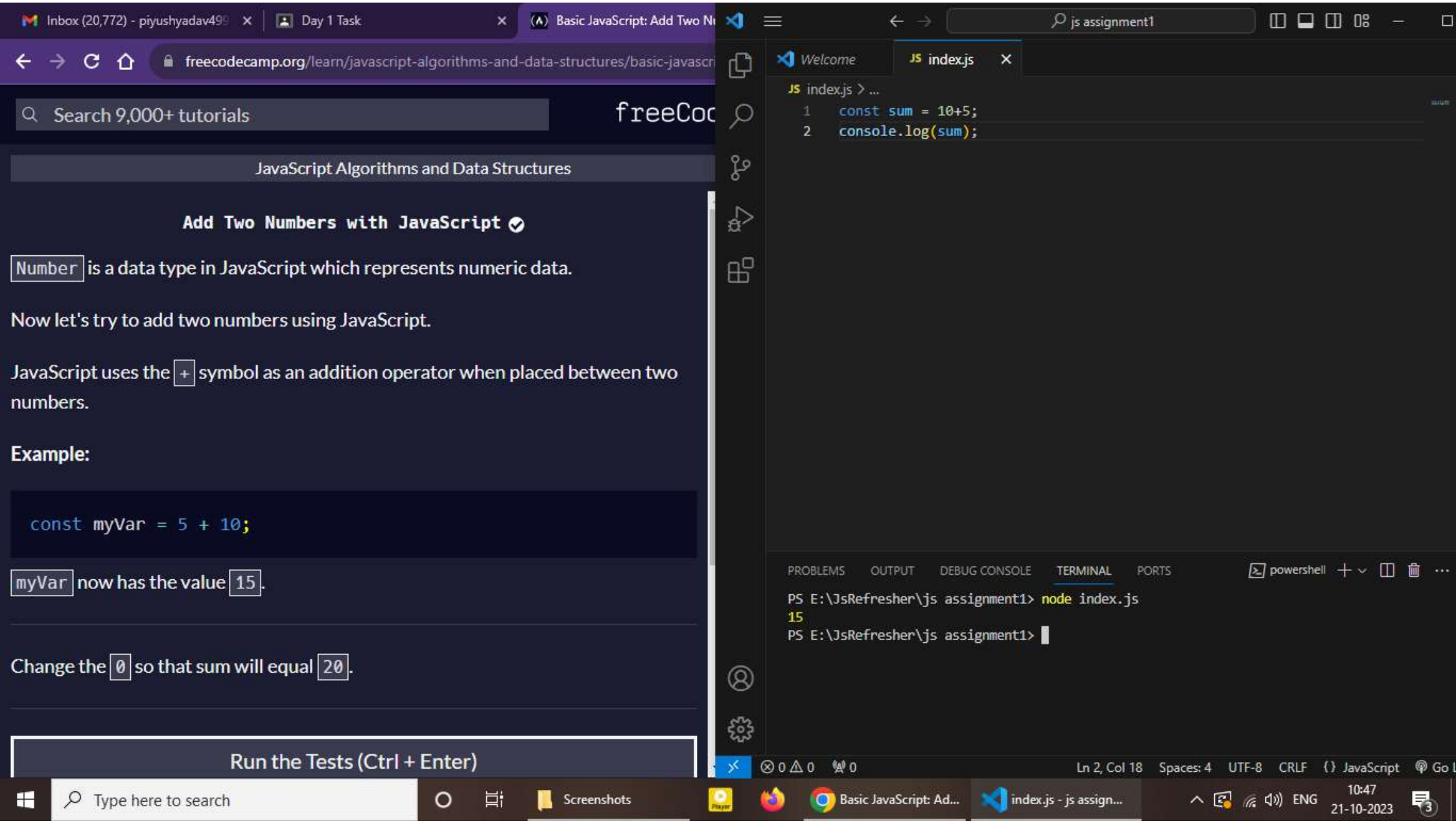
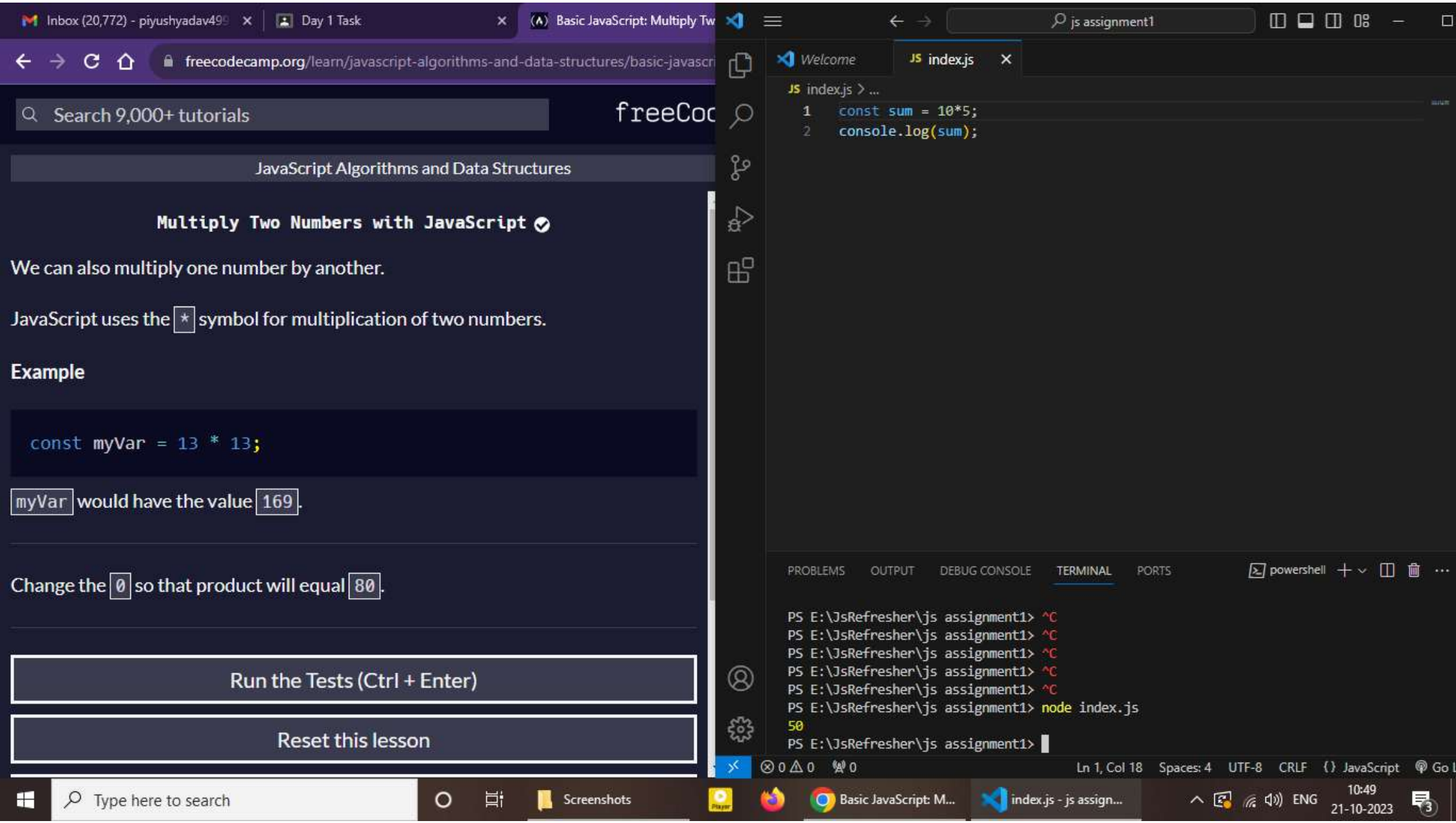
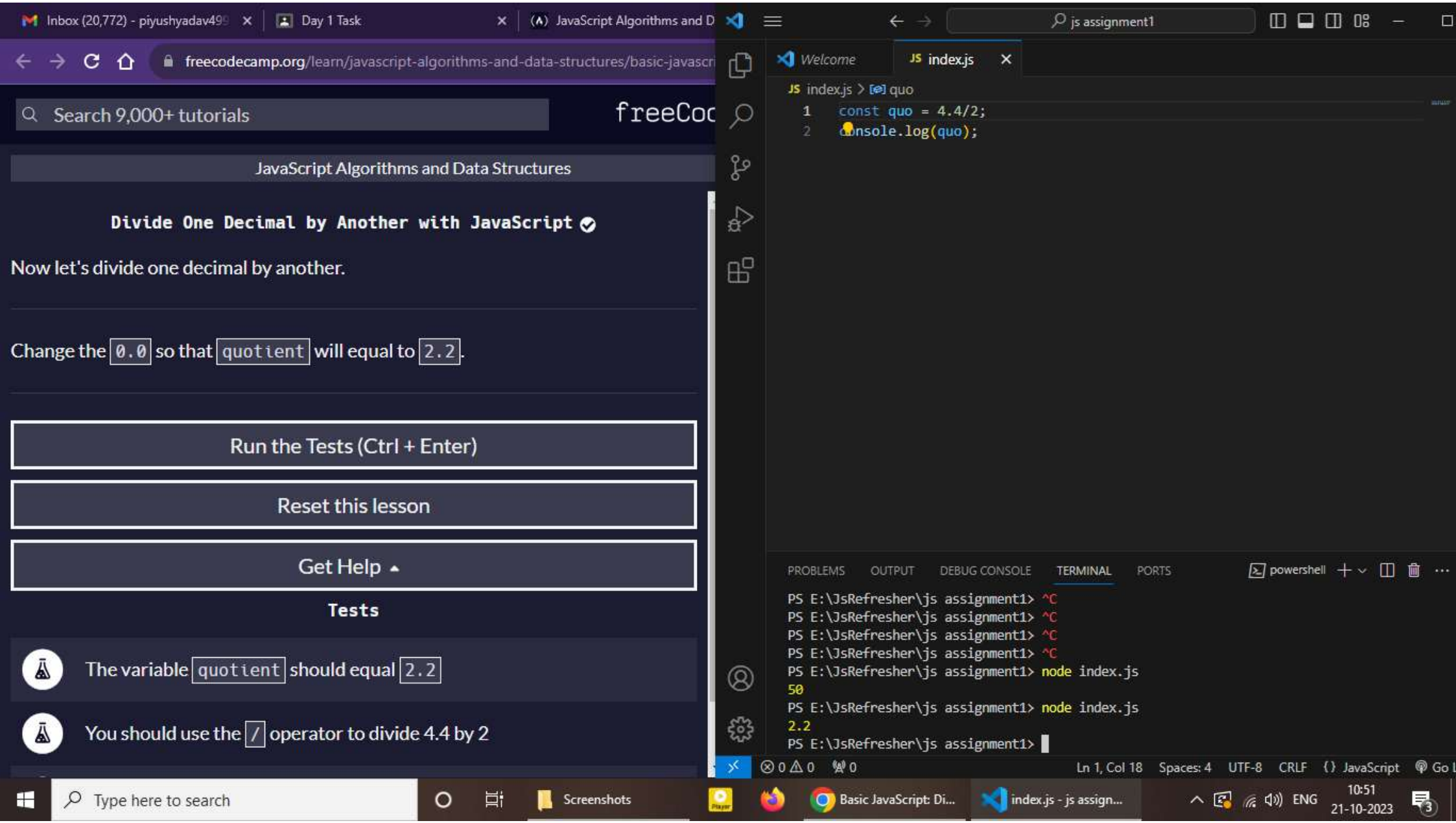

Inbox (20,772) - piyushyadav49
Day 1 Task
Basic JavaScript: Add Two Nu
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri
Search 9,000+ tutorials
freeCod
JavaScript Algorithms and Data Structures
Add Two Numbers with JavaScript
Number is a data type in JavaScript which represents numeric data.
Now let's try to add two numbers using JavaScript.
JavaScript uses the + symbol as an addition operator when placed between two numbers.
Example:
const myVar = 5 + 10;
myVar now has the value 15.
Change the 0 so that sum will equal 20.
Run the Tests (Ctrl + Enter)
js assignment1
Welcome
index.js
index.js > ...
1 const sum = 10+5;
2 console.log(sum);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> node index.js
15
PS E:\JsRefresher\js assignment1>
Ln 2, Col 18
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Ad...
index.js - js assign...
ENG
10:47
21-10-2023


Inbox (20,772) - piyushyadav49
Day 1 Task
Basic JavaScript: Multiply Tw
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri
Search 9,000+ tutorials
freeCod
JavaScript Algorithms and Data Structures
Multiply Two Numbers with JavaScript
We can also multiply one number by another.
JavaScript uses the * symbol for multiplication of two numbers.
Example
const myVar = 13 * 13;
myVar would have the value 169.
Change the 0 so that product will equal 80.
Run the Tests (Ctrl + Enter)
Reset this lesson
js assignment1
Welcome
index.js
index.js > ...
1 const sum = 10*5;
2 console.log(sum);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> node index.js
50
PS E:\JsRefresher\js assignment1>
Ln 1, Col 18
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: M...
index.js - js assign...
ENG
10:49
21-10-2023


Inbox (20,772) - piyushyadav49
Day 1 Task
JavaScript Algorithms and D
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri
Search 9,000+ tutorials
freeCod
JavaScript Algorithms and Data Structures
Divide One Decimal by Another with JavaScript
Now let's divide one decimal by another.
Change the 0.0 so that quotient will equal to 2.2.
Run the Tests (Ctrl + Enter)
Reset this lesson
Get Help
Tests
The variable quotient should equal 2.2
You should use the / operator to divide 4.4 by 2
js assignment1
Welcome
index.js
index.js > quo
1 const quo = 4.4/2;
2 console.log(quo);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> ^C
PS E:\JsRefresher\js assignment1> node index.js
50
PS E:\JsRefresher\js assignment1> node index.js
2.2
PS E:\JsRefresher\js assignment1>
Ln 1, Col 18
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Di...
index.js - js assign...
ENG
10:51
21-10-2023

Inbox (20,772) - piyushyadav49 | Day 1 Task | JavaScript Algorithms and D

freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri

Search 9,000+ tutorials

freeCoc

JavaScript Algorithms and Data Structures

Finding a Remainder in JavaScript

The *remainder* operator `%` gives the remainder of the division of two numbers.

Example

```
5 % 2 = 1
5 / 2 = 2 remainder 1
2 * 2 = 4
5 - 4 = 1
```

Usage

In mathematics, a number can be checked to be even or odd by checking the remainder of the division of the number by `2`. Even numbers have a remainder of `0`, while odd numbers a remainder of `1`.

```
17 % 2 = 1
48 % 2 = 0
```

Note: The *remainder* operator is sometimes incorrectly referred to as the modulus operator. It is very similar to modulus, but does not work properly with negative

js assignment1

Welcome | index.js

index.js > ...

```
1  const rem = 5%2;
2  console.log(rem);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

```
PS E:\JsRefresher\js assignment1> node index.js
50
PS E:\JsRefresher\js assignment1> node index.js
2.2
PS E:\JsRefresher\js assignment1> node index.js
2.5
PS E:\JsRefresher\js assignment1> node index.js
1
PS E:\JsRefresher\js assignment1>
```

Ln 2, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

Type here to search | Screenshots | Basic JavaScript: Fi... | index.js - js assign... | ENG | 10:56 21-10-2023

Inbox (20,772) - piyushyadav49 | Day 1 Task | JavaScript Algorithms and D

freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri

Search 9,000+ tutorials

freeCoc

JavaScript Algorithms and Data Structures

One such operator is the `+=` operator.

```
let myVar = 1;
myVar += 5;
console.log(myVar);
```

`6` would be displayed in the console.

Convert the assignments for `a`, `b`, and `c` to use the `+=` operator.

Run the Tests (Ctrl + Enter)

Reset this lesson

Get Help

Tests

js assignment1

Welcome | index.js

index.js > ...

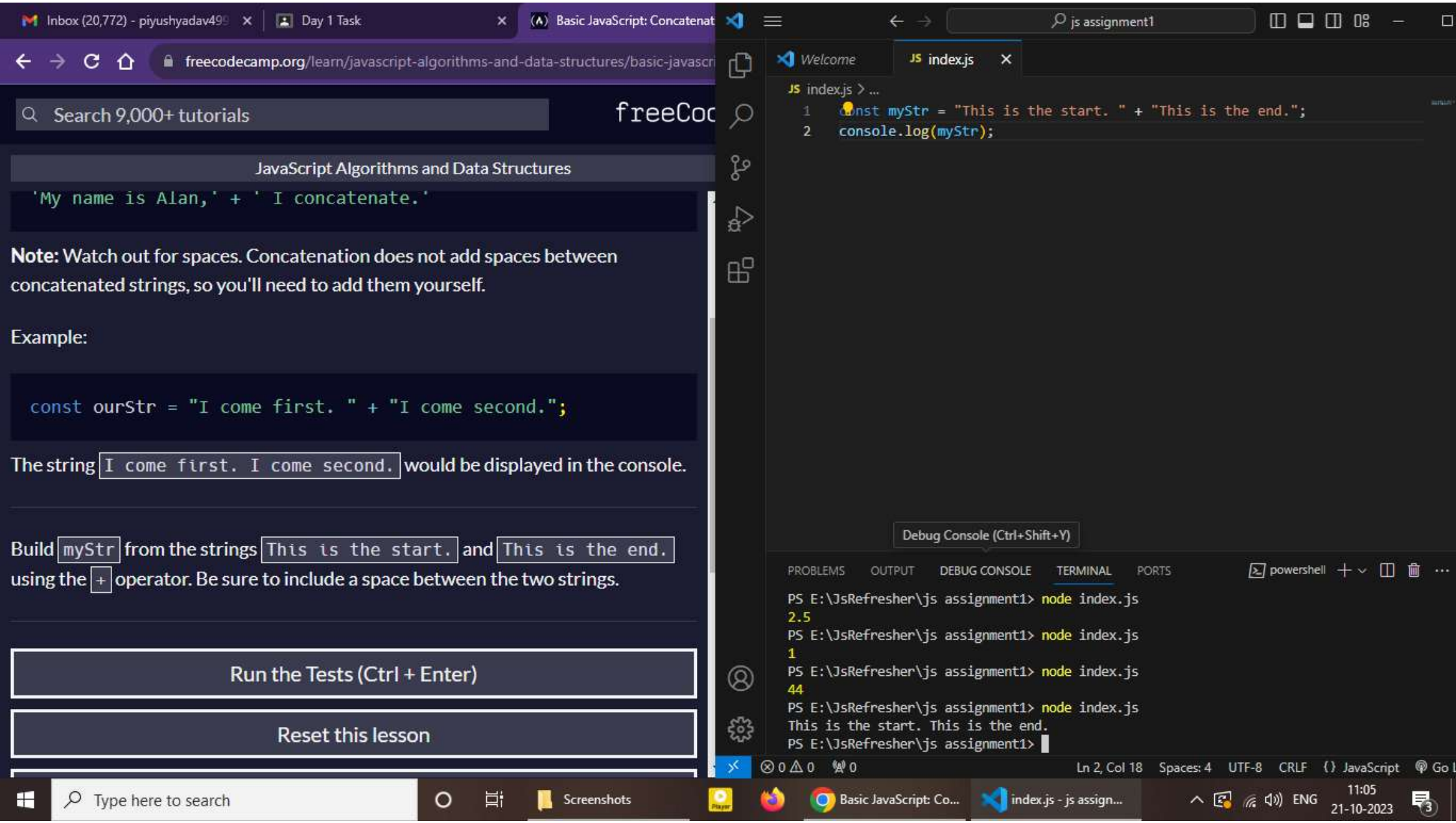
```
1  let myVar = 1;
2  myVar +=43;
3  console.log(myVar);
```

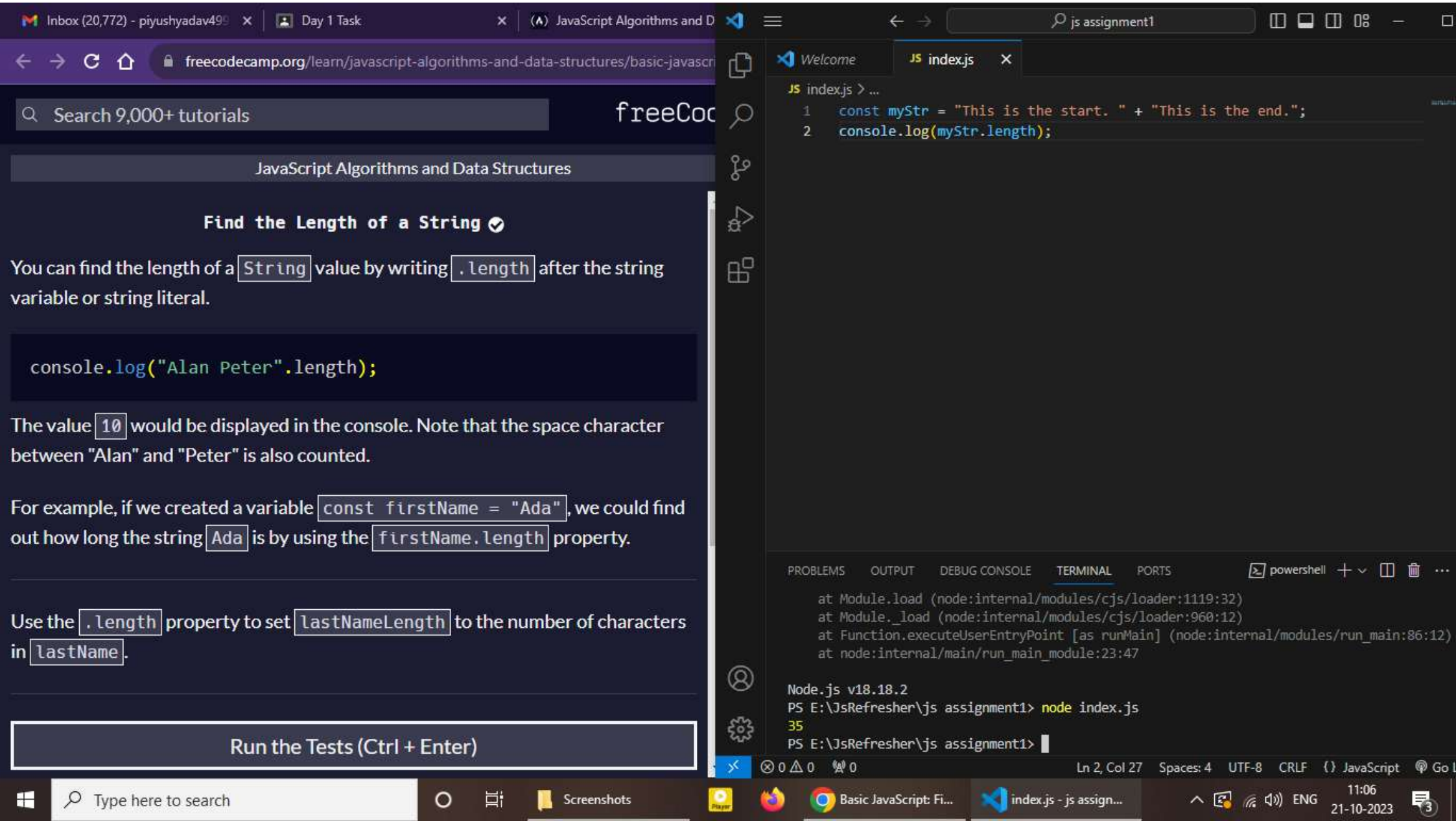
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

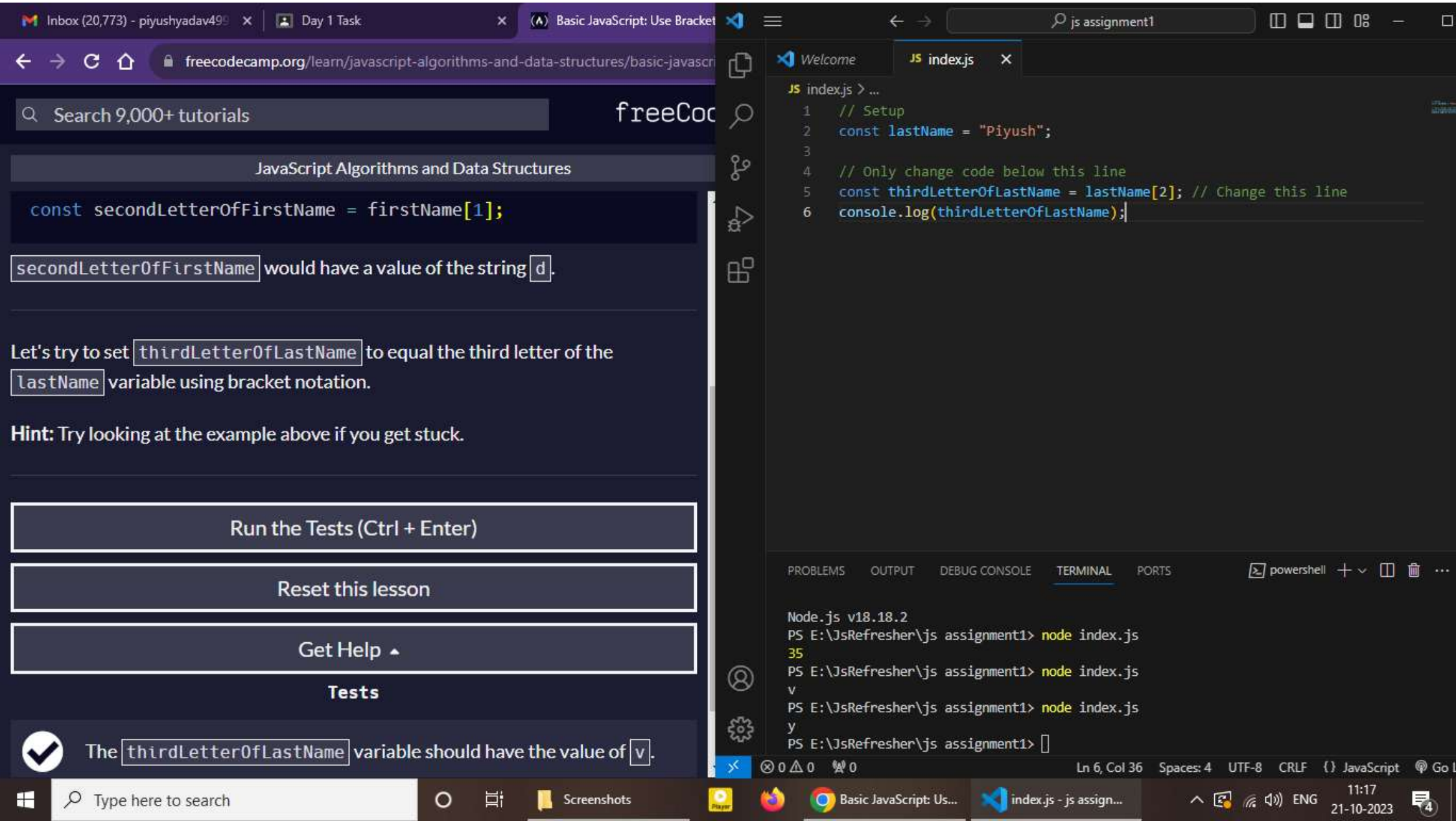
```
PS E:\JsRefresher\js assignment1> node index.js
2.2
PS E:\JsRefresher\js assignment1> node index.js
2.5
PS E:\JsRefresher\js assignment1> node index.js
1
PS E:\JsRefresher\js assignment1> node index.js
44
PS E:\JsRefresher\js assignment1>
```

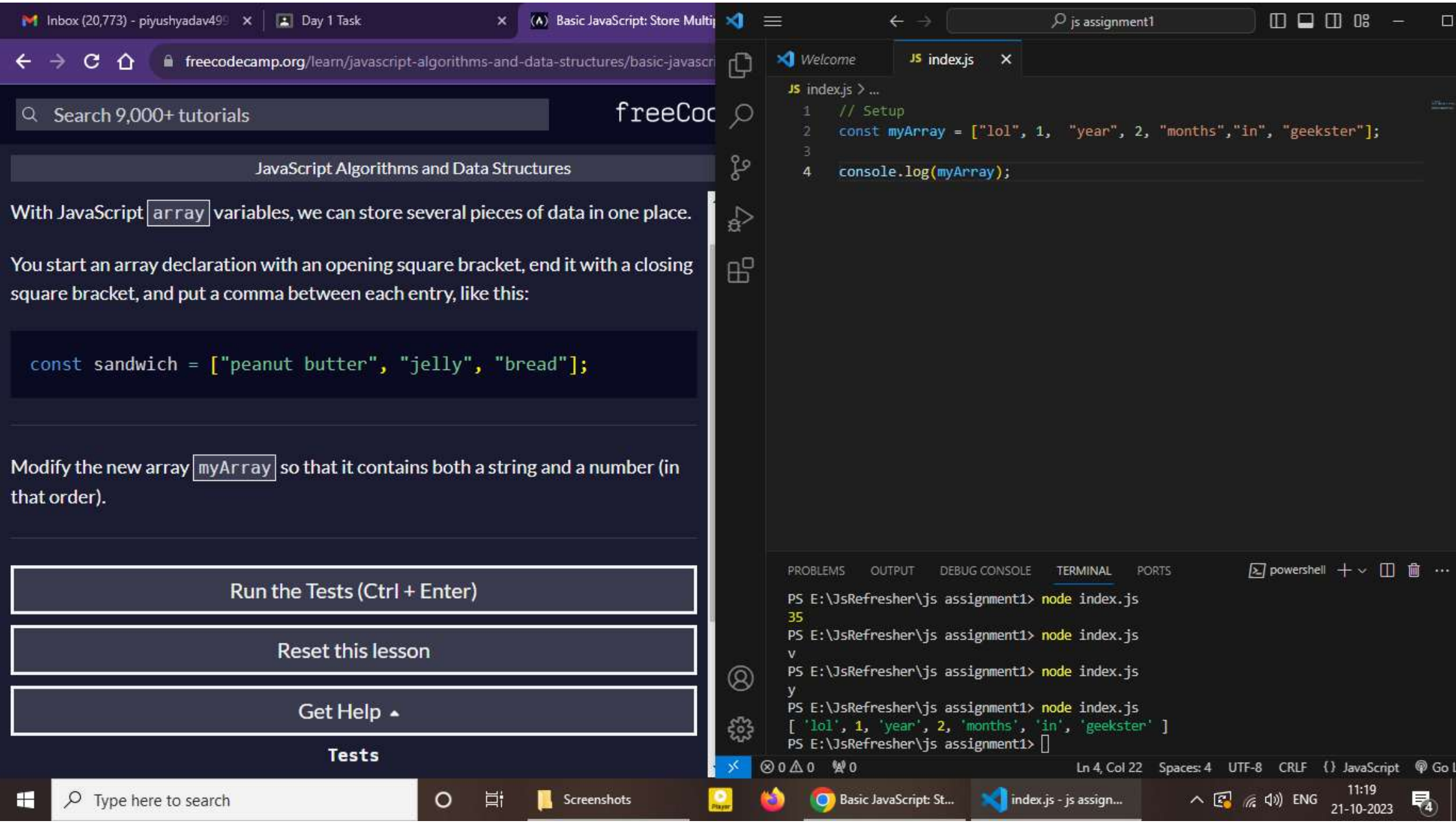
Ln 3, Col 18 Spaces: 4 UTF-8 CRLF JavaScript

Type here to search Screenshots Basic JavaScript: Co... index.js - js assign... ENG 11:01 21-10-2023


Inbox (20,772) - piyushyadav49
Day 1 Task
Basic JavaScript: Concatenat
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri
Search 9,000+ tutorials
freeCod
JavaScript Algorithms and Data Structures
'My name is Alan,' + ' I concatenate.'
Note: Watch out for spaces. Concatenation does not add spaces between concatenated strings, so you'll need to add them yourself.
Example:
const ourStr = "I come first. " + "I come second.";
The string I come first. I come second. would be displayed in the console.
Build myStr from the strings This is the start. and This is the end. using the + operator. Be sure to include a space between the two strings.
Run the Tests (Ctrl + Enter)
Reset this lesson
js assignment1
Welcome
index.js
index.js > ...
1 const myStr = "This is the start. " + "This is the end.";
2 console.log(myStr);
Debug Console (Ctrl+Shift+Y)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> node index.js
2.5
PS E:\JsRefresher\js assignment1> node index.js
1
PS E:\JsRefresher\js assignment1> node index.js
44
PS E:\JsRefresher\js assignment1> node index.js
This is the start. This is the end.
PS E:\JsRefresher\js assignment1>
Ln 2, Col 18
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Co...
index.js - js assign...
ENG
11:05
21-10-2023


Inbox (20,772) - piyushyadav49
Day 1 Task
JavaScript Algorithms and D
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri
Search 9,000+ tutorials
JavaScript Algorithms and Data Structures
Find the Length of a String
You can find the length of a String value by writing .length after the string variable or string literal.
console.log("Alan Peter".length);
The value 10 would be displayed in the console. Note that the space character between "Alan" and "Peter" is also counted.
For example, if we created a variable const firstName = "Ada", we could find out how long the string Ada is by using the firstName.length property.
Use the .length property to set lastNameLength to the number of characters in lastName.
Run the Tests (Ctrl + Enter)
js assignment1
Welcome
index.js
const myStr = "This is the start. " + "This is the end.";
console.log(myStr.length);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
at Module.load (node:internal/modules/cjs/loader:1119:32)
at Module._load (node:internal/modules/cjs/loader:960:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:86:12)
at node:internal/main/run_main_module:23:47
Node.js v18.18.2
PS E:\JsRefresher\js assignment1> node index.js
35
PS E:\JsRefresher\js assignment1>
Ln 2, Col 27
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Fi...
index.js - js assign...
ENG
11:06
21-10-2023


Inbox (20,773) - piyushyadav49
Day 1 Task
Basic JavaScript: Use Bracket
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascr
Search 9,000+ tutorials
freeCod
JavaScript Algorithms and Data Structures
const secondLetterOfFirstName = firstName[1];
secondLetterOfFirstName would have a value of the string d.
Let's try to set thirdLetterOfLastName to equal the third letter of the lastName variable using bracket notation.
Hint: Try looking at the example above if you get stuck.
Run the Tests (Ctrl + Enter)
Reset this lesson
Get Help
Tests
The thirdLetterOfLastName variable should have the value of v.
js assignment1
Welcome
index.js
index.js > ...
1 // Setup
2 const lastName = "Piyush";
3
4 // Only change code below this line
5 const thirdLetterOfLastName = lastName[2]; // Change this line
6 console.log(thirdLetterOfLastName);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
Node.js v18.18.2
PS E:\JsRefresher\js assignment1> node index.js
35
PS E:\JsRefresher\js assignment1> node index.js
v
PS E:\JsRefresher\js assignment1> node index.js
y
PS E:\JsRefresher\js assignment1>
Ln 6, Col 36
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Us...
index.js - js assign...
ENG
11:17
21-10-2023


Inbox (20,773) - piyushyadav49
Day 1 Task
Basic JavaScript: Store Multi
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascr
Search 9,000+ tutorials
freeCod
JavaScript Algorithms and Data Structures
With JavaScript array variables, we can store several pieces of data in one place.
You start an array declaration with an opening square bracket, end it with a closing square bracket, and put a comma between each entry, like this:
const sandwich = ["peanut butter", "jelly", "bread"];
Modify the new array myArray so that it contains both a string and a number (in that order).
Run the Tests (Ctrl + Enter)
Reset this lesson
Get Help
Tests
js assignment1
Welcome
index.js
index.js > ...
1 // Setup
2 const myArray = ["lol", 1, "year", 2, "months","in", "geekster"];
3
4 console.log(myArray);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> node index.js
35
PS E:\JsRefresher\js assignment1> node index.js
v
PS E:\JsRefresher\js assignment1> node index.js
y
PS E:\JsRefresher\js assignment1> node index.js
['lol', 1, 'year', 2, 'months', 'in', 'geekster']
PS E:\JsRefresher\js assignment1>
Ln 4, Col 22
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: St...
index.js - js assign...
ENG
11:19
21-10-2023

Inbox (20,773) - piyushyadav49 | Day 1 Task | Basic JavaScript: Access Mult

freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascri

Search 9,000+ tutorials

freeCod

JavaScript Algorithms and Data Structures

In this example, `subarray` has the value `[[10, 11, 12], 13, 14]`, `nestedSubarray` has the value `[10, 11, 12]`, and `element` has the value `11`.

Note: There shouldn't be any spaces between the array name and the square brackets, like `array [0][0]` and even this `array [0] [0]` is not allowed. Although JavaScript is able to process this correctly, this may confuse other programmers reading your code.

Using bracket notation select an element from `myArray` such that `myData` is equal to `8`.

Run the Tests (Ctrl + Enter)

Reset this lesson

Get Help

js assignment1

Welcome | index.js

index.js > ...

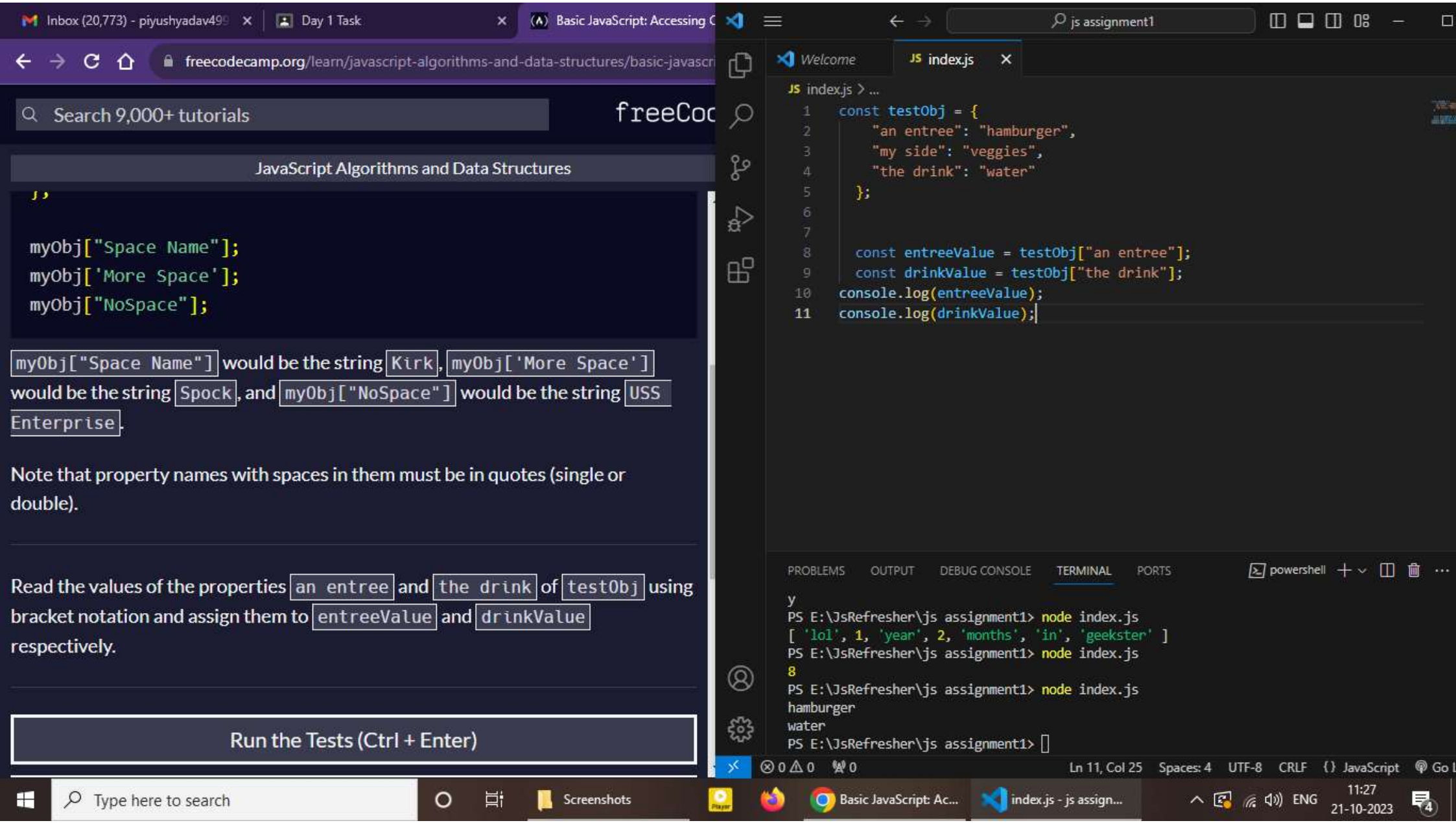
```
1  // Setup
2  const myArray = [
3      [1, 2, 3],
4      [4, 5, 6],
5      [7, 8, 9],
6      [[10, 11, 12], 13, 14],
7    ];
8
9    const myData = myArray[2][1];
10
11 console.log(myData);
```

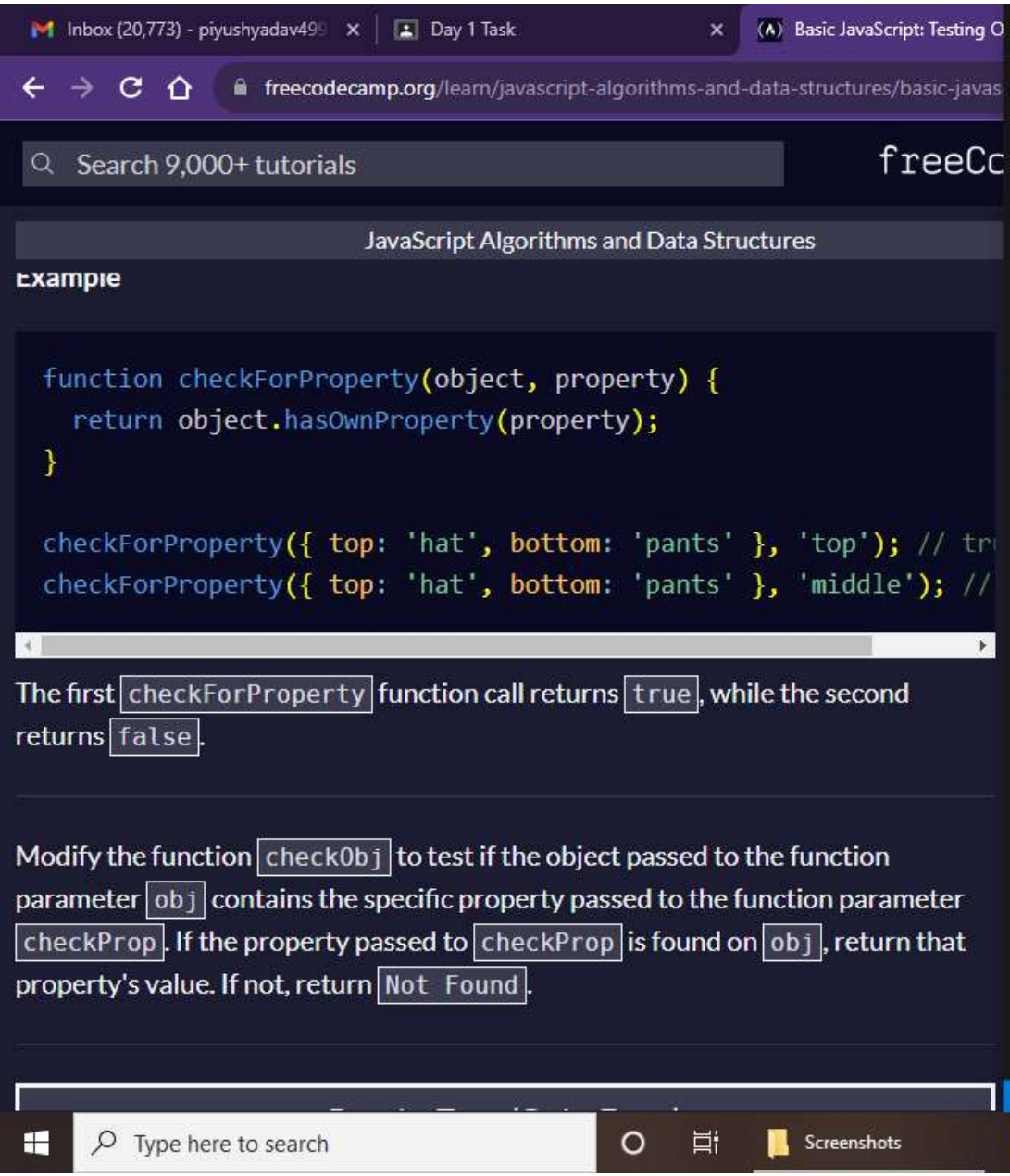
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell

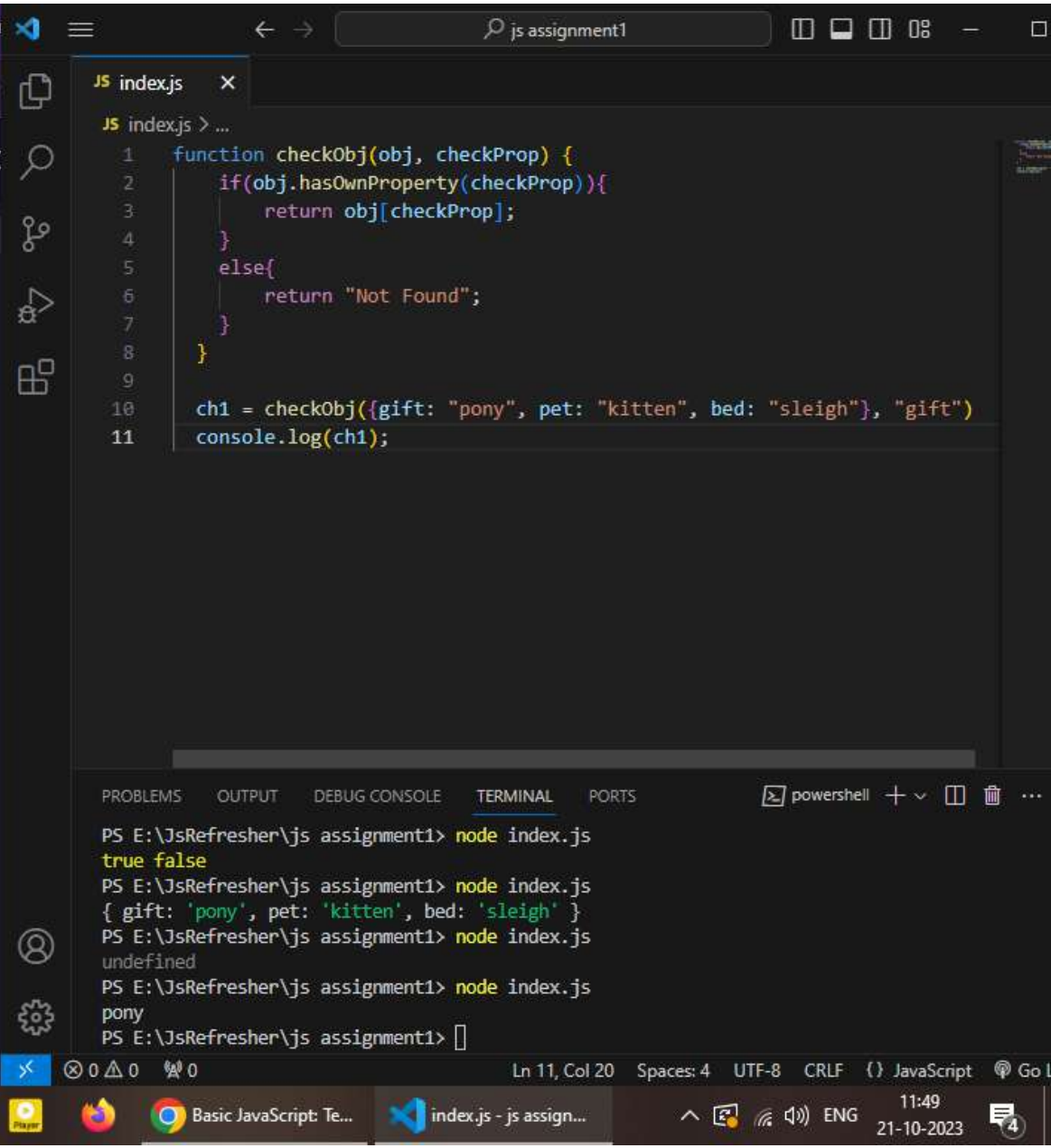
```
PS E:\JsRefresher\js assignment1> node index.js
v
PS E:\JsRefresher\js assignment1> node index.js
y
PS E:\JsRefresher\js assignment1> node index.js
[ 'lol', 1, 'year', 2, 'months', 'in', 'geekster' ]
PS E:\JsRefresher\js assignment1> node index.js
8
PS E:\JsRefresher\js assignment1>
```

Ln 11, Col 19 Spaces: 4 UTF-8 CRLF JavaScript

Type here to search | Screenshots | Basic JavaScript: Ac... | index.js - js assign... | ENG 11:24 21-10-2023


Inbox (20,773) - piyushyadav49
Day 1 Task
Basic JavaScript: Accessing
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascr
Search 9,000+ tutorials
JavaScript Algorithms and Data Structures
myObj["Space Name"];
myObj['More Space'];
myObj["NoSpace"];
myObj["Space Name"] would be the string Kirk, myObj['More Space'] would be the string Spock, and myObj["NoSpace"] would be the string USS Enterprise.
Note that property names with spaces in them must be in quotes (single or double).
Read the values of the properties an entree and the drink of testObj using bracket notation and assign them to entreeValue and drinkValue respectively.
Run the Tests (Ctrl + Enter)
js assignment1
Welcome
index.js
const testObj = {
"an entree": "hamburger",
"my side": "veggies",
"the drink": "water"
};
const entreeValue = testObj["an entree"];
const drinkValue = testObj["the drink"];
console.log(entreeValue);
console.log(drinkValue);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> node index.js
['lol', 1, 'year', 2, 'months', 'in', 'geekster']
PS E:\JsRefresher\js assignment1> node index.js
8
PS E:\JsRefresher\js assignment1> node index.js
hamburger
water
PS E:\JsRefresher\js assignment1>
Ln 11, Col 25
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Ac...
index.js - js assign...
ENG
11:27
21-10-2023


Inbox (20,773) - piyushyadav49
Day 1 Task
Basic JavaScript: Testing O
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javas
Search 9,000+ tutorials
freeCc
JavaScript Algorithms and Data Structures
Example
function checkForProperty(object, property) {
return object.hasOwnProperty(property);
}
checkForProperty({ top: 'hat', bottom: 'pants' }, 'top'); // tr
checkForProperty({ top: 'hat', bottom: 'pants' }, 'middle'); //
The first checkForProperty function call returns true, while the second returns false.
Modify the function checkObj to test if the object passed to the function parameter obj contains the specific property passed to the function parameter checkProp. If the property passed to checkProp is found on obj, return that property's value. If not, return Not Found.


js assignment1
index.js
index.js > ...
1 function checkObj(obj, checkProp) {
2 if(obj.hasOwnProperty(checkProp)){
3 return obj[checkProp];
4 }
5 else{
6 return "Not Found";
7 }
8 }
9
10 ch1 = checkObj({gift: "pony", pet: "kitten", bed: "sleigh"}, "gift")
11 console.log(ch1);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS E:\JsRefresher\js assignment1> node index.js
true false
PS E:\JsRefresher\js assignment1> node index.js
{ gift: 'pony', pet: 'kitten', bed: 'sleigh' }
PS E:\JsRefresher\js assignment1> node index.js
undefined
PS E:\JsRefresher\js assignment1> node index.js
pony
PS E:\JsRefresher\js assignment1>
Ln 11, Col 20 Spaces: 4 UTF-8 CRLF JavaScript
Type here to search
Screenshots
Basic JavaScript: Te...
index.js - js assign...
ENG
11:49
21-10-2023

Inbox (20,773) - piyushyadav49 | Day 1 Task | Basic JavaScript: Global vs

freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javas

Search 9,000+ tutorials

freeCc

JavaScript Algorithms and Data Structures

do this, the local variable takes precedence over the global variable.

In this example:

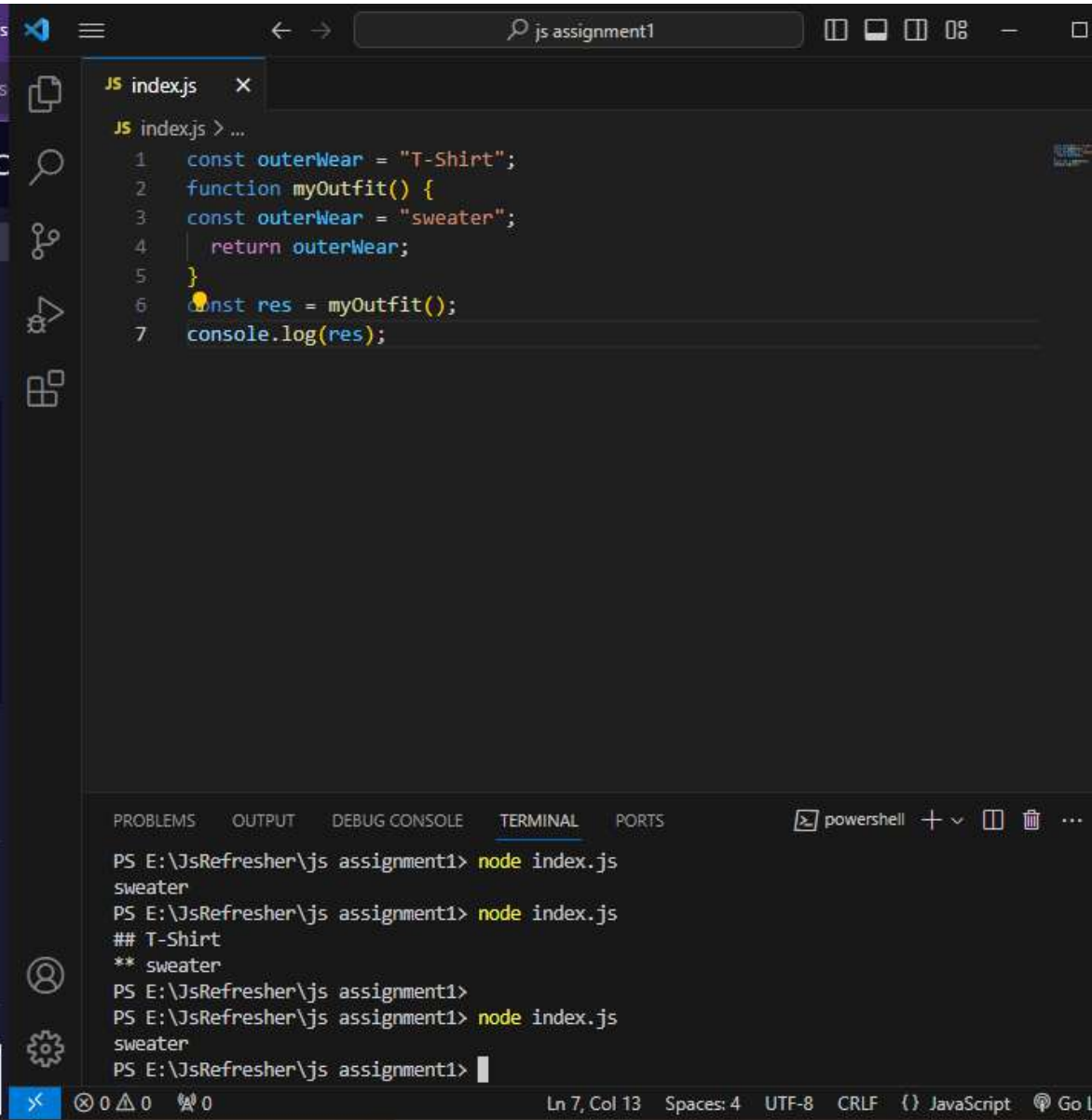
```
const someVar = "Hat";

function myFun() {
  const someVar = "Head";
  return someVar;
}
```

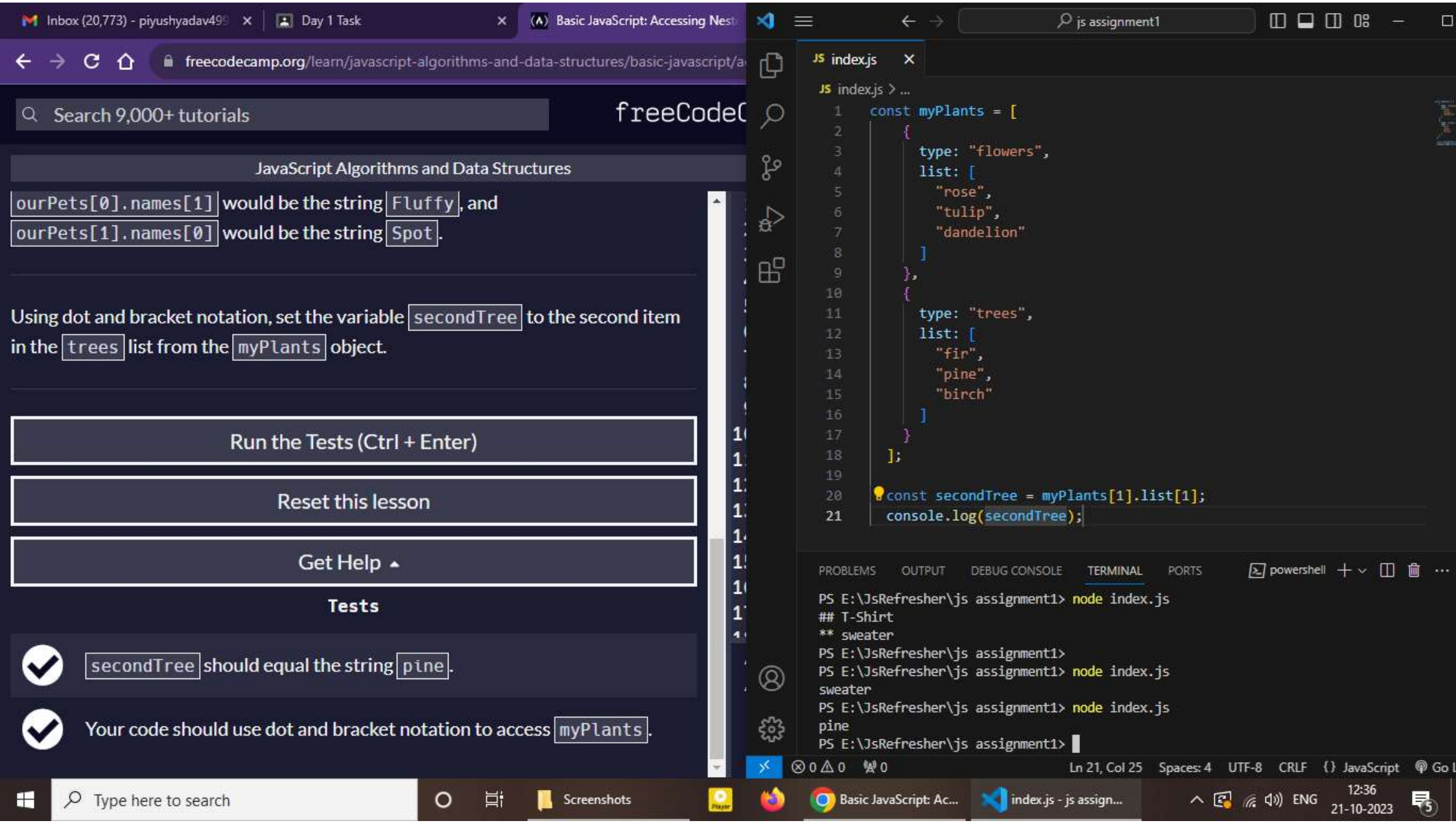
The function `myFun` will return the string `Head` because the local version of the variable is present.

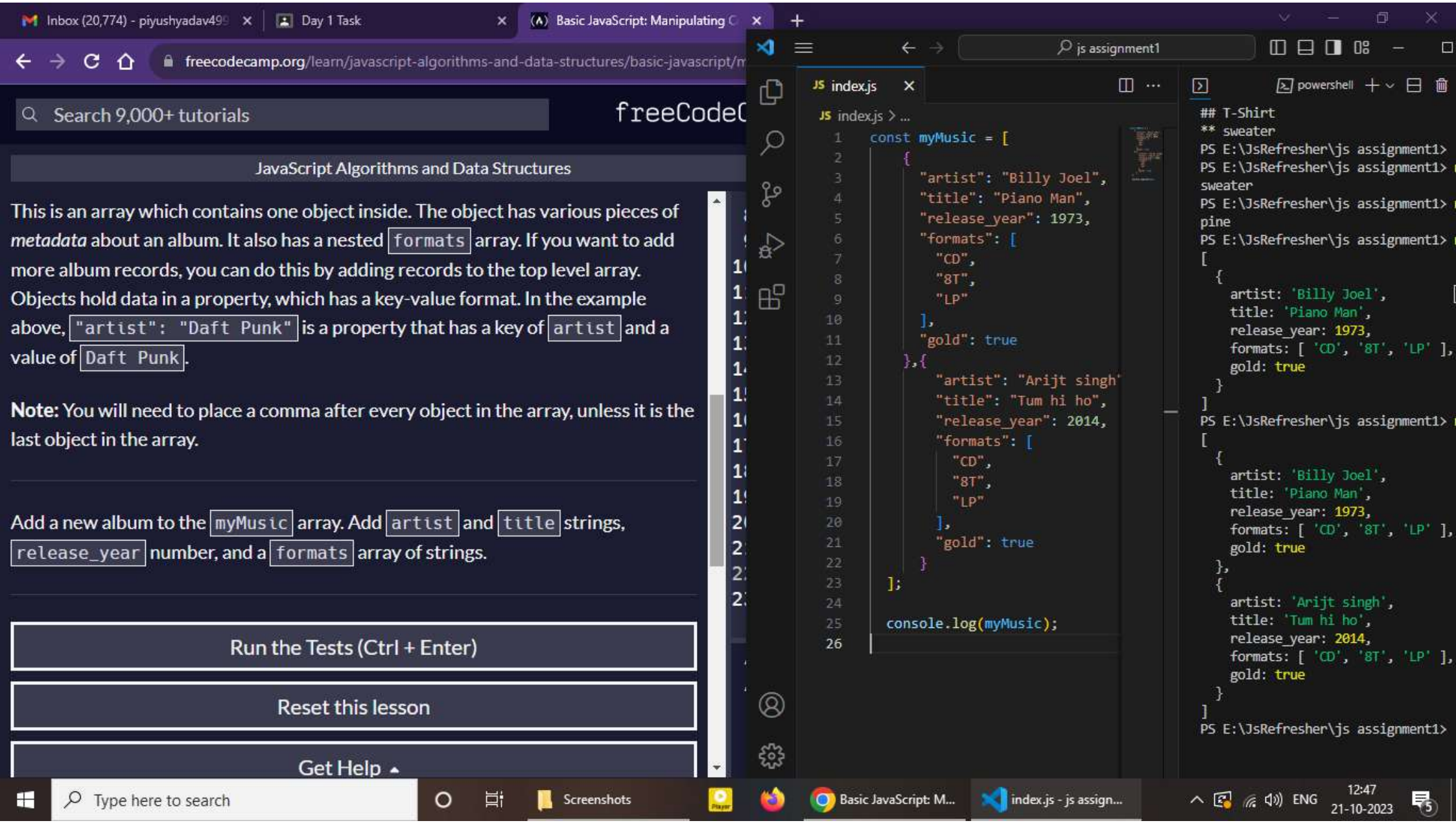
Add a local variable to `myOutfit` function to override the value of `outerWear` with the string `sweater`.

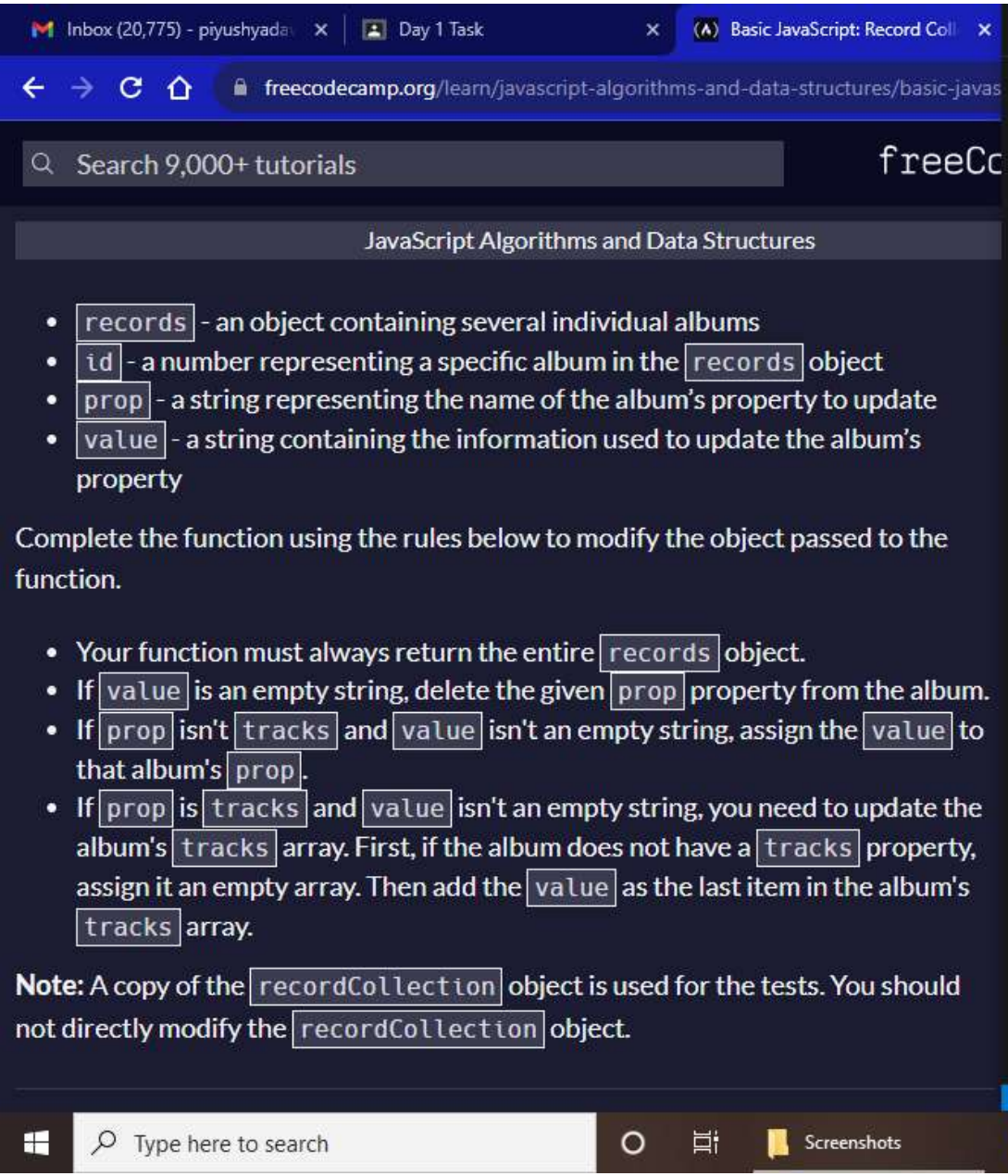
Run the Tests (Ctrl + Enter)

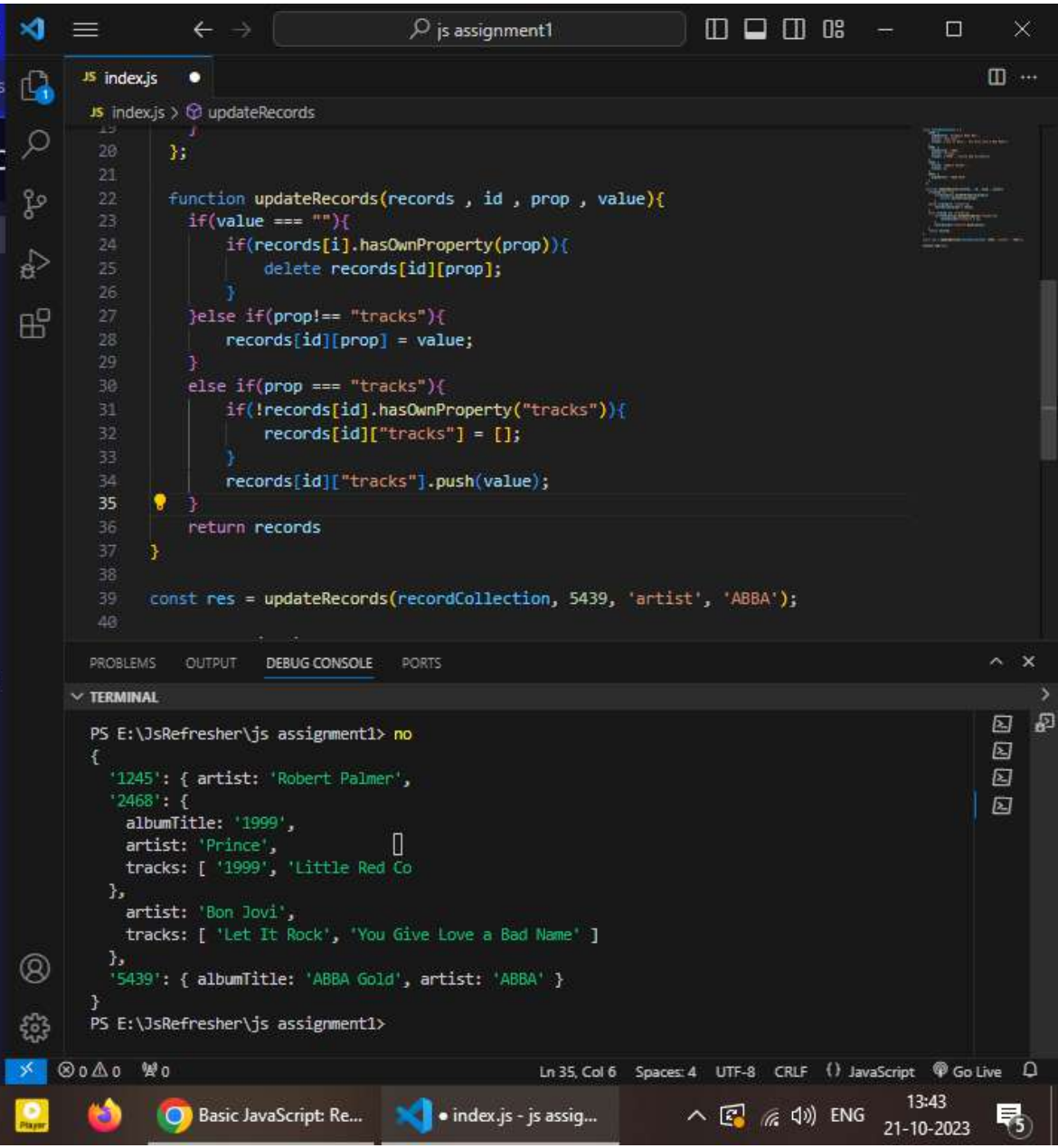


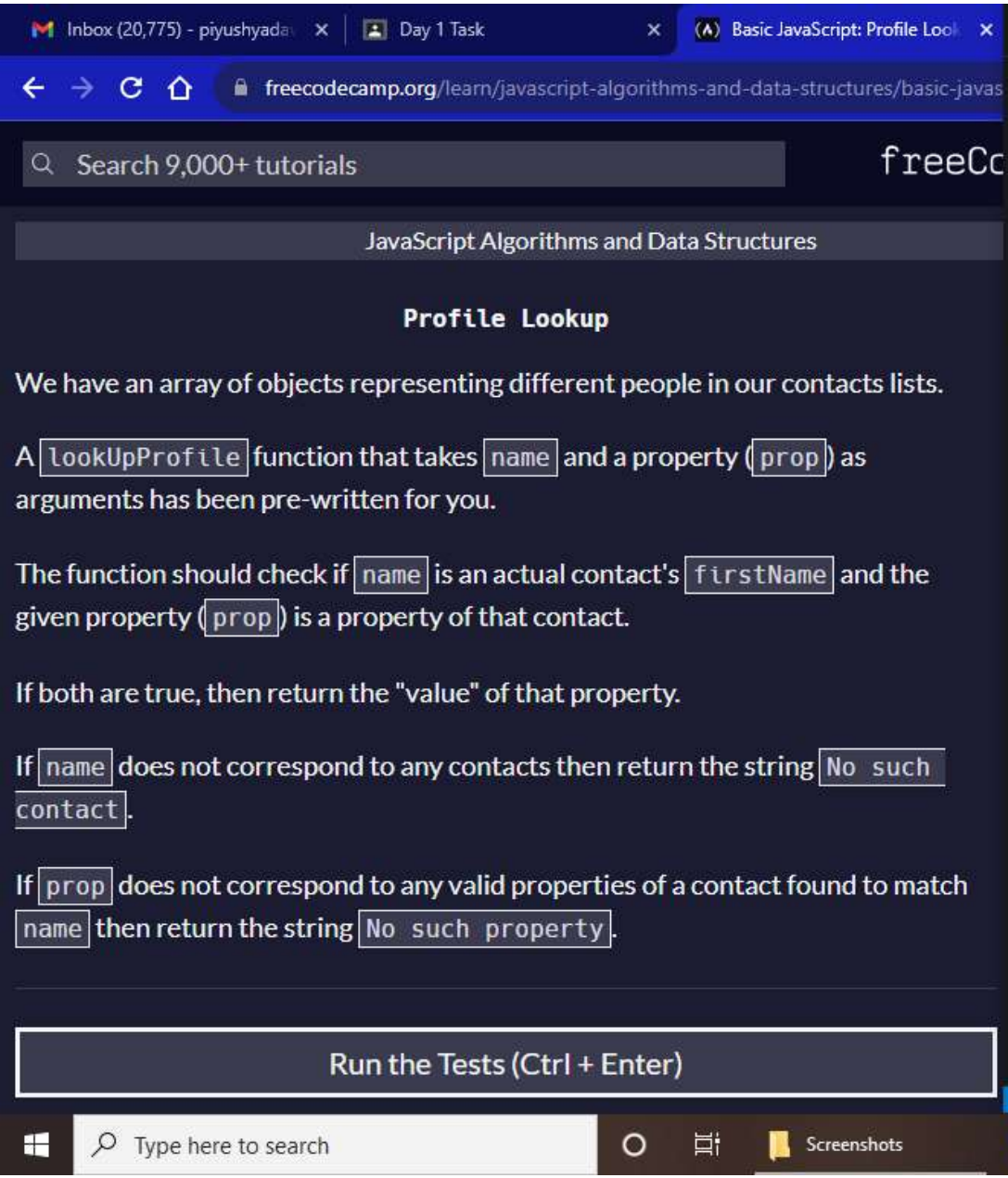
Type here to search | Screenshots | Basic JavaScript: Gl... | index.js - js assign... | ENG | 11:54 | 21-10-2023

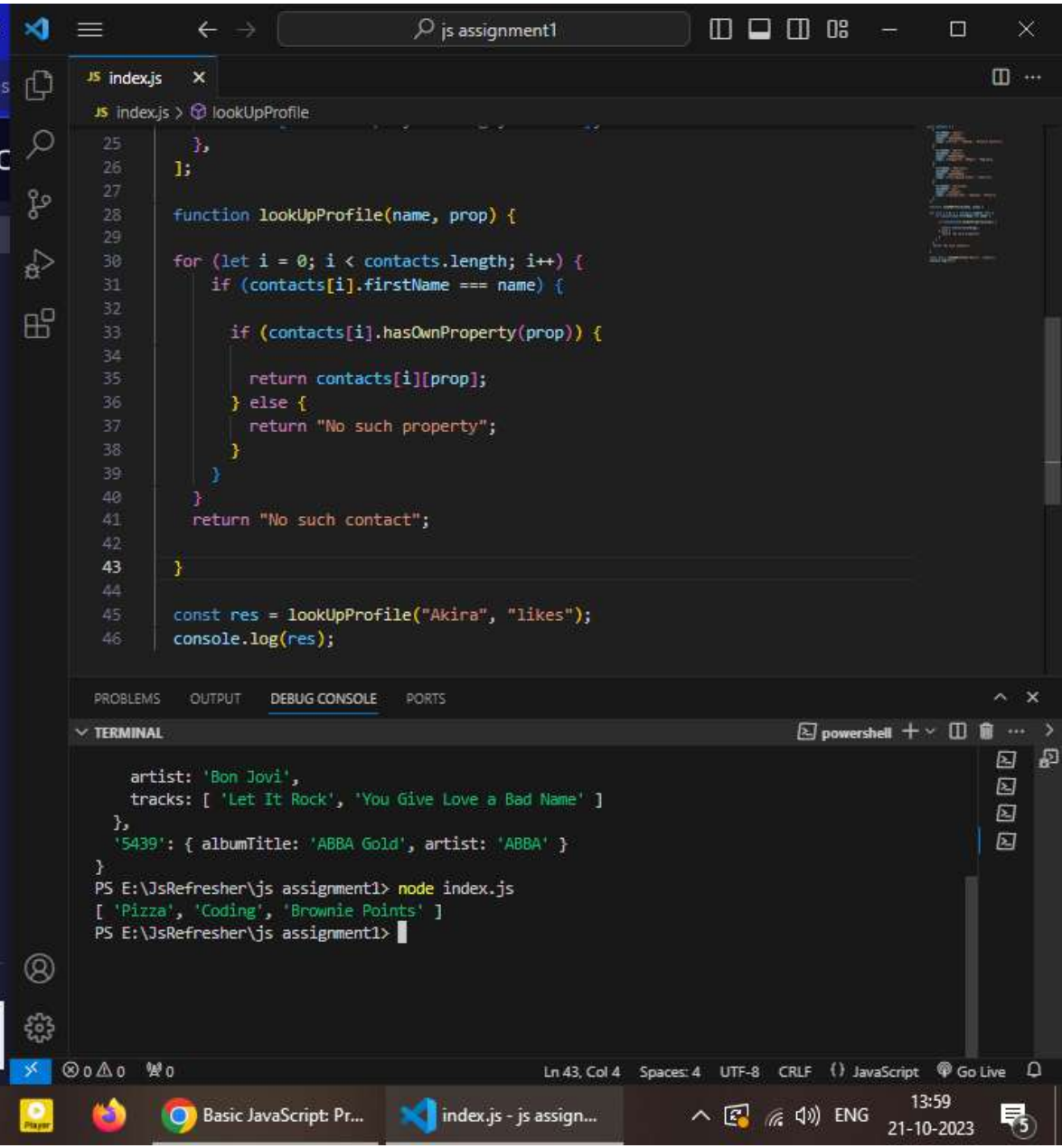

Inbox (20,773) - piyushyadav49
Day 1 Task
Basic JavaScript: Accessing Nest
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/a
Search 9,000+ tutorials
freeCodeC
JavaScript Algorithms and Data Structures
ourPets[0].names[1] would be the string Fluffy, and
ourPets[1].names[0] would be the string Spot.
Using dot and bracket notation, set the variable secondTree to the second item in the trees list from the myPlants object.
Run the Tests (Ctrl + Enter)
Reset this lesson
Get Help
Tests
secondTree should equal the string pine.
Your code should use dot and bracket notation to access myPlants.
js assignment1
index.js
index.js > ...
1 const myPlants = [
2 {
3 type: "flowers",
4 list: [
5 "rose",
6 "tulip",
7 "dandelion"
8]
9 },
10 {
11 type: "trees",
12 list: [
13 "fir",
14 "pine",
15 "birch"
16]
17 }
18];
19
20 const secondTree = myPlants[1].list[1];
21 console.log(secondTree);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell
PS E:\JsRefresher\js assignment1> node index.js
T-Shirt
** sweater
PS E:\JsRefresher\js assignment1>
PS E:\JsRefresher\js assignment1> node index.js
sweater
PS E:\JsRefresher\js assignment1> node index.js
pine
PS E:\JsRefresher\js assignment1>
Ln 21, Col 25
Spaces: 4
UTF-8
CRLF
JavaScript
Type here to search
Screenshots
Basic JavaScript: Ac...
index.js - js assign...
ENG
12:36
21-10-2023


Inbox (20,774) - piyushyadav49
Day 1 Task
Basic JavaScript: Manipulating C
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/m
Search 9,000+ tutorials
freeCodeC
JavaScript Algorithms and Data Structures
This is an array which contains one object inside. The object has various pieces of metadata about an album. It also has a nested formats array. If you want to add more album records, you can do this by adding records to the top level array. Objects hold data in a property, which has a key-value format. In the example above, "artist": "Daft Punk" is a property that has a key of artist and a value of Daft Punk.
Note: You will need to place a comma after every object in the array, unless it is the last object in the array.
Add a new album to the myMusic array. Add artist and title strings, release_year number, and a formats array of strings.
Run the Tests (Ctrl + Enter)
Reset this lesson
Get Help
index.js
const myMusic = [
{
"artist": "Billy Joel",
"title": "Piano Man",
"release_year": 1973,
"formats": [
"CD",
"8T",
"LP"
],
"gold": true
},{
"artist": "Arijt singh",
"title": "Tum hi ho",
"release_year": 2014,
"formats": [
"CD",
"8T",
"LP"
],
"gold": true
}
];
console.log(myMusic);
powershell
T-Shirt
** sweater
PS E:\JsRefresher\js assignment1>
sweater
pine
artist: 'Billy Joel',
title: 'Piano Man',
release_year: 1973,
formats: ['CD', '8T', 'LP'],
gold: true
artist: 'Arijt singh',
title: 'Tum hi ho',
release_year: 2014,
Type here to search
Screenshots
Basic JavaScript: M...
index.js - js assign...
ENG
12:47
21-10-2023

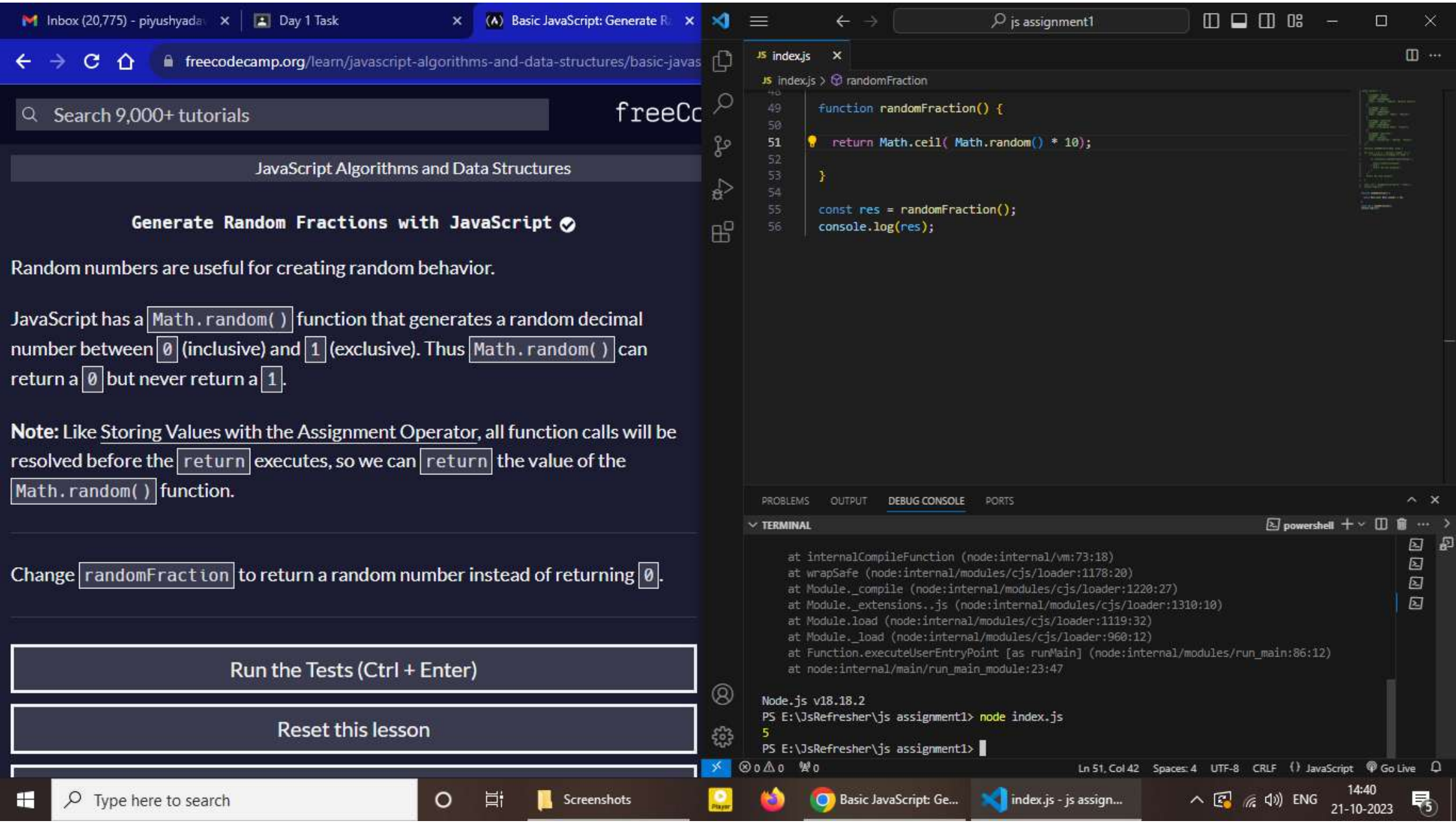

Inbox (20,775) - piyushyada
Day 1 Task
Basic JavaScript: Record Coll
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javas
Search 9,000+ tutorials
freeCo
JavaScript Algorithms and Data Structures
records - an object containing several individual albums
id - a number representing a specific album in the records object
prop - a string representing the name of the album's property to update
value - a string containing the information used to update the album's property
Complete the function using the rules below to modify the object passed to the function.
Your function must always return the entire records object.
If value is an empty string, delete the given prop property from the album.
If prop isn't tracks and value isn't an empty string, assign the value to that album's prop.
If prop is tracks and value isn't an empty string, you need to update the album's tracks array. First, if the album does not have a tracks property, assign it an empty array. Then add the value as the last item in the album's tracks array.
Note: A copy of the recordCollection object is used for the tests. You should not directly modify the recordCollection object.

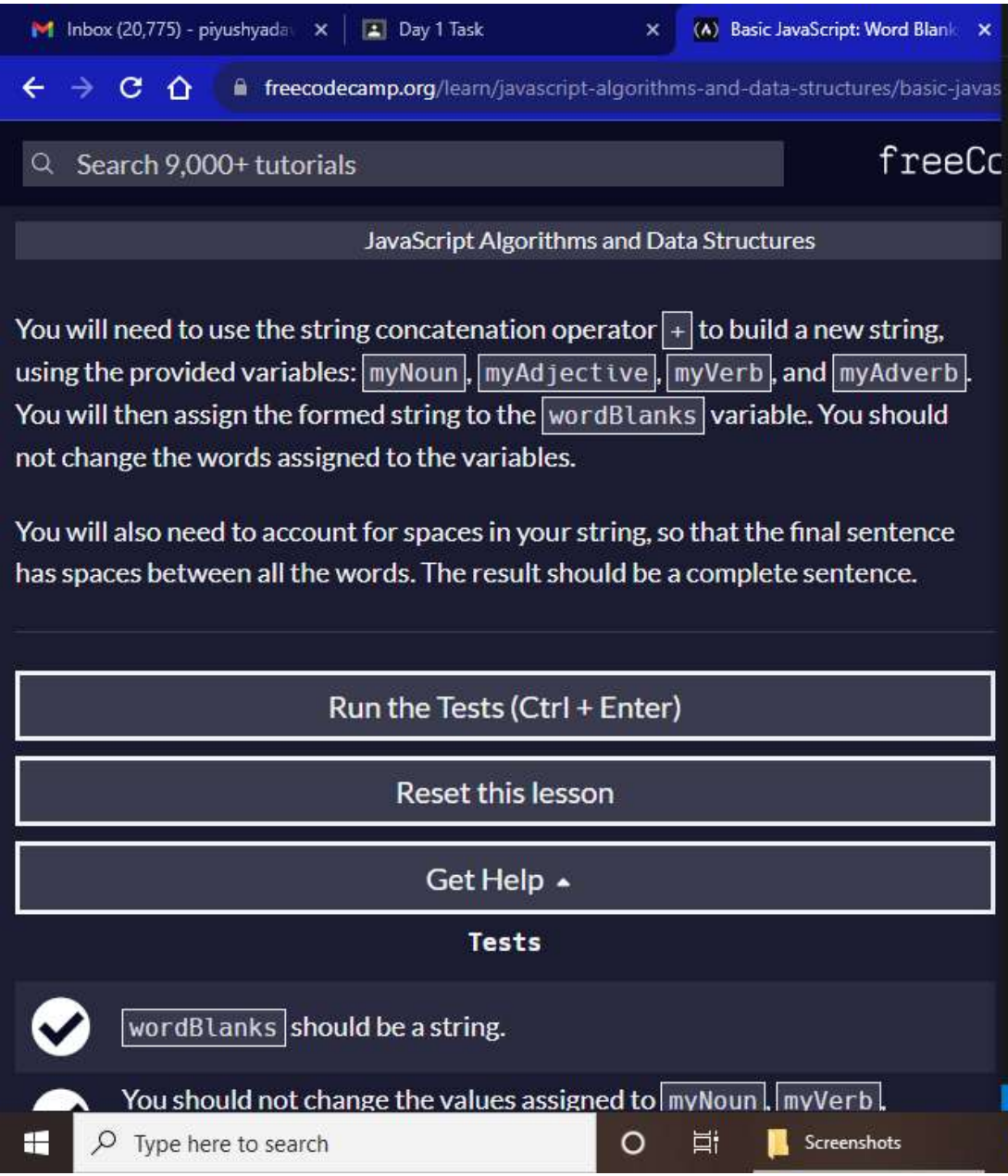

js assignment1
index.js
index.js > updateRecords
};
function updateRecords(records , id , prop , value){
if(value === ""){
if(records[i].hasOwnProperty(prop)){
delete records[id][prop];
}
}else if(prop!== "tracks"){
records[id][prop] = value;
}
else if(prop === "tracks"){
if(!records[id].hasOwnProperty("tracks")){
records[id]["tracks"] = [];
}
records[id]["tracks"].push(value);
}
return records
}
const res = updateRecords(recordCollection, 5439, 'artist', 'ABBA');
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
PS E:\JsRefresher\js assignment1> no
{
'1245': { artist: 'Robert Palmer',
'2468': {
albumTitle: '1999',
artist: 'Prince',
tracks: ['1999', 'Little Red Co
},
artist: 'Bon Jovi',
tracks: ['Let It Rock', 'You Give Love a Bad Name']
},
'5439': { albumTitle: 'ABBA Gold', artist: 'ABBA' }
}
PS E:\JsRefresher\js assignment1>
Ln 35, Col 6
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
Screenshots
Basic JavaScript: Re...
• index.js - js assig...
ENG
13:43
21-10-2023

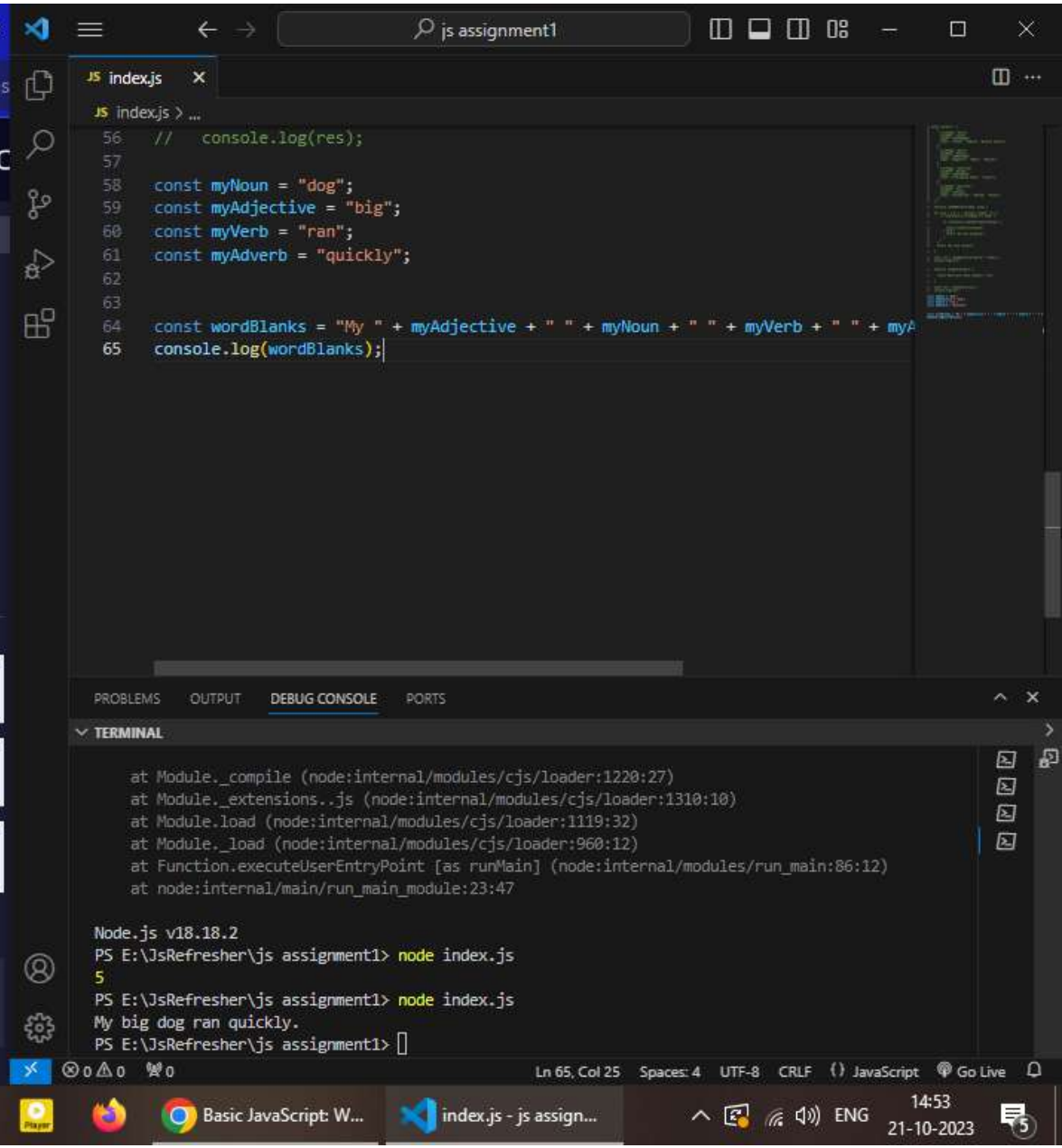

Inbox (20,775) - piyushyada
Day 1 Task
Basic JavaScript: Profile Look
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javas
Search 9,000+ tutorials
freeCo
JavaScript Algorithms and Data Structures
Profile Lookup
We have an array of objects representing different people in our contacts lists.
A lookUpProfile function that takes name and a property (prop) as arguments has been pre-written for you.
The function should check if name is an actual contact's firstName and the given property (prop) is a property of that contact.
If both are true, then return the "value" of that property.
If name does not correspond to any contacts then return the string No such contact.
If prop does not correspond to any valid properties of a contact found to match name then return the string No such property.
Run the Tests (Ctrl + Enter)


js assignment1
index.js
index.js > lookUpProfile
25 },
26];
27
28 function lookUpProfile(name, prop) {
29
30 for (let i = 0; i < contacts.length; i++) {
31 if (contacts[i].firstName === name) {
32
33 if (contacts[i].hasOwnProperty(prop)) {
34
35 return contacts[i][prop];
36 } else {
37 return "No such property";
38 }
39 }
40 }
41 return "No such contact";
42
43 }
44
45 const res = lookUpProfile("Akira", "likes");
46 console.log(res);
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
powershell
artist: 'Bon Jovi',
tracks: ['Let It Rock', 'You Give Love a Bad Name']
},
'5439': { albumTitle: 'ABBA Gold', artist: 'ABBA' }
}
PS E:\JsRefresher\js assignment1> node index.js
['Pizza', 'Coding', 'Brownie Points']
PS E:\JsRefresher\js assignment1>
Ln 43, Col 4
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live

Type here to search
Screenshots
Basic JavaScript: Pr...
index.js - js assign...
ENG
13:59
21-10-2023


Inbox (20,775) - piyushyada
Day 1 Task
Basic JavaScript: Generate R
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javas
Search 9,000+ tutorials
JavaScript Algorithms and Data Structures
Generate Random Fractions with JavaScript
Random numbers are useful for creating random behavior.
JavaScript has a Math.random() function that generates a random decimal number between 0 (inclusive) and 1 (exclusive). Thus Math.random() can return a 0 but never return a 1.
Note: Like Storing Values with the Assignment Operator, all function calls will be resolved before the return executes, so we can return the value of the Math.random() function.
Change randomFraction to return a random number instead of returning 0.
Run the Tests (Ctrl + Enter)
Reset this lesson
js assignment1
index.js
index.js > randomFraction
function randomFraction() {
return Math.ceil(Math.random() * 10);
}
const res = randomFraction();
console.log(res);
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
powershell
at internalCompileFunction (node:internal/vm:73:18)
at wrapSafe (node:internal/modules/cjs/loader:1178:20)
at Module._compile (node:internal/modules/cjs/loader:1220:27)
at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
at Module.load (node:internal/modules/cjs/loader:1119:32)
at Module._load (node:internal/modules/cjs/loader:960:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:86:12)
at node:internal/main/run_main_module:23:47
Node.js v18.18.2
PS E:\JsRefresher\js assignment1> node index.js
5
PS E:\JsRefresher\js assignment1>
Ln 51, Col 42
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Type here to search
Screenshots
Basic JavaScript: Ge...
index.js - js assign...
ENG
14:40
21-10-2023


Inbox (20,775) - piyushyada
Day 1 Task
Basic JavaScript: Word Blank
freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javas
Search 9,000+ tutorials
freeCo
JavaScript Algorithms and Data Structures
You will need to use the string concatenation operator + to build a new string, using the provided variables: myNoun, myAdjective, myVerb, and myAdverb. You will then assign the formed string to the wordBlanks variable. You should not change the words assigned to the variables.
You will also need to account for spaces in your string, so that the final sentence has spaces between all the words. The result should be a complete sentence.
Run the Tests (Ctrl + Enter)
Reset this lesson
Get Help
Tests
wordBlanks should be a string.
You should not change the values assigned to myNoun, myVerb,


js assignment1
index.js
index.js > ...
56 // console.log(res);
57
58 const myNoun = "dog";
59 const myAdjective = "big";
60 const myVerb = "ran";
61 const myAdverb = "quickly";
62
63
64 const wordBlanks = "My " + myAdjective + " " + myNoun + " " + myVerb + " " + myA
65 console.log(wordBlanks);
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
at Module._compile (node:internal/modules/cjs/loader:1220:27)
at Module._extensions..js (node:internal/modules/cjs/loader:1310:10)
at Module.load (node:internal/modules/cjs/loader:1119:32)
at Module._load (node:internal/modules/cjs/loader:960:12)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:86:12)
at node:internal/main/run_main_module:23:47
Node.js v18.18.2
PS E:\JsRefresher\js assignment1> node index.js
5
PS E:\JsRefresher\js assignment1> node index.js
My big dog ran quickly.
PS E:\JsRefresher\js assignment1>
Ln 65, Col 25
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live

Type here to search
Screenshots
Basic JavaScript: W...
index.js - js assign...
ENG
14:53
21-10-2023

Inbox (20,775) - piyushyada | Day 1 Task | Basic JavaScript: Returning B | Actualizar colección de álbu | js function to round off - Go

freecodecamp.org/learn/javascript-algorithms-and-data-structures/basic-javascript/returning-boolean-values-from-functions

Search 9,000+ tutorials

freeCodeCamp

Menu

JavaScript Algorithms and Data Structures | Basic JavaScript

But there's a better way to do this. Since `===` returns `true` or `false`, we can return the result of the comparison:

```
function isEqual(a, b) {
  return a === b;
}
```

Fix the function `isLess` to remove the `if/else` statements.

Run the Tests (Ctrl + Enter)

Reset this lesson

Get Help

Tests

```
1  function isLess(a, b) {
2    // Only change code below this line
3    return a  < b;
4    // Only change code above this line
5  }
6
7  isLess(10, 15);
```

```
// running tests
// tests completed
```

Type here to search | Screenshots | Basic JavaScript: Re... | index.js - js assign... | ENG | 14:56 | 21-10-2023

```
  }
  else {
    return "b is greater or equal";
  }
}
```

This can be re-written using the conditional operator:

```
function findGreater(a, b) {
  return a > b ? "a is greater" : "b is greater or equal";
}
```

Use the conditional operator in the `checkEqual` function to check if two numbers are equal or not. The function should return either the string `Equal` or the string `Not Equal`.

Run the Tests (Ctrl + Enter)

```
1  function checkEqual(a, b) {
2    return a === b ? "Equal" : "Not Equal";
3  }
4
5  checkEqual(1, 2);
```

```
// running tests
// tests completed
```