1234",
        "office": "555-5678"
      }
    },
    "projects": [
      {
        "name": "Project A",
        "description": "Developing a web application",
        "status": "In Progress"
      },
      {
        "name": "Project B",
        "description": "Implementing new features",
        "status": "Completed"
      }
    ]}}
```

JSON example with information for 3 employees

```json
{
  "employees": [
    {
      "id": 101,
      "name": "Alice Smith",
      "position": "Software Developer",
      "department": "Engineering",
      "salary": 75000
    },
    {
      "id": 102,
      "name": "Bob Johnson",
      "position": "Data Analyst",
      "department": "Analytics",
      "salary": 60000
    },
    {
      "id": 103,
      "name": "Charlie Brown",
      "position": "UI/UX Designer",
      "department": "Design",
      "salary": 70000
    },
  ]}
```

Parsing JSON in JavaScript

- JSON is primarily a text-based data interchange format.
- When data is transmitted or received, it is often represented as a JSON string.
- We need to parse the JSON string into a usable JavaScript object.

Parsing JSON in JavaScript

- Parsing is the process of converting a JSON string into a JavaScript object.

- This process is called Parsing JSON.

- JSON.parse() method is used for parsing JSON in JavaScript.

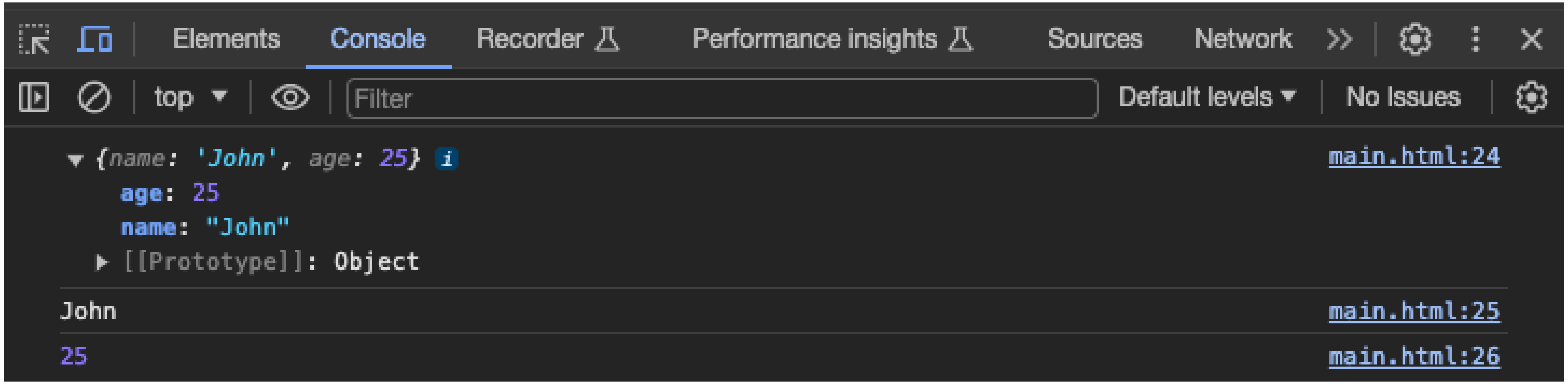
Parsing JSON in JavaScript cont.

- Once parsed, the data becomes a JavaScript object that can be easily manipulated.
- Access values using dot notation or square bracket notation.

Example on parsing simple JSON

```
<script>

const jsonString = '{"name": "John", "age": 25}';

const parsedObject = JSON.parse(jsonString);

console.log(parsedObject)
console.log(parsedObject.name)
console.log(parsedObject.age)

</script>
```

Example on parsing simple JSON

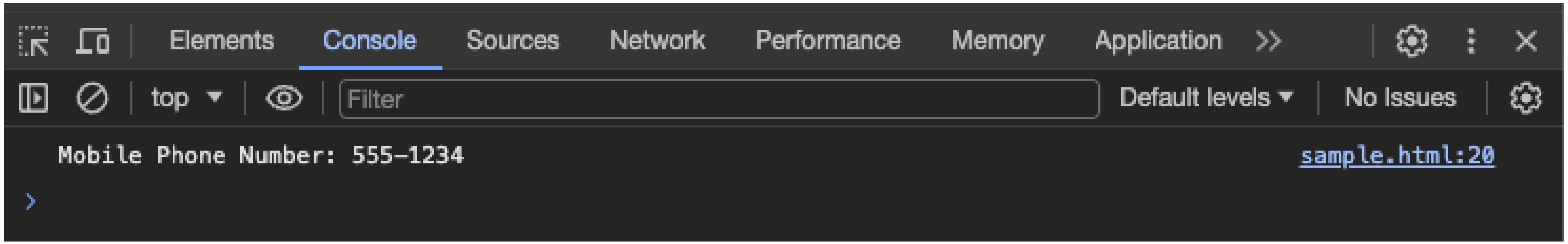


```
▼ {name: 'John', age: 25} i                         main.html:24
    age: 25
    name: "John"
  ▶ [[Prototype]]: Object
John                                                main.html:25
25                                                  main.html:26
```

Accessing nested elements…

```javascript
const employeeDataString = '{"employee": {
"id": 101,
"name": { "first": "John", "last": "Doe"},
"position": "Software Engineer",
"skills": ["JavaScript", "Python", "React"],
"contact": {
"email": "john.doe@example.com",
"phone": {"mobile": "555-1234", "office": "555-5678"}
},
"projects": [
{"name": "Project A", "description": "Developing a web application", "status": "In Progress"},
{"name": "Project B", "description": "Implementing new features", "status": "Completed"}]}}';

// Parsing the JSON string to convert it to an object
const employeeData = JSON.parse(employeeDataString);

// Accessing the phone number
const phoneNumber = employeeData.employee.contact.phone.mobile;

console.log("Mobile Phone Number:", phoneNumber);
```

Accessing nested elements…



```
Mobile Phone Number: 555-1234                sample.html:20
```

What is the output of this code snippet?

```
</script>

const jsonString = '{"name": "Bob", "age": "25"}';
const parsedObject = JSON.parse(jsonString);

console.log(parsedObject.age + 5);

</script>
```

The result will be the string "255" instead of the arithmetic sum 30.

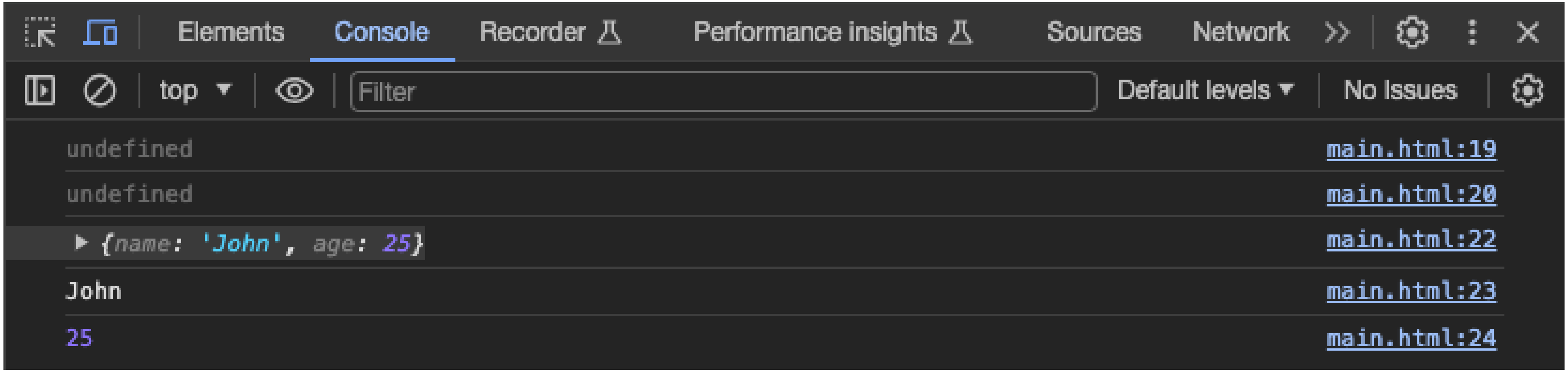
Parsing JSON in JavaScript cont.

- Once parsed, the data becomes a JavaScript object that can be easily manipulated.
- Access values using dot notation or square bracket notation.
- We cannot access the objects before converting them to JSON.

What is the output of this code snippet?

```
<script>

const jsonString = '{"name": "John", "age": 25}';
const parsedObject = JSON.parse(jsonString);

console.log(jsonString.name)
console.log(jsonString.age)

console.log(parsedObject)
console.log(parsedObject.name)
console.log(parsedObject.age)

</script>
```

What is the output of this code snippet?



```
undefined                                   main.html:19
undefined                                   main.html:20
▶ {name: 'John', age: 25}                   main.html:22
John                                        main.html:23
25                                          main.html:24
```

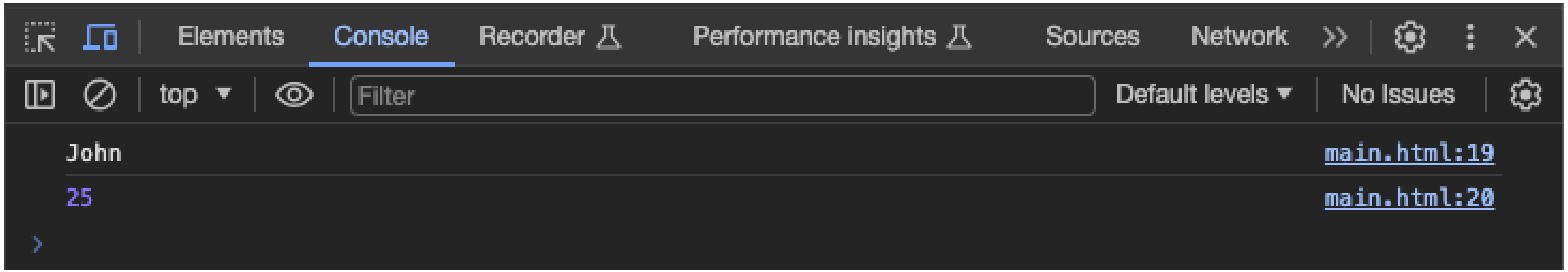
Parsing JSON in JavaScript cont.

- Once parsed, the data becomes a JavaScript object that can be easily manipulated.
- Access values using dot notation or square bracket notation.
- We cannot access the objects before converting them to JSON.
- Also we can **destructure** the JSON object.

Destructuring JSON object

```
<script>

const jsonString = '{"name": "John", "age": 25}';
let {name, age}  = JSON.parse(jsonString);

console.log(name)
console.log(age)

</script>
```

Destructuring JSON object

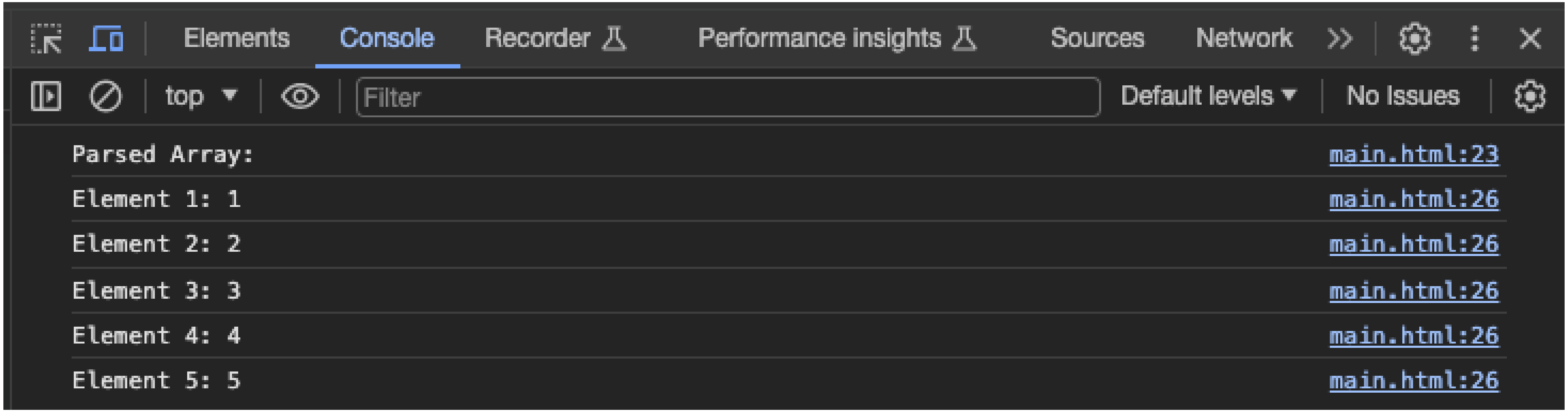


```
John                                                    main.html:19
25                                                      main.html:20
>
```

Example of parsing a simple JSON array

```
<script>

// Simple JSON array string
const jsonArrayString = '[1, 2, 3, 4, 5]';

// Parse JSON array string into a JavaScript array
const parsedArray = JSON.parse(jsonArrayString);

// Iterate through the parsed array and print each element
console.log("Parsed Array:");

for (let i = 0; i < parsedArray.length; i++) {
  console.log(`Element ${i + 1}: ${parsedArray[i]}`);
}
</script>
```

Example of parsing a simple JSON array

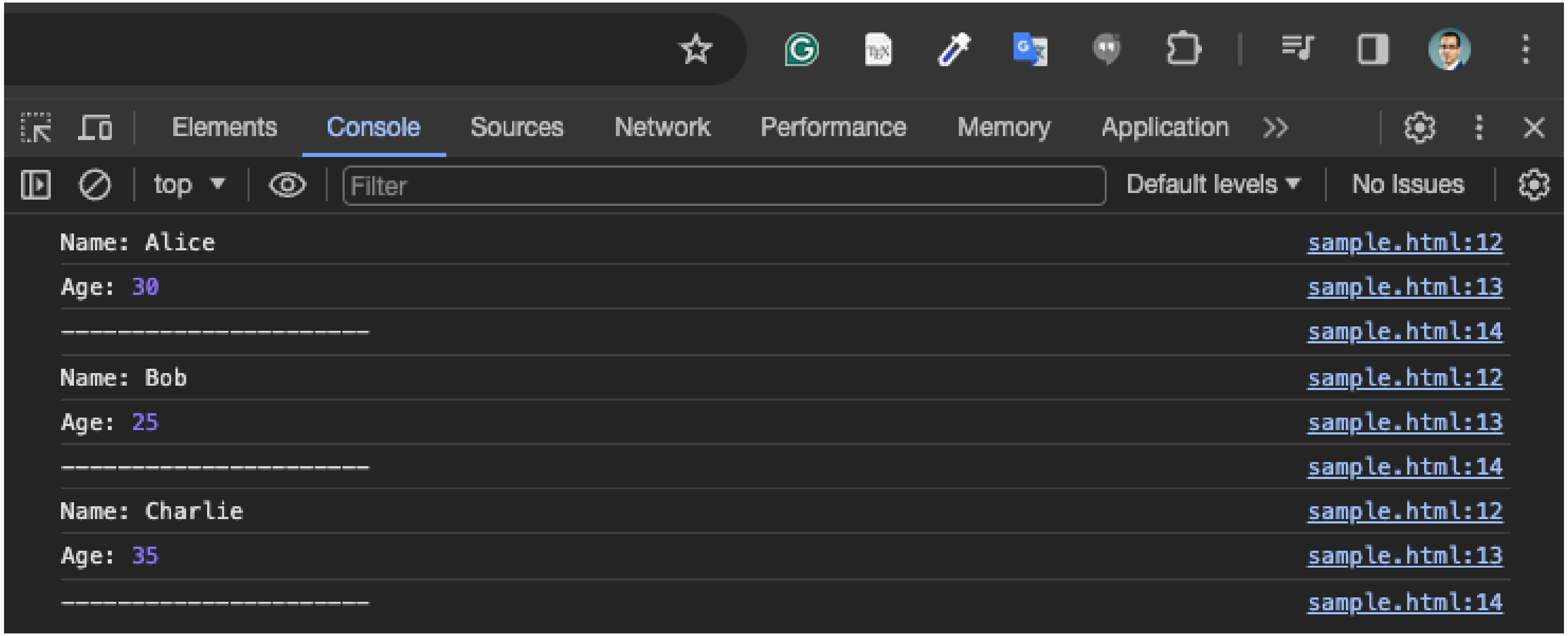


```
Parsed Array:          main.html:23
Element 1: 1           main.html:26
Element 2: 2           main.html:26
Element 3: 3           main.html:26
Element 4: 4           main.html:26
Element 5: 5           main.html:26
```

Example of parsing a JSON array of objects

```
<script>
  function extract_data(person){
      console.log("Name:", person.name);
      console.log("Age:", person.age);
      console.log("----------------------");
  }

  // Sample JSON array string
  const jsonArrayString = '[{"name": "Alice", "age": 30}, {"name":
"Bob", "age": 25}, {"name": "Charlie", "age": 35}]';
  // Parse JSON array string into a JavaScript array of objects
  const parsedArray = JSON.parse(jsonArrayString);

  // Accessing values from the parsed array
  parsedArray.forEach(extract_data);
</script>
```

Example of parsing a JSON array of objects

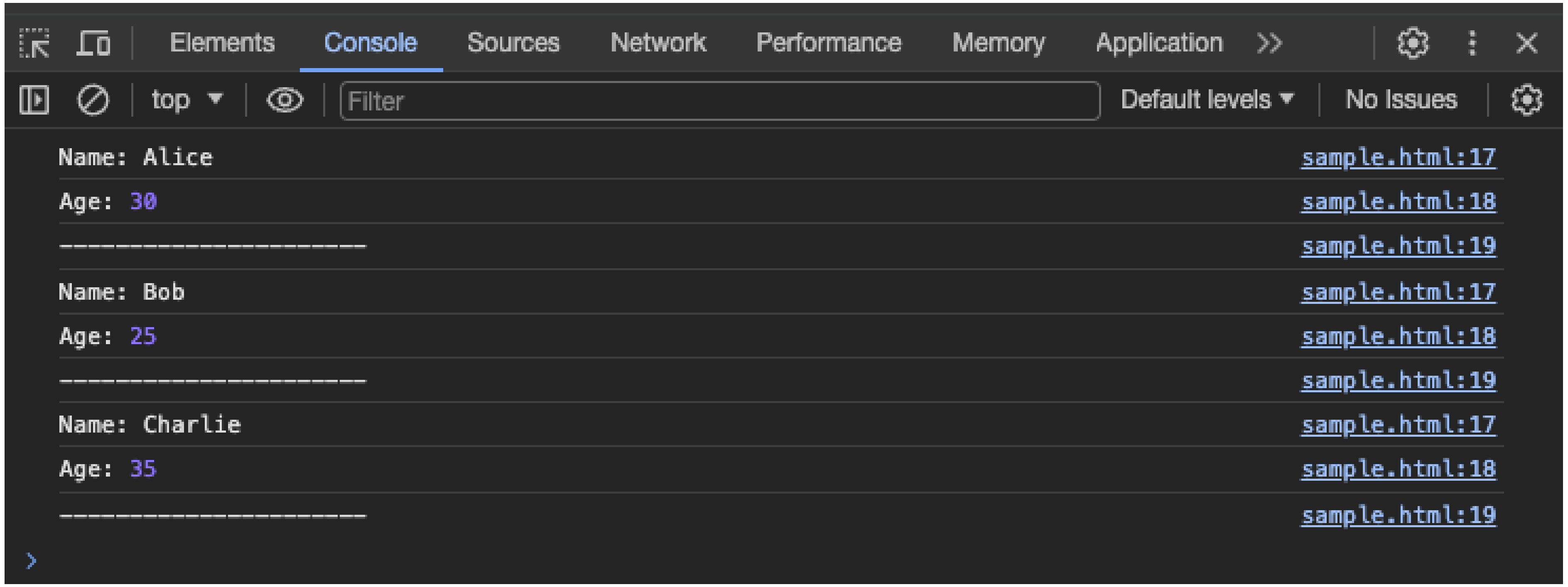


```
Name: Alice                                   sample.html:12
Age: 30                                       sample.html:13
----------------------                        sample.html:14
Name: Bob                                     sample.html:12
Age: 25                                       sample.html:13
----------------------                        sample.html:14
Name: Charlie                                 sample.html:12
Age: 35                                       sample.html:13
----------------------                        sample.html:14
```

Example of parsing a JSON array of objects

```
<script>

 // Sample JSON array string
  const jsonArrayString = '[{"name": "Alice", "age": 30}, {"name": "Bob",
"age": 25}, {"name": "Charlie", "age": 35}]';
  // Parse JSON array string into a JavaScript array of objects
  const parsedArray = JSON.parse(jsonArrayString);

  // Accessing values from the parsed array
  parsedArray.forEach((person) => {
  console.log("Name:", person.name);
  console.log("Age:", person.age);
  console.log("----------------------");
});

</script>
```

Example of parsing a JSON array of objects



```
Name: Alice                                    sample.html:17
Age: 30                                        sample.html:18
----------------------                         sample.html:19
Name: Bob                                      sample.html:17
Age: 25                                        sample.html:18
----------------------                         sample.html:19
Name: Charlie                                  sample.html:17
Age: 35                                        sample.html:18
----------------------                         sample.html:19
>
```

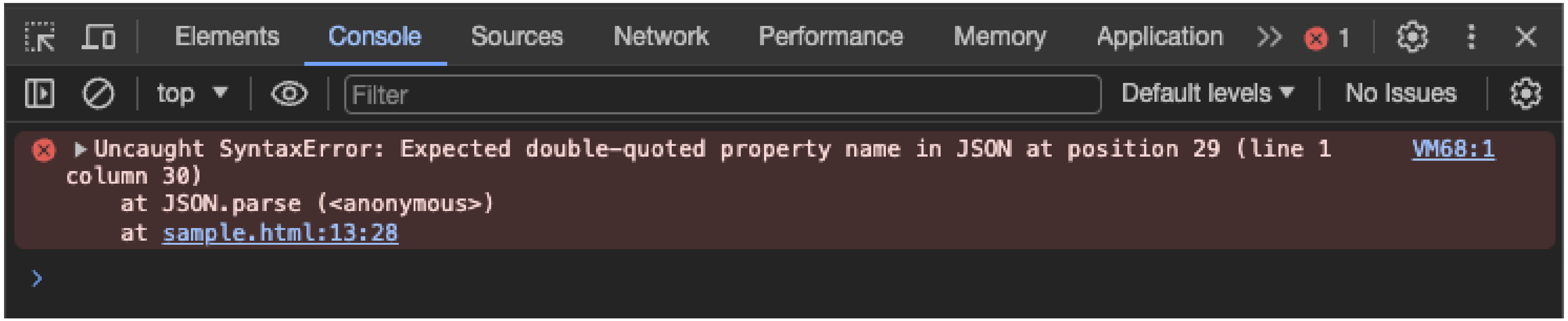
What happens when the data is malformed?

Error Handling - example

The JSON string has an extra comma

```
<script>

const malformedJSON = '{ "name": "John", "age": 25, }';

const parsedObject = JSON.parse(malformedJSON);

</script>
```

Error Handling - example



```
▶Uncaught SyntaxError: Expected double-quoted property name in JSON at position 29 (line 1      VM68:1
column 30)
    at JSON.parse (<anonymous>)
    at sample.html:13:28
```

Error Handling

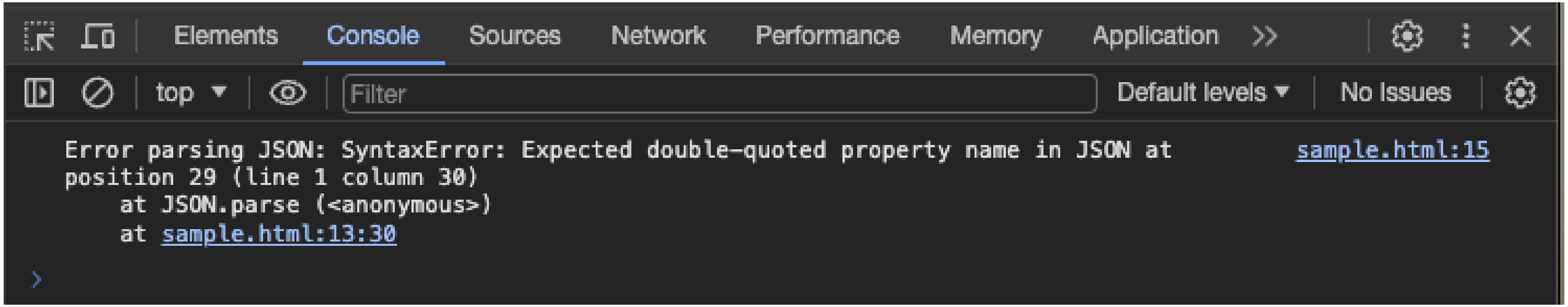
- JSON.parse() can throw exceptions if the JSON string is malformed.
- Use try-catch blocks to handle parsing errors gracefully.

```
try {
// Code block to attempt execution.
} catch (error) {
// Code block to handle any error.
}
```

Error Handling - example

```
<script>

  const malformedJSON = '{ "name": "John", "age": 25, }';
  try {
    const parsedObject = JSON.parse(malformedJSON);
  } catch(error){
    console.log('Error parsing JSON:', error);
  }

</script>
```

Error Handling - example



```
Error parsing JSON: SyntaxError: Expected double-quoted property name in JSON at        sample.html:15
position 29 (line 1 column 30)
    at JSON.parse (<anonymous>)
    at sample.html:13:30
```

Questions