**18BCS045**

**LASHYA P**

**LIST OUT VARIOUS JAVASCRIPT DEBUGGERS:**

Debugging is not easy. But fortunately, all modern browsers have a built-in JavaScript debugger.

Built-in debuggers can be turned on and off, forcing errors to be reported to the user.

With a debugger, you can also set breakpoints (places where code execution can be stopped), and examine variables while the code is executing.

Normally, otherwise, follow the steps at the bottom of this page, you activate debugging in your browser with the F12 key, and select "Console" in the debugger menu.

Here, we will find out errors using a built-in web browser debugger. To perform debugging, we can use any of the following approaches:

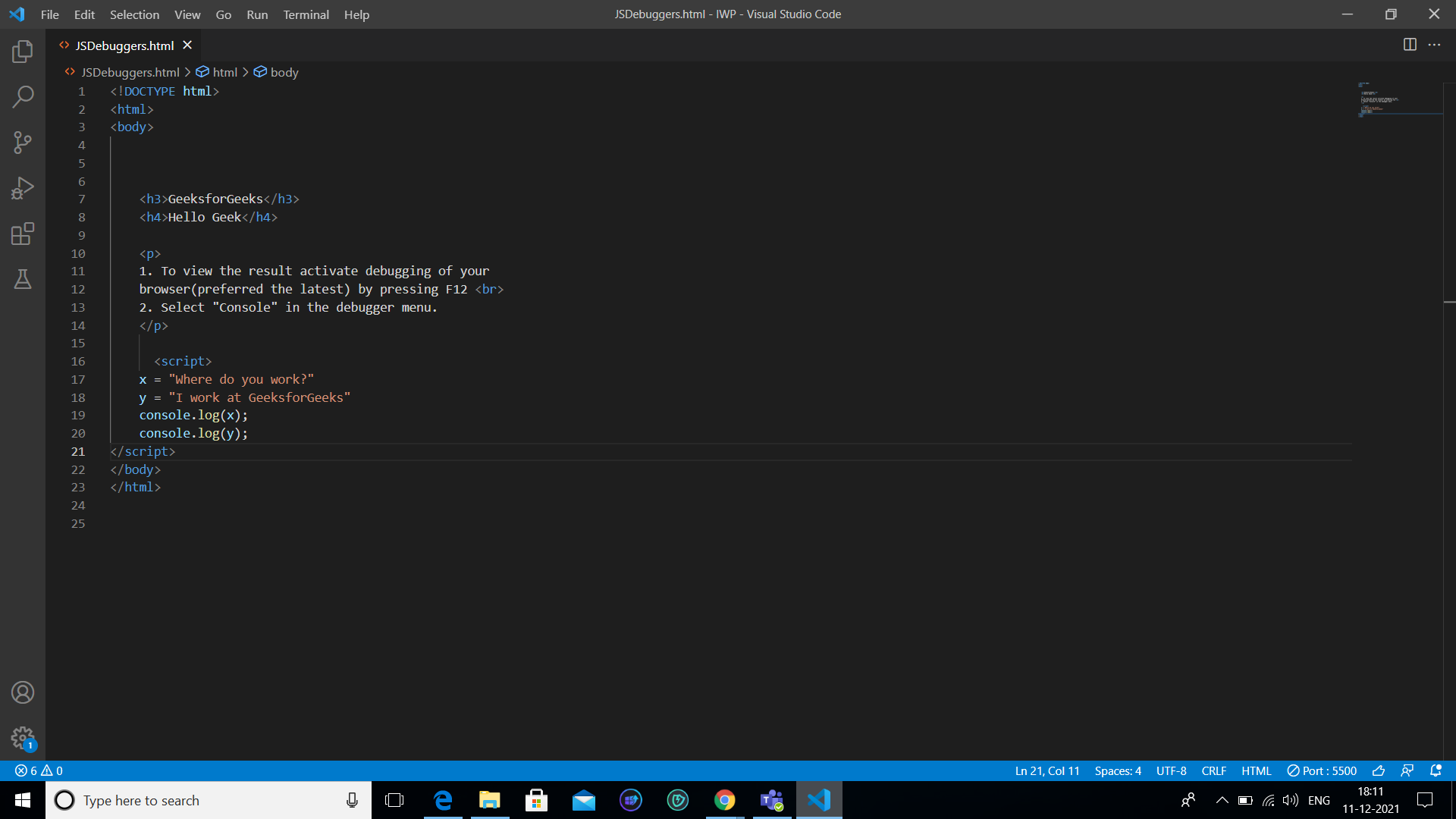
* Using console.log() method
* Using debugger keyword

**USING CONSOLE.LOG() METHOD**

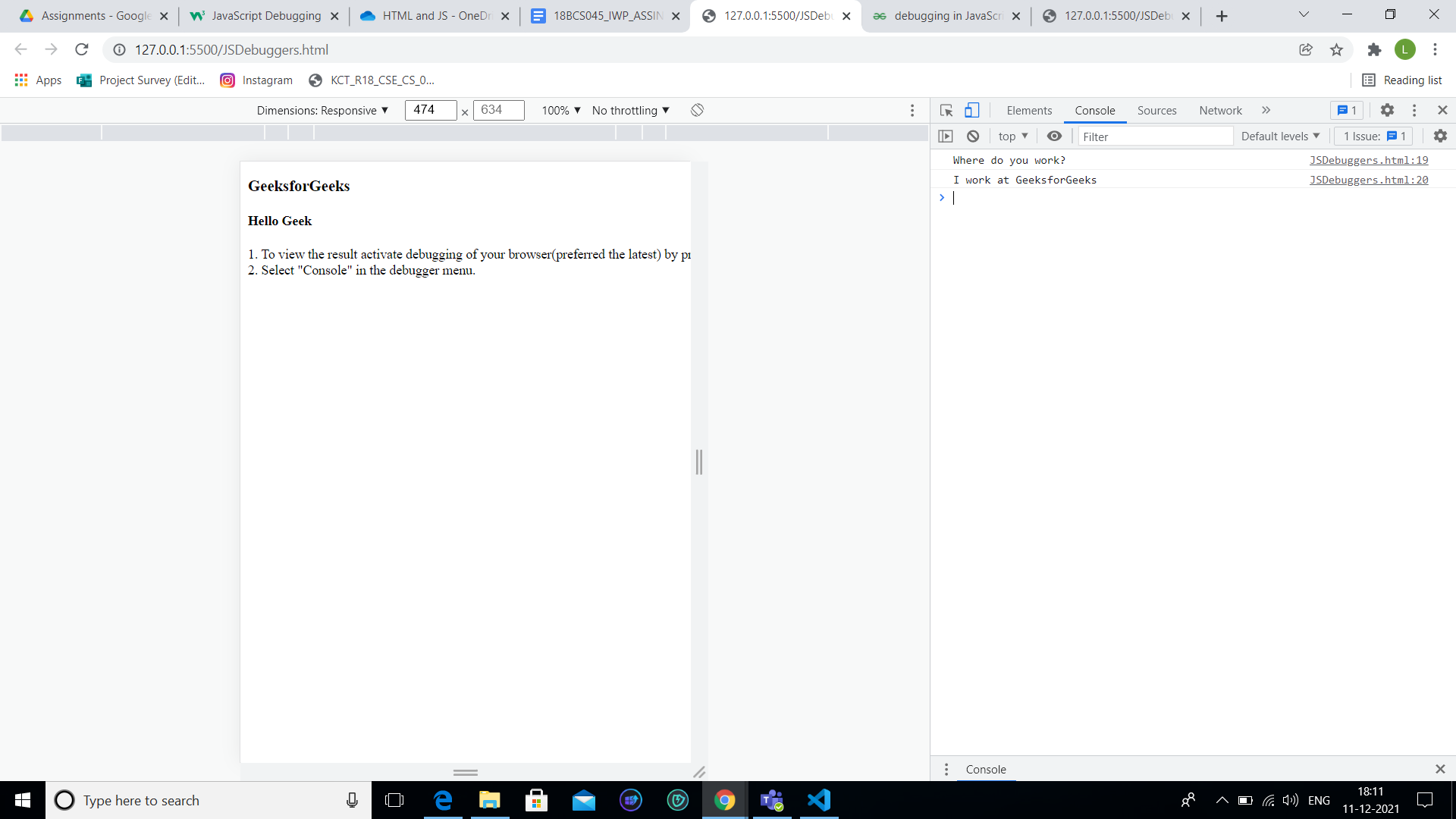
The **console.log()** method displays the result in the console of the browser. If there is any mistake in the code, it generates an error message.

Example:

Code :

****

**Output:**



**THE DEBUGGER KEYWORD:**

The debugger keyword stops the execution of JavaScript, and calls (if available) the debugging function.

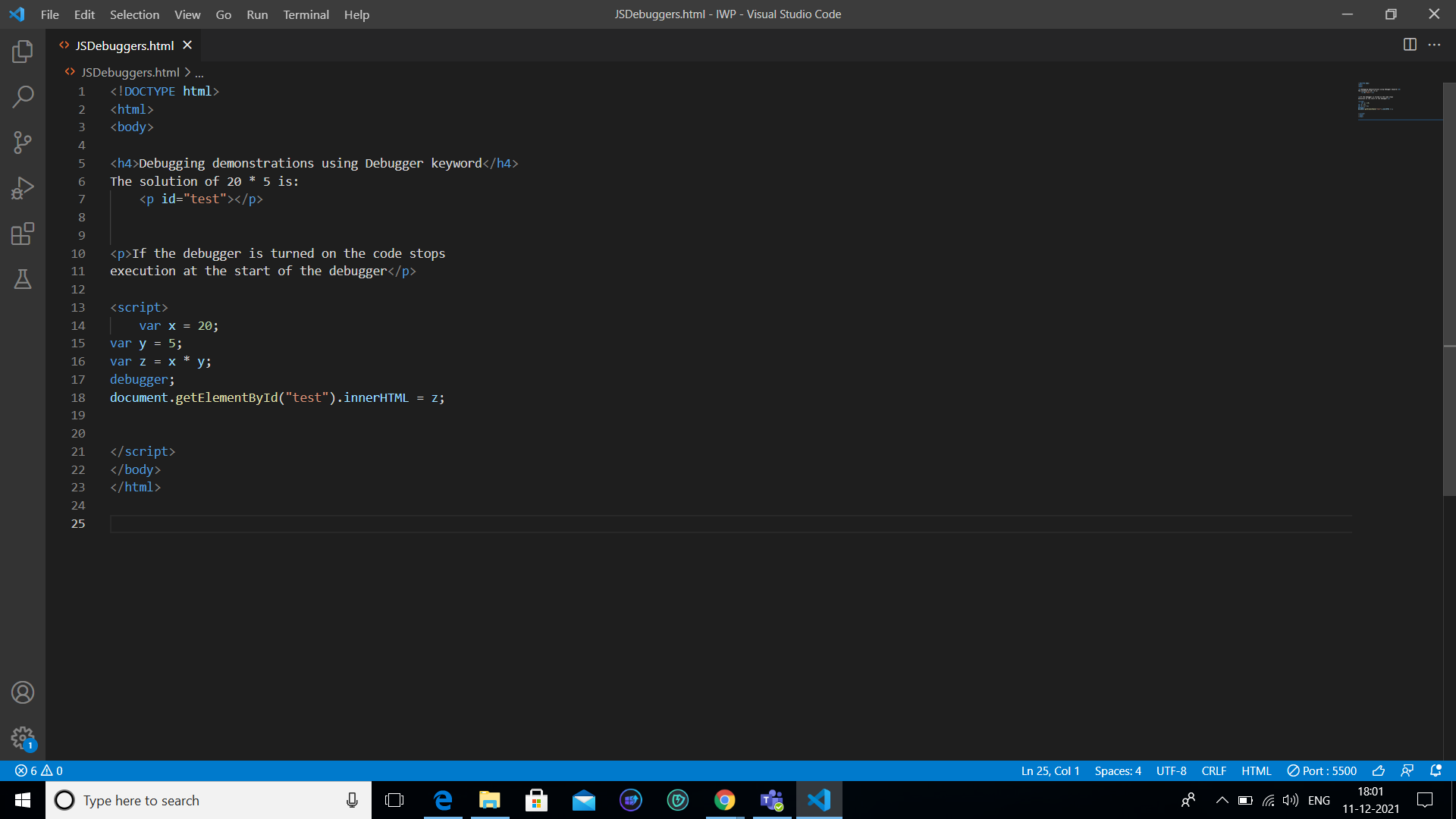
This has the same function as setting a breakpoint in the debugger.

If no debugging is available, the debugger statement has no effect.

With the debugger turned on, this code will stop executing before it executes the third line.

Example:

Code :



Output:

