

### ### Exercise 1: Variables and Data Types

**\*\*Task:\*\*** Declare variables of different data types and log them to the console.

**\*\*Solution:\*\***

```
``javascript
```

```
let myString = "Hello, World!";
```

```
let myNumber = 42;
```

```
let myBoolean = true;
```

```
let myNull = null;
```

```
let myUndefined;
```

```
let myObject = { name: "John", age: 30 };
```

```
console.log(myString);
```

```
console.log(myNumber);
```

```
console.log(myBoolean);
```

```
console.log(myNull);
```

```
console.log(myUndefined);
```

```
console.log(myObject);
```

```
``
```

### ### Exercise 2: Functions

**\*\*Task:\*\*** Write a function that takes two numbers as arguments and returns their sum.

**\*\*Solution:\*\***

```
``javascript
```

```
function sum(a, b) {  
    return a + b;  
}
```

```
console.log(sum(5, 3)); // Output: 8
```

```
````
```

### ### Exercise 3: Arrays

**\*\*Task:\*\*** Create an array of fruits and add a new fruit to the end. Then, remove the first fruit from the array and log the updated array.

**\*\*Solution:\*\***

```
``javascript
```

```
let fruits = ["Apple", "Banana", "Cherry"];
```

```
fruits.push("Dragonfruit");
```

```
fruits.shift();
```

```
console.log(fruits); // Output: ["Banana", "Cherry",  
"Dragonfruit"]
```

```
``
```

### ### Exercise 4: Objects

**\*\*Task:\*\*** Create an object representing a car with properties `make``, `model``, and `year``. Add a new property `color``, and then log the object.

**\*\*Solution:\*\***

```
``javascript
```

```
let car = {
```

```
  make: "Toyota",
```

```
  model: "Corolla",
```

```
    year: 2020
};

car.color = "Red";

console.log(car);
``
```

### ### Exercise 5: DOM Manipulation

**\*\*Task:\*\*** Select an HTML element with the ID `demo` and change its text content to "Hello, JavaScript!".

**\*\*Solution:\*\***

```
``html
<!DOCTYPE html>
<html>
<head>
  <title>DOM Manipulation</title>
</head>
```

```
<body>
  <div id="demo">Original Text</div>
  <script>
    document.getElementById("demo").textContent
= "Hello, JavaScript!";
  </script>
</body>
</html>
'''
```

### ### Exercise 6: Event Handling

**\*\*Task:\*\*** Create a button that alerts "Button Clicked!" when clicked.

**\*\*Solution:\*\***

```
```html
<!DOCTYPE html>
<html>
<head>
  <title>Event Handling</title>
```

```
</head>
<body>
  <button id="myButton">Click Me</button>
  <script>

document.getElementById("myButton").addEventListener("click", function() {
    alert("Button Clicked!");
});
</script>
</body>
</html>
```
```

### ### Exercise 7: Error Handling

**\*\*Task:\*\*** Write a function that divides two numbers and handles the error if the second number is zero.

**\*\*Solution:\*\***

```
```javascript
```

```
function divide(a, b) {  
  try {  
    if (b === 0) {  
      throw new Error("Division by zero is not  
allowed.");  
    }  
    return a / b;  
  } catch (error) {  
    return error.message;  
  }  
}
```

```
console.log(divide(10, 2)); // Output: 5
```

```
console.log(divide(10, 0)); // Output: Division by  
zero is not allowed.
```

```
...
```

### ### Exercise 8: ES6 Features

**\*\*Task:\*\*** Use template literals to create a greeting message and log it to the console.

**\*\*Solution:\*\***

```
``javascript
```

```
let name = "Alice";
```

```
let greeting = `Hello, ${name}! Welcome to  
JavaScript.`;
```

```
console.log(greeting); // Output: Hello, Alice!  
Welcome to JavaScript.
```

```
``
```

### ### Exercise 9: JSON

**\*\*Task:\*\*** Convert a JavaScript object to a JSON string and then parse it back to an object.

**\*\*Solution:\*\***

```
``javascript
```

```
let user = {  
  name: "John",  
  age: 30
```



```
};
```

```
let jsonString = JSON.stringify(user);
```

```
let parsedObject = JSON.parse(jsonString);
```

```
console.log(jsonString); // Output:
```

```
{"name":"John","age":30}
```

```
console.log(parsedObject); // Output: { name: 'John',  
age: 30 }
```

```
``
```

### ### Exercise 10: Form Validation

**\*\*Task:\*\*** Create a simple form with an input field and validate that the input is not empty when the form is submitted.

**\*\*Solution:\*\***

```
``html
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Form Validation</title>
</head>
<body>
  <form id="myForm">
    <input type="text" id="myInput"
placeholder="Enter something">
    <button type="submit">Submit</button>
  </form>
  <script>

document.getElementById("myForm").addEventListener("submit", function(event) {
  let input =
document.getElementById("myInput").value;
  if (input === "") {
    alert("Input cannot be empty");
    event.preventDefault();
  }
});
</script>
</body>
</html>
```