Faculty of Computing

BSc (Hons) in Software Engineering
Year 2 Semester 2 (2026)

SE2020 - Web and Mobile Technologies Lab 03

JavaScript Basics – Lab Sheet

Why you will learn

After completing this lab, students will be able to:

- Understand the role of JavaScript in web development
- Use the `<script>` tag correctly in an HTML file
- Write basic JavaScript syntax
- Declare and use variables
- Identify valid identifiers
- Work with basic data types
- Use conditional statements and loops
- Perform comparisons and logical operations
- Use dialog boxes (alert, confirm, prompt)
- Understand basic JavaScript objects, arrays, and functions

Pre Lab-Requirements

Before starting this lab, students should have:

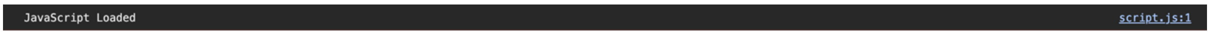
- Basic understanding of HTML and CSS
- A computer with a modern web browser (Chrome, Firefox)

Introduction

1. Script Tag and Syntax

- **Explanation:** The `<script>` tag connects JavaScript to your HTML. Statements end with ;. Use `console.log()` to print messages in the console.
- **Example:**

```
console.log("JavaScript Loaded");
```



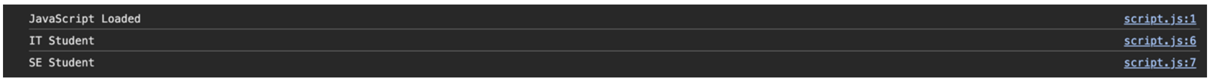
```
JavaScript Loaded                                                    script.js:1
```

- **Task:** Print your name in the console.

2. JavaScript is Case-Sensitive

- name and Name are different.
- **Example:**

```
let name = "IT Student";
let Name = "SE Student";
console.log(name); // IT Student
console.log(Name); // SE Student
```



```
JavaScript Loaded                                                    script.js:1
IT Student                                                           script.js:6
SE Student                                                           script.js:7
```

3. Reserved Keywords

- Words like let, const, var, if, else, function cannot be used as variable names.
- Wrong: `let if = 10;`

4. Variables

- **let:** Can change, block-scoped
- `let age = 25;`
- `age = 26;`

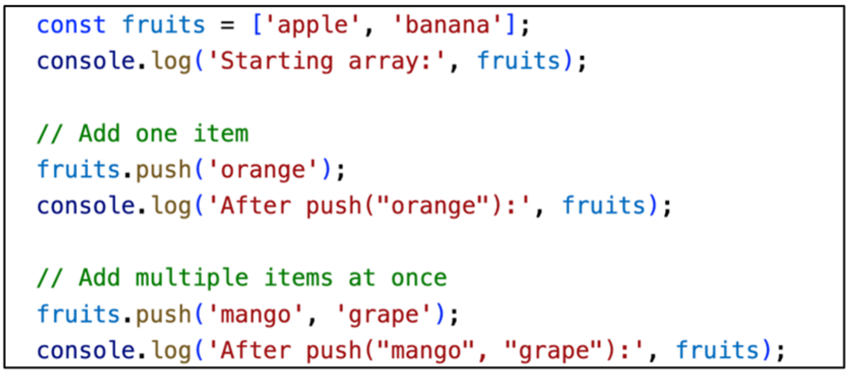
- **const:** Cannot change, used for constants
- `const country = "Sri Lanka";`
- **var:** Older, function-scoped, avoid if possible
- **Special variable names:** _count, $price, $$total, $myMoney

5. Data Types

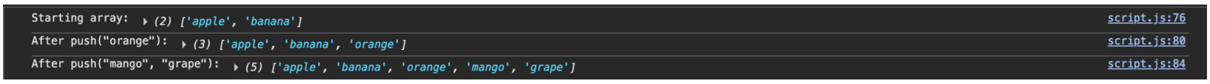
- **String:** ”ITXXXXXXXX”
- **Number:** 95
- **Boolean:** true or false
- **Null:** `let value = null;`
- **Undefined:** `let result;`

6. Arrays

An array is used to store multiple values in one variable.



```
const fruits = ['apple', 'banana'];
console.log('Starting array:', fruits);

// Add one item
fruits.push('orange');
console.log('After push("orange"):', fruits);

// Add multiple items at once
fruits.push('mango', 'grape');
console.log('After push("mango", "grape"):', fruits);
```

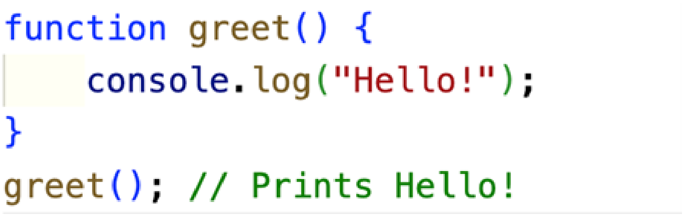


```
Starting array:  ▸ (2) ['apple', 'banana']                                          script.js:76
After push("orange"):  ▸ (3) ['apple', 'banana', 'orange']                          script.js:80
After push("mango", "grape"):  ▸ (5) ['apple', 'banana', 'orange', 'mango', 'grape']  script.js:84
```

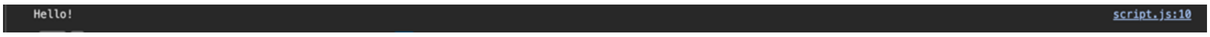
Task: Print the array values in the console and observe the output. Then, try to access individual elements of the array using the array concepts you learned earlier and check what happens.

7. Functions

- Blocks of code that do tasks, can be reused.
- **Example:**



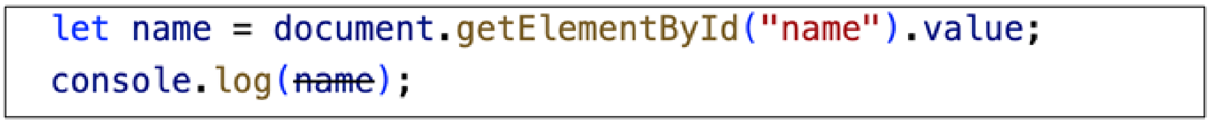
```
function greet() {
    console.log("Hello!");
}
greet(); // Prints Hello!
```



```
Hello!                                                    script.js:10
```

8. DOM (Document Object Model)

- Lets JavaScript **access HTML elements**.
- **Example:**



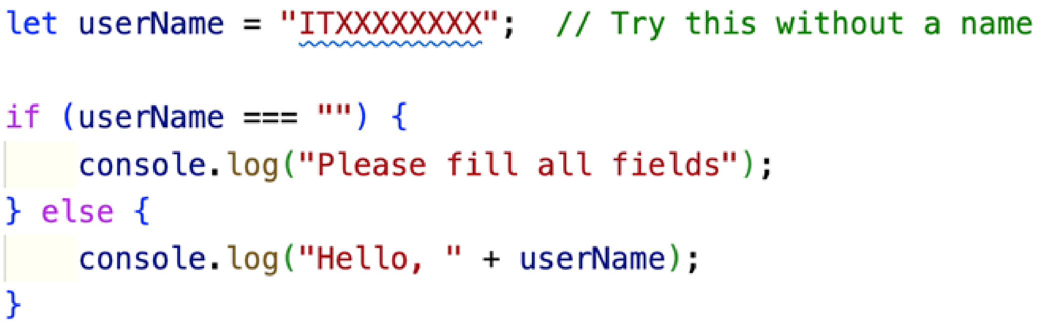
```
let name = document.getElementById("name").value;
console.log(~~name~~);
```

Task: Create an input field in HTML and print its value to the console. Will learn more about next lab sheet.

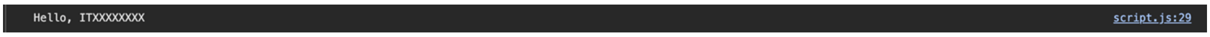
9. Conditional Statements

- **If else:** Decide based on conditions

Task – Try this without a name

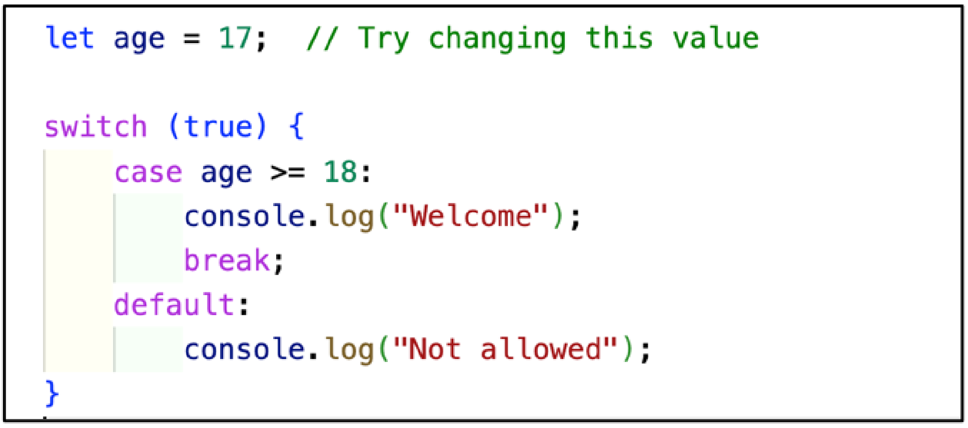


```
let userName = "ITXXXXXXXX";  // Try this without a name

if (userName === "") {
    console.log("Please fill all fields");
} else {
    console.log("Hello, " + userName);
}
```

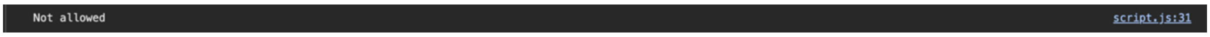


```
Hello, ITXXXXXXXX                                         script.js:29
```

- **Switch:** Check multiple conditions



```
let age = 17;  // Try changing this value

switch (true) {
    case age >= 18:
        console.log("Welcome");
        break;
    default:
        console.log("Not allowed");
}
```

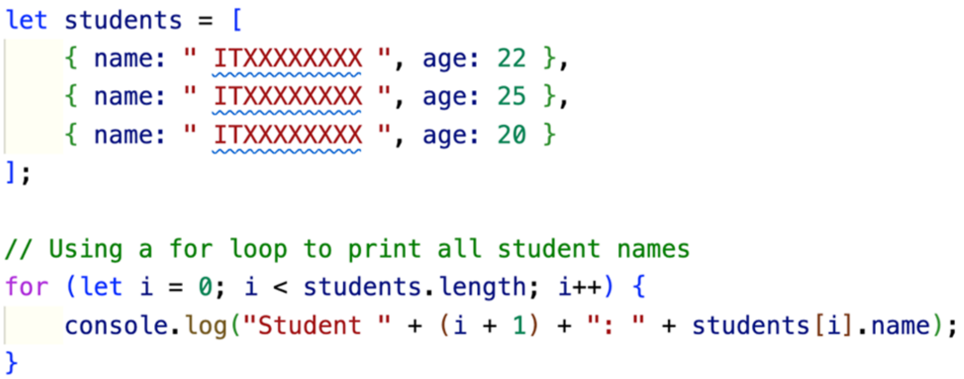


```
Not allowed                                                    script.js:31
```

10. Loops

- Repeat tasks multiple times
- **For loop example:**

```
// Array to store student objects
```



```
let students = [
    { name: " ITXXXXXXXX ", age: 22 },
    { name: " ITXXXXXXXX ", age: 25 },
    { name: " ITXXXXXXXX ", age: 20 }
];

// Using a for loop to print all student names
for (let i = 0; i < students.length; i++) {
    console.log("Student " + (i + 1) + ": " + students[i].name);
}
```

11. Comparison Operators

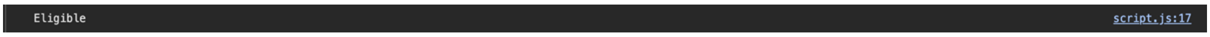
- == equal, === strict equal, != not equal, ¿ greater than, ¡ less than
- **Example:**

```
console.log(5 === "5"); // false
```

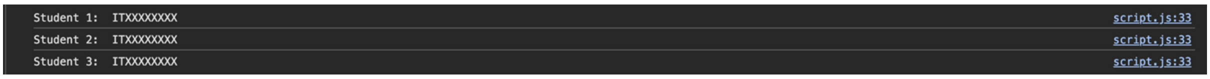
12. Logical Operators

- && AND, —— OR, ! NOT
- **Example:**

```
let age = 20;
let isPassed = true;
if (age > 18 && isPassed)
console.log("Eligible");
```



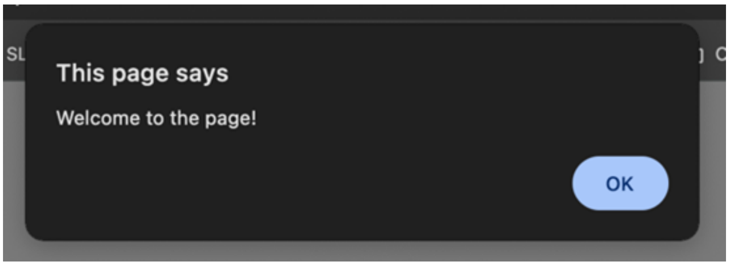
```
Eligible                                                        script.js:17
```



```
Student 1:  ITXXXXXXXX                                          script.js:33
Student 2:  ITXXXXXXXX                                          script.js:33
Student 3:  ITXXXXXXXX                                          script.js:33
```

13. Dialog Boxes

- **Alert:** Show message



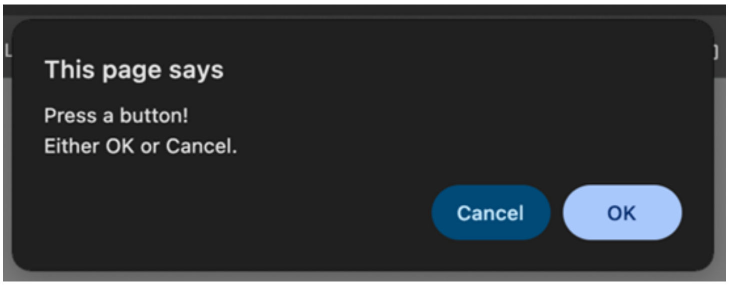
This page says

Welcome to the page!

```
alert("Welcome to the page!");
```

- **Confirm:** Yes/No dialog

```
confirm("Press a button!OK or Cancel.");
```

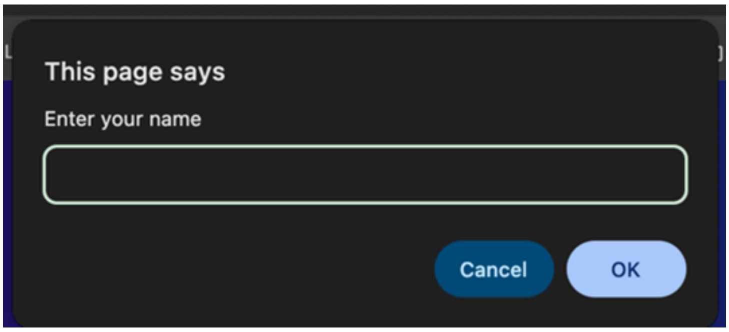


This page says

Press a button!
Either OK or Cancel.

- **Prompt:** Ask for input

```
prompt("Enter your name");
```

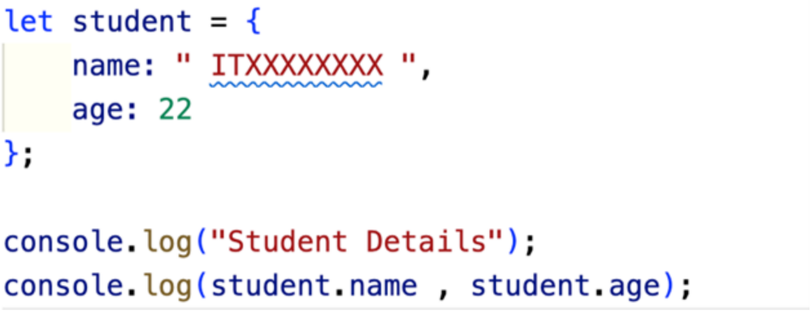


This page says

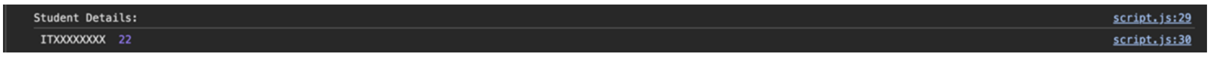
Enter your name

13. Objects

- Store related data as key-value pairs



```
let student = {
    name: " ITXXXXXXXX ",
    age: 22
};

console.log("Student Details");
console.log(student.name , student.age);
```



```
Student Details:                                   script.js:29
 ITXXXXXXXX  22                                    script.js:30
```

Lab Exercise

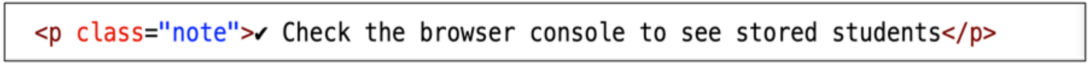
Project Setup

1. Use the folder from your previous lab.
2. Create a file named script.js for JavaScript.
3. Make sure index.html and styles.css exist from previous labs.

Create the HTML Form

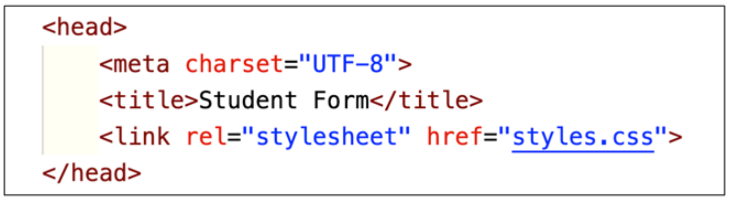
1. Add a form in index.html with **four input fields**:
   - Name
   - Age [minage = 18]
   - Email
   - Phone Number
2. Add a **submit button**. [*Make sure index.html and styles.css exist from previous labs*]
3. Add a note below the form:

✓ Check the browser console to see stored students



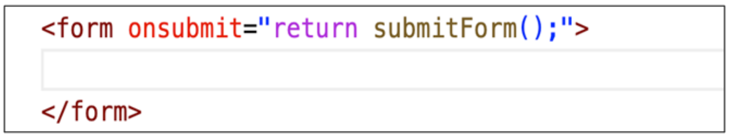
```
<p class="note">✓ Check the browser console to see stored students</p>
```

4. Link the CSS and JS files.



```
<head>
    <meta charset="UTF-8">
    <title>Student Form</title>
    <link rel="stylesheet" href="styles.css">
</head>
```

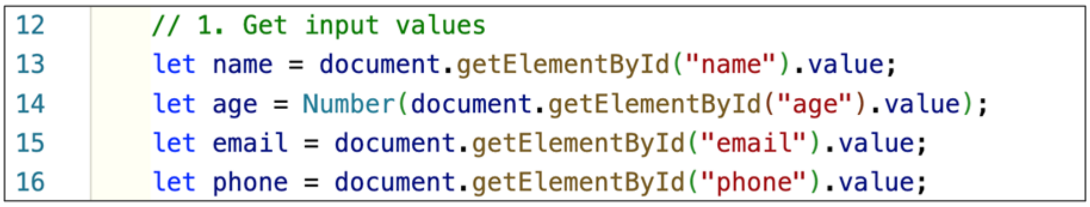
5. Add onsubmit=”return submitForm();” to the form tag.



```
<form onsubmit="return submitForm();">

</form>
```

Capture Input

1. Create a constant variable called minAge and assign it the value **18**.
2. Create an empty array called students to store student objects.
3. Write a **function** submitForm() that reads the values entered in the form:
   a. Name
   b. Age
   c. Email
   d. Phone Number



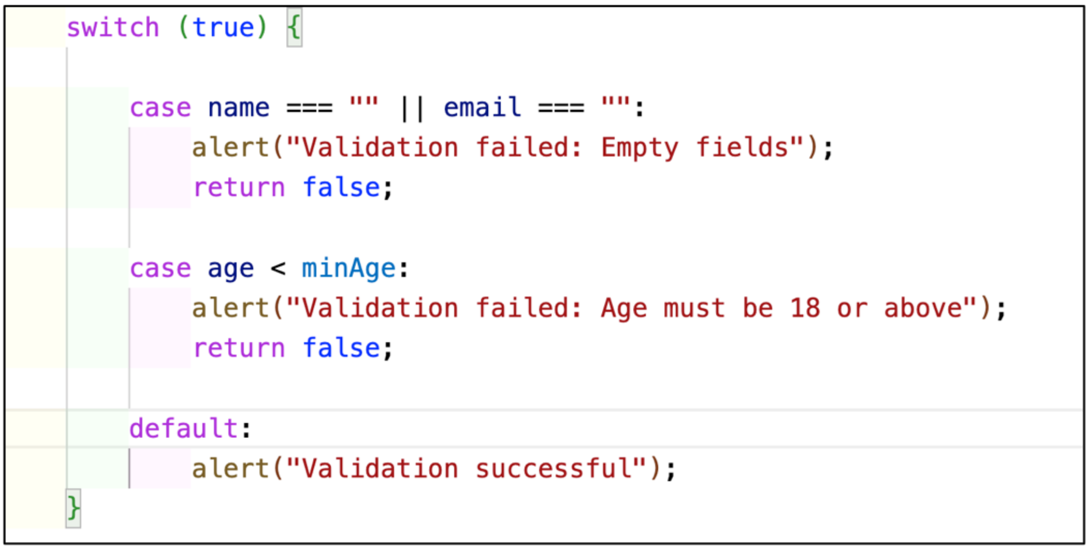
```
    // 1. Get input values
    let name = document.getElementById("name").value;
    let age = Number(document.getElementById("age").value);
    let email = document.getElementById("email").value;
    let phone = document.getElementById("phone").value;
```

Hint – Use **document.getElementById(”id”).value** to get the value entered in an input field.

```
Number(document.getElementById("age").value);
```

Validate Input

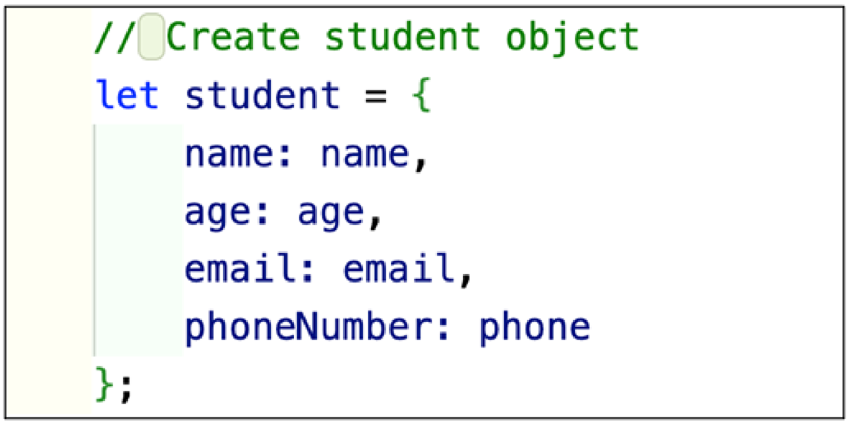
1. Use a **switch statement** for validation: [Only check the conditions according to the given]
   - If Name is empty **or** email is empty → display **alert**: ”Validation failed: Empty fields”
   - If Age < 18 → display **alert**: ”Validation failed: Age below 18”
   - If all data is valid → display **alert**: ”Validation successful”
   - Otherwise print ”Validation successful”



```
switch (true) {

    case name === "" || email === "":
        alert("Validation failed: Empty fields");
        return false;

    case age < minAge:
        alert("Validation failed: Age must be 18 or above");
        return false;

    default:
        alert("Validation successful");
}
```

Create a Student Object

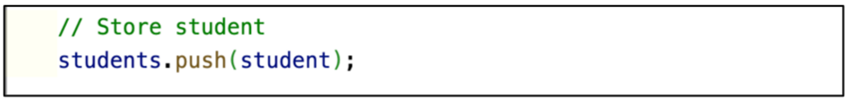
1. If validation passes, create a **student object** with:
   - Name
   - Age
   - Email
   - Phone Number



```
// Create student object
let student = {
    name: name,
    age: age,
    email: email,
    phoneNumber: phone
};
```

Store Students

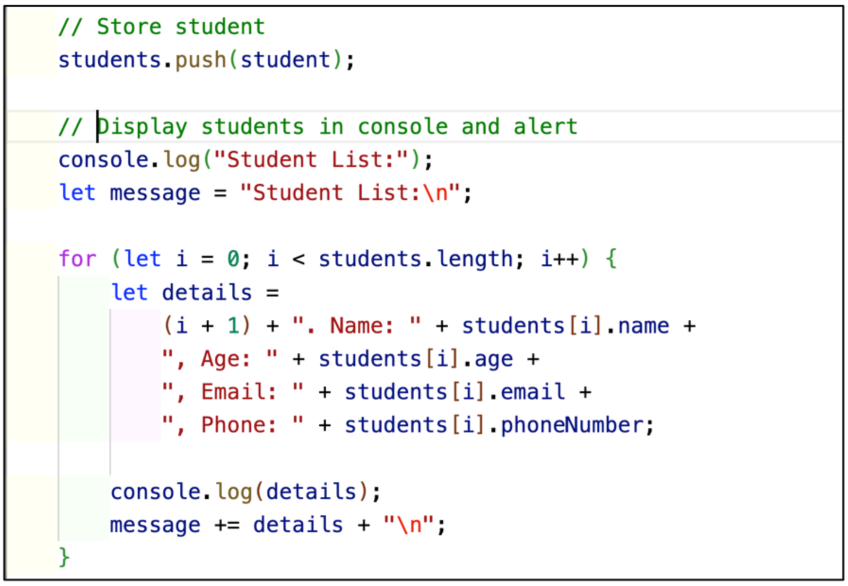
1. Add each student object to an **array** for storage.



```
// Store student
students.push(student);
```

Display Students

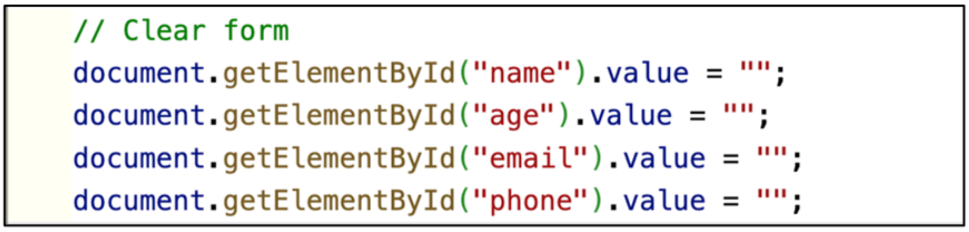
1. Use a **for loop** to print all stored students in the browser console in this format:



```
// Store student
students.push(student);

// Display students in console and alert
console.log("Student List:");
let message = "Student List:\n";

for (let i = 0; i < students.length; i++) {
    let details =
        (i + 1) + ". Name: " + students[i].name +
        ", Age: " + students[i].age +
        ", Email: " + students[i].email +
        ", Phone: " + students[i].phoneNumber;

    console.log(details);
    message += details + "\n";
}
```

Clear Form

1. After successful submission:
   - Clear all form fields.
   - Set focus back to the Name field.
   - Prevent the page from refreshing.



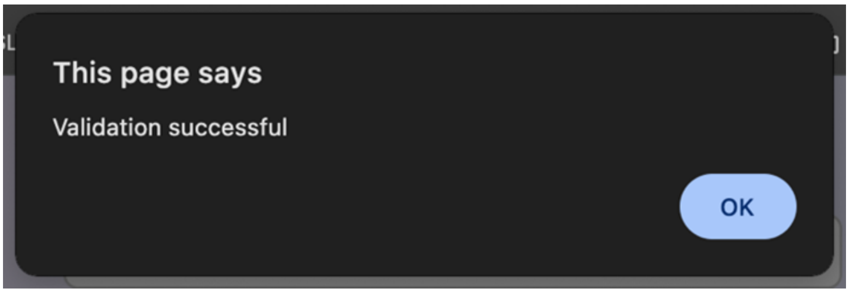
```
// Clear form
document.getElementById("name").value = "";
document.getElementById("age").value = "";
document.getElementById("email").value = "";
document.getElementById("phone").value = "";
```

Test the Form

1. Open index.html in a browser.
2. Test the form with valid and invalid data:
   - Alerts should show validation errors.
   - ”Validation successful” should show when input is valid.
   - Console should display the full student list.

- Form fields should clear after each successful submission

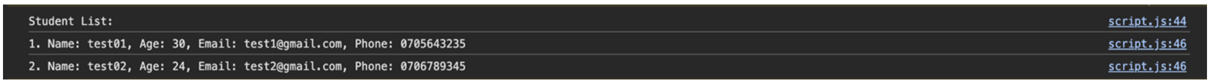
Success message:



This page says

Validation successful

Console Output:



```
Student List:                                                              script.js:44
1. Name: test01, Age: 30, Email: test1@gmail.com, Phone: 0705643235        script.js:46
2. Name: test02, Age: 24, Email: test2@gmail.com, Phone: 0706789345        script.js:46
```

In class Test

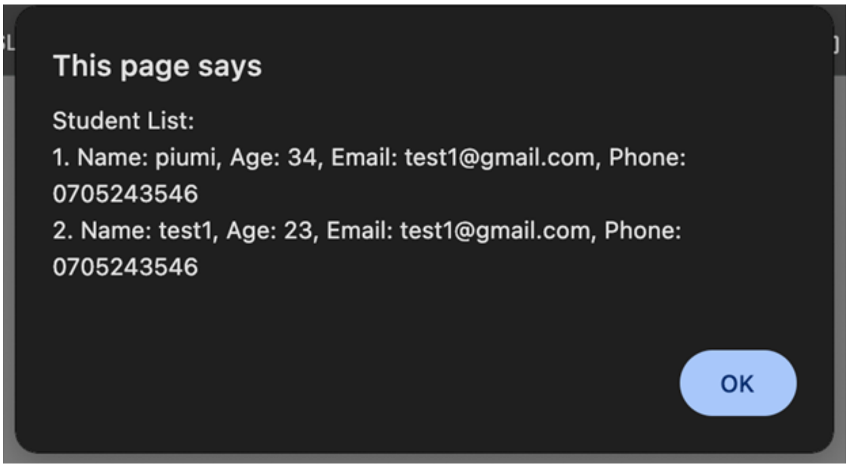
Improve the above code to show the

A. Add a validation part to check whether the phone number have 10 digits.

B. Phone number must contain 10 numbers -¿ display ”Validation failed: Phone number must be at least 10 digits”

C. Use alert to display info to user as a list

Output:



This page says

```
Student List:
1. Name: piumi, Age: 34, Email: test1@gmail.com, Phone:
0705243546
2. Name: test1, Age: 23, Email: test1@gmail.com, Phone:
0705243546
```