|  |
| --- |
| **Q. What is Primitive Data Type ?**  Ans : A value stored easily in variable that called "Primitive Data Type" |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Q. How may Primitive data type ?**  Ans : There are 7 types of Primitive data type   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 1.String | 2.number | 3. boolean | 4.undefined | 5.null | 6.bigInt | 7.Symboll | |  |  |  |  |  |  |  | |

|  |  |  |
| --- | --- | --- |
| **Q. How to convert String to Number ?**  **Ans :** use **parseInt()** and **+** Operator   |  |  | | --- | --- | | **Example with parseInt()**  let a = "Rajnish"  let output = parseInt(a)  console.log(typeof output) | **Example with + Operator**  let a = "Rajnish"  let output = +a  console.log(typeof output) |   **Q. what is data type when convert string to number ?**  Ans : Number |

|  |
| --- |
| **Q. How to convert number to String ?**  **Ans :** use double comma and single comma  **Ex**  let a = "9874565"; console.log(typeof a) |

|  |
| --- |
| **Q. what is return when we convert “true” value to number ?**  **Ans** : 1 |

|  |
| --- |
| **Q. what is return when we convert “false” value to number ?**  **Ans** : 0 |

|  |
| --- |
| **Q. what is return when we convert “undefined” value to number ?**  **Ans** : NaN |

|  |
| --- |
| **Q. what is return when we convert “null” value to number ?**  **Ans** : 0 |

|  |
| --- |
| **Q. What is data type of null?**  **Ans** : Object |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Q. Different between “var” , “let” & “Const” ?**  **Ans** :   |  |  |  | | --- | --- | --- | | var | Let | const | | 1. Var multiple declare 2. Var multiple assign 3. Var use for hoisting | 1. Let single declare 2. Let multiple assign 3. Let not use for hoisting | 1 const single time deceleare  2. const single time assing |   **Q. What give error when we use multiple same type late variable? Suppose we use multiple time Variable**  **Ans** : Syntax Error : 'a' has already been declared  **Q. What give error when we not assign const variable ?**  **Ans** : Syntax Error : Missing initializer in const declaration |

|  |
| --- |
| **Q. What is alert ()?**  **Ans** : alert() is a method that show some message or warning with the dialog box on browser. It has one button **ok** |

|  |
| --- |
| **Q. What is prompt ()?**  **Ans** : prompt() is a method that the access value of user with the dialog box on browser. It has two button **ok** & **cancel.**   * When we **click ok** then return “user value” * When we **click cance**l then return “null” |

|  |
| --- |
| **Q. What is confirm ()?**  **Ans** : confirm() is a method that show some message for confirm or not with the dialog box on browser. It has two button **ok** & **cancel**. It is return Boolean false   * When we **click ok** then return “true” * When we **click cance**l then return “false” |

|  |
| --- |
| **Q. How to get String length ?**  **Ans** : length  Ex = "Hello My Name is Rajnish".length // output 24 |

|  |
| --- |
| **Q. How to convert string in uppercase ?**  **Ans** :toUpperCase()  Ex = let text = "Hello, World"  console.log(text.toUpperCase()) //Output HELLO, WORLD |

|  |
| --- |
| **Q. How to remove unwanted spacing from start & end in any string ?**  **Ans** :trim()  Ex console.log(" Hello World ".trim()) //Output Hello World |
| **Q. How to replace value from String ?**  **Ans** :replace()  Ex console.log("Hello World".replace("H", "X")) //Output 'Xello World' |

|  |
| --- |
| **Q. How to add two string value ?**  **Ans** :concat()  Ex  let a = "Hello "  let b = "World"  console.log(a.concat(b)) |

|  |  |  |
| --- | --- | --- |
| **Q. How to find “char code” of any character ?**  **Ans** :concat()  Ex   |  |  | | --- | --- | | Console.log("B".charCodeAt()) //Output 65  Console.log("B".charCodeAt()) //Output 66 | console.log("THIS IS A MY BOOK".charCodeAt(8)) //output65  console.log("AB".charCodeAt(1)) //outpu t66 | |

|  |
| --- |
| **Q. How to use padStart() ?**  **Ans** :with the help of padStart() we want special character until we want  Ex  let str = "123";  let paddedStr = str.padStart(10, '\*');  console.log(paddedStr); // output 123\*\*\*\*\*\*\* |

|  |
| --- |
| **Q. Write falsie value list ?**  **Ans** : 0, -0, undefined, null , “ “ are falsie value |

|  |
| --- |
| **Q. What is comparison ?**  **Ans** : Comparison operator compares between two or more value with the help of  greater then, less then, equal to, not equal etc |

|  |
| --- |
| **Q. Write the List of Comparison Operator ?**  **Ans** : |

|  |
| --- |
| **Q. Different between == between === ?**  **Ans** : |

|  |
| --- |
| **Q. Write the List of Logical Operator ?**  **Ans** : |

|  |
| --- |
| **Q. How to set default value in prompt ?**  **Ans** : |

|  |
| --- |
| **Q. Write the logical operator in javaScript ?**  **Ans** :   * && = both statement must should be true * || = both statement must should be true * ! |

|  |
| --- |
| **Q. What is a ternary operator in JavaScript ?**  **Ans** : |

|  |
| --- |
| **Q. What is an Object?**  **Ans** : Object is a collection of key-value and parirs. Each key has of many type refer value like String, array, function |

|  |
| --- |
| **Q. What is an Object?**  **Ans** : Object is a collection of key-value and parirs. Each key has of many type refer value like String, array, function |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q. How do you add a new property to an object in JavaScript?**  **Ans** :   1. Dot nation = user.name = “Rajnish” 2. Bracket nation = user[“name”] = “Rajnish”  |  |  | | --- | --- | | Dot nation Example | Bracket nation | | Let user ={}  **user.name** = “Rajnish” | Let user ={}  user[“name”] = “Rajnish” | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q. How to access value of object in JavaScript?**  **Ans** :   1. Dot nation = user.name 2. Bracket nation = user[“name”]  |  |  | | --- | --- | | Dot nation Example | Bracket nation | | let user ={  fname : “Rajnish”,  [“lname”] : “Bharti”  }  console.log(**user.fname** ) | Let user ={  fname : “Rajnish”,  [“l-name”] : “Bharti”  }  console.log(**user.[”fname”]** )  console.log(**user.[”l-name”]** ) | |

|  |
| --- |
| **Q. How do you remove a property from an object in JavaScript?**  **Ans** : with **delete** key word  let user ={  fname : "Rajnish",  ["lname"] : "Bharti"  }  **delete user.fname**  console.log(user) |

|  |
| --- |
| **Q. How do you check if an object has a property in JavaScript ?**  **Ans** : with **Object. hasOwnProperty('**keyName**')**  let user ={  fname : "Rajnish",  ["lname"] : "Bharti"  }  delete user.fname  console.log(user.hasOwnProperty('keyName')) //false  console.log(user.hasOwnProperty('lname')) //true |

|  |
| --- |
| **Q. How do you check if an object has a property in JavaScript ?**  **Ans** : with **Object. hasOwnProperty('**keyName**')**  let user ={  fname : "Rajnish",  ["lname"] : "Bharti"  }  delete user.fname  console.log(user.hasOwnProperty('keyName')) //false  console.log(user.hasOwnProperty('lname')) //true |

|  |
| --- |
| **Q. What is an Array in JavaScript ?**  **Ans** : Array is reference type data. This is stores multiple value in a single variable. Array has key for each value. |

|  |
| --- |
| **Q. How to Access array of value ?**  **Ans** : with the help of indexing we can access array of value  **Ex =** arr[0] |

|  |
| --- |
| **Q. How to update array of value ?**  **Ans** : with the help of array indexing  **Ex =**  arr[0] = 'Shyam Narayan' |

|  |
| --- |
| **Q. what is usages pop() ?**  **Ans** : with the help of pop() method we can remove last element from current array  **Ex =**  const student =["Ram", "Shyam", "Suresh", "Gagan"]  student.pop()  console.log(student) |

|  |
| --- |
| **Q. what is usages push() ?**  **Ans** : with the help of push() method we can add last element in current array  **Ex =**  const student =["Ram", "Shyam", "Suresh", "Gagan"]  student.push("Mayan", "Narayan")  console.log(student) //Output = ['Ram', 'Shyam', 'Suresh', 'Gagan', 'Mayan', 'Narayan'] |

|  |
| --- |
| **Q. what is usages shift() ?**  **Ans** : with the help of shift() method we can remove first element in current array  **Ex =**  const student =["Ram", "Shyam", "Suresh", "Gagan"]  student.shift()  console.log(student) // out put = ['Shyam', 'Suresh', 'Gagan'] |

|  |
| --- |
| **Q. what is usages unShift() ?**  **Ans** : with the help of shift() method we can remove first element in current array  **Ex =**  const student =["Ram", "Shyam", "Suresh", "Gagan"]  student. unshift ("Kunal", "Manohar")  console.log(student) // out put =['Kunal', 'Manohar', 'Ram', 'Shyam', 'Suresh', 'Gagan'] |

|  |
| --- |
| **Q. what is usages concat() ?**  **Ans** : add two array  **Ex =**  let arr1 =["A", "B"]  let arr2 = ["C", "D"]  let arr3 = ["D", "E"]  let newArr = arr1.concat(arr2, arr3);  console.log(newArr) // out put = ['A', 'B', 'C', 'D', 'D', 'E'] |

|  |
| --- |
| **Q. what is slice() ?**  **Ans** : with the help create a new array from specific index to end  **Ex =**  let arr1 =["A", "B", "C", "D", "E", "F", "G"]  let newArr = arr1.slice(2, 5);  console.log(newArr) // out put = ['C', 'D', 'E'] |

|  |
| --- |
| **Q. what is indeOf() ?**  **Ans** : with the help find Index of array from searching value  **Ex =**  let arr1 =["Ram", "Suresh", "Mayank", "Gagan", "Kunal", "Nayan", "Papu"]  let newArr = arr1.indexOf( "Suresh");  console.log(newArr) // out put = 1 |

|  |
| --- |
| **Q. what is includes () ?**  **Ans** : includes() method return **true / false** after search value from index  **Ex =**  let arr1 =["Ram", "Suresh", "Mayank", "Gagan", "Kunal", "Nayan", "Papu"]  console.log(arr1.includes( "Suresh")) // out put = true  console.log(arr1.includes( "Hello")) // out put = false |

|  |
| --- |
| **Q. what is reverse () ?**  **Ans** : with the help of referse() method we can reverse reverence in original array  **Ex =**  let arr1 =["Ram", "Suresh", "Mayank", "Gagan", "Kunal", "Nayan", "Papu"]  console.log(arr1.reverse( )) // out put = ['Papu', 'Nayan', 'Kunal', 'Gagan', 'Mayank', 'Suresh', 'Ram'] |

|  |
| --- |
| **Q. what is shallow copy ?**  **Ans** : shallow coppy is a copy of top label of an object.  Object.assing(), sperade(…), slice(), concat method helps to copy of top label object or array  **Ex with source =**  let target =["Ram", "Suresh", "Mayank", "Gagan", "Kunal", "Nayan", "Papu"]  let source=[]  Object.assign(target, source)  console.log(target)  **Example with concat**  let target =["Ram", "Suresh", "Mayank", "Gagan", "Kunal", "Nayan", "Papu"]  let source=[].concat(target)  console.log(source)  **Example with slice**  let target =["Ram", "Suresh", "Mayank", "Gagan", "Kunal", "Nayan", "Papu"]  let source=target.slice (0, target.lastIndexOf())  console.log(source) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Q. Different between freez & sheal?**   |  |  | | --- | --- | | **freez** | seal | | With the help seal we can’t any type change in existing object | With the help freez we can only update in existing proper. Not add or update | | Ex  let user = {  fname : "Rajnish",  lname : "Kumar"  }  Object.freeze(user)  user.fname = "Lakshma"  console.log(user) //{fname: 'Rajnish', lname: 'Kumar'} | Ex  let user = {  fname : "Rajnish",  lname : "Kumar"  }  Object.seal(user)  user.fname = "Lakshma"  console.log(user) //{fname: 'Lakshma', lname: 'Kumar'} | |

|  |
| --- |
| **Q. what is Context () ?**  **Ans** : javaScript engine create a context. Where our code exist that is global context.  There are two type of context   1. Global context = var, let , const and object key value is global contaxt 2. Local context = function has local context   There are two type phase of execution   1. Memory creation phase = it is created a memory for all variable, object and function   ex : let a = undefined  let b = undefined   1. Code Execution phase= where our code exist in memory   ex : a = 20  b = 30 |
| **Note :**   1. **The global execution context is created when a JavaScript script first starts to run** 2. function memory creation phase create when memory create 3. when function return then local exesition phase has destroy |

|  |
| --- |
| **Q. what is CallStack () ?**  **Ans** : The call stack keeps track execution of then function **.** When function is call then it is added in stack. When the function returns any value then it’s remove from stack.  . It is working on **Last in and first out (LIFO) .**  function funcA(){  console.log('func A Execute')  funcB()  console.log('out = A')  }  function funcB(){  console.log('func B Execute')  funcC()  console.log('Output = B')  }  function funcC(){  console.log('func C Execute')  console.log('Output = C')  }  funcA() |
| **Note :**  **We see stack when we use debugger** |

|  |
| --- |
| **Q. what is Hoisting ?**  **Ans** : Hoisting is behavior of javaScript declaration. Hoisting Declaration is top label scope.  **Ex =**  **Hoisting with var**  console.log('a') //output =50  var a = 50;  console.log('b') //output ='Hello, World';  var b = 'Hello World'  **Hoisting with Regular function**  console.log('myfunc()') //output ='My Name is Rajnish Kumar';  function abc(){  console.log(' My Name is Rajnish Kumar')  } |
| **Note:**   1. **Hoisting in not working with let & const. Because he is temperal dead zone** |

|  |
| --- |
| **Q. ls let, const the hoisitng ?**  **Ans** : Behind the scene yes, but due to behavior of javaScript let, const, is temperal Dead zone. So let, const is not part of hoisting  **Ex**  console.log(a) //ReferenceError: a is not defined  let a = 50  console.log(b) //ReferenceError: b is not defined  const b = 'Hello World' |

|  |
| --- |
| **Q. What is a temporal Dead zone ?**  **Ans** : A variable is declared but not accessible that is colled temporal Dead Zone |

|  |
| --- |
| **Q. What is Scope ?**  **Ans** : A range of accessible of code is called Dead Zone |

|  |
| --- |
| **Q. What is a global Scope ?**  **Ans** : A range of access anywhere that is called global Scope  Ex  let a = 10; //This is global Scope  function abc (){  //"a" is use in block Scope  console.log('inner output ' +(a+2))  }  abc()  //"a" is use in global Scope  console.log('inner output ' +(a)) |

|  |
| --- |
| **Q. What is a local Scope ?**  **Ans** : A range of access only block of code  Ex  function cde() {  let b= 10; //This is local Scope  console.log(b)  }  cde()  // console.log(b) //Error : Uncaught ReferenceError: b is not defined |

|  |
| --- |
| **Q. What is a lexical Scope in JavaScript ?**  **Ans** : A block that the access same scope and outer scope that is called lexical of local scope  Ex  function a1(){  let a = 10;  console.log(a)  function a2(){  let b = 20;  console.log(a+b) // "a2()" access "a1()" variable  }  a2()  }  a1() |
| **Note:** Sibling Scope not access each other variable  **For Example**  function b2(){  console.log('output')  function b2\_1(){  let x = 20  }  function b2\_2(){  let y = 10  console.log(x\*y) // Error : x is not defined  }  b2\_1()  b2\_2()  }  b2() |

|  |
| --- |
| **Q. What is call back() function ?**  **Ans** : a function that is used as a function argument known as “call back” function  Ex =  function pannel(a, b, c){  c(a,b)  }  pannel("Good Morning", "Rajnish", **callback**)  function **callback**(x, y){  console.log(`Hello, ${x} ${y}`)  } |
| **Q. What is high order function ?**  **Ans** : High order function takes one or more function as a callback function and return a function that is called High order function. map(), filter(), reduce() is a high order function  Ex =  function outer(fun, num1, num2){  return fun(num1, num2)  }  outer(add, 50, 40) // outer function call with "callback()" argument  // add() function is a callback function  //This is a callback function  function add(num1, num2){  let sumValue = num1+num2  console.log(`You out out is ${sumValue}`)  } ` |

|  |
| --- |
| **Q. What is Event loop ?**  **Ans** : Event loop is allow to asynchronous programing language. It is monitor between **call Stack** & **call que**  **Q. How to work Event loop ?**  Ans : Better to know understand  CallStacK :When a function call then **“call Stack”** Create. When function return then callStack removed.  WebAPI : WebApi Provide asynchronous function like setTimeOut, fetch  Call back que: Callback que stores callback function to complete asynchronous opratin  EventLoop : Event lop moniters between callStack & CallQue. When **Call Stack** empty, then  “call Que” transfer call que to callStack.  **Example**  console.log('program Start')  setTimeout(()=>{  console.log('B')  }, 2000)    setTimeout(()=>{  console.log('C')  }, 5000)    console.log('program End') |
| **Q. Different Between method and function ?**  **Ans** : method uses with object. They are written as it is a function but they are define in object  **Example**  let method ={  sum: function (a, b){  return a+b  },  sub: function (a, b){  return a-b  },  mul: function (a, b){  return a\*b  }  }  console.log(method.sum(10, 50))  console.log(method.sub(90, 50))  console.log(method.mul(6, 5)) |

|  |
| --- |
| **Q. What is a closer ?**  **Ans** : closer is a function that returns a new function with own Scope  **Q. Is Every function return Scope (clouser) ?**  **Ans** : No, Every function is not return Scope. When need return a child function that time return scope(clouser)  **Q. Is Every function return Scope (clouser) ?**  **Ans** : No, Every function is not return Scope. When need return a child function that time return scope(clouser)  **Example**  debugger  function parent(){  let a= 10;  let b= 20;  let c = 'Garbage value'  return function child(){  return a+b  }  }  let sum = parent();  console.log(sum()) |

|  |
| --- |
| **Q. Different between for of and for in loop ?**  **Ans** : for of loop **iterate value of array**  let books = ["Hindi", "English", "Math", "Science"]  for (let item of books ) {  console.log(item)  }  console.log('----------------------')  but for in loop **iterate value of object**  Example  let sutdent ={  name: "Rajnish",  age :20,  class :6  }  for(let stdKey in sutdent){  console.log(stdKey) //pring name, age, class  console.log(sutdent.stdKey) // "Rajnish", 20, 6  }  console.log('power')  let a = 2;  console.log(a\*\*2) |

|  |
| --- |
| **Q. What is a forEach() ?**  **Ans** : forEach() method is high Order function. That iterates value of array. It is return undefined  **Example :**  let stundent = ["Ram", "Shyam", "Suresh", "Mahesh"];  stundent.forEach((x)=>{  console.log(x)  })  **Example : return undefined**  let returnIndex = stundent.forEach((x)=>{  console.log(x)  })  console.log(returnIndex) |

|  |
| --- |
| **Q. What is a map () ?**  **Ans** : map () method is high order function it is return copy of array or update version of array.  It is not updated in Original Array  **Example (Copy Array) :**  let num = [10, 20, 30, 50, 100, 120, 200, 250];  let CoppyArr =num.map((x)=>{  return x  })  console.log(CoppyArr)  **Example (Updated Array) :**  let num = [10, 20, 30, 50, 100, 120, 200, 250];  let powerArray =num.map((x)=>{  return x\*\*2  }) |

|  |
| --- |
| **Q. What is a filter() ?**  **Ans** : filter () method is high order function it is return new copy array after some conditional state . when not use conditional statement that time it returns copy of original array  It is not updated in Original Array  **Example (Copy Array) :**  let num = [10, 20, 30, 50, 100, 120, 200, 250];  let updatedCondition =num.filter((x)=>{  return x>18  })  console.log(updatedCondition)  **Example (Updated Array) :**  let num = [10, 20, 30, 50, 100, 120, 200, 250];  let powerArray =num.map((x)=>{  return x\*\*2  }) |

|  |
| --- |
| **Q. What is a filter() ?**  **Ans** : filter () method is high order function it is return new copy array after some conditional state . when not use conditional statement that time it returns copy of original array  It is not updated in Original Array  **Example (Copy Array) :**  let num = [10, 20, 30, 50, 100, 120, 200, 250];  let updatedCondition =num.filter((x)=>{  return x>18  }) |

Logical Question

|  |
| --- |
| Q 1. print “Hello” from let a= “Hello World” |
| let a = "Hello World"  let output = a.slice(0, 5)  console.log(output) |

|  |
| --- |
| Q 2. print “XXXXXXX12” from let b= 12999999999 |
| let b= 12999999999  let output = b.toString().slice(0, 2).padStart(10, "X")  console.log(output) |

|  |
| --- |
| Q 3. print “12XXXXXXX” from let b= 12999999999 |
| let b= 12999999999  let output = b.toString().slice(0, 2).padEnd(10, "X")  console.log(output) |

|  |
| --- |
| Q 4. Create an Array [Ram , Suresh, Mahesh]  from given from let a = "Ram Suresh Mahes is my Friend" |
| let a = "Ram Suresh Mahes is my Friend"  let output = a.split(" ", 3)  console.log(output) |

|  |
| --- |
| Q 5. Find Character |
| let a = "Ram Suresh Mahes is my Friend"  let output = a.split(" ", 3)  console.log(output) |

|  |
| --- |
| Q 6. Write a JavaScript program that prompts the user to input their name and age, and then categorizes the user based on their age. The program should:  **Categorize the user based on the following age ranges:**   1. 1 to 5 years: Kid 2. 6 to 17 years: School Student 3. 18 to 24 years: College Student 4. 25 to 60 years: Working Professional 5. 61 to 80 years: Retired Person 6. Over 80 years: Angel |
| - |

|  |
| --- |
| Q 7. Write the age program with the switch statement |
| let a = "Ram Suresh Mahes is my Friend"  let output = a.split(" ", 3)  console.log(output) |

|  |
| --- |
| Q 8. What is output when  let a={name: " Ram"}  let b={name: " Ram"}  console.log(a==b) |
| Ans : false |

|  |
| --- |
| Q 9. What is output when  let a={name: " Ram"}  let b={name: " Ram"}  console.log(a.name==b.name) |
| Ans : false |

|  |
| --- |
| Q 10. Remove "Mayank", "Suresh from Geffen array  let employ = ["Ram", "Mayank", "Suresh", "Gagan", "Kamla"];  employ.splice(1, 2)  console.log(employ) |
|  |

|  |
| --- |
| Q 11. Reverse array value from given array  let a=[10, 1, 2, 20, 3, 30]  a.reverse()  console.log(a) |
|  |

|  |
| --- |
| Q 12. create one new array when number is greater than 10 in given array using while loop  let a=[10, 1, 2, 20, 3, 30]  let source = [1, 2, 12, 3, 33, 4, 44, 5, 55, 6, 66, 7, 77, 8, 88, 9, 99, 10, 100];  let i=0;  let target = []  while(i<source.length){  target.push(source[i])  i++  }  console.log(target) |
|  |

|  |
| --- |
| Q 13. Find the unique number from given array using while loop ?  let arr = ["Ram", "Shyam", "Suresh", "Ram"]  let i = 0;  let newArr = []  while (i < arr.length) {  if (newArr.indexOf(arr[i]) == -1) {  console.log()  newArr.push(arr[i])  }  i++  }  console.log(newArr) |
|  |

|  |
| --- |
| Q 14. Find the maximum number from given array ?  let arr1 = [10, 20, 50, 2, 3, 9, 90, 6, 60, 100]  let maxVal = arr1[0]  let i =0;  while(i<arr1.length){  if(maxVal<arr1[i]){  maxVal = arr1[i]  }  i++  }  console.log(maxVal) |
|  |

|  |
| --- |
| Q 15. Find even number from 1 to 100 ?  for(let i= 0; i<=100; i++){  if(i%2===0){  console.log(`Even Number is ${i}`)  }  } |
|  |
| Q 16. Find the unique number from given array using for loop ?  let taks2 = ["Ram", "Shyam", "Suresh", "Ram", "Shyam", "Suresh"];  let uniqueArry = ['Ram', 'Shyam', 'Suresh']  for(let i = 0; i<taks2.length; i++){  uniqueArry.indexOf(taks2[i]) === -1 ?uniqueArry.push(taks2[i]) :null    }  console.log(uniqueArry) |
|  |