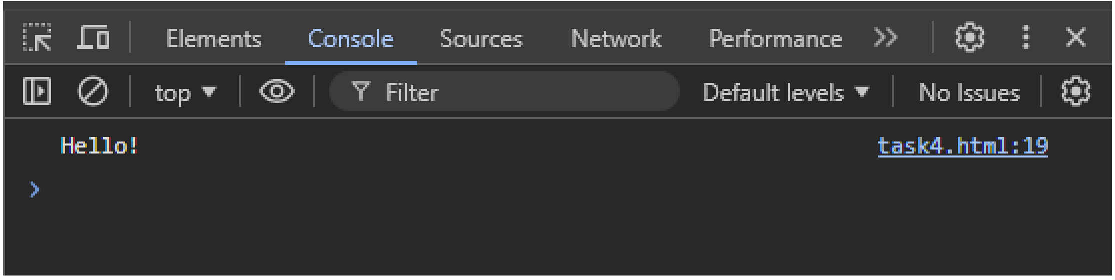
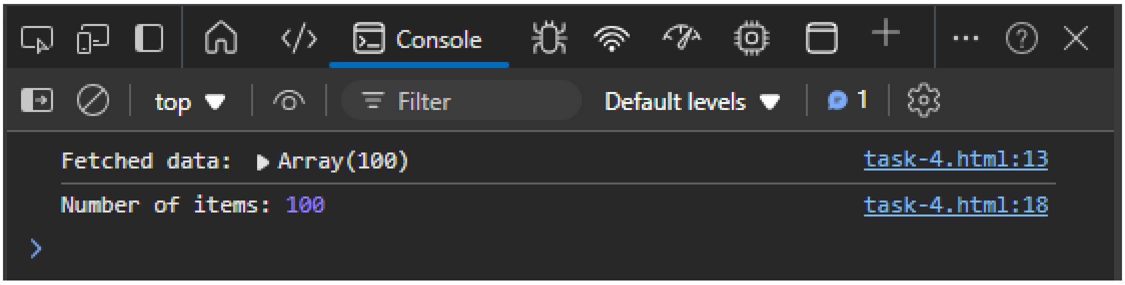
```
16.<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name ="viewport" content="width=device-width,initial-scale=1">
        <title> document</title>
    </head>
<body>
    <script>
function greetAfterSeconds() {
return new Promise((resolve) => {
setTimeout(() => {
resolve("Hello!");
}, 1000);
});
}
greetAfterSeconds().then((greeting) => {
console.log(greeting);
});
</script>
</body>
```



```
Hello!                                          task4.html:19
```

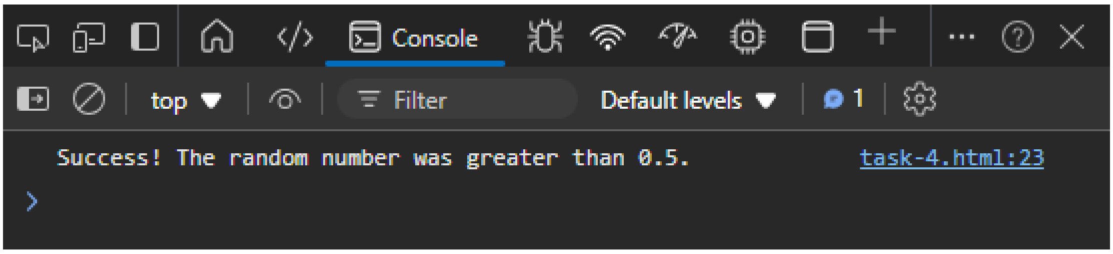
```
17.<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
 <script>
         async function fetchData(url) {
            try {
                const response = await fetch(url);

                if (!response.ok) {
                    throw new Error(`HTTP error! Status: ${response.status}`);
                }

                const data = await response.json();
                console.log('Fetched data:', data);
                const count = data.length;
                console.log('Number of items:', count);
            } catch (error) {
                console.log('Error:', error);
            }
        }

        const apiUrl = 'https://jsonplaceholder.typicode.com/posts';
        fetchData(apiUrl);
 </script>
</body>
</html>
```



```
Fetched data:  ▶ Array(100)          task-4.html:13
Number of items: 100                 task-4.html:18
```

```
18.<!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 randomPromise() {
    const randomNumber = Math.random();
    if (randomNumber > 0.5) {
        return "Success! The random number was greater than 0.5.";
    } else {
        throw new Error("Failure! The random number was less than or equal to 
0.5.");
    }
}

async function run() {
    try {
        const message = await randomPromise();
        console.log(message);
    } catch (error) {
        console.log(error.message);
    }
}
```

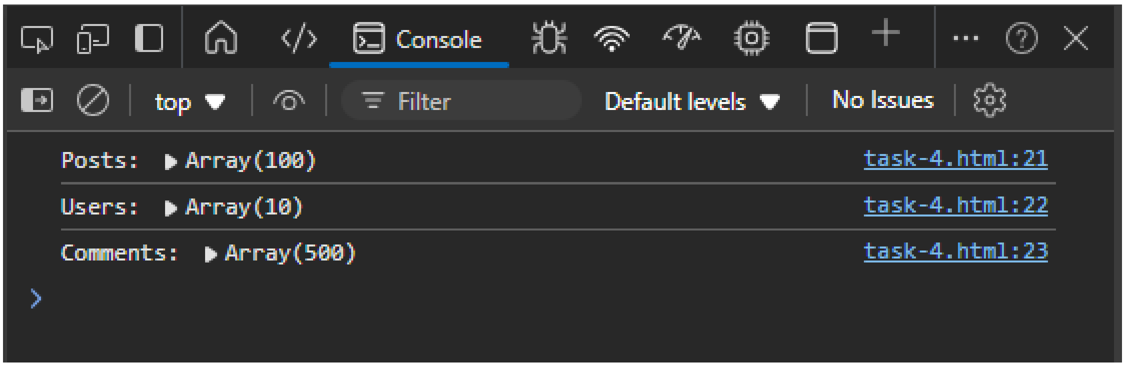
```
run();
</script>

</body>
</html>
```



```
Success! The random number was greater than 0.5.          task-4.html:23
```

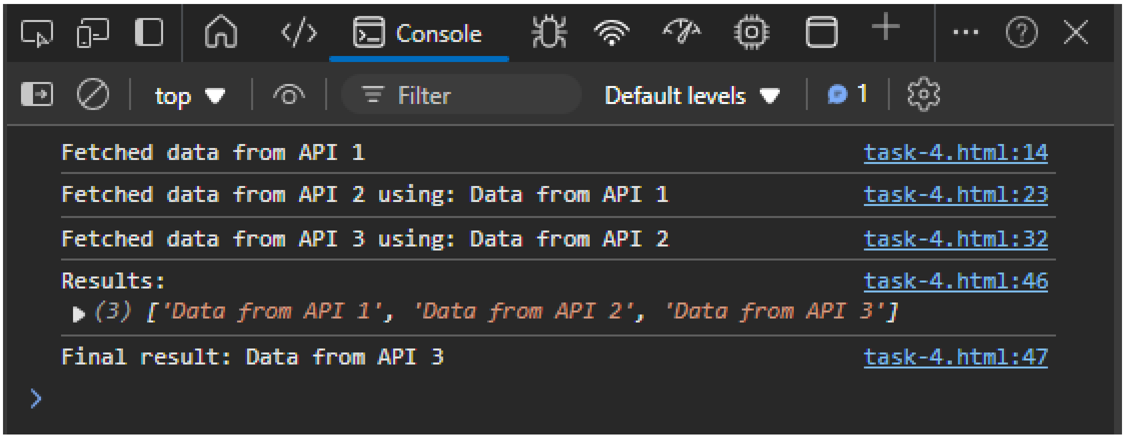
```
19.<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Webpage</title>
</head>
<body>

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

    try {
        const [posts, users, comments] = await Promise.all(urls.map(url =>
fetch(url).then(response => response.json())));

        console.log('Posts:', posts);
        console.log('Users:', users);
        console.log('Comments:', comments);

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

fetchMultipleResources();
</script>

</body>
```

```
</html>
```



```
Posts:  ▶ Array(100)                                task-4.html:21
Users:  ▶ Array(10)                                 task-4.html:22
Comments:  ▶ Array(500)                             task-4.html:23
>
```

```
20.<!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>
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);
    });
}

function fetchAllDataInParallel() {

    const promise1 = fetchDataFromAPI1();
    const promise2 = promise1.then(data => fetchDataFromAPI2(data));
    const promise3 = promise2.then(data => fetchDataFromAPI3(data));

    Promise.all([promise1, promise2, promise3])
        .then((results) => {
            console.log('Results:', results);
            console.log('Final result:', results[2]);
        })
        .catch((error) => {
            console.error("Error:", error);
        });
}

fetchAllDataInParallel();
</script>

</body>
</html>
```



```
Fetched data from API 1                                   task-4.html:14
Fetched data from API 2 using: Data from API 1            task-4.html:23
Fetched data from API 3 using: Data from API 2            task-4.html:32
Results:                                                  task-4.html:46
 ▶ (3) ['Data from API 1', 'Data from API 2', 'Data from API 3']
Final result: Data from API 3                             task-4.html:47
>
```