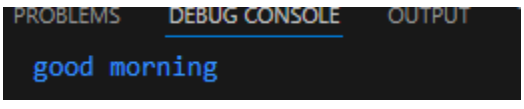
ASYNC/AWAIT:

TASK 1;

```html
<!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>
        async function fetchData() {
            return new Promise((resolve, reject) => {
                setTimeout(() => {
                    resolve('good morning');

                }, 1000);
            });
        }

        async function processData() {
            try {
                const data = await fetchData();
                console.log(data);
            } catch (error) {
                console.log('Error in fetching data:', error);
            }
        }
        processData();
    </script>
</body>

</html>
```



TASK 2:

```
<!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>
         async function fetchandprocessData() {
         try {
              const response = await
fetch('https://jsonplaceholder.typicode.com/posts');
              const data = await response.json();
              console.log( data);
              return data;
              } catch (error) {
              document.writeln('Error fetching data:', error);
              }
           }
            fetchandprocessData();
     </script>
</body>

</html>
```

```
(100) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}
{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}
{…}, {…}, {…}, {…}, {…}, {…}]
```

TASK 3:

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

</html>
```

```
PROBLEMS    DEBUG CONSOLE    OUTPUT    TERMINAL    PORTS    SPELL CHECKER 2          Filter (e.g. text, !exclude, \escape)
 Fetched data: (100) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
 {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
 {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…},
 {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}]
```

TASK 4:

```
<!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>
         async function fetchData() {
           const response = await fetch(
'https://jsonplaceholder.typicode.com/posts');
           const data = await response.json();
           return data;
         }

         async function fetchMultipleData() {
           const urls = [
             'https://jsonplaceholder.typicode.com/posts',
             'https://jsonplaceholder.typicode.com/users',
            ];

         try {
           const [posts, users] = await Promise.all(urls.map(url =>
fetchData(url)));
           console.log('Posts:', posts);
           console.log('Users:', users);
          } catch (error) {
           console.error('Error fetching data:', error);
          }
        }
        fetchMultipleData();
    </script>
</body>

</html>
```