**Sessions Roadmap – Day 01 Task**

1. **Write a blog on Difference between HTTP/1.1 vs HTTP/2.**

HTTP stands for **Hyper Text Transfer Protocol** and it is used in client-server communication. By using HTTP user sends the request to the server & the server sends the response to the user. There are several stages of development of HTTP but we will focus mainly on HTTP/1.1 which was created in 1997 & the new one is HTTP/2 which was created in 2015.

**Difference between HTTP/1.1 vs HTTP/2:**

**HTTP/1.1 :**

* It Works on the textual format. i.e., Human-readable characters in ASCII encoding.
* There is head of line that blocks all the requests behind it until it doesn’t get its all resources.
* It compresses data by itself.
* Time taken by HTTP/1.1 to load the data is comparatively higher than HTTP/2.

**HTTP/2 :**

* It works on the binary protocol. i.e., Binary protocols are intended to be read by a machine rather than a human being.
* It allows multiplexing so one TCP connection is required for multiple requests.
* It uses HPACK for data compression. HPACK compression reduces the size of your headers by over 30% on average. There is less data now being sent which in turn will increase your content delivery speeds.
* Time taken by HTTP/2 to load the data is half of time taken by HTTP/1.1.

1. **Write a blog about objects and its internal representation in JavaScript.**

Objects, in JavaScript, is most important data-type and forms the building blocks for modern JavaScript. These objects are quite different from JavaScript’s primitive data-types (Number, String, Boolean, null, undefined and symbol)  in the sense that while these primitive data-types all store a single value each (depending on their types).

Objects in JavaScript may be defined as an unordered collection of related data, of primitive or reference types, in the form of “key: value” pairs. These keys can be variables or functions and are called properties and methods, respectively, in the context of an object.

For example, If your object is a student, it will have properties like name, age, address, id, etc.

**Objects and properties**

* A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object.
* The properties of an object define the characteristics of the object. You access the properties of an object with a simple dot-notation.

|  |
| --- |
| objectName.propertyName |

* Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. You can define a property by assigning it a value.

For example, let’s create an object named “myCar” and give it properties named “make, model, and year” as follows:

|  |
| --- |
| var myCar = new Object(); myCar.make = 'Ford'; myCar.model = 'Mustang'; myCar.year = 1969; |

**Creating Objects In JavaScript :**

One of easiest way to create a JavaScript object is object literal, simply define the property and values inside curly braces as shown below

|  |
| --- |
| let bike = {name: 'SuperSport', maker:'Ducati', engine:'937cc'}; |

**Using the JavaScript Keyword new :**

The following example also creates a new JavaScript object with four properties:

Example :

|  |
| --- |
| var person = new Object(); person.firstName = “John”; person.lastName = “Doe”; person.age = 50; person.eyeColor = “blue”; |

1. **CODEKATA Practice :**

**Category - Mathematics :**

**Question 01 :**

You are given a task to tell whether the number is pure or not. A pure number is a number whose sum of digits is multiple of 3.

O(1) time and O(1) space

**Input Description:**  
You are given a number n.

**Output Description:**  
Print yes if it is pure else not

**Sample Input :**  
13

**Sample Output :**  
not

**Logical Coding :**

inp.on("close", () => {

let number = userInput[0].split('');

let a = parseInt(number[0]);

let b = parseInt(number[1]);

let sum = a+b;

if((sum%3) === 0){

console.log('yes');

} else {

console.log('not');

}

});

**Category - Mathematics :**

**Question 04 :**

Given 3 numbers a,b,c print a\*b mod c.  
Sample Testcase :  
**INPUT**  
5 3 2  
**OUTPUT**  
1

**Logical Coding :**

inp.on("close", () => {

let number = userInput[0].split(' ');

let a = parseInt(number[0]);

let b = parseInt(number[1]);

let c = parseInt(number[2]);

let result = ((a%c)\*(b%c))%c;

console.log(result);

});

**Category - Mathematics :**

**Question 06 :**

Given a number N, print yes if the number is a multiple of 7 else print no.  
Sample Testcase :  
**INPUT**  
49  
**OUTPUT**  
yes

**Logical Coding :**

inp.on("close", () => {

let n = parseInt(userInput[0]);

if((n%7 === 0)){

console.log('yes');

} else {

console.log('no');

}

});

**Category - Mathematics :**

**Question 07 :**

You are given a large number made of only 0’s and 1’s.Your task is to find the max no of consecutive 1’s. If there are no 1’s print -1

**Input Description:**  
A large number ‘n’

**Output Description:**  
Print the max no of consecutive 1 in the number

**Sample Input :**  
101011111

**Sample Output :**  
5

**Logical Coding :**

inp.on("close", () => {

let number = userInput[0].split('');

let consecutive = 0;

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

let n = parseInt(number[i]);

if(i<number.length-1){

if(n==1){

consecutive++ ;

} else {

consecutive = 0;

continue;

}

}

if(i==number.length-1){

if(n==1){

consecutive++ ;

} else {

consecutive = -1;

break;

}

}

}

console.log(consecutive);

});

**Category - Mathematics :**

**Question 11 :**

Pk finds it difficult to judge the minimum element in the list of elements given to him. Your task is to develop the algorithm in order to find the minimum element.

Note:Don’t use sorting

**Input Description:**  
You are given ‘n’ number of elements. Next line contains n space separated numbers.

**Output Description:**  
Print the minimum element

**Sample Input :**  
5  
3 4 9 1 6

**Sample Output :**  
1

**Logical Coding :**

inp.on("close", () => {

let n = parseInt(userInput[0]);

let elementsOfN = userInput[1].split(' ')

let min = 0;

for(let i=0; i<n; i++){

if(i===0){

min = parseInt(elementsOfN[i]);

} else {

if(parseInt(elementsOfN[i])<min){

min = parseInt(elementsOfN[i]);

}

}

}

console.log(min);

});