

JavaScript Practice 001

Module 1.1

1. // Types, Values, and Variables/ Functions

Instruction:

- Write function **findMin(arr)** : Find min number in an array : Return min number

Initiate code:

```
let numbers = [34, 1, 58, 8, 21, 5, 13, 3, 89, 7];  
console.log(findMin(arr)); // 1  
// Insert your code here
```

2. // Types, Values, and Variables

Instruction:

- Convert **numString** to an integer and store it in a new variable **num**.

Initiate code:

```
// XXXXX is your code  
let numString = "1234";  
let num = XXXXX;  
console.log(typeof num); // Number
```

3. // Types, Values, and Variables/ Arrays

Instruction:

- Write code to swap their values using **array destructuring**.

Initiate code:

```
// XXXXX is your code  
let a = 5, b = 10;  
XXXXX = [a, b];  
console.log(a); // 10  
console.log(b); // 5
```

4. // Types, Values, and Variables/ Arrays

Instruction:

- Check if **arrays** are **empty**.

Initiate code:

```
let value1 = [1, 2, 3];  
let value2 = [];  
let value3;  
// XXXXX is the code to check each variable  
console.log(XXXXX); // value1 false  
console.log(XXXXX); // value2 true  
console.log(XXXXX); // value3 false
```

5. // Types, Values, and Variables/ Arrays

Instruction:

- Write code to display only the **object elements** in the **mixedArray**.

Initiate code:

```
let mixedArray = [42, "hello", {name: "Joe", surname: "Doe"}, true, 56, false, {name: "John", surname: "Smith"}, {name: "Joy", surname: "Stein"}];
```

// Insert your code here

Output:

```
{name: "Joe ", surname: "Doe"}  
{name: "John ", surname: "Smith"}  
{name: "Joy ", surname: "Stein"}
```

6. // Types, Values, and Variables/ Expressions, Operators, and Control Structures

Instruction:

- No coding, What is **result** and **type of result** ?

Initiate code:

// XXXXX : What is you think about result

```
let x = 10;
```

```
let y = "20";
```

```
let result = x + y;
```

```
console.log(result); // XXXXX
```

```
console.log(typeof result); // XXXXX
```

7. // Objects

Instruction:

- Create an object **person** with properties: **name** (a string), **age** (a number), and **isStudent** (a boolean).

7.1 Add a new property **hobbies**, an array of strings, to the **person** object.

7.2 Change the **age** property to a new value.

7.3 Delete the **isStudent** property.

Initiate code:

// Insert your code here

8. // Expressions, Operators, and Control Structures

Instruction:

- No coding, What is the result of the code below, and why is it displayed?

Initiate code:

// XXXXX : What is you think about result

```
if(1){
  console.log(1); // XXXXX
}
if({}){
  console.log({}); // XXXXX
}
if(''){
  console.log(''); // XXXXX
}

if(null){
  console.log(null); // XXXXX
}
if(undefined){
  console.log(undefined); // XXXXX
}
```

9. // Expressions, Operators, and Control Structures / Objects

Instruction:

- Compare three objects with name properties (case-insensitive).

Initiate code:

// XXXXX is your code

```
let peopleObj1 = { name: "John", surname: "Doe" },
let peopleObj2 = { name: "Sarah", surname: "Wilson" },
let peopleObj3 = { name: "David", surname: "Brown" },
if(XXXXX){
  console.log("These two objects have the same name.");
}
```

10. // Arrays / Arrays Functions / Objects

Instruction:

- Display **all the names of people objects** (case-insensitive) that have the **same name**; return an array of strings with the names that are the same.

Initiate code:

```
// XXXXX is your code
let people = [
  { name: "John", surname: "Doe" },
  { name: "olivia", surname: "Smith" },
  { name: "john", surname: "Johnson" },
  { name: "Sarah", surname: "Wilson" },
  { name: "David", surname: "Brown" },
  { name: "Emily", surname: "Davis" },
  { name: "David", surname: "Miller" },
  { name: "Olivia", surname: "Taylor" },
  { name: "david", surname: "Anderson" },
  { name: "Sophia", surname: "Clark" }
];
let result = XXXXX
console.log(result); // ["john", "olivia", "david"]
```

11. // Types, Values, and Variables

Instruction:

- No coding, What is the result of the code below, and why is it displayed?

Initiate code:

```
// XXXXX : What is you think about result
const arr = [1, 2, 3];
arr.push(4); // XXXXX
arr = [1, 2, 3, 4]; // XXXXX
const obj = { name: 'Alice' };
obj.age = 30; // XXXXX
obj = { name: 'Alice', age: 30 }; // XXXXX
```

12.// Functions / Arrays / Arrays Functions

Instruction:

- Write a function **uniqueElements(arrayValue)** to return array of the unique elements from an array.

Initiate code:

```
let numbers = [1, 1, 3, 8, 21, 8, 13, 3, 8, 7];
// Insert your code here
// array numbers after use function uniqueElements will be [1, 3, 8, 21, 13, 7]
```

13. // Functions / Arrays / Arrays Functions

Instruction:

- Write a function **reverseArray(arrayValue)** to reverse an array.

Initiate code:

```
let numbers = [1, 1, 3, 8, 21, 8, 13, 3, 8, 7];  
// Insert your code here  
// return value will be [7, 8, 3, 13, 8, 21, 8, 3, 1, 1]
```

14. // Expressions, Operators, and Control Structures / Arrays / Arrays Functions

Instruction:

- Given an array of objects, use destructuring to extract and display all properties from each object.

Initiate code:

```
// XXXXX is your code  
const users = [  
  { id: 1, name: 'John', age: 28 },  
  { id: 2, name: 'Jane', age: 32 },  
  { id: 3, name: 'Dave', age: 24 }  
];  
users.forEach(XXXXX);
```

15. // Expressions, Operators, and Control Structures / Arrays

Instruction:

- Destructure the first two elements and the last element of an array into new variables.

Initiate code:

```
const numbers = [1, 2, 3, 4, 5];  
// Insert your code here
```

16. // Expressions, Operators, and Control Structures

Instruction:

- Create a function named **multiplicationTable(number)** that generates a multiplication table based on the provided **number**. The function should return the results as an array of strings.

Initiate code:

```
multiplicationTable(number){  
  // Insert your code here
```

```
}  
console.log(multiplicationTable(2));
```

output

```
[  
  "2 x 1 = 2",  
  "2 x 2 = 4",  
  "2 x 3 = 6",  
  "2 x 4 = 8",  
  "2 x 5 = 10",  
  "2 x 6 = 12",  
  "2 x 7 = 14",  
  "2 x 8 = 16",  
  "2 x 9 = 18",  
  "2 x 10 = 20",  
  "2 x 11 = 22",  
  "2 x 12 = 24",  
]
```

17. // Expressions, Operators, and Control Structures / Arrays / Arrays Functions / Functions

Instruction:

- Create a function **calculateGrade(scores)** that takes an array of numerical scores and returns the corresponding letter grade (A, B, C, D, or F) based on the average score.

17.1 Calculate the average score.

17.2 Based on the average score, determine the letter grade:

- 90-100: A
- 80-89: B
- 70-79: C
- 60-69: D
- Below 60: F

Return the letter grade.

Initiate code:

```
function calculateGrade(scores) {  
  // Insert your code here  
}  
console.log(calculateGrade([80, 95, 90, 70, 100]));
```

output

B

18. // Expressions, Operators, and Control Structures / Functions

Instruction:

- Write a JavaScript function **convertTemperature** that converts a temperature from Fahrenheit to Celsius, or from Celsius to Fahrenheit, based on a mode parameter.

18.1 The function should accept two arguments: **temperature** (a number) and **mode** (a string). The **mode** argument can be either **'FtoC'** for Fahrenheit to Celsius conversion or **'CtoF'** for Celsius to Fahrenheit conversion.

18.2 Implement the conversion logic:

- Fahrenheit to Celsius: **$(\text{temperature} - 32) * 5/9$**
- Celsius to Fahrenheit: **$(\text{temperature} * 9/5) + 32$**

18.3 Return the converted temperature as a number. If the mode is neither **'FtoC'** nor **'CtoF'**, return a message 'Invalid mode. Please use "FtoC" or "CtoF"'.

Initiate code:

```
// Insert your code here  
console.log(convertTemperature(68, 'FtoC'));  
console.log(convertTemperature(20, 'CtoF'));  
console.log(convertTemperature(100, 'Unknown'));
```

output

20

68

Invalid mode. Please use "FtoC" or "CtoF"

19. // Expressions, Operators, and Control Structures / Arrays / Arrays Functions / Functions

Instruction:

- Write a JavaScript function, **sumPositiveNumbers(numbers)**, that calculates the sum of all positive numbers in an array. The function should return the sum of positive numbers. If the array is empty or contains no positive numbers, the function should return 0.

Initiate code:

// Insert your code here

```
console.log(sumPositiveNumbers([1, -4, 12, 0, -3, 29, -150]));
```

```
console.log(sumPositiveNumbers([-1, -2, -3]));
```

```
console.log(sumPositiveNumbers([]));
```

output

42

0

0

20. // Expressions, Operators, and Control Structures / Arrays / Arrays Functions / Functions

Instruction:

- Write a JavaScript function **sameNumbers(numArray1, numArray2)** to receive two arrays of numbers and return an array that contains the numbers from both arrays without repeating numbers.

Initiate code:

// Insert your code here

```
console.log(sameNumbers([1,2,3,4,5,2,1,4,1,5,2,5,8],[1,2,5,8,4,1,5,1,47,2,3,56,5,8]));
```

output

[

1, 2, 3, 4,

5, 8, 47, 56

]

