Vallidation in javascript

Validate data in javascript

validatedropdown	In dropdown box it stores property by name selectedIndex, it stores the index position of the selected option
	
	It also stores a array by name options, hence to find the value of selected dropdown box in javascript
	
	//get reference of dropdown box centers
	var c=document.getElementById(id);
	//retrieve the value of index of the selected option
	var pos=c.selectedIndex;
	//alert(c.selectedIndex)
	//retrieve the value of selected option
	var val=c.options[pos].value;

validateradio	to find which radio button is checked we use checked property, In radio button we can select only one option, so as soon as we get the checked radio button we may break the loop
	const validateradio=(nm,errid)=>{
	var arr=document.getElementsByName("gender");
	for(var i=0;i<arr.length;i++){
	if(arr[i].checked){
	document.getElementById(errid).innerHTML="";
	return true;
	}
	}
	document.getElementById(errid).innerHTML="pls select one choice"
	return false;
	}

validatecheckbox	to find which check is checked, we use checked property, checkboxes are used to select multiple options, we use the checked property to find which checkboxes are checked
	const validatecheckbox=(nm,errid)=>{
	var subarr=document.getElementsByName(nm);
	var cnt=0;
	var str="";
	for(var i=0;i<subarr.length;i++){
	if(subarr[i].checked){
	cnt++;
	str=str+subarr[i].value+",";
	}
	/*if(cnt>=2){
	break;
	}*/
	}
	if(cnt>=2){
	//clear the message
	document.getElementById(errid).innerHTML="";

	alert(str);
	return true;
	}
	//display error message
	document.getElementById(errid).innerHTML="pls select minimum 2 subjects"
	return false;
	}

validatefile	to validate file size, we may use size property
	
	function validateSize(){
	alert("in validatesize")
	var f=document.getElementById("files");
	//f.files[0].size it will give size in bytes
	alert(f.files[0].size/1024+"KB");
	if(f.files[0].size/1024 < 10){
	alert("in if")
	document.getElementById("errfile").innerHTML="";
	//return true;
	}
	document.getElementById("errfile").innerHTML="file size exceeds the limit 10KB"
	//return false;
	}

Types in javascript

1. Number
2. String
3. Date
4. Object
5. Null
6. Undefined

To find the datatype of the variable

typeof(n)

functions on numeric data

isNaN(data)	if the data contains atleast one alphabet, then it returns true, otherwise it returns false
isFinite(data)	if the data contains all digits between 0-9, then it returns true, otherwise it returns false
toFixed(n)	To display only n digits after decimal point Var num=123.456 Var n=num.toFixed(2)

	Console.log(n);
Math.round(number)	It will round the number upto 2 decimal places

Functions on Strings

s.trim()	to remove leading and trailing spaces	s=" aaaaa " s.trim() aaaaa
s.length	to find length of the string	s.length
s.charAt(i)	to find the character at specific index position	s="testing" s.charAt(4)=i
s.concat(str1,str2,.....)	to concat multiple strings	s="taste" s.concat("is", "good") tasteisgood

s.indexOf(str)	to find the position of first occurrence of the given str, indexing starts with 0	s="Hello World!!"
		s.indeXOf("Hello") //0
		s.indexOf("World) //6

s.split(delimiter)	to break the string into array of strings at delimiter position	s="rain in spain, is plain" s.split(" ") ["rain", "in","spain,","is","plain"] s.split(",") ["rain in spain"," is plain"]
s.toUpperCase()	convert the string into uppercase	s="abcd" s.toUpperCase() ABCD
s.toLowerCase()	convert the string into lowercase	s="ABcd" s.toLowerCase() abcd

s.slice(startIndex[,endIndex])	it retrieves the portion of the string from startIndex to endIndex, endIndex is excluded	s="Hello world!!"
		s.slice(0,1) //H
		
		s.slice(4) //o world!!

s.includes(substr)	it returns true, if substring exists, false otherwise	s="Hello world!!"
		s.includes("ell") //true
		
		s.includes("xxx") //false

using regular expression

s.match(regular expression)	it checks the specified pattern in the string, and return all occurences	s=”rain in SPAIN is plain”
		pattern=/a.*?n/
		s.match(pattern) //ain
		
		pattern=/a.*?n/gi -→ values enclosed in / / is treated as regular expression
		i is flag for ignore index
		g is a flag for global
		s.match(pattern)
		//ain,AIN,ain

s.replace(regexpression, newstring)	It will search the given pattern and replaces it with the newstring	s=”To be or not to be” s.replace(/be/,”exists”) –>replace first occurence s.replace(/be/g,”exists”)-> flag g, will replace all occurences

s.search(regexpression)	it checks the specified pattern in the string, and return the position of first occurrence	var test=”testing: 1, 2, 3”
		var pattern= /\d+/g
		s.search(pattern) //9
		s.match(pattern) // [“1”,”2”,”3”]
		s.replace(pattern,”#”) // testing: #, #,#

RegExp.test(str)	It will return true if the RegExp exists in the string, otherwise false	var str=”testing: 1, 2, 3” var pattern= /\d+/g pattern.test(str) true

Date functions

Var dt=new Date();

new Date()	Give you today’s date	Var dt=new Date() Var dt1=new Date(2024,10,21)
Dt.getMonth()	To get month portion of the date, moth numbering starts with 0	Var dt=new Date() Var m=dt.getMonth(); 10 --→ november
Dt.getDate()	To get date portion of the date	Var dt=new Date() Var m=dt.getDate(); 21--→ november
Dt.getFullYear()	To get 4 digit year	Var dt=new Date() Var m=dt.getFullYear();

		2025
Dt.getDay()	To get day number in the week	Counting starts from Sunday Sunday-→0
Dt.getTime()	To convert date in milliseconds starting from 1 st jan 1970	

Array functions

Var arr=new Array(12,12,"xxxx",45)

Var arr1=[1,2,3,"dddd",34]

Var arr2=[12,1,3,arr1] //length 4

Var arr3=[12,1,3,...arr1] //length 8 spread operator

arr.push(val)	to add the value at the end of the array	Var=arr1=[10,20,30] arr1.push(300) [10,20,30,300]
arr.pop()	to delete the value from the end	Var=arr1=[10,20,30] arr1.pop() [10,20]
arr.unshift(val)	to add the value at the beginning	Var=arr1=[10,20,30] arr1.unshift(300) [300,10,20,30]
arr.shift()	to delete the value from the beginning	Var=arr1=[10,20,30] arr1.shift() [20,30]
arr.splice(position, number of elements to delete)	delete the number of elements starting from the given position	arr=[1,2,3,4,5,6] arr.splice(3,2) //4 and 5 will be deleted [1,2,3,6]
arr.splice(position, number,list of values)	delete the number of elements starting from the given position, and replace it	arr=[1,2,3,4,5,6] arr.splice(3,2,100,200,300,400,500) //4 and 5 will be replaced by the list of values [1,2,3, 100,200,300,400,500,6]

	with the list of values	

arr.splice(position, 0,list of values)	to add the values at the given position	arr=[1,2,3,4,5,6]
		
		arr.splice(3,0,100,200,300,400,500) //all the values will be added at the position
		[1,2,3, 100,200,300,400,500,4,5,6]

arr.indexOf(value)	find the position of the first occurrence of the given value This function is useful when the value is known	arr=[12,13,12,15,13] arr.indexOf(13) 1 arr.indexOf(100) -1

arr.findIndex(predicate function)	predicate function - → accepts one parameter and returns true / false
	
	findIndex function will find the position of the value for which predicate function returns true
		arr=[12,13,12,15,13,20]
		
		//to find index of first value which is divisible by 5
		
		//findindex will give the index of 15,
		
		arr.findIndex((val,index,arr)=>val%5==0)
		//use return keyword inside {}
		arr.findIndex((val,index,arr)=>{return val%5==0})
		
		arr.findIndex(val=>val%5==0)

arr.find (predicate function)	predicate function - → accepts one parameter and returns true / false
	
	find function
		arr=[12,13,12,15,13,20]
		
		//to find the first value which is divisible by 5
		
		//find will give the value 15,
		
		arr.find ((val,index,arr)=>val%5==0)
		//use return keyword inside {}
		arr.find ((val,index,arr)=>{return val%5==0})

	will find the value in the array for which predicate function returns true	arr.find (val=>val%5==0)

arr.filter(predicate function)	predicate function - → accepts one parameter and returns true / false
	
	filter function will find the all the values for which predicate function returns true
		arr=[12,13,12,15,13,20]
		
		//to find all values which is divisible by 5
		
		//filter will give the array of all values which are divisible by 5
		
		arr.filter((val,index,arr)=>val%5==0)
		//use return keyword inside {}
		arr.filter((val,index,arr)=>{return val%5==0})
		
		arr.filter(val=>val%5==0)

arr.map(coverter function)	map function will apply the given expression on every value in the array and return a new value filter function will find the all the values for which predicate function returns true	arr=[12,13,12,15,13,20]
		
		//to find squares of all the numbers
		
		arr.map((val,index,arr)=>val*val)
		//use return keyword inside {}
		arr.map((val,index,arr)=>{return val*val})
		
		arr.filter(val=>val*val)

Arr.reduce((acc,num)=>acc+num)	To convert many values into	var addition=arr1.reduce((acc,num)=>acc+num)

	one value we use reduce function	console.log("Addition : "+addition)

Objects

Var ob={num:12,name:”xxxx”}