

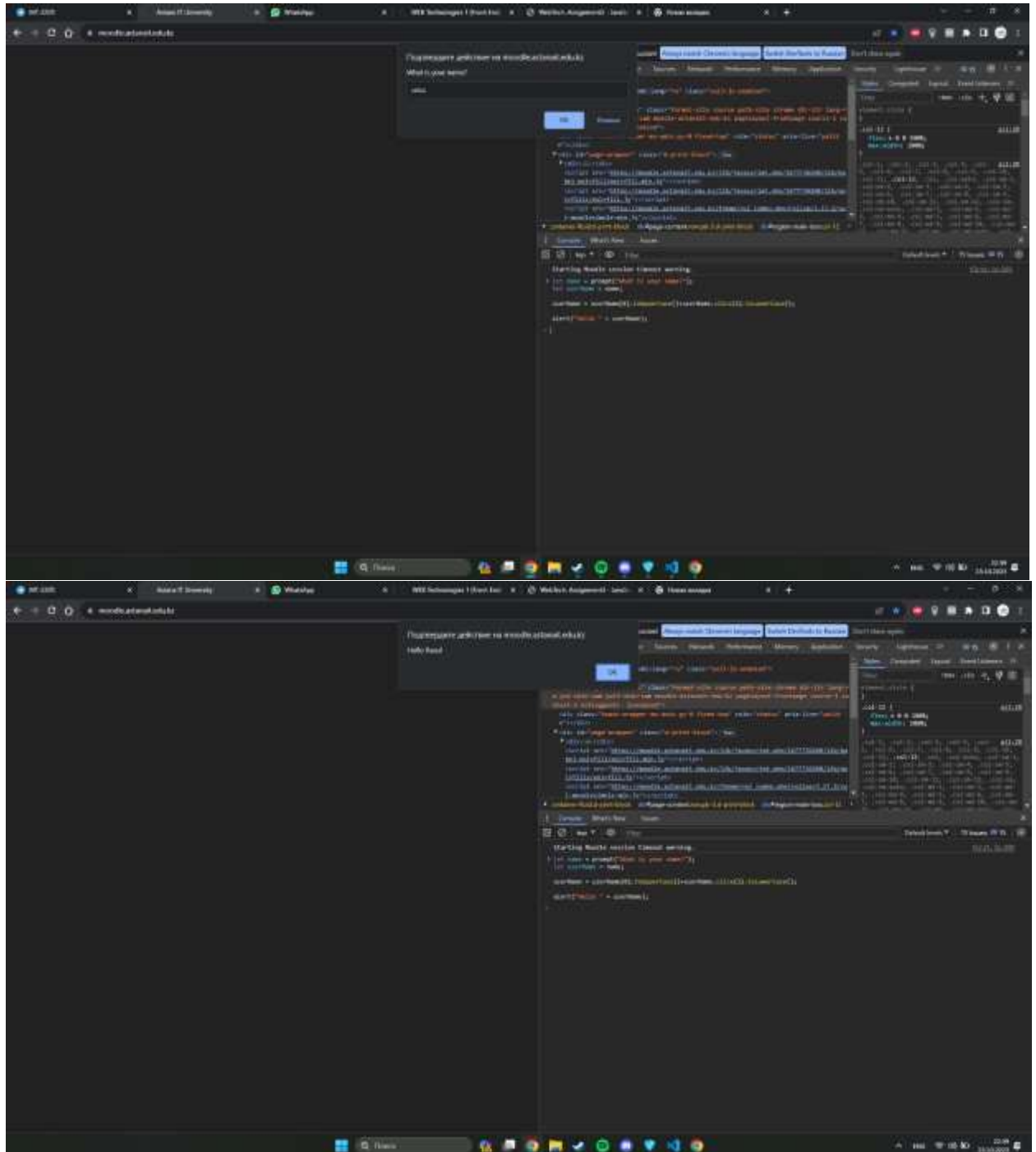
Task 1-Strings

```
let name = prompt("What is your name?");
```

```
let userName = name;
```

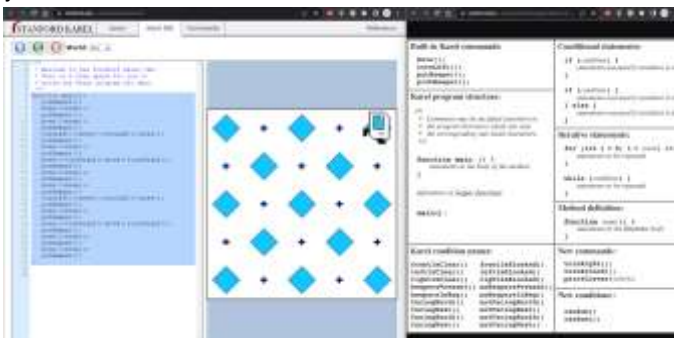
```
userName = userName[0].toUpperCase()+userName.slice(1).toLowerCase();
```

```
alert("Hello " + userName);
```

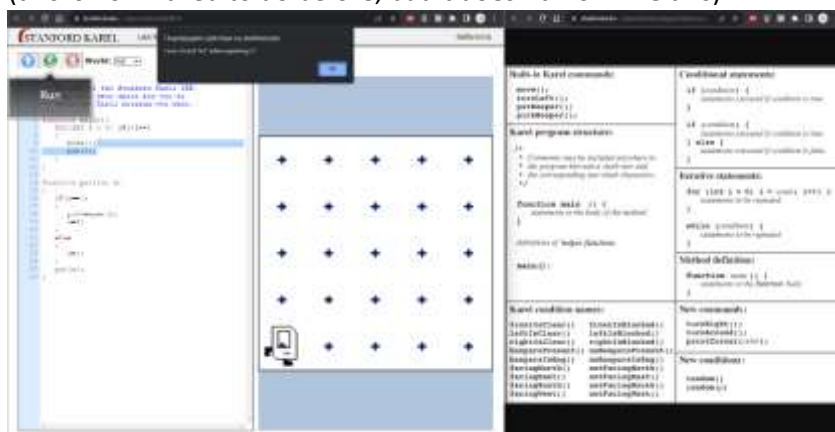


Task 2- Functions.

```
function main(){
    putBeeper();
    move();move();
    putBeeper();
    move();move();
    putBeeper();
    turnLeft();move();turnLeft();move();
    putBeeper();
    move();move();
    putBeeper();
    move();turnRight();move();turnRight();
    putBeeper();
    move();move();
    putBeeper();
    move();move();
    putBeeper();
    turnLeft();move();turnLeft();move();
    putBeeper();
    move();move();
    putBeeper();
    move();turnRight();move();turnRight();
    putBeeper();
    move();move();
    putBeeper();
    move();move();
    putBeeper();
}
```



(this is how I tried to do before, but it doesn't work like this)



Task 3 - Functions.

the program eventually got frozen

The screenshot shows the Stanford Karel IDE. The code editor on the left contains the following program:

```
1 // the Stanford Karel IDE.  
2 // Free space for you to  
3 // Karel program you want.  
4  
5 while (notBeepersPresent())  
6 {  
7   if (frontIsClear())  
8     move();  
9   else  
10    if (leftIsClear())  
11    {  
12      turnLeft();  
13    }  
14   else if (rightIsClear())  
15     turnRight();  
16 }  
17  
18 // End of program
```

The Karel world on the right shows a 10x10 grid with a robot at (1,1) facing right. There are beepers at (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (1,8), (1,9), (1,10), (2,1), (2,2), (2,3), (2,4), (2,5), (2,6), (2,7), (2,8), (2,9), (2,10), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (3,7), (3,8), (3,9), (3,10), (4,1), (4,2), (4,3), (4,4), (4,5), (4,6), (4,7), (4,8), (4,9), (4,10), (5,1), (5,2), (5,3), (5,4), (5,5), (5,6), (5,7), (5,8), (5,9), (5,10), (6,1), (6,2), (6,3), (6,4), (6,5), (6,6), (6,7), (6,8), (6,9), (6,10), (7,1), (7,2), (7,3), (7,4), (7,5), (7,6), (7,7), (7,8), (7,9), (7,10), (8,1), (8,2), (8,3), (8,4), (8,5), (8,6), (8,7), (8,8), (8,9), (8,10), (9,1), (9,2), (9,3), (9,4), (9,5), (9,6), (9,7), (9,8), (9,9), (9,10), and (10,1). A blue diamond is at (10,10).

The reference panel on the right contains the following information:

Boilt-in Karel commands:	Conditional statements:
<pre>move(); turnLeft(); putBeeper(); pickBeeper();</pre>	<pre>if (condition) { statements executed if condition is true } if (condition) { statements executed if condition is true } else { statements executed if condition is false }</pre>
Karel program structure:	Iterative statements:
<pre>/* * Comments may be included anywhere in * the program between a slash-star and * the corresponding star-slash characters. */ function main () { statements in the body of the method } definitions of helper functions main();</pre>	<pre>for (int i = 0; i < count; i++) statements to be repeated } while (condition) { statements to be repeated }</pre>
Karel condition names:	Method definition:
<pre>frontIsClear() frontIsBlocked() leftIsClear() leftIsBlocked() rightIsClear() rightIsBlocked() beepersPresent() noBeepersPresent() beepersInBag() noBeepersInBag() facingNorth() notFacingNorth() facingEast() notFacingEast() facingSouth() notFacingSouth() facingWest() notFacingWest()</pre>	<pre>function name () { statements in the function body }</pre>
New commands:	New conditions:
<pre>turnRight(); turnAround(); paintCorner (color);</pre>	<pre>random(); random(p);</pre>

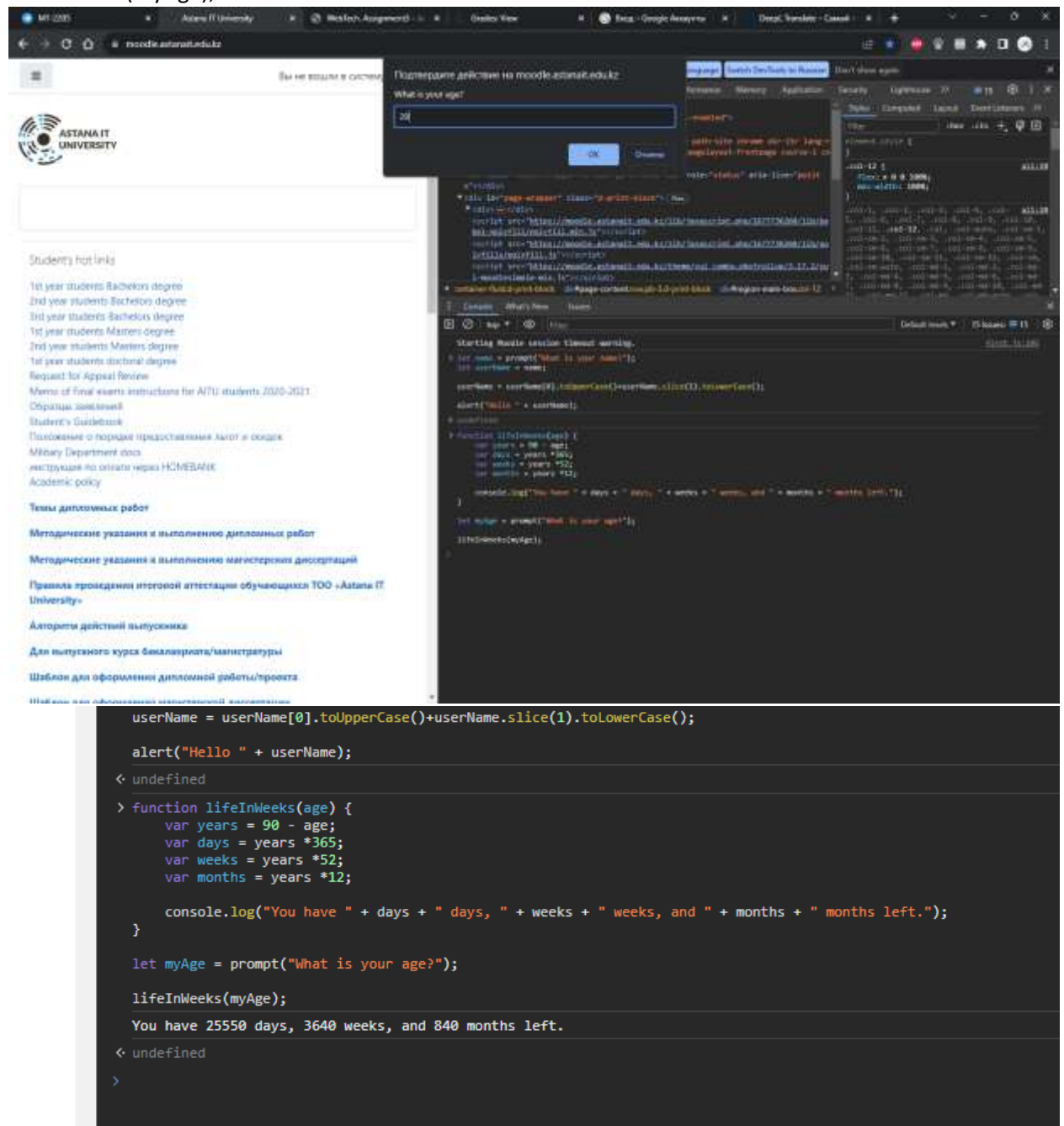
Task 4

```
function lifeInWeeks(age) {
  var years = 90 - age;
  var days = years * 365;
  var weeks = years * 52;
  var months = years * 12;

  console.log("You have " + days + " days, " + weeks + " weeks, and " + months + " months left.");
}
```

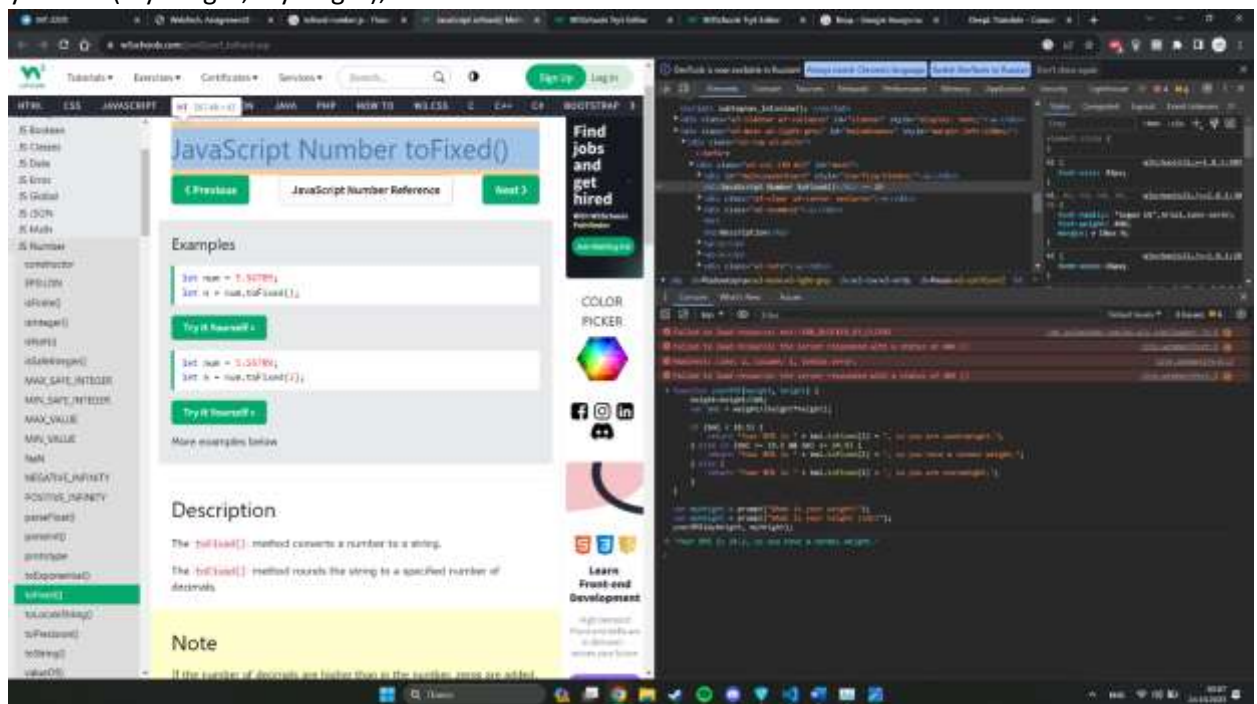
```
let myAge = prompt("What is your age?");
```

```
lifeInWeeks(myAge);
```



Task 5

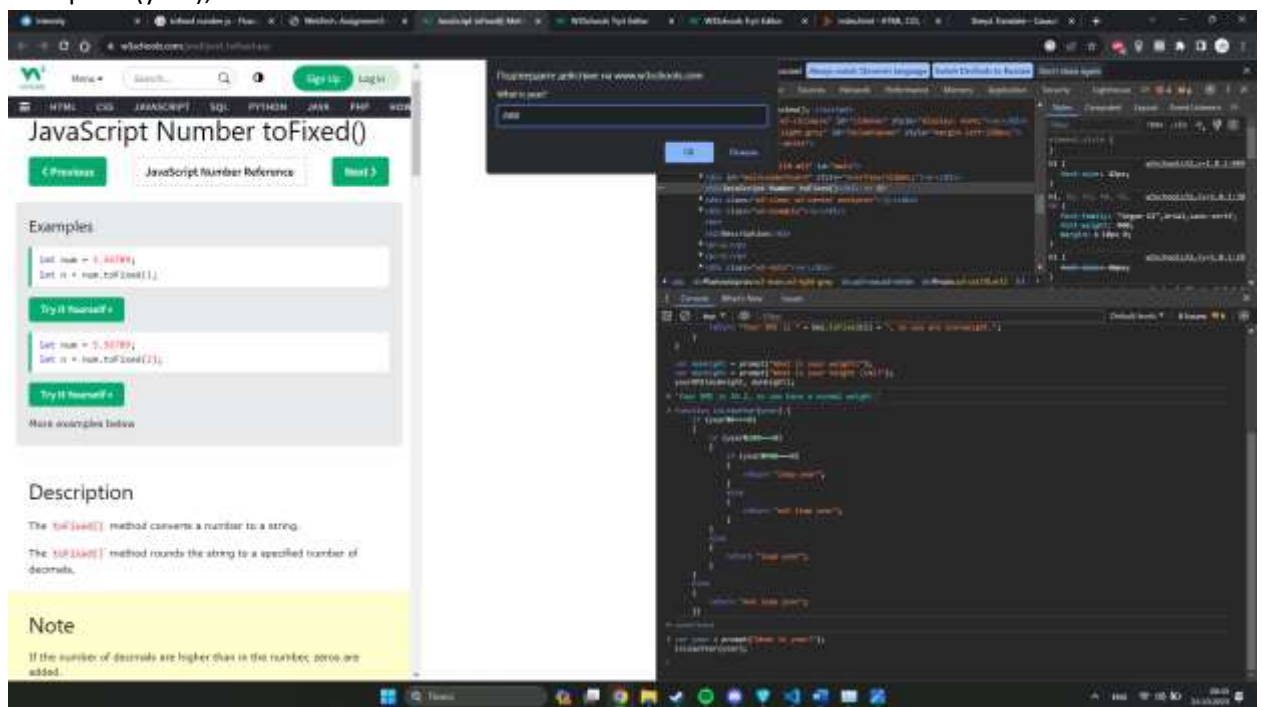
```
function yourBMI(weight, height) {  
    height=height/100;  
    var bmi = weight/(height*height);  
  
    if (bmi < 18.5) {  
        return "Your BMI is " + bmi.toFixed(1) + ", so you are underweight.";  
    } else if (bmi >= 18.5 && bmi <= 24.9) {  
        return "Your BMI is " + bmi.toFixed(1) + ", so you have a normal weight.";  
    } else {  
        return "Your BMI is " + bmi.toFixed(1) + ", so you are overweight.";  
    }  
}  
  
var myWeight = prompt("What is your weight?");  
var myHeight = prompt("What is your height (cm)?");  
yourBMI(myWeight, myHeight);
```



task 6

```
function isLeapYear(year) {  
  if (year%4===0)  
  {  
    if (year%100===0)  
    {  
      if (year%400===0)  
      {  
        return "leap year";  
      }  
      else  
      {  
        return "not leap year";  
      }  
    }  
    else  
    {  
      return "leap year";  
    }  
  }  
  else  
  {  
    return "not leap year";  
  }  
}
```

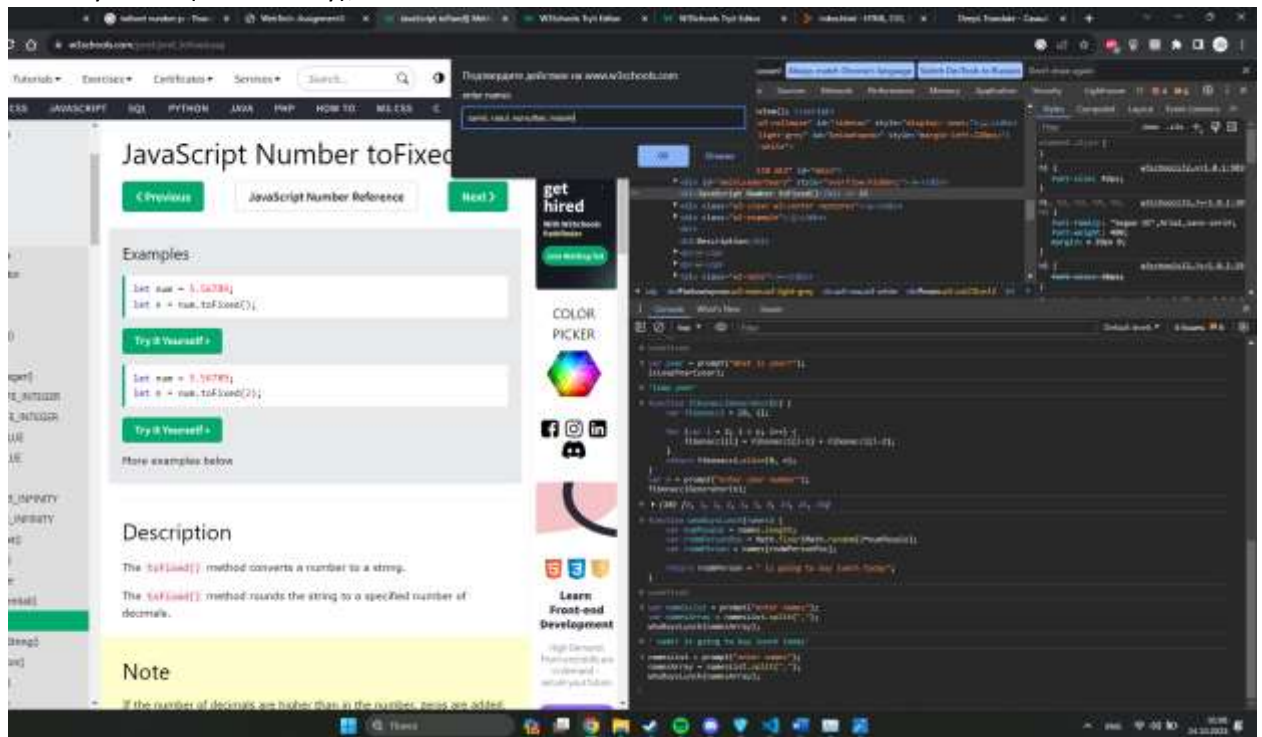
```
var year = prompt("What is year?");  
isLeapYear(year);
```



Task 7

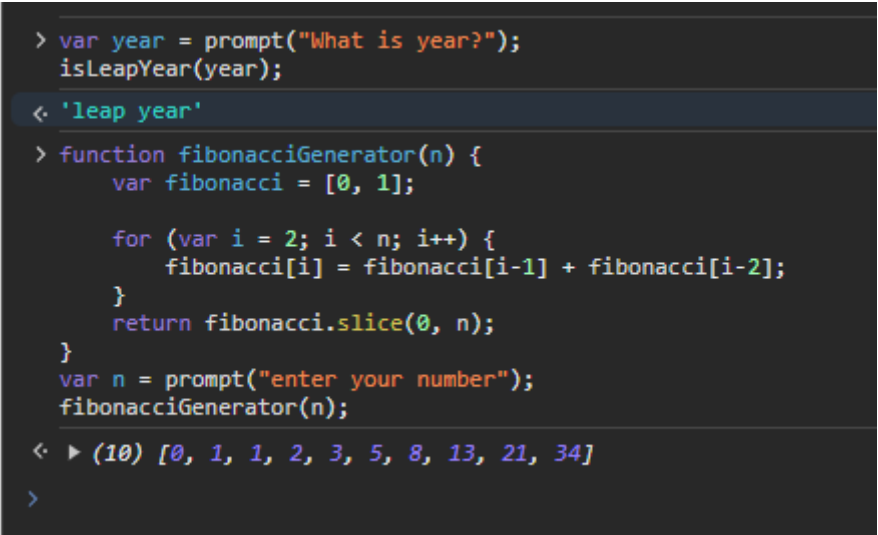
```
function whoBuysLunch(names) {  
  var numPeople = names.length;  
  var rndmPersonPos = Math.floor(Math.random()*numPeople);  
  var rndmPerson = names[rndmPersonPos];  
  
  return rndmPerson + " is going to buy lunch today";  
}
```

```
var namesList = prompt("enter names");  
var namesArray = namesList.split(",");  
whoBuysLunch(namesArray);
```



Task 8

```
function fibonacciGenerator(n) {  
  var fibonacci = [0, 1];  
  
  for (var i = 2; i < n; i++) {  
    fibonacci[i] = fibonacci[i-1] + fibonacci[i-2];  
  }  
  return fibonacci.slice(0, n);  
}  
var n = prompt("enter your number");  
fibonacciGenerator(n);
```



```
> var year = prompt("What is year?");  
isLeapYear(year);  
< 'leap year'  
> function fibonacciGenerator(n) {  
  var fibonacci = [0, 1];  
  
  for (var i = 2; i < n; i++) {  
    fibonacci[i] = fibonacci[i-1] + fibonacci[i-2];  
  }  
  return fibonacci.slice(0, n);  
}  
var n = prompt("enter your number");  
fibonacciGenerator(n);  
< ▶ (10) [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]  
>
```