Arrow function=> in arrow function does not need to curly braces and return if one line of code

1)how many ways defined object?

1. Object Literals.
2. New operator or constructor.
3. Object.create method.
4. Class.

https://dzone.com/articles/easy-javascript-part-13-four-ways-to-create-object#:~:text=There%20are%20four%20ways%20to,as%20using%20a%20function%20constructor).

2)how many ways to define array

<https://medium.com/javascript-in-plain-english/six-ways-to-create-an-array-in-javascript-ac4aa115b926>

Using assignment operator

# Using new operator

# Using Array.from

# Usign Spread operator

# Using Array

# Using Array.of

3) hoisting?

Variable and function both are moved to the top of the scope (declare part)

Example:

Sum(5,10)

Function sum(5,10){

Add= a+b;

Console.log(Add)

}

Example:

Var a;

a=10

console.log(a)

6) find prime no?

https://www.programiz.com/javascript/examples/prime-number

const number = parseInt(prompt("Enter a positive number: "));

let isPrime = true;

// check if number is equal to 1

if (number === 1) {

console.log("1 is neither prime nor composite number.");

}

// check if number is greater than 1

else if (number > 1) {

// looping through 2 to number-1

for (let i = 2; i < number; i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

console.log(`${number} is a prime number`);

} else {

console.log(`${number} is a not prime number`);

}

}

// check if number is less than 1

else {

console.log("The number is not a prime number.");

}

7) find smaller no?

Math.min(…[1,2,3])

Math.max(…[1,2,3])

8) what is clousure and why?

<https://medium.com/@deepakmankotiacse/what-is-closures-in-javascript-7acc626281ab>

A closure is the combination of a function and the lexical environment within which that function was declared. i.e, It is an inner function that has access to the outer or enclosing function’s variables. The closure has three scope chains

1. Own scope where variables defined between its curly brackets
2. Outer function’s variables
3. Global variables Let's take an example of closure concept,

function Welcome(name){

var greetingInfo = function(message){

console.log(message+' '+name);

}

return greetingInfo;

}

var myFunction = Welcome('John');

myFunction('Welcome '); //Output: Welcome John

myFunction('Hello Mr.'); //output: Hello Mr.John

As per the above code, the inner function(greetingInfo) has access to the variables in the outer function scope(Welcome) even after the outer function has returned.

Closures are very helpful to hide the implementation details in javascript. Closures can be useful to create private variables and functions.

9) difference b/w map and forEach?

* map() allocates memory and stores return values. forEach() throws away return values and always returns undefined.
* forEach() will allow a callback function to mutate the current array. map() will instead return a new array.
* Map is the chainable method.it transformed the element
* Foreach not chainableot transformed the element. And n

map() is faster than forEach()

10)what is event bubling?

Event bubbling is a type of event propagation where the event first triggers on the innermost target element, and then successively triggers on the ancestors (parents) of the target is element in the same nesting hierarchy till it reaches the outermost DOM element.

11)What is preventDefault?

It stop the default behaviour of browser

12) What is stop Propogation?

It stop the event bubbling

13. high order function?

Higher-order function is a function that accepts another function as an argument or returns a function as a return value.

14) diifernce b/w let and var?

Var=

It is been available from the beginning of JavaScript

Variables will be hoisted

It has function scope

Let =

Introduced as part of ES6

Hoisted but not initialized

It has block scope

15)Differnce b/w some and every?

Both are return true or false

Some check if one condition is true so come is true

Every check all condition are true then it is true

16) which life cycle in react is deprecated?

17) what is return type of filter and find?

Filter->array

Find -> value return

18)Reducer?

Creating the reducer for dispatch only not select

Dispatch -> post , put, delete (useDispatch)

Select-> get (useSelector)

19) Action?

Any action we performed , add, update, delete user

20)Differnce b/w substring,slice, splice?

21) Convert Object To String?

JSON.Stringify(Object)

22) check string/array is there or not?

Includes()

indexOf()

23) all instance of character built in methods?(in string particular character is in uppercase)

24) remove particular item from array?

|  |
| --- |
| const value = 3 |
| ing | let arr = [1, 2, 3, 4, 5, 3] |
|  | arr = arr.filter(item => item !== value) |
|  | console.log(arr) // [ 1, 2, 4, 5 ] |

25) remove element by using the key?

|  |
| --- |
| const arr = [1, 2, 3, 4, 5, 6]; |
|  | delete arr[4]; // delete element with index 4 |
|  | console.log( arr ); // [1, 2, 3, 4, undefined, 6] |
|  |  |

26) check duplicate value in array?

|  |
| --- |
| a=[1,2,3,1,2,3]; |
|  | b=[]; |
|  | a.filter(data=>{ |
|  | if(b.includes(data)==false){ |
|  | b.push(data) |
|  | } |
|  |  |
|  | }) |
|  | console.log(b); |

Using includes, indexOf(), set()

27) sorting alphabetically?

var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.sort();        // First sort the elements of fruits  
fruits.reverse();

28) sorting numerically?

var points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return a - b});

28) generate random no?

Math.floor(Math.random() \* 10);     // returns a random integer from 0 to 9

Math.floor(Math.random() \* 11);      // returns a random integer from 0 to 10

This JavaScript function always returns a random number between min (included) and max (excluded)

function getRndInteger(min, max) {  
  return Math.floor(Math.random() \* (max - min) ) + min;  
}

This JavaScript function always returns a random number between min and max (both included)

function getRndInteger(min, max) {  
  return Math.floor(Math.random() \* (max - min + 1) ) + min;  
}

29) // program to extract value as an array from an array of objects

function extractValue(arr, prop) {

// extract value from property

let extractedValue = arr.map(item => item[prop]);

return extractedValue;

}

const objArray = [{a: 1, b: 2}, {a: 4, b: 5}, {a: 8, b: 9}];

// passing an array of objects and property 'a' to extract

const result = extractValue(objArray, 'a');

console.log(result);

[1, 4, 8]

## 30) Get Random Item From an Array

// program to get a random item from an array

function getRandomItem(arr) {

// get random index value

const randomIndex = Math.floor(Math.random() \* arr.length);

// get random item

const item = arr[randomIndex];

return item;

}

const array = [1, 'hello', 5, 8];

const result = getRandomItem(array);

console.log(result);

'hello'

## 31 Factors of Positive Number

for(let i = 1; i <= 12; i++) {

// check if number is a factor

if(12 % i == 0) {

console.log(i);

}

}

## 32 Fibonacci Series Up to n Terms

// program to generate fibonacci series up to n terms

// take input from the user

const number = parseInt(prompt('Enter the number of terms: '));

let n1 = 0, n2 = 1, nextTerm;

console.log('Fibonacci Series:');

for (let i = 1; i <= number; i++) {

console.log(n1);

nextTerm = n1 + n2;

n1 = n2;

n2 = nextTerm;

}  
  
0, 1, 1, 2, 3, 5, 8, 13, 21,

## 33 Find Factorial

// program to find the factorial of a number

// take input from the user

const number = parseInt(prompt('Enter a positive integer: '));

// checking if number is negative

if (number < 0) {

console.log('Error! Factorial for negative number does not exist.');

}

// if number is 0

else if (number === 0) {

console.log(`The factorial of ${number} is 1.`);

}

// if number is positive

else {

let fact = 1;

for (i = 1; i <= number; i++) {

fact \*= i;

}

console.log(`The factorial of ${number} is ${fact}.`);

}

**Output**

Enter a positive integer: 5

The factorial of 5 is 120.

## 34 Check Armstrong Number of Three Digits

// program to check an Armstrong number of three digits

let sum = 0;

const number = prompt('Enter a three-digit positive integer: ');

// create a temporary variable

let temp = number;

while (temp > 0) {

// finding the one's digit

let remainder = temp % 10;

sum += remainder \* remainder \* remainder;

// removing last digit from the number

temp = parseInt(temp / 10); // convert float into integer

}

// check the condition

if (sum == number) {

console.log(`${number} is an Armstrong number`);

}

else {

console.log(`${number} is not an Armstrong number.`);

}

**Output**

Enter a three-digit positive integer: 153

153 is an Armstrong number.

## 35. Swaping

//JavaScript program to swap two variables

//take input from the users

let a = parseInt(prompt('Enter the first variable: '));

let b = parseInt(prompt('Enter the second variable: '));

// addition and subtraction operator

a = a + b;

b = a - b;

a = a - b;

console.log(`The value of a after swapping: ${a}`);

console.log(`The value of b after swapping: ${b}`);

**Output**

Enter the first variable: 4

Enter the second variable: 2

The value of a after swapping: 2

The value of b after swapping: 4

## 36.Check Palindrome using built-in Functions

// program to check if the string is palindrome or not

function checkPalindrome(str) {

// convert string to an array

const arrayValues = string.split('');

// reverse the array values

const reverseArrayValues = arrayValues.reverse();

// convert array to string

const reverseString = reverseArrayValues.join('');

if(string == reverseString) {

console.log('It is a palindrome');

}

else {

console.log('It is not a palindrome');

}

}

//take input

const string = prompt('Enter a string: ');

checkPalindrome(string);

**Output**

Enter a string: hello

It is not a palindrome

## 37. Sort Words in Alphabetical Order

// program to sort words in alphabetical order

// take input

const string = prompt('Enter a sentence: ');

// converting to an array

const words = string.split(' ');

// sort the array elements

words.sort();

// display the sorted words

console.log('The sorted words are:');

for (const element of words) {

console.log(element);

}

**Output**

Enter a sentence: I am learning JavaScript

The sorted words are:

I

JavaScript

am

learning

## 38. Reverse a String Using built-in Methods

// program to reverse a string

function reverseString(str) {

// return a new array of strings

const arrayStrings = str.split("");

// reverse the new created array elements

const reverseArray = arrayStrings.reverse();

// join all elements of the array into a string

const joinArray = reverseArray.join("");

// return the reversed string

return joinArray;

}

// take input from the user

const string = prompt('Enter a string: ');

const result = reverseString(string);

console.log(result);

**Output**

Enter a string: hello

olleh

## 39. Count the Number of Vowels Using Regex

// program to count the number of vowels in a string

function countVowel(str) {

// find the count of vowels

const count = str.match(/[aeiou]/gi).length;

// return number of vowels

return count;

}

// take input

const string = prompt('Enter a string: ');

const result = countVowel(string);

console.log(result);

**Output**

Enter a string: JavaScript program

5

## 40 Convert First letter to UpperCase

// program to convert first letter of a string to uppercase

function capitalizeFirstLetter(str) {

// converting first letter to uppercase

const capitalized = str.charAt(0).toUpperCase() + str.slice(1);

return capitalized;

}

// take input

const string = prompt('Enter a string: ');

const result = capitalizeFirstLetter(string);

console.log(result);

**Output**

Enter a string: javaScript

JavaScript

## 41 Check Occurrence of a Character Using for Loop

// program to check the number of occurrence of a character

function countString(str, letter) {

let count = 0;

// looping through the items

for (let i = 0; i < str.length; i++) {

// check if the character is at that position

if (str.charAt(i) == letter) {

count += 1;

}

}

return count;

}

// take input from the user

const string = prompt('Enter a string: ');

const letterToCheck = prompt('Enter a letter to check: ');

//passing parameters and calling the function

const result = countString(string, letterToCheck);

// displaying the result

console.log(result);

**Output**

Enter a string: school

Enter a letter to check: o

2

42. clone of Object?

## Clone the Object Using Object.assign()

Newperson = Object.assign({},person)

// declaring object

const person = {

name: 'John',

age: 21,

}

// cloning the object

const clonePerson = Object.assign({}, person);

console.log(clonePerson);

## 43 Clone the Object Using Spread Syntax

// program to clone the object

// declaring object

const person = {

name: 'John',

age: 21,

}

// cloning the object

const clonePerson = { ... person}

console.log(clonePerson);

## 44Clone the Object Using JSON.parse()

// program to clone the object

// declaring object

const person = {

name: 'John',

age: 21,

}

// cloning the object

const clonePerson = JSON.parse(JSON.stringify(person));

console.log(clonePerson);

45. reduce method?

46. str=”abc”;

47. program

var arr = [10, 30, 90, 70, 80, 20, 50, 40, 60];

var arr1=[];

for(let i=0; i<=arr.length; i++){

  for(let j=1; j<arr.length; j++){

        if(arr[i]+arr[j]===100 ){

          arr1.push(arr[i])

          console.log(arr[i],arr[j])

         }

  }

}

48.We want up 10 character long string.

str="abc";

n=10;

console.log(str.repeat(4).slice(0,10))

49. dataytpe of null and undefined

50. what is prototype in javascript?

51)arr=[“one”,”two”,”three”,”four”,”five”];

Index=2

Result= [“three”,”four”,”five”,”One”,”Two”];

52) unique no in array?

### 53) What is the difference between Call, Apply and Bind?

54) What is first order function?

55) What is high order function?

56) what is unary function?

57) what is curring function?

58.Deep copying?

59.Shallow copying?

**Some important function:-**

charCodeAt—character to unicode

fromCharCode—Unicode to characters

getTime()--- convert time to number

isInteger() --- check the integer

isNaN()----check not a number

window.location.href()--- get the current url

document.URL---- get the current url

charAt-----return single character

Math.floor(Math.random()\*100)----- Generate random no

indexOf-----

lastIndexOf()------ return index

lastIndexOf(20,planet)------second planet from str. Started search position 20

startWith------

slice(1,4)-------ell

splice(whatposition ,ho many removed, new item)------

ex-Hello

substring(1,4)-----ell

substring(4,1)---ell

substring(-3)----Hello

substring(2)----llo

substr(1,4)----ello

btoa()----convert string to base64

atob()-----convert base64 to string

toDateString()----convert to date Object

toLocaleTimeString()----convert to time object

getDate()

getMonth()+1

getFullYear()

getTime()------ it return the millisecond

hasOwnProperty()---- check this key is exist or not

Intl.NumberFormat object enables language-sensitive number formatting.

.toFixed() round up the value of 2 decimal places

Ascii value:

A-Z=65-90

a-z=97-122

0-9=48-57

Newperson = Object.assign({},person)------clone of object