

## JavaScript Async - Practice Code

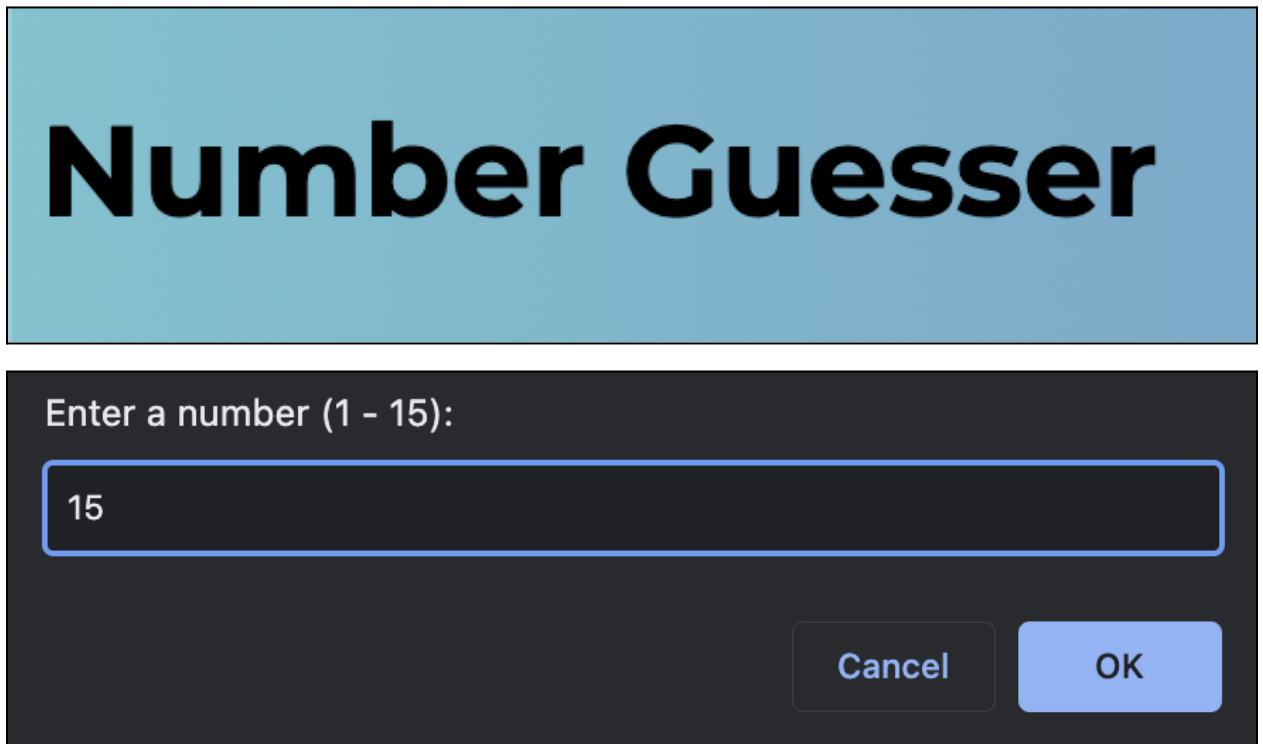
### Steps to see the output:

- Open VS code.
- Select New File in the opened folder...
- Save the file in the appropriate folder/ create the folder and save the file with extension.
- Copy and Add the below given JS code in the file.
- View the file in the browser, by right click on the file name in the left pane.
- View of file in browser.

### Problem Statement 1: Number Guesser

Write a javascript file to make a number guesser game where you ask the user for the input number from 1 to 15 and check whether the number guessed is right or wrong?

### Output:-



**Number Guesser**

Enter a number (1 - 15):

15

Cancel OK

Number: 10: you got 0 score

OK

Do you want to continue?

Cancel

OK

Enter a number (1 - 15):

Number|

Cancel

OK

Error: Wrong Input Type

OK

Game ends

OK

## Problem Statement 2: Random Image Generator

Write a javascript file to make a random image generator using an API call?

Output:-

# Random Image Generator



## Solutions

### Problem Statement 1:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Number Guesser</title>
  <style>
    body{
      height:100%;
      background-image: linear-gradient(to right, rgb(113, 196, 209),
rgb(102, 89, 187));
      font-family: "Montserrat","Helvetica Neue","sans-serif";
    }
  </style>
</head>
<body>
  <h1>Number Guesser</h1>
  <script>
    const enterNumber = () => {
      return new Promise((resolve, reject) => {
        const userNumber = Number(window.prompt("Enter a number (1 -
15):")); // Ask the user to enter a number
        const randomNum = Math.floor(Math.random() * 15 + 1); // Pick
a random number between 1 and 6

        if (isNaN(userNumber)) {
          reject(new Error("Wrong Input Type")); // If the user enters
a value that is not a number, run reject with an error
        }

        if (userNumber === randomNum) { // If the user's number
matches the random number, return 2 score
          resolve({
            score: 2,
            randomNum,
          });
        } else if (
          userNumber === randomNum - 1 ||
          userNumber === randomNum + 1
        ) { // If the user's number is different than the random
number by 1, return 1 point
          resolve({
            score: 1,
            randomNum,
          });
        } else { // Else return 0 score
          resolve({
            score: 0,
            randomNum,
          });
        }
      });
    }
  </script>

```

```

        });
    };

    const continueGame = () => {
        return new Promise((resolve) => {
            if (window.confirm("Do you want to continue?")) { // Ask if
the user want to continue the game with a confirm modal
                resolve(true);
            } else {
                resolve(false);
            }
        });
    };

    const handleGuess = async () => {
        try {
            const result = await enterNumber(); // Instead of the then
method, we can get the result directly by just putting await before the
promise

            alert(`Number: ${result.randomNum}: you got ${result.score}
score`);

            const isContinuing = await continueGame();

            if (isContinuing) {
                handleGuess();
            } else {
                alert("Game ends");
            }
        } catch (error) { // Instead of catch method, we can use the
try, catch syntax
            alert(error);
        }
    };
    handleGuess(); // Run handleGuess function
</script>
</body>
</html>

```

## Problem Statement 2:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <title>Random Image Generator</title>
    <style>
        body{
            height:100%;
            background-image: linear-gradient(to right, rgb(161, 113, 209),
rgb(18, 8, 85));
            font-family: "Montserrat","Helvetica Neue","sans-serif";

```

```

    }
    .container{
        display:flex;
        flex-direction: column;
        align-items:center;
        justify-content: center;
    }
    #image{
        width:350px;
        height:350px;
        border: 5px solid #fff;
        box-sizing: border-box;
        overflow: hidden;
    }
</style>
</head>
<body>
    <div class="container">
        <h1 style="text-align:center; color:azure;">Random Image
Generator</h1>
        <img id="image"/>
    </div>
    <script>
        const url = 'https://source.unsplash.com/random'
        const img = document.getElementById("image");
        const getImage = async (url) => {
            return await fetch(url).then(res => res.url)
        }
        const img_url=getImage;
        getImage(url).then(result => {
            img.setAttribute('src', result)
        });
    </script>
</body>
</html>

```