**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 squar e = ( num) => num \* num ;

const numbe r = 10 ;

const squar eclNumbe r = squa re (number ) ;

console .log ( " Squa re : " , squaredNumbe r )

1. 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 . sor t ( ( a , b ) = > a - b ) ;

canst minAge = ages [ 0 ] ;

canst ma xAge = ages [ ages . length - 1 ] ;

let med ian Age ;

if ( ages . lengt h % 2 === 0 ) {

con st m id1 = ages [ M at h . f loo r ( ages . lengt h *I* 2 ) - 1 ] ; con st m id 2 = ages [ M at h . f loo r ( ages . lengt h *I* 2 ) ] ;

med ian Age = ( mid1 + mid 2 ) *J* 2 ;

} else {

med ian Age = ages [ Ma t h . f loo r ( a ges . length *I* 2 ) ] :

}

con st sumAges = ages . red uce ( ( sum , a ge) => sum + a ge, 0 ) ; canst ave r ageAge = sumAges *I* a ges . len gth ;

con st age Ran ge = ma xAge - minAge ;

con st min Diff e rence = Ma t h . abs ( minAge - a ve r a geAge ) ;

con st ma x Diff e rence = Ma t h . abs ( ma xAge - a ve r a geAge ) ;

console . log ( ' Sor ted Ages : ' , ages ) ; console . log ( ' Min Age : ' , minAge ) ; console . log ( ' Max Age : ' , ma xAge ) ; console . log ( ' Media n Age : ' , med ia nAge ) ;

console . log ( ' Ave r a ge Age : ' , ave r ageAge ) ; console . log ( ' Age Ra nge : ' , ageRange) ;

console . log ( ' M in - Ave rage Dif f e rence : ' , minDif f e r ence ) ; console . log ( ' Ma x - Ave r age Dif f e rence : ' , maxDif f e r ence ) ;

1. 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”));

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

**Answer:**

###### con st pe r son1 = { na me : "Alice " , age : 25 } ; con st pe r son2 = { na me : " Bob " , age : 30 } ;

funct ion int r oduce ( ) {

console . log ( ' Hello , I ' m ${th is . n ame}, and I ' m ${this . age} yea r s old . ' ) ;

}

###### int r od u ce . ca ll ( pe r son2 ) ;

1. 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 un iq u eNumbe r s = n ew Set ( [ 3 , 7 , 5 , 7 , 2 , 3 , 8 ]) ; *I I* R eplace wit h you r u niq u e n umbe r s

let n umbe rSq u a r eMap = n ew Ma p ( ) ;

un iqueN umbe r s .f or Each ( n umbe r = > {

n um be rSq u a reMa p . set ( n umbe r , n um be r \* n umbe r ) ;

} ) ;

###### con sole . log ( "Un ique Num be r s : " ) ;

console . log ( A r r ay . frorn ( u niq u eNumbe r s) . join ( ' , ' ) ) ;

console . log ( " \ nNu mbe r -Squa r e Ma p : " ) ;

n um be rSq ua r eMap . for E a c'h ( ( squ a re , n umbe r ) => { console . log ( '${numbe r } : ${squa re} ' ) ;

6

* + Create a simple JavaScript function named displaylnfo that takes two parameters (name and role) and logs a message.
  + Use call to invoke the displaylnfofunction with specific arguments.
  + Use apply to invoke the displaylnfo 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 bound Greet *=* greet.birld (user ); boundGreet ();

output:

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

1. 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:

###### co n st ca lcu la to r = {

ad d : fu nc tion ( a , b ) { r etu r n a + b ;

} I

###### subt r a ct : f u n ct ion ( a , b ) { r etu r n a - b ;

} I

###### mu ltiply : f u nct ion ( a , b ) { r etu rn a \* b ;

} I

###### ca lcu la te : f u nct ion ( ope r a t ion , a , b ) { if ( ope r a t ion == = ' ad d ' ) {

ret u rn t h is . ad d ( a , b ) ;

} else if ( ope ra t ion == = ' subt r act ' ) {

return this .subtract (a, b);

}else if (ope ration === 'multiply '){ return this.multiply(a , b);

}

} I

};

const additionResult = calculator.calculate.call(calcu lator, ·add ·, 5, 3); console.log ('Addition Result : ${additionResult} '); *I I* Addition Resu lt: 8

##### const multiplicationResult = calculator.calculate ..apply (calculator, [·multiply ·,

4 , 2 ] ) ;

##### console.log( " Multiplication Result: ${multiplicationResult} '); //Multiplication Result: 8

const discountCalculator = {

discountPercentage: 1 0, applyDiscount: function (amoun t ){

return amount - (a11Tiount \* this.discountPe rcentage) *I* 100;

} I

};

##### canst calculateDiscount =

discountCalculator. applyDiscount.bind( discountCalculator) ;

const discountedAmount = calculateDiscount (100);

console.log( " Discounted Amount: ${discountedAmount} '); *I I* Discounted Amount: 90