1. Create an arrow function called square that takes a number as an argument and returns its square. Use the arrow function to calculate the square of a given number and display the result.

Answer:

```
J
const square = (num) => num * num;

const number = 10;
const squarecINumber = square(number);
console.bg("Square:", squaredNumber)
```

2. The following is an array of 10 students ages:

=> **const ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]**

- **Sort the array and find the min and max age.**
- **Find the median age{one middle item or two middle items divided by two)**
- **Find the average age{all items divided by number of items)**
- **Find the range of the ages(max minus min)**
- **Compare the value of (min - average) and (max - average), use absQ method**

Answe1

```
canst ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24];

ages.sort((a, b) => a - b);

canst minAge = ages[0];
canst maxAge = ages[ages.length - 1];

let medianAge;
if (ages.length % 2 === 0) {
 const mid1 = ages[Math.floor(ages.length / 2) - 1];
 const mid2 = ages[Math.floor(ages.length / 2)];
 medianAge = (mid1 + mid2) J 2;
} else {
 median Age = ages[Math.floor(ages.length / 2)]:
}
const sumAges = ages.reduce((sum, age) => sum + age, 0); canst averageAge = sumAges / ages.length;

const ageRange = maxAge - minAge;

const minDifference = Math.abs(minAge - averageAge);
const maxDifference = Math.abs(maxAge - averageAge);

console.log('Sorted Ages:', ages);
console.log('Min Age:', minAge);
console.log('Max Age:', maxAge);
console.log('Median Age:', medianAge);
console.log('Average Age:', averageAge);
console.log('Age Range:', ageRange);
console.log('Min - Average Difference:', minDifference);
console.log('Max - Average Difference:', maxDifference);
```

3. Create a Map to store contact information (name, age, email, location) and implement a function to retrieve contact details by name.

Answer:

```
Const contactMap = new Map();
contactMap.set(“John” , { age: 30, email: “John@gmail.com”, location: “Bangalore”,});

contactMap.set(“Bob”, {age: 35, email:”bob@example.com”, location: “UP”,});

function getContact(name){ return contactMap.get(name);}

console.log(getContact(“John”));
```

4. Create two objects personland person2 with properties name and age. Create a function "introduce" that prints "Hello, Im [name], and I'm [age] years old." Use the call method to make person2 introduce itself using the introduce function.

Answer:

```
const person1 = { name: "Alice", age: 25 };
const person2 = { name: "Bob", age: 30 };


function introduce() {
 console.log('Hello, I'm ${this.name}, and I'm ${this.age} years old.');
}

introduce.call(person2);
```

5. You are developing a program to manage a list of unique items. Write a JavaScript program that uses a Set to store a collection of unique numbers. Use the Map object to associate each number with its square. Finally, print both the unique numbers and their corresponding squares.

Answer:

```
let uniqueNumbers = new Set([3, 7, 5, 7, 2, 3, 8]); // Replace with your unique
numbers

let numberSquareMap = new Map();

uniqueNumbers.forEach(number => {
 numberSquareMap.set(number, number * number);
});

console.log("Unique Numbers:");
console.log(Array.frorn(uniqueNumbers).join(', '));

console.log("\nNumber-Square Map:");
numberSquareMap.forEac'h((square, number) => {
 console.log('${number} : ${square}');
```

6

- **Create a simple JavaScript function named displayInfo that takes two parameters (name and role) and logs a message.**
- **Use call to invoke the displayInfo function with specific arguments.**
- **Use apply to invoke the displayInfo function with arguments passed as an array.**
- **Create another function named greet that logs a greeting with this context.**
- **Use bind to create a new function boundGreet with a specific context and log the greeting.**

Solution:

```
function displayinfo (name, role){
console.log("Name : ${name}, Role : ${role}" );
}

displayinfo. call (null, "'Prabir'", "Developer");

displayinfo. apply (null, ["'Mithun'", "SDE"J);

function greet (){
 console.log( "Hello, ${this.name}!");
}

const user = {name : "'PK"' };
const boundGreet = greet.birld (user);
boundGreet ();


output:
Name: Prabir, Role: Developer
Name: Mithun, Role: SDE
Hello, PK!
```

7. Tasks:

- **Create an object named calculator with methods add, subtract, and multiply.**
- **Implement the calculate method in the calculator object, which takes an operation ('add', 'subtract', or 'multiply') and two numbers.**
- **Use call to perform an addition operation using the calculate method.**
- **Use apply to perform a multiplication operation using the calculate method with arguments as an array.**
- **Create another object named discountCalculator with a discount percentage property and a method applyDiscount.**
- **Use bind to create a new function calculateDiscount that is bound to the discountCalculator object and can be reused.**

Solution:

```
const calculator = {
 add: function (a, b) {
  return a + b;
 },
 subtract: function (a, b) {
  return a - b;
 },
 multiply: function (a, b) {
  return a * b;
 },
 calculate: function (operation, a, b) {
  if (operation === 'add') {
   return this.add(a, b);
  } else if (operation === 'subtract') {
    return this.subtract(a, b);
   } else if (operation === 'multiply') {
    return this.multiply(a, b);
   }
 },
};

const additionResult = calculator.calculate.call(calculator, 'add', 5, 3);
console.log(`Addition Result: ${additionResult}`); // Addition Result: 8
```

```
const multiplicationResult = calculator.calculate..apply(calculator, ['multiply', 4, 2]);
console.log("Multiplication Result: ${multiplicationResult}"); //Multiplication Result: 8

const discountCalculator = {
 discountPercentage: 10,
 applyDiscount: function (amount){
  return amount - (a11Tiount * this.discountPercentage) / 100;
 }'
};

canst calculateDiscount =
discountCalculator.applyDiscount.bind(discountCalculator);
const discountedAmount = calculateDiscount(100);
console.log("Discounted Amount: ${discountedAmount}"); // Discounted Amount: 90
```