**JAVASCRIPT Basics**

What is JavaScript?

* Scripting language, low level programming language.
* Dynamically typed language.
* Scripts are basically some functions which tells how to do and what to do?
* Client-side language, which can be run in browser.
* Developed by Netscape (in just 10 days, in 1995), initially Mocha was named.
* Some developer adds this into C++ and it became Node.js so that we can use it outside the browser server – side.

What can we do with JS?

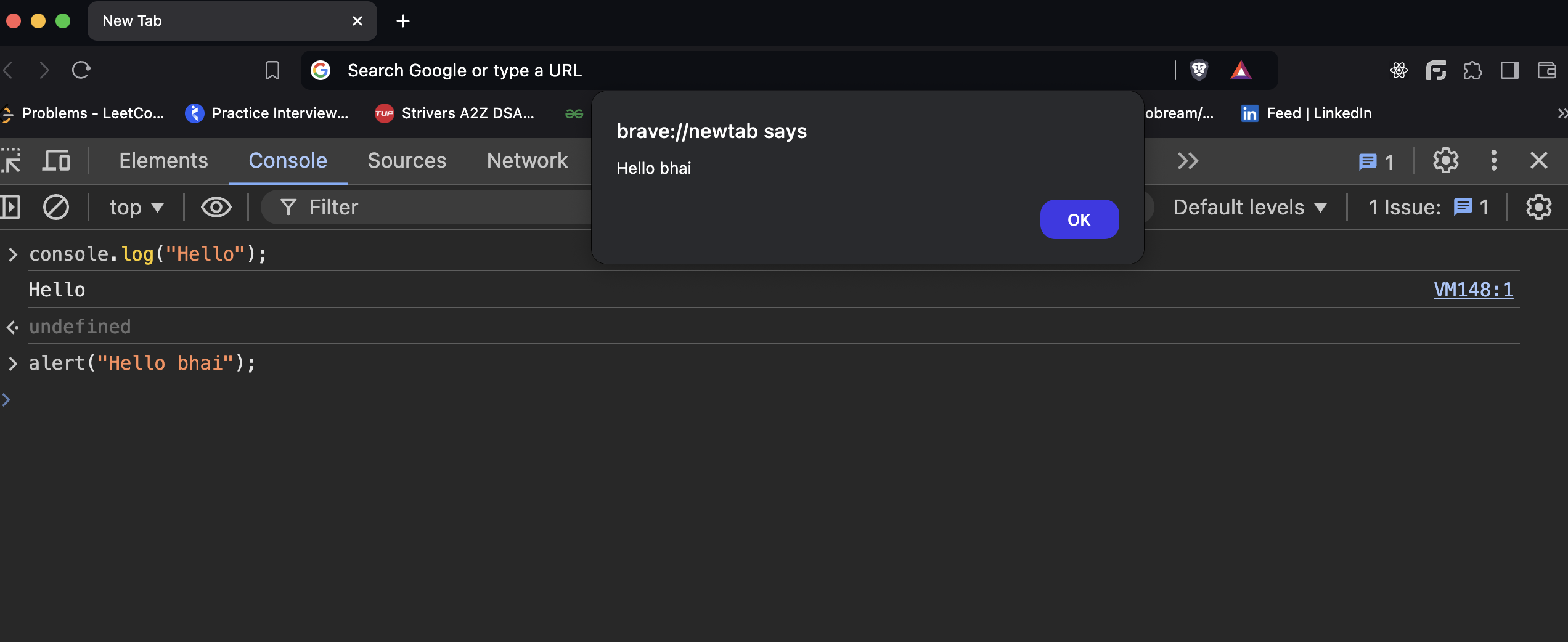
* Webapp / mobile app / network apps
* CLI tools
* Games

Do we need compiler?

* We need only JS Engine.
* Like in firefox Spider Monkey, Google V8.

Simple way to run a JS Code in Browser.

* Open any browser.
* Right click on anywhere on the screen.
* Click on inspect
* Find the console window.

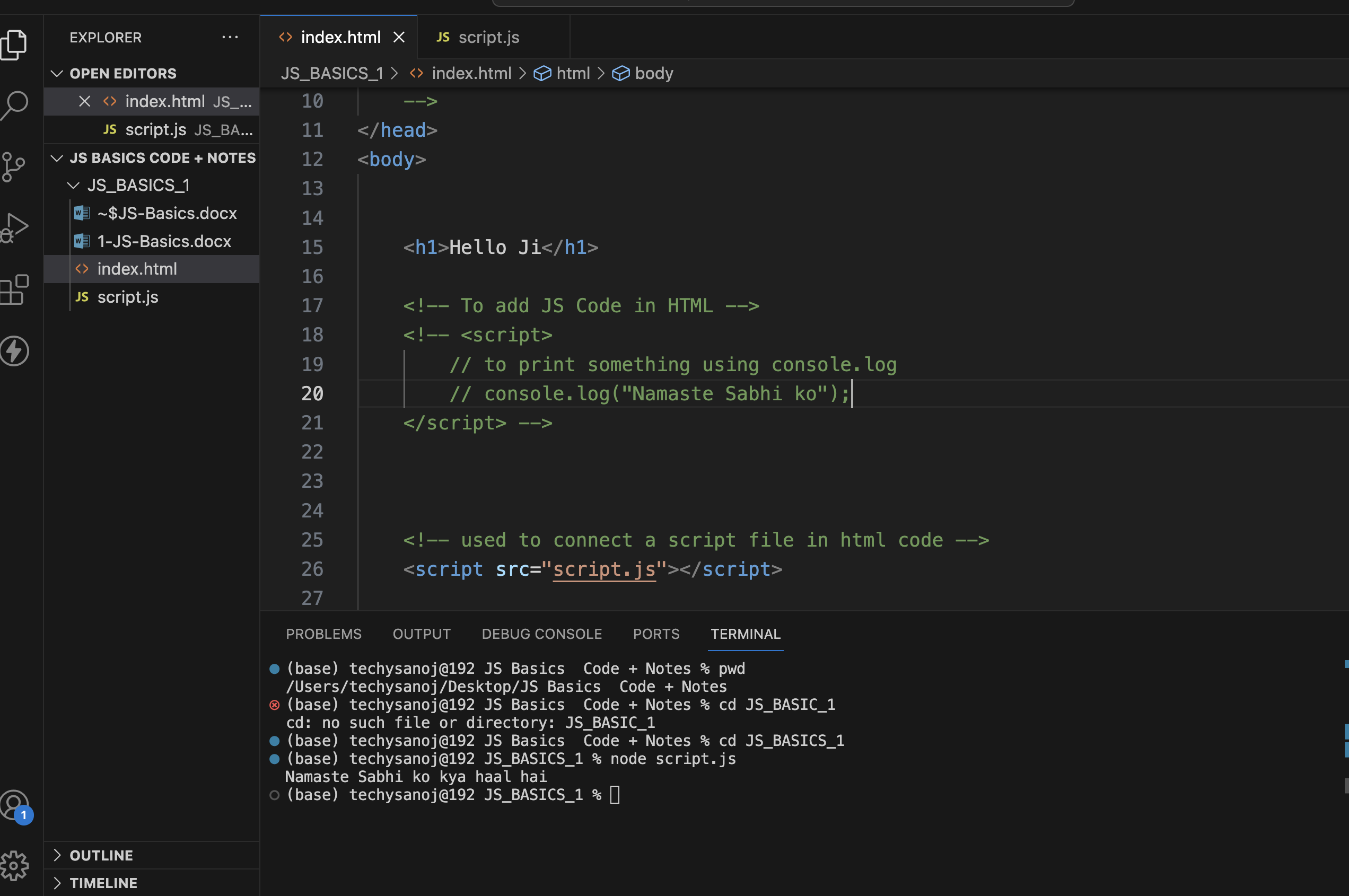


JS Setup?

* VS Code
* Node js

**Adding JS Code in HTML + CSS + File**

* Check file in JS\_BASIC\_1
* String – sequence of characters
* console.log() – use to print something in the console.
* ; - is used to end the line.
* // - is used to add comment in the JS Code.
* Create a separate file for the JS code.
* Lower ss is telling about how we can run code in terminal



**Variable**

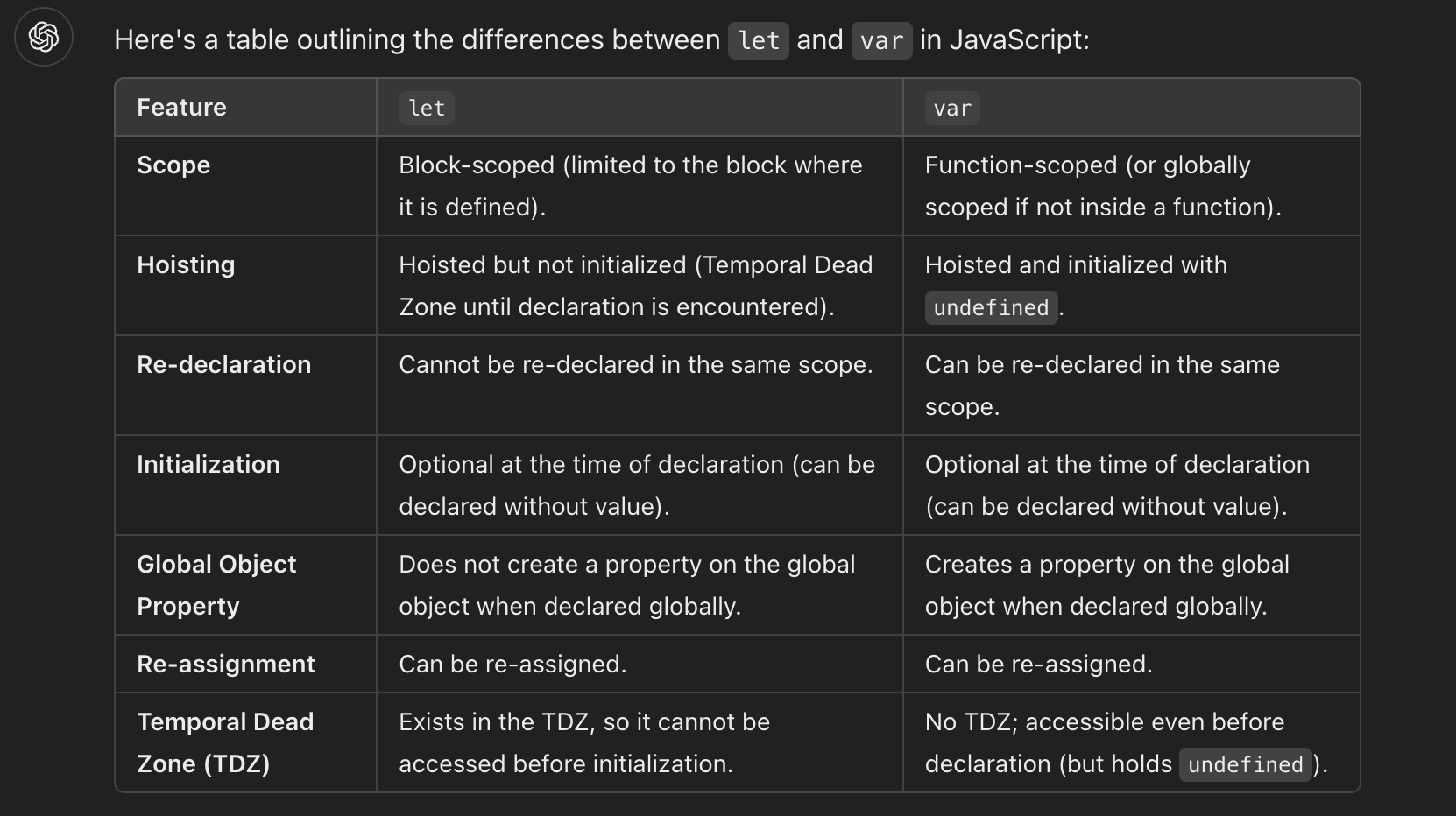
* Named memory location is called as variable
* Can be created using “let” and “var” keyword. Some of the example
  + let a = 5;
  + let name = “Sanoj”;
  + let status = true;
  + var check = false;
  + var name = “Sanoj”;
* Difference between var and let are simple scoping of the variables.
  + let are basically scoped in the closed curly bracket, var are basically global variable.
  + Example:
    - {

let a = 5; // here ‘a’ is only scoped in these curly bracket

var b = 5; // but ‘b’ can be accessed outside these bracket

}

* + let cannot be redeclared but var can be redeclared.



**Constant**

* fixed value which cannot be changed further
* using const keyword

**Variable Naming Convention**

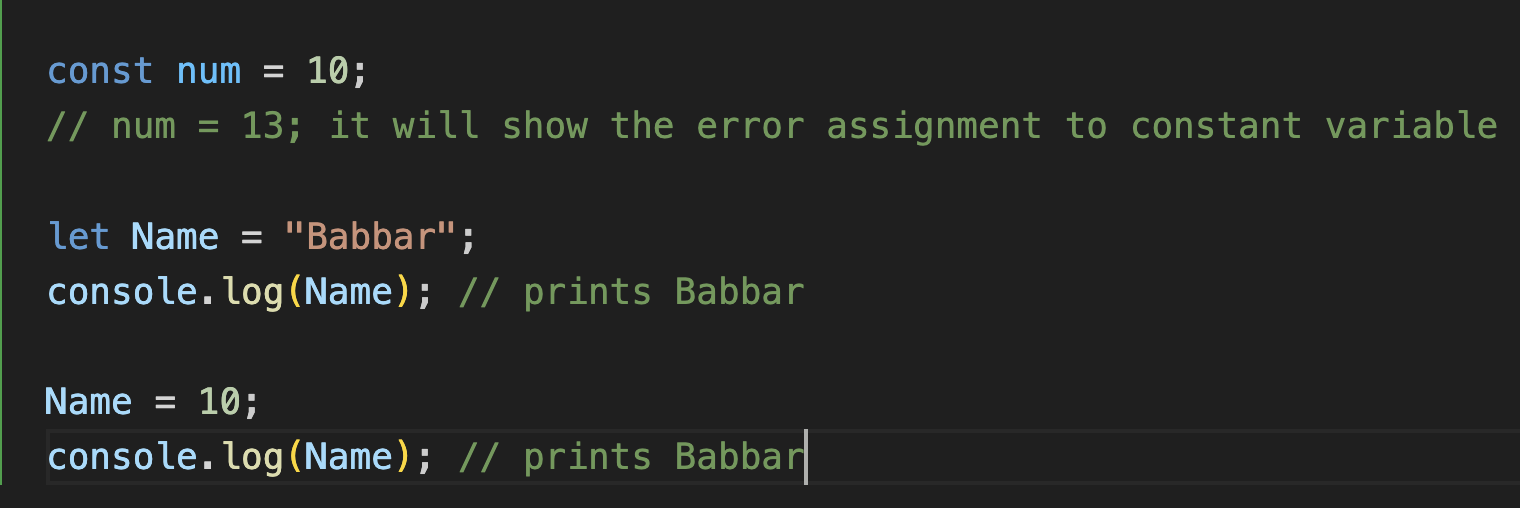
* Cannot be a reserved keyword like we cannot use let as a variable name, if as a variable name etc.
* Name should be meaning full
* Cannot start with a number.
* Cannot contain space or ‘-‘.
* Naming should be done in CamelCase. Example SanojKumar etc.
* Declaring multiple variables let a, b, c; but best practice is used to create all these variables in different lines.

**Primitive Types**

* String: Sequence of characters example “sanoj” etc.
* Number: any number 1, 2, 3, 1.4, 1200, etc.
* Boolean: True, False.
* Undefined: Which is declared but not defined. Example let a; here a is undefined.
* Null: Value is declared and defined but the value is NULL means no value is present.

**Dynamic Typing**

* Means at the runtime we can change the type of the variable.
* Increases the flexibility.



**Reference Data Types**

* Objects: Any element having properties and behavior. Like Person having name, age as properties
* Key – Value pairs are basically stored in object.
  + let person = {

firstName: “Sanoj”,

age: 24

};

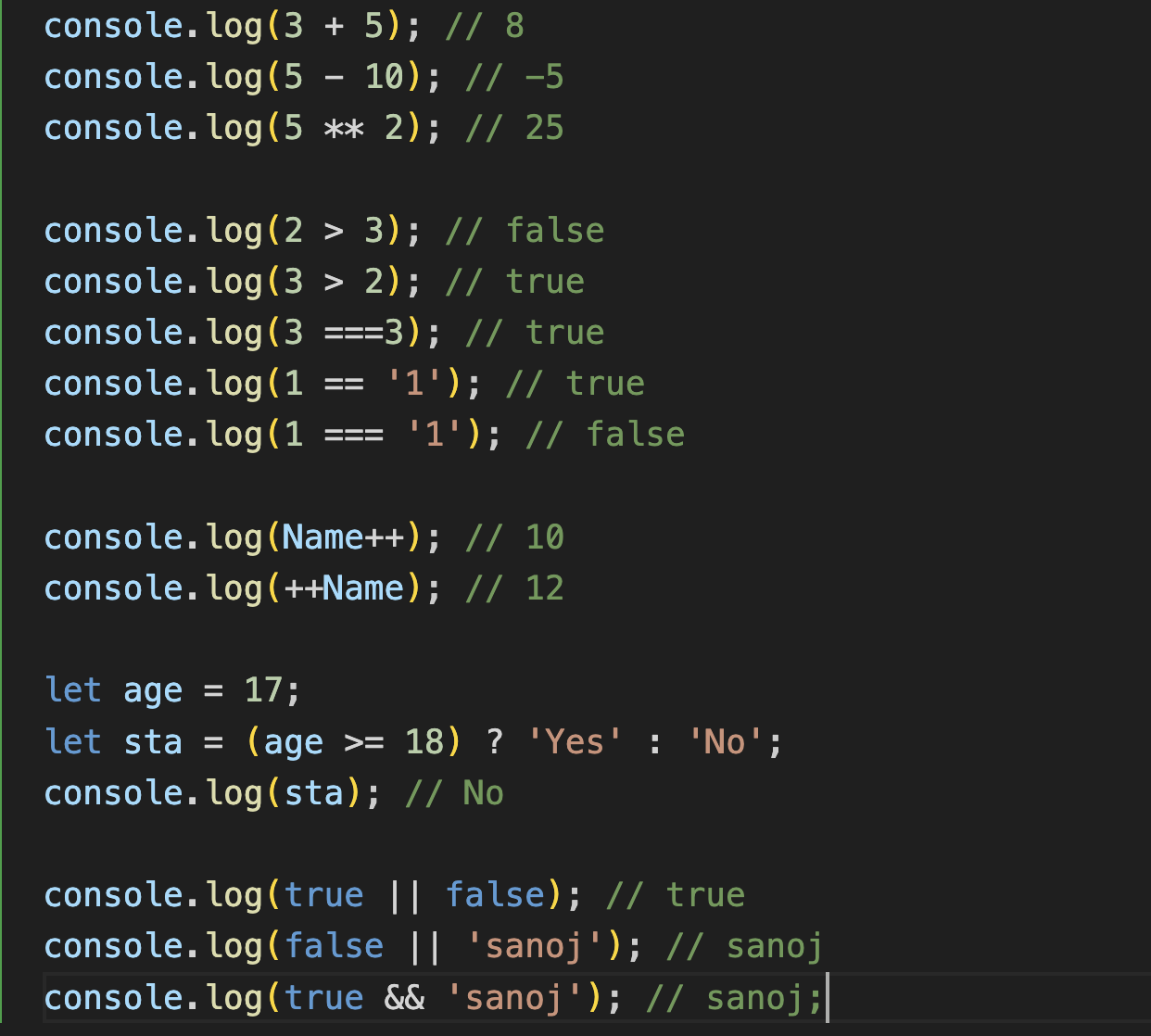
* + Access using dot notation, person.age; etc.
  + Access using bracket notation, person[‘age’];
  + There is difference of using these two notations.
* Arrays: A list of of element (items). Can be same kind of items and can be different.
  + let arr = [1, 2, true, ‘Sanoj’];
  + let arr = [“sanoj”, “kumar”];
  + Accessing element using Indexes. Starting with 0 index till len – 1;
  + What if we try to access index which is not present it will give me undefined value.
* Functions: Later we will use it.

**Operators**

* Arithmetic: +, -, /, %, \*\* (exponential/power)
* Assignment: = (equal to operator), += , -=, \*=, /=.
* Comparison: <, >, <=, >=, = = = (to check equality – strict equality), !== (to check not equality – strict not equality)
  + Loose equality ( = =): Only value is checked.
  + Strict Equality ( = = = ): Value + Type is checked.
* Bitwise:
  + Bitwise AND (&): works on the bits of the number.
  + Bitwise OR ( | ): works on the bits of the number.
  + Example: 0010 & 0011 = it will give me 0010
  + Example: 0110 | 0011 = it will give me 0111
* Logical: AND, OR, NOT
  + When we are having multiple condition and we want to process them using some condition there we use Logical Operator.
  + AND (&&): All condition must be true
  + OR ( | | ): Any condition is true. Short circuiting (means it will execute its execution when it will find the first truthy value).
  + NOT ( ! ): Change the Boolean value.
  + It can also be used with non-Boolean value.
  + Check using the code. There is something which are known as Falsie and Truthy Value.
  + Falsie: undefined, null, 0, false, ‘’, NaN
  + Truthy: Anything that is not falsie.
* Pre / Post Increment / Decrement operators:
  + Pre (++ x) first increment then use
  + Post (x ++) first use then increment
  + Similar in Decrement
* Ternary Operator
  + Condition? val1 : val2
  + If condition is true then it will be assigned val1 else val2
  + Like let status = (age >= 18) ? ‘I can vote’ : ‘I cannot Vote’;

Operator Precedence

* Using brackets
* Example: let c = a + b \* d / c;
* Use brackets: let c = a + ((b \* d) / c);



**Control Statements**

* if – else:
  + if (condition) {

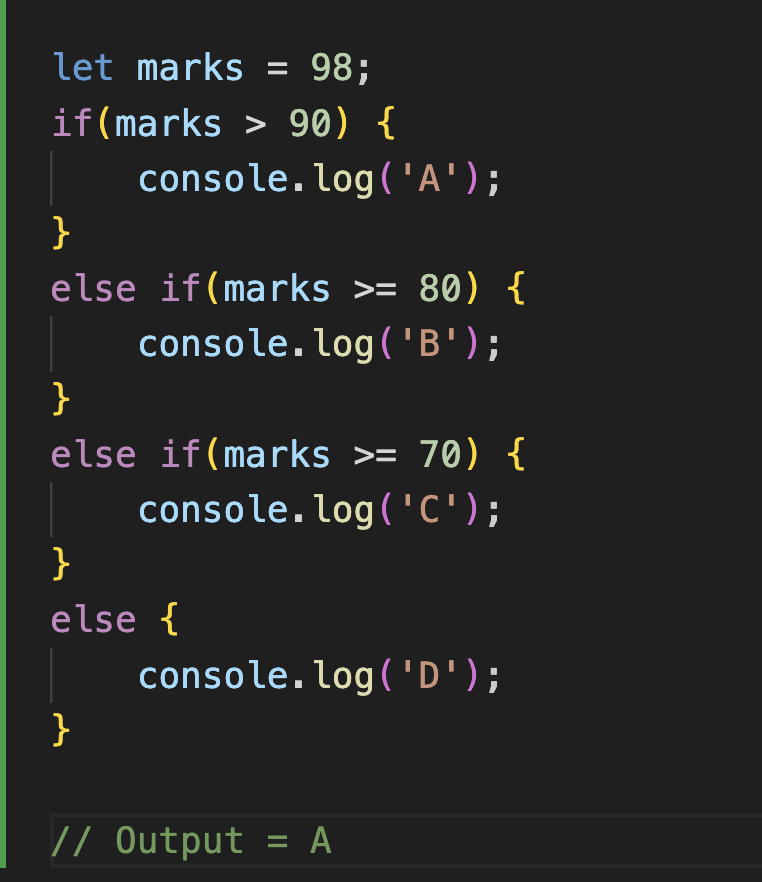
}

Elseif (condition) {

}

else {

}



* Switch:
  + switch(expression) {

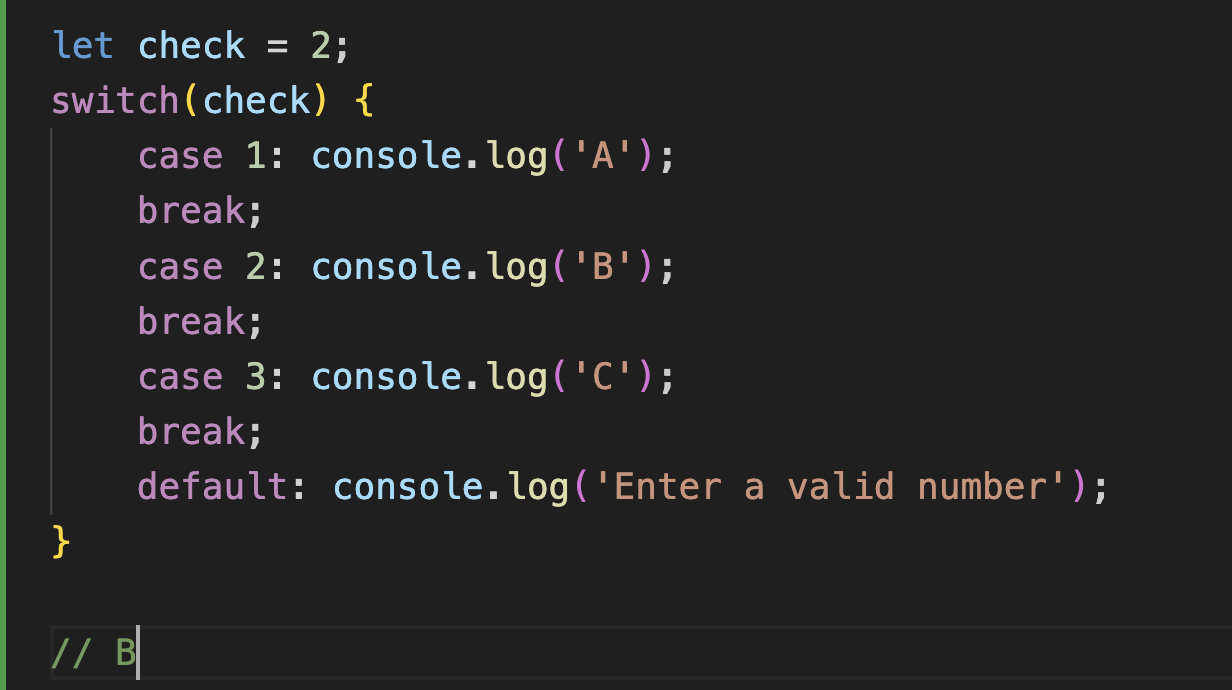
case 1: ---

case 2: --

case 3: --

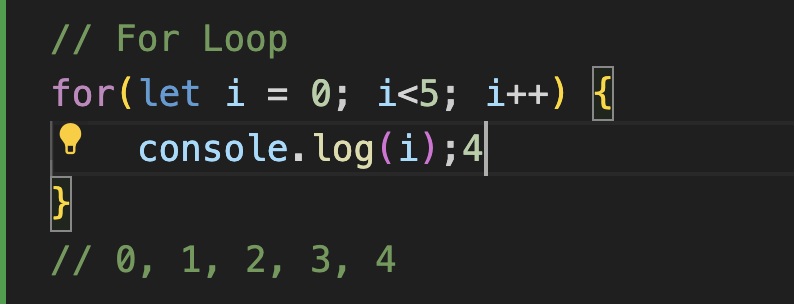
default: --

}

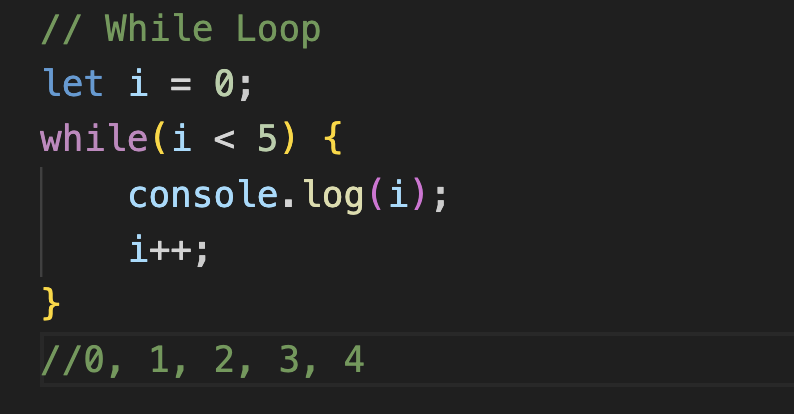


**Loops**

* For Loop:
  + When we want to do repetition of task.
  + for (let i = 0; i <= 5; i++) { - - -}
  + first part is initialization, then condition, then increment or decrement or updating.



* While Loop:
  + Only condition is checking and updation is done inside the block and initialization is done outside the while loop
  + While(condition) { - - - , updation}



* Do While Loop:
  + Same as while loop but it will run 1 time for sure.
  + Do {

} while(condition)

