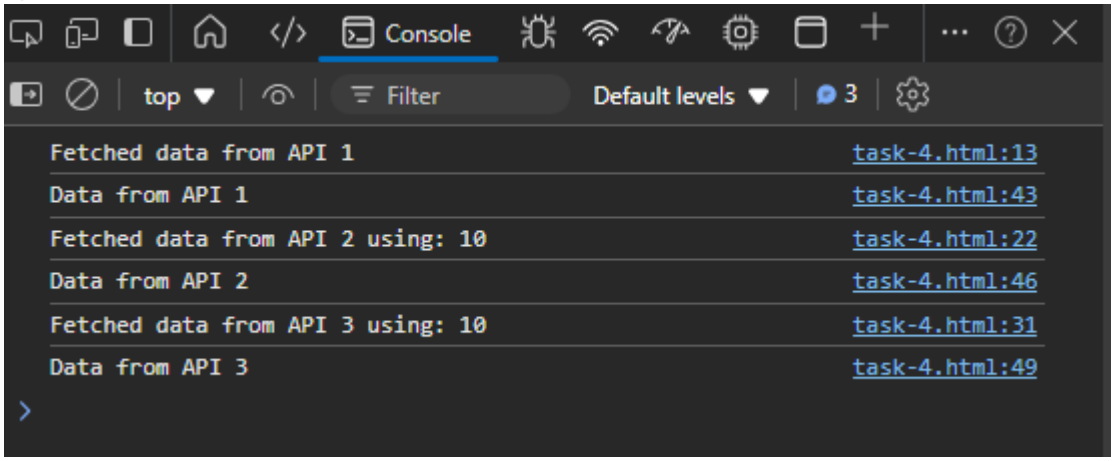
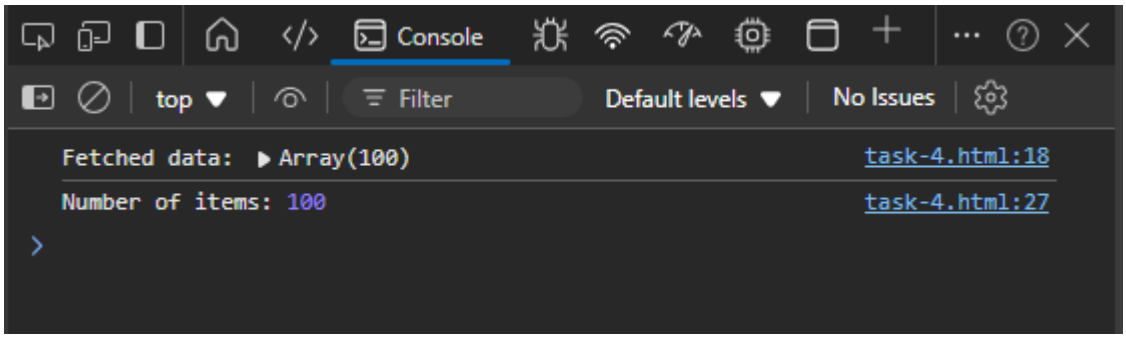
21.
```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Document</title>
</head>
<body>
<script>
    function fetchDataFromAPI1() {
        return new Promise((resolve) => {
            setTimeout(() => {
                console.log("Fetched data from API 1");
                resolve("Data from API 1");
            }, 1000);
        });
    }

    function fetchDataFromAPI2(data) {
        return new Promise((resolve) => {
            setTimeout(() => {
                console.log(`Fetched data from API 2 using: ${data}`);
                resolve("Data from API 2");
            }, 1000);
        });
    }

    function fetchDataFromAPI3(data) {
        return new Promise((resolve) => {
            setTimeout(() => {
                console.log(`Fetched data from API 3 using: ${data}`);
                resolve("Data from API 3");
            }, 1000);
        });
    }

    async function chainPromises() {
        let data1 = 10;

        try {

            const a = await fetchDataFromAPI1();
            console.log(a);

            const b = await fetchDataFromAPI2(data1);
            console.log(b);

            const c = await fetchDataFromAPI3(data1);
```

```
            console.log(c);

            document.writeln("All data fetched successfully.");
        } catch (error) {
            console.error("Error:", error);
            document.writeln("An error occurred during data fetching.");
        }
    }
    chainPromises();
</script>
</body>
</html>
```



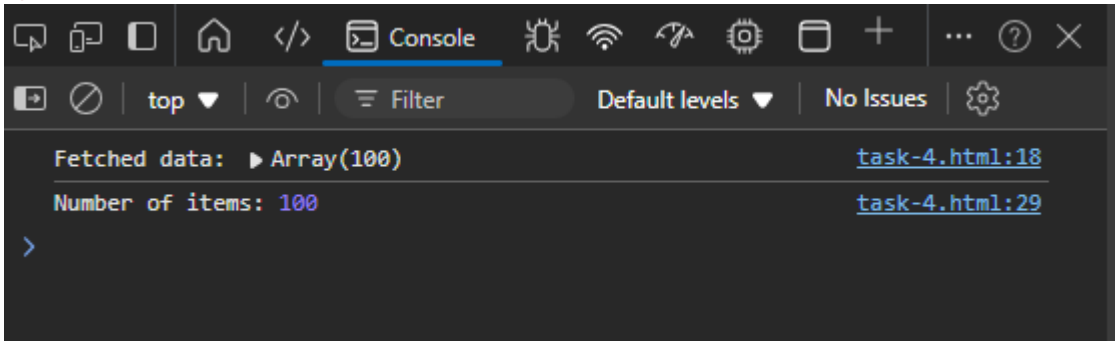
22.
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Webpage</title>
</head>
<body>

<script>
async function fetchDataAndProcess(url) {
try {
const response = await fetch(url);
if (!response.ok) {
throw new Error("Network response was not ok");
}
const data = await response.json();
console.log("Fetched data:", data);
return data.length;
} catch (error) {
console.error("Error fetching data:", error);
}
}
const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
```

```
fetchDataAndProcess(apiUrl).then((result) => {
if (result !== undefined) {
console.log('Number of items:', result);
}
});
</script>

</body>
</html>
```




```
23.<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Webpage</title>
</head>
<body>

<script>
async function fetchDataAndProcess(url) {
    try {
        const response = await fetch(url);
        if (!response.ok) {
            throw new Error("Network response was not ok");
        }
        const data = await response.json();
        console.log("Fetched data:", data);
        return Array.isArray(data) ? data.length : 0;
    } catch (error) {
        console.error("Error fetching data:", error);
        return 0;
    }
}

const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
fetchDataAndProcess(apiUrl).then((result) => {
```

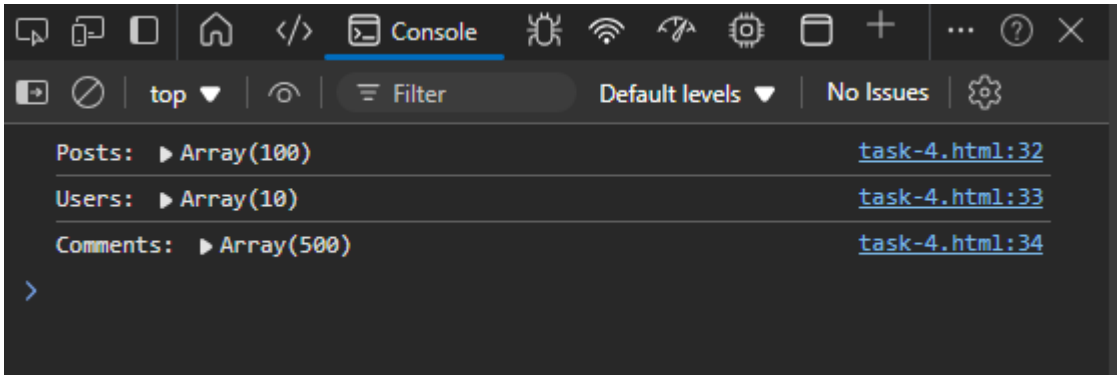
```
    if (result > 0) {
        console.log('Number of items:', result);
    } else {
        console.log('No items fetched or an error occurred.');
    }
});
</script>

</body>
</html>
```



24.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Webpage</title>
</head>
<body>

<script>
async function fetchMultipleResources() {
    const urls = [
        'https://jsonplaceholder.typicode.com/posts',
        'https://jsonplaceholder.typicode.com/users',
        'https://jsonplaceholder.typicode.com/comments'

    ];

    try {

        const fetchPromises = urls.map(async (url) => {
            const response = await fetch(url);
            if (!response.ok) {
                throw new Error(`Failed to fetch from ${url}`);
            }
            return response.json();
        });
```

```
        const results = await Promise.all(fetchPromises);

        console.log('Posts:', results[0]);
        console.log('Users:', results[1]);
        console.log('Comments:', results[2]);


    } catch (error) {
        console.error('Error fetching data:', error);
    }
}
fetchMultipleResources();

</script>
</body>
</html>
```




```
25.<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Webpage</title>
</head>
<body>

<script>
async function executeMultipleTasks() {
  try {
    const task1 = new Promise((resolve) => setTimeout(() => resolve('Task 1
Finished'), 2000));
    const task2 = new Promise((resolve) => setTimeout(() => resolve('Task 2
Finished'), 3000));
    const task3 = new Promise((resolve) => setTimeout(() => resolve('Task 3
Finished'), 1000));
```

```
    const outcomes = await Promise.all([task1, task2, task3]);

    console.log('All tasks have been completed:');
    outcomes.forEach(outcome => console.log(outcome));
  } catch (err) {
    console.error('An error occurred:', err);
  }
}

executeMultipleTasks();

</script>
</body>
</html>
```