**Name: Alaa Gamal**

**ID:20184859**

**Front end training**

**TASK1**

**Question3:**

**1-**string

**2-**NaN property representing :Not-A-Number

**- there are different types of operations that return NaN:**

1-Number cannot be parsed e.g. parseInt("jgjthbg")

2-Math operation (e.g. Math.sqrt(-1))

3-Any operation that involves a string and is not an addition operation (e.g. "ok" / 3)

**How can you reliably test if a value is equal to NaN?** By using  value !== value, which would only produce true if the value is equal to NaN. Also, Number.isNaN() function.

**3-**JavaScript code runs in one line at a time manner and there is no possibility of running code in parallel.

**4-**polymorphism is one of the behaviours of Object Oriented Programming (OOP). It is the practice of designing objects to share behaviors and to be able to override shared behaviors

**Question4:**

**1-**function sumobjectvalues ( obj ) {

    var sum = 0;

    for( var el in obj ) {

      if( obj.hasOwnProperty( el ) ) {

        sum += parseFloat( obj[el] );

      }

    }

    return sum;

  }

  var sample = { a: 1 , b: 2 , c:3 };

  var summed = sumobjectvalues( sample );

  console.log( "sum: "+summed );

**3-**  
  function print24(str)

{

    // Get hours

    var h1 = Number(str[1] - '0');

    var h2 = Number(str[0] - '0');

    var hh = (h2 \* 10 + h1 % 10);

    // If time is in "AM"

    if (str[8] == 'A')

    {

        if (hh == 12)

        {

            document.write("00");

            for (var i = 2; i <= 7; i++)

                document.write(str[i]);

        }

        else

        {

            for (var i = 0; i <= 7; i++)

                document.write(str[i]);

        }

    }

    // If time is in "PM"

    else

    {

        if (hh == 12)

        {

            document.write("12");

            for (var i = 2; i <= 7; i++)

                document.write(str[i]);

        }

        else

        {

            hh = hh + 12;

            document.write(hh);

            for (var i = 2; i <= 7; i++)

                document.write(str[i]);

        }

    }

}

    var str = "07:05:45PM";

    print24(str);

4-

class Date

{

    constructor(d,m,y)

    {

        this.d = d;

            this.m = m;

            this.y = y;

    }

}

// To store number of days in   all months from January to Dec

let monthDays=[31, 28, 31, 30, 31, 30,

                            31, 31, 30, 31, 30, 31];

//counting leap years

function countLeapYears(d)

{

    let years = d.y;

        // Check if the current year needs to be considered

        if (d.m <= 2)

        {

            years--;

        }

        return Math.floor(years / 4) - Math.floor(years / 100) +

        Math.floor(years / 400);

}

function getDifference(dt1,dt2)

{

        let n1 = dt1.y \* 365 + dt1.d;

        // Add days for months in given date

        for (let i = 0; i < dt1.m - 1; i++)

        {

            n1 += monthDays[i];

        }

        // Since every leap year is of 366 days,

        // Add a day for every leap year

        n1 += countLeapYears(dt1);

        let n2 = dt2.y \* 365 + dt2.d;

        for (let i = 0; i < dt2.m - 1; i++)

        {

            n2 += monthDays[i];

        }

        n2 += countLeapYears(dt2);

        return (n2 - n1);

}

let dt1 = new Date(1, 2, 2000);

let dt2 = new Date(1, 2, 2004);

document.write("Difference between two dates is " +

                            getDifference(dt1, dt2));

5-let num=[3,4,5,9,2];

const maxvalue=math.max(...num);

console.log(num.indexOf(maxvalue));

console.log(maxvalue);

**Question5:**

**What is the output?**

**3**

**1**

**2**

**4**

**5**

**BONUS:**

**1-**function add(a){

   return function(b){

       console.log(a + b);

   }

}

add(2)(3);

**2-** var arr = [{ id: 1, username: 'ali' },

  { id: 2, username: 'mohamed'},

  { id: 3, username: 'ahmed' }];

function userExists(username) {

  return arr.some(function(ue) {

    return ue.username === username;

  });

}

console.log(userExists('ali')); // true

console.log(userExists('maher')); // false