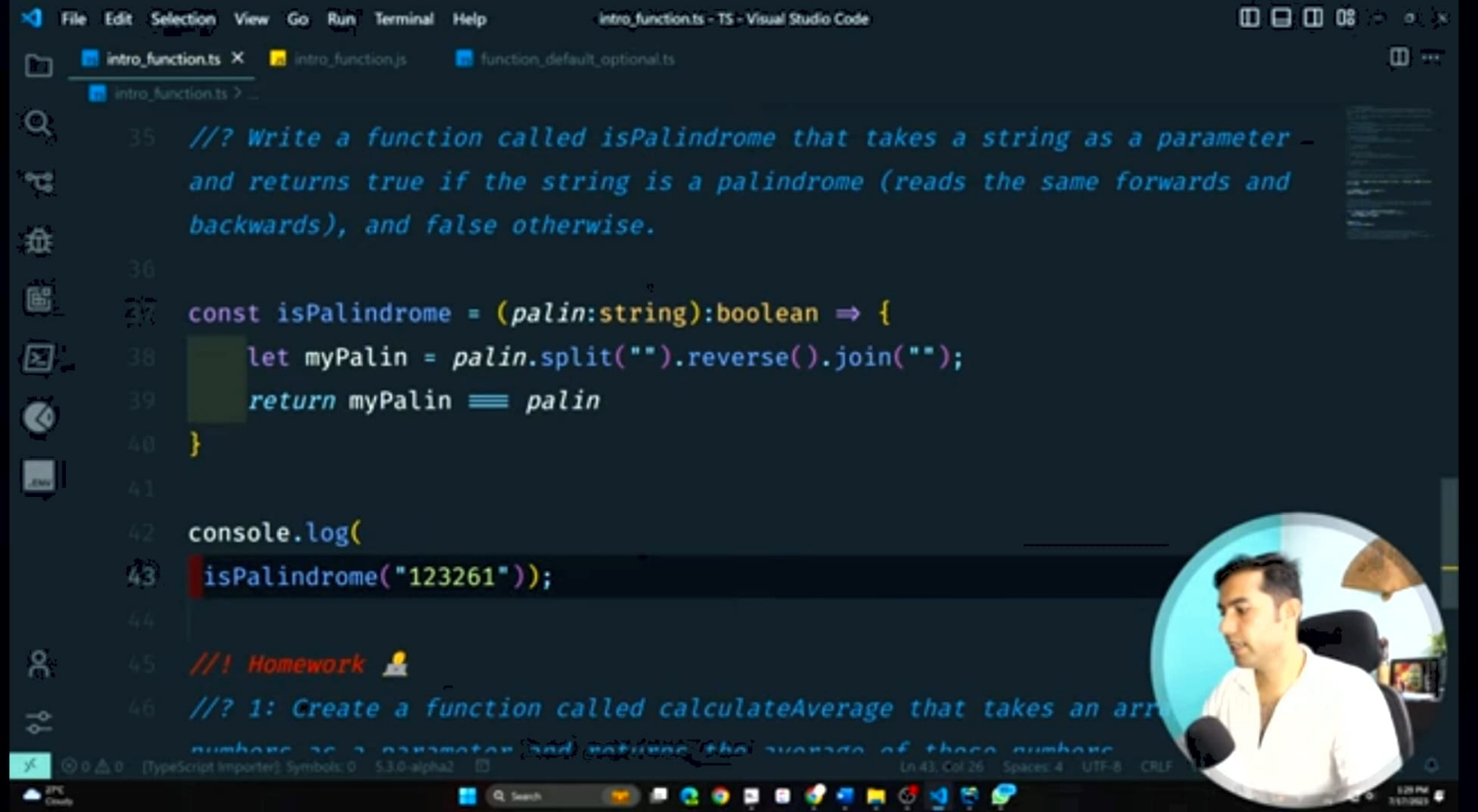

File Edit Selection View Go Run Terminal Help
intro_function.ts - TS - Visual Studio Code
intro_function.ts
intro_function.js
function_default_optional.ts
intro_function.ts > ...
//? Write a function called isPalindrome that takes a string as a parameter and returns true if the string is a palindrome (reads the same forwards and backwards), and false otherwise.
const isPalindrome = (palin:string):boolean => {
let myPalin = palin.split("").reverse().join("");
return myPalin === palin
}
console.log(
isPalindrome("123261"));
//! Homework
//? 1: Create a function called calculateAverage that takes an arr
numbers as a parameter and returns the average of these numbers

File Edit Selection View Go Run Terminal Help • intro_function.ts - TS - Visual Studio Code

intro_function.ts • intro_function.js function_default_optional.ts

intro_function.ts >

```
console.log(
isPalindrome("12321"));

//! Homework
//? 1: Create a function called calculateAverage that takes an array of numbers as a parameter and returns the average of those numbers.

//? 2: Write a function called findMaxValue that takes an array of numbers as a parameter and returns the maximum value in the array.
```

Ln 45, Col 69 (5 selected) Spaces: 4 UTF-8 CRLF

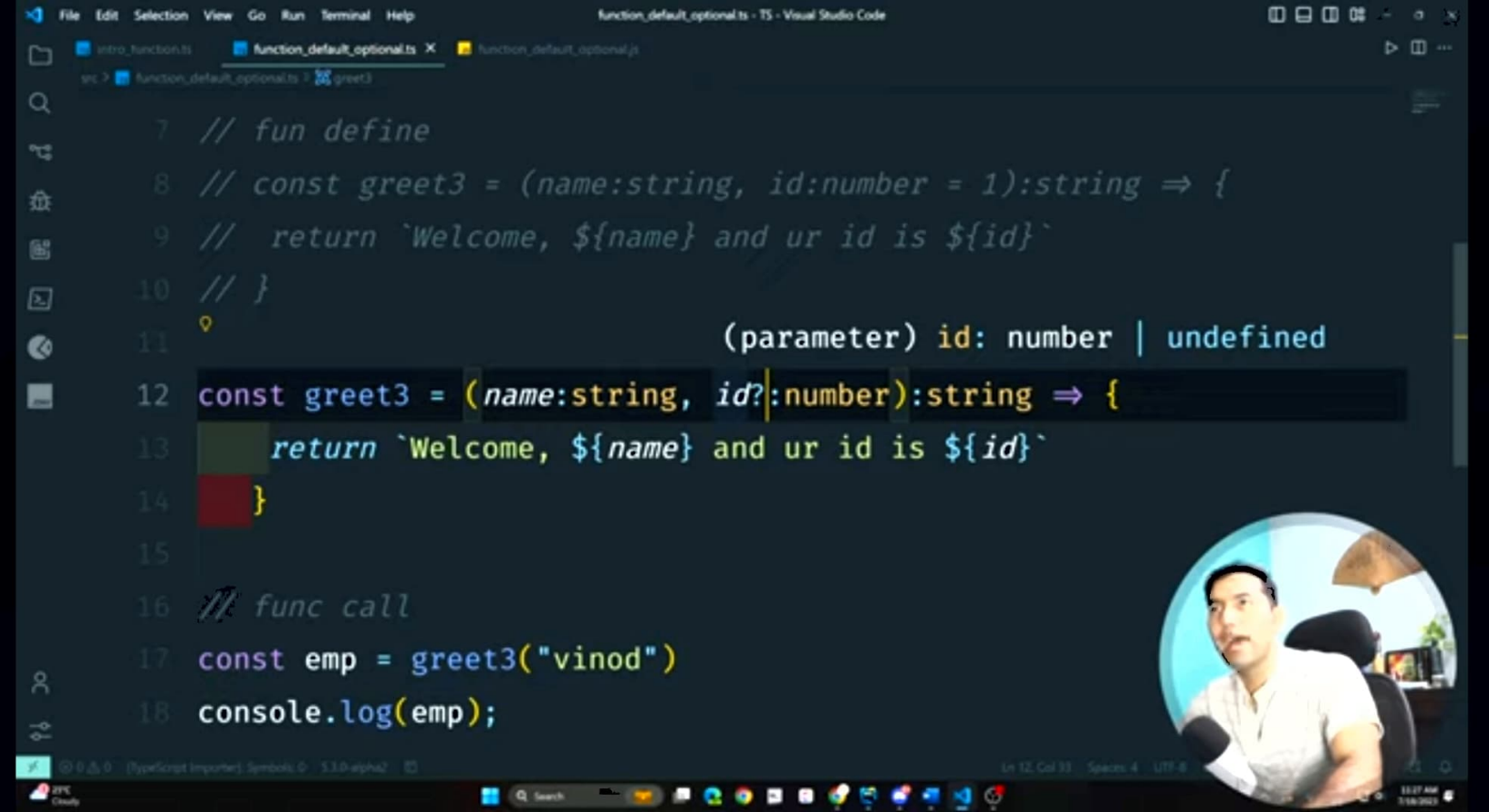
File Edit Selection View Go Run Terminal Help typeinference.ts - TS - Visual Studio Code

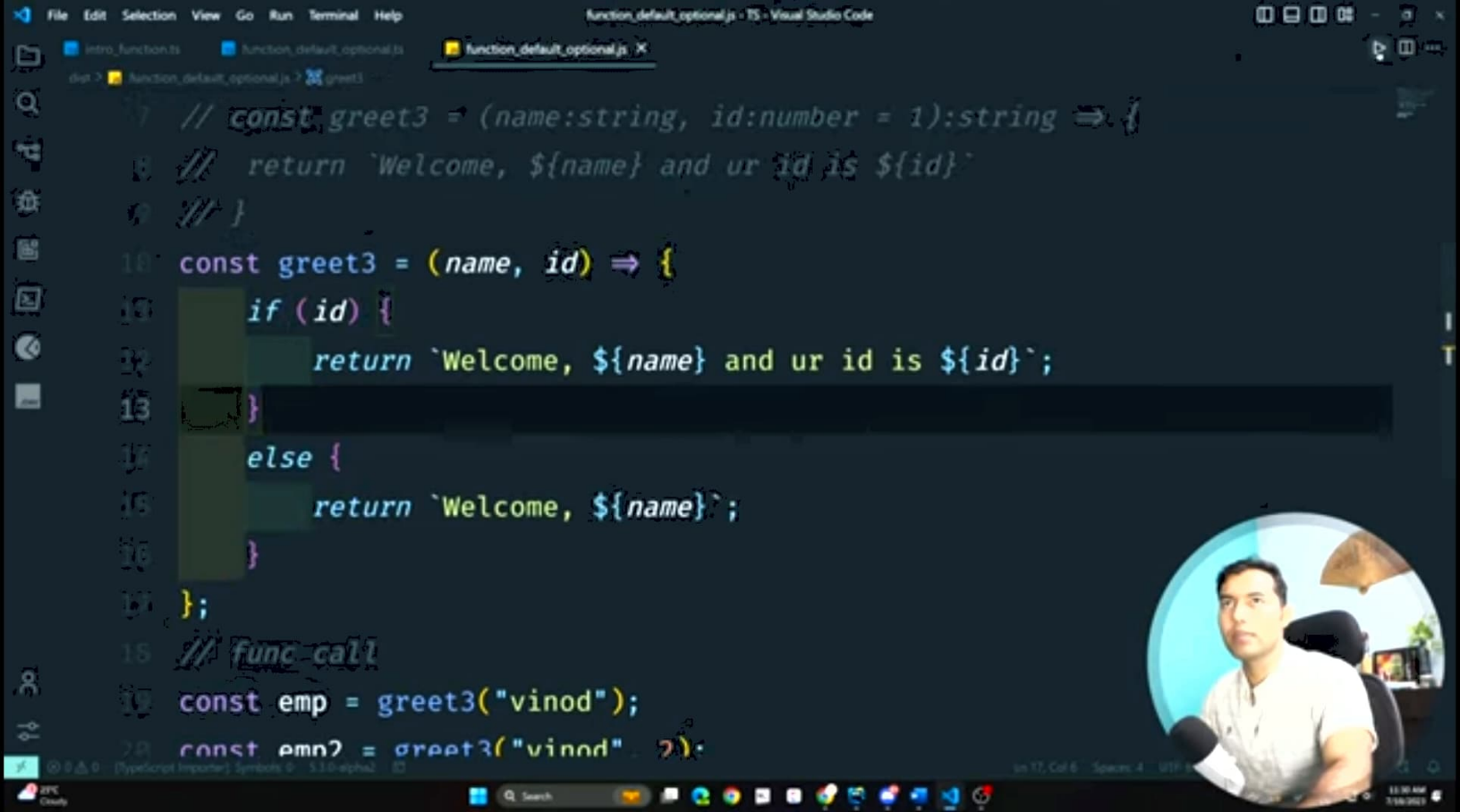
typeinference.ts

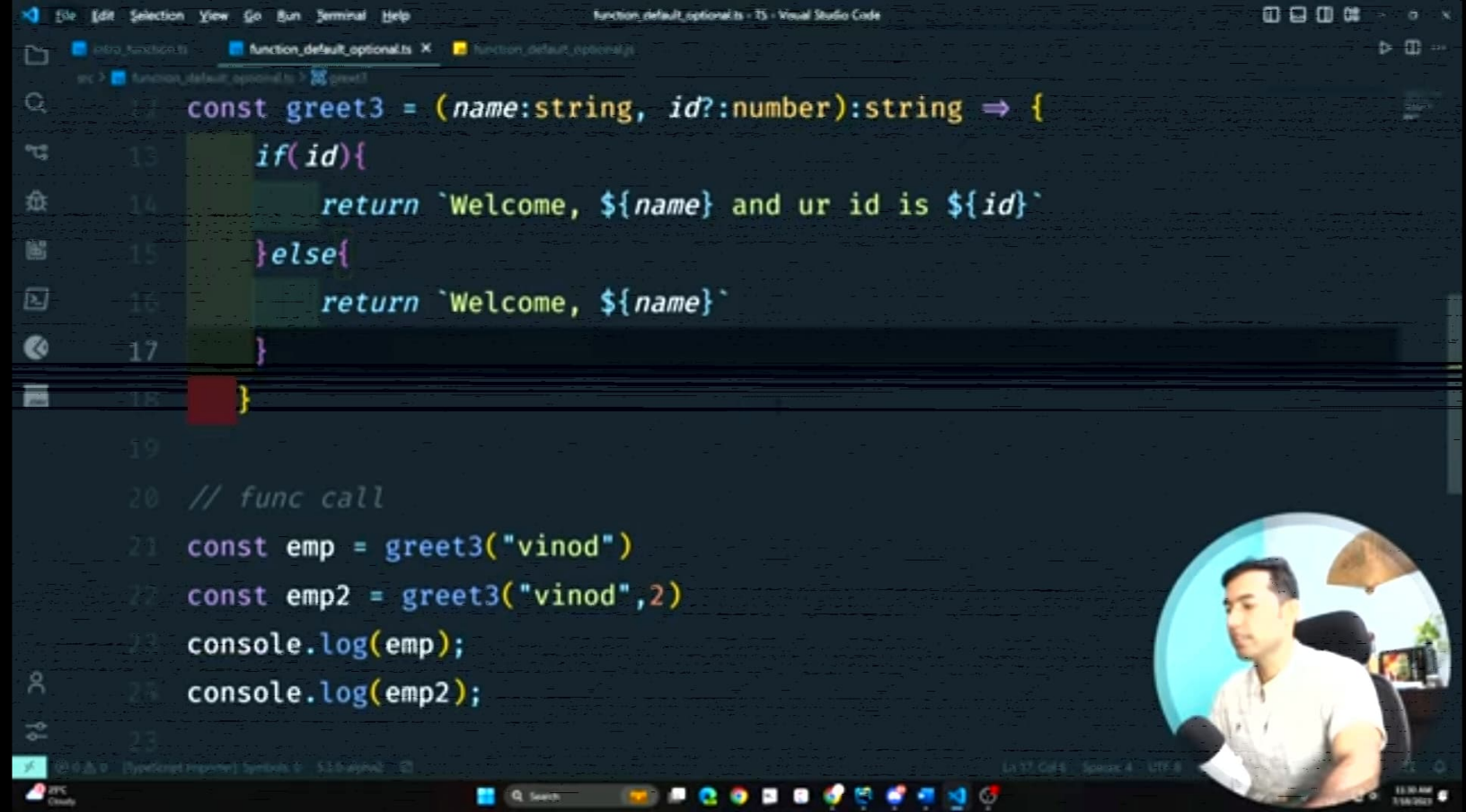
```
//* Type inference in TypeScript refers to the ability of the TypeScript compiler to
automatically determine and assign types to variables, expressions, and function
return values based on their usage and context in the code.

// todo 👉 What are some best practices for using type inference in TypeScript?
//? Use type inference for simple cases where the assigned value clearly indicates the
intended type.
// ? When in doubt, provide explicit type annotations to make your intentions clear.
// ? Avoid relying too heavily on type inference when the assigned value is complex or
ambiguous.
// ? Regularly review and refactor your code to ensure the inferred types align with
your intentions.

//! 🔔 Here are the two questions for practices 🔔
//* 1: Declare a variable message and initialize it with the value "Hello,
TypeScript!". Infer the type of message using type inference.
//* 2: Write a function calculateArea that takes the length and width parame
```


File Edit Selection View Go Run Terminal Help
function_default_optional.ts - TS - Visual Studio Code
intro_function.ts
function_default_optional.ts
function_default_optional.js
src > function_default_optional.ts > greet3
// fun define
// const greet3 = (name:string, id:number = 1):string => {
// return `Welcome, ${name} and ur id is ${id}`
// }
(parameter) id: number | undefined
const greet3 = (name:string, id?:number):string => {
return `Welcome, ${name} and ur id is ${id}`
}
// func call
const emp = greet3("vinod")
console.log(emp);


File Edit Selection View Go Run Terminal Help
function_default_optional.js - TS - Visual Studio Code
intro_function.ts
function_default_optional.ts
function_default_optional.js
dist > function_default_optional.js > greet3
// const greet3 = (name:string, id:number = 1):string => {
// return `Welcome, ${name} and ur id is ${id}`
// }
const greet3 = (name, id) => {
if (id) {
return `Welcome, ${name} and ur id is ${id}`;
}
else {
return `Welcome, ${name}`;
}
};
// func call
const emp = greet3("vinod");
const emp2 = greet3("vinod", 2);


File Edit Selection View Go Run Terminal Help
function_default_optional.ts - TS - Visual Studio Code
intro_function.ts
function_default_optional.ts
function_default_optional.js
src > function_default_optional.ts > greet3
const greet3 = (name:string, id?:number):string => {
if(id){
return `Welcome, ${name} and ur id is ${id}`
}else{
return `Welcome, ${name}`
}
}
// func call
const emp = greet3("vinod")
const emp2 = greet3("vinod",2)
console.log(emp);
console.log(emp2);

File Edit Selection View Go Run Terminal Help

● 10_intro_array.ts - TS - Visual Studio Code

intro_function.ts | 10_intro_array.ts ●

src > 10_intro_array.ts > numbers

```ts
//! Array in TS

//? In TypeScript, you can create and initialize arrays using various approaches

// a) Using square brackets:

const numbers = [1,2,3,4,5,6]

//b) Using the Array constructor:
//c) Using the Array.of method:

//Operation: Accessing elements using index:
```

Ln 7, Col 33 Spaces: 4 UTF-8

File Edit Selection View Go Run Terminal Help

10_intro_array.ts - TS - Visual Studio Code

intro_function.ts | 10_intro_array.ts

src > 10_intro_array.ts > numbers

```ts
//! Array in TS

//? In TypeScript, you can create and initialize arrays using various approaches

// a) Using square brackets:

const numbers:number[] = [1,2,3,4,5,6]

//(b) Using the Array constructor:
//(c) Using the Array.of method:

//Operation: Accessing elements using index:
```

File Edit Selection View Go Run Terminal Help

10_intro_array.ts - TS - Visual Studio Code

intro_function.ts | 10_intro_array.ts

src > 10_intro_array.ts > names

```ts
const numbers:number[] = [1,2,3,4,5,6]

//b) Using the Array constructor:

const numbers1:number[] =  new Array(1,2,3,4,5,6);

//c) Using the Array.of method:
    const names: string[]
const names = Array.of("vinod","thapa","youtube")


//Operation: Accessing elements using index:

//TODO HomeWork Time
```

File Edit Selection View Go Run Terminal Help

10_intro_array.ts - TS - Visual Studio Code

intro_function.ts | 10_intro_array.ts | 10_intro_array.js | 2_array.ts

src > 10_intro_array.ts > ...

```
//TODO HomeWork Time
const value1 = [1, 2, 3, 4, 5];
// // is value1 an array? Yes or No?
const value2 = "Hello, world!";
// // is value2 an array? Yes or No?
const value3 = { name: "John", age: 30 };
// // is value3 an array? Yes or No?
const value4 = [true, false, true];
// // is value4 an array? Yes or No?
const value5 = 42;
// // is value5 an array? Yes or No?
```

File Edit Selection View Go Run Terminal Help

array_operations.ts - TS - Visual Studio Code

array_operations.ts

src > array_operations.ts > ...

```ts
//* Arrays in TypeScript come with built-in methods that allow you to perform common operations

const fruits: string[] = ["apple", "banana", "orange", "mango"];

// a) Adding elements to an array using push:
const newUpdatedFruits = fruits.push("Kiwi")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

powershell

JavaScript Debug Terminal

src tsc

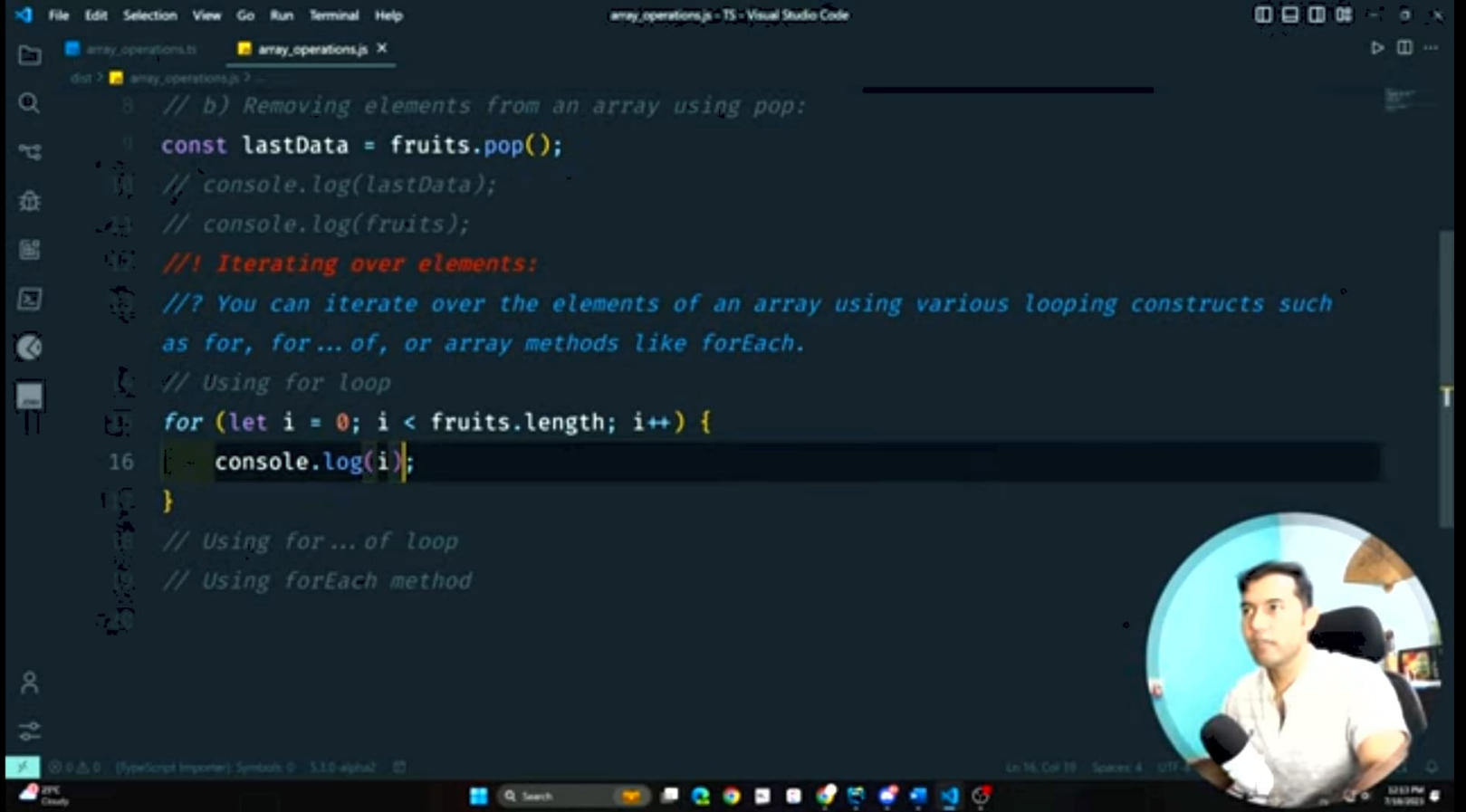
src tsc

src tsc

src tsc

Thapa src

in push at 12:00


File Edit Selection View Go Run Terminal Help
array_operations.js - TS - Visual Studio Code
array_operations.ts
array_operations.js
dist > array_operations.js > ...
// b) Removing elements from an array using pop:
const lastData = fruits.pop();
// console.log(lastData);
// console.log(fruits);
//! Iterating over elements:
//? You can iterate over the elements of an array using various looping constructs such as for, for...of, or array methods like forEach.
// Using for loop
for (let i = 0; i < fruits.length; i++) {
console.log(i);
}
// Using for...of loop
// Using forEach method

```
const arr = [1, 2, 3, 4];

// for...in loop (iterates over indices)
for (const index in arr) {
  console.log(index); // Output: 0, 1, 2, 3  }

// for...of loop (iterates over values)
for (const value of arr) {
  console.log(value); // Output: 1, 2, 3, 4 }
```

File Edit Selection View Go Run Terminal Help

array_operations.ts 2 ● array_operations.js

src > array_operations.ts > ...

```
// for(const fruit of fruits){
//     console.log(fruit);
// }

// Using forEach method

fruits.forEach((curVal) => console.log(curVal)
)
```

Ln 33, Col 46 Spaces: 4 UTF-8