

# Tutorial 4

You may use repl.it (Node.JS template), Visual Studio Code or simply use your browser to work on your exercises. Place completed answers into a word document or zip up your project folder and submit. You can download the tutorial04-students.zip as a base template

## Reference:

- [First Steps with Arrays](#)
- [Learn about Arrow Functions](#)
- [Learn about IIFE](#)

## Exercise 1: Initialize and Access a 2D Array

**Objective:** Learn to create, access, and modify elements in a 2D array.

### Instructions:

- Create a 2D array of integers.
- Access and print the element at the second row and third column.
- Change the value of this element and print the entire array.

## Exercise 2: Remove Elements from an Array

### Initialize and Print a 2D Array

**Objective:** Learn to create and display a 2D array.

### Instructions:

- Initialize a 2D array with specific values.
- Write a nested loop to print each element in the format of its coordinates and value.

## Exercise 3: Classroom Seating Chart

**Objective:** Create and modify a 2D array to represent a classroom seating chart.

### Instructions

- Initialize a 2D array representing seats in a classroom (5 rows by 4 columns).
- Assign student names to each seat.
- Change a student's seat and print the updated seating chart.

## Exercise 4: Sum of All Elements in a 2D Array

**Objective:** Calculate and print the sum of all elements in a 2D array.

### Instructions:

Use nested loops to iterate through the array and sum all the elements.

### Exercise 5: Printing Diagonal Elements

**Objective:** Print elements on the diagonal of a square 2D array.

**Instructions:**

For a square matrix ( $n \times n$ ), print elements where the row index equals the column index.  
Discuss how this would change if printing the secondary diagonal.

### Exercise 6: Find the Largest Number in Each Row

**Objective:** Identify and print the largest number in each row of a 2D array.

**Instructions:**

- Iterate through each row of the array.
- For each row, find the largest number.
- Print the largest number for each row.

### Exercise 7: Product Inventory Grid

**Objective:** Manage a small product inventory using a 2D array.

**Instructions**

Create a 2D array to represent types of products in rows and their properties (name, price, quantity) in columns.

Update the inventory quantity for a product and print the updated inventory.

### Exercise 8: Basic IIFE Usage

**Objective:** Write a simple IIFE that prints "Hello, World!" using both traditional function syntax and arrow function syntax.

**Instructions:**

Create an **IIFE** using traditional function syntax.

```
(function() {  
  
})();
```

Create an **IIFE** using arrow function syntax.

```
(( ) => {  
  
})();
```

### Exercise 9: IIFE with Parameters

**Objective:** Pass parameters to an IIFE and use them inside the function to perform a calculation.

**Instructions:**

Write an IIFE that takes two numbers as parameters and logs their sum

### Exercise 10: Convert to Arrow Function

**Objective:** Convert a traditional function expression to an arrow function.

**Instructions:**

Convert the following traditional function into an arrow function:

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

### Exercise 11: Single Argument Arrow Function

**Objective:** Write an arrow function that takes one argument and returns the square of that number.

**Instructions:**

Create an arrow function named square that takes a single argument and returns its square.

### Exercise 12: No Argument Arrow Function

**Objective:** Write an arrow function that returns a default greeting message.

**Instructions:**

Create an arrow function that takes no arguments and returns the string "Hello, World!".

## Exercise 16: Debugging Challenge

### Task

- Identify and fix issues so the function correctly adds two numbers and returns the sum.
- Ensure the function handles different data types appropriately.

```
function addNumbers(a, b) {
  var sum = a + b;
  console.log(sum);
}
```

```
addNumbers(10, '20'); // Incorrectly prints "1020" instead of 30
```