TASK 16:

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>Document</title>
</head>
<body>
  <SCRIPT>
        function greet(sec) {
    return new Promise((resolve) => {
        setTimeout(() => {
            resolve('Hello!');
        }, sec * 1000);
    });
}
greet(3)
    .then((message) => {
        document.writeln(message);
    });
  </SCRIPT>
</body>
</html>
```

OUTPUT:

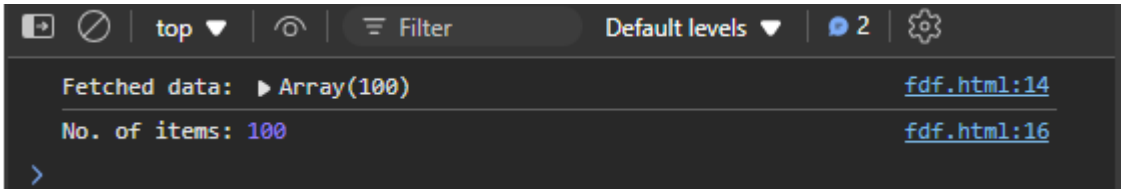
Hello!

TASK 17:

```
<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 res= await fetch(url);
if (!res.ok) {
throw new Error(`error! Status: ${res.status}`);
}
const data = await res.json();
console.log('Fetched data:', data);
const count = data.length;
console.log('No. of items:', count);
} catch (error) {
console.log('Error:', error);
```

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

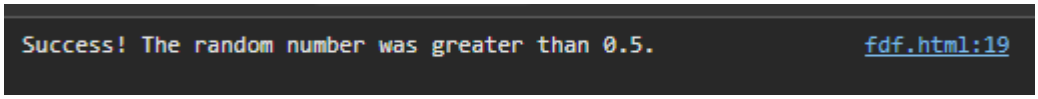
OUTPUT:



TASK 18:

```
<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 rNumber = Math.random();
if (rNumber > 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 msg = await randomPromise();
console.log(msg);
} catch (error) {
console.log(error.msg);
}
}
run();
</script>
</body>
</html>
```

OUTPUT:

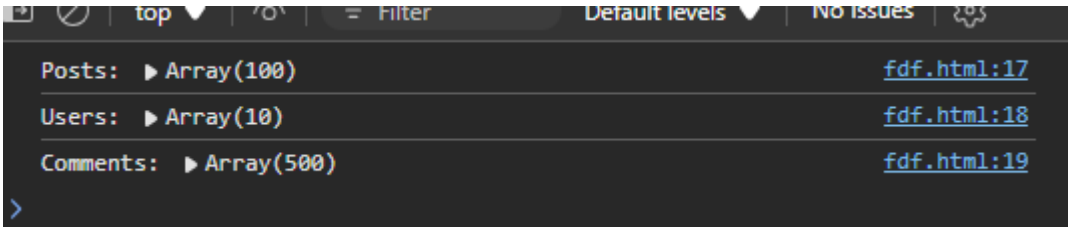


TASK 19:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```html
<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>
```

OUTPUT:



TASK 20:

```html
<!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>
```

OUTPUT: