

# **HTML5\_Day4\_Hands\_On\_Koduru\_Ushasree**

## **Problem 1**

**Assessment Goal:** Ensure learners understand responsiveness and screen adaptability.

### **Hands-on Tasks:**

1. Add viewport meta tag to the HTML page
2. Use media queries to:
  - o Change background color on mobile screen
  - o Adjust font size for smaller screens
3. Convert navigation into vertical layout on mobile
4. Test the page using browser responsive mode

### **Expected Outcome:**

A webpage that looks different and readable on mobile and desktop screens.

### **Code :-**

```
Problem1 > index.html > html > body > header.header
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Responsive Page</title>
7      <link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <header class="header">
11         <h1>Responsive Webpage</h1>
12     </header>
13     <nav class="navbar">
14         <a href="#">Home</a>
15         <a href="#">About</a>
16         <a href="#">Contact</a>
17     </nav>
18
19     <div class="content">
20         <h1>Welcome to my Website</h1>
21         <p>This is responsive webpage</p>
22     </div>
23
24 </body>
25 </html>
```

```
1  /* Default styles (Desktop) */
2  body {
3      background-color: #lightblue;
4      font-size: 20px;
5      font-family: Arial;
6  }
7
8
9  .header {
10     background-color: #darkblue;
11     color: #white;
12     text-align: center;
13     padding: 20px;
14 }
15
16 .navbar {
17     display: flex;
18     justify-content: space-around;
19     background-color: #black;
20     padding: 15px;
21 }
22
23 .navbar a {
24     color: #white;
25     text-decoration: none;
26 }
27
28 .content {
29     text-align: center;
30     margin-top: 50px;
31 }
32
33
34 /* Mobile styles */
35 @media (max-width: 600px) {
36     body {
37         background-color: #lightgreen;
38
39         font-size: 15px;
40     }
41
42     .header {
43         font-size: 18px;
44         padding: 15px;
45     }
46
47     .navbar { /* Co
48         flex-direction: column;
49         align-items: center;
50     }
51
52     .navbar a {
53         padding: 10px;
54     }
55 }
```

## **Output :-**



## **Technical Requirements :-**

The webpage must include the `<meta name="viewport">` tag to ensure proper scaling on different screen sizes. CSS media queries must be used to modify styles based on screen width, such as changing background color and adjusting font size for smaller screens. The navigation layout should switch from horizontal (desktop) to vertical (mobile) using responsive CSS rules like `flex-direction: column`. The page must be tested using the browser's responsive mode to verify adaptability across devices.

## **Problem 2: Student Grade Evaluator (Level-1)**

### **Scenario**

A school wants a simple JavaScript program to evaluate a student's performance based on marks obtained in a subject.

### **📌 Requirements**

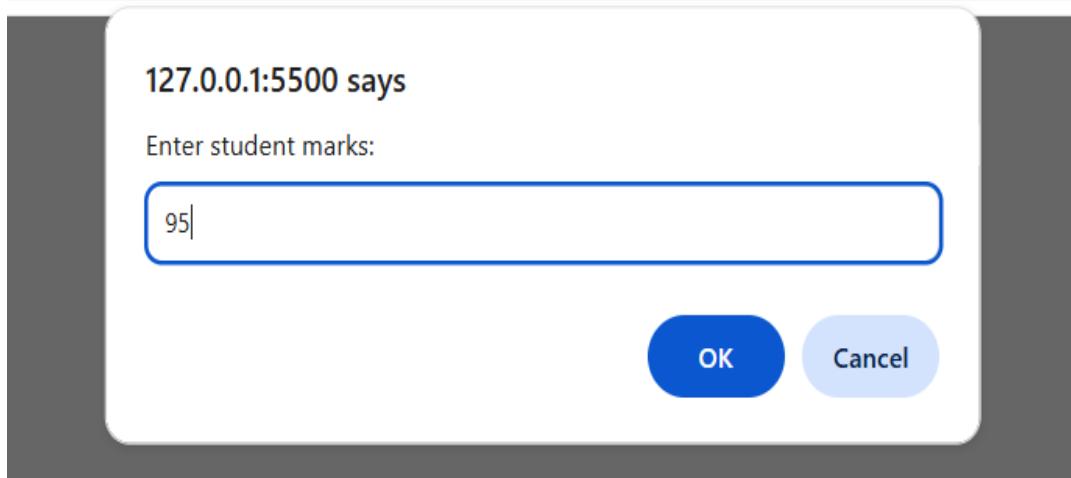
- Accept the student's marks as a variable
- Use if–else statements to assign grades:
  - Marks  $\geq 75 \rightarrow$  Grade A
  - Marks  $\geq 60 \rightarrow$  Grade B
  - Marks  $\geq 40 \rightarrow$  Grade C
  - Marks  $< 40 \rightarrow$  Fail

Display the grade on the web page or console

**Code :-**

```
Problem2 > index.html > html > body > script
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Document</title>
7     </head>
8     <body>
9       <h1>Student Grade Evaluator</h1>
10      <script>
11        let input = prompt("Enter student marks:");
12        let marks = Number(input);
13        if (isNaN(marks)) {
14          document.write("Invalid input");
15        } else if (marks >= 75) {
16          document.write("Grade A");
17        } else if (marks >= 60) {
18          document.write("Grade B");
19        } else if (marks >= 40) {
20          document.write("Grade C");
21        } else {
22          document.write("Fail");
23        }
24
25
26      </script>
27    </body>
28  </html>
```

**Output :-**



# **Student Grade Evaluator**

Grade A

## **Technical Requirements :-**

The program must use JavaScript with variables declared using let or const and store marks as a numeric data type. Conditional logic must be implemented using if–else statements and comparison operators ( $\geq$ ,  $<$ ). Logical flow must strictly follow the grading criteria without using functions or arrays. The result should be displayed using console.log() or document.write(), and only basic JavaScript syntax is allowed.

## **Problem 3: Simple Discount Calculator (Level-1)**

### **Scenario**

An online store wants to apply a discount based on the total purchase amount.

### **Requirements**

- Store purchase amount in a variable
- Apply discount rules:
  - Amount  $\geq$  5000  $\rightarrow$  20% discount
  - Amount  $\geq$  3000  $\rightarrow$  10% discount
  - Amount  $<$  3000  $\rightarrow$  No discount
- Calculate and display:
  - Discount amount
  - Final payable amount

### **Code :-**

```

Problem3 > index.html > html > body > h1
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document3</title>
7  </head>
8  <body>
9      <h1>Simple Discount Calculator </h1>
10     <script>
11         let amount = 5000;
12         let discount = 0;
13         let finalAmount = 0;
14         if (amount >= 5000) {
15             discount = amount * 0.20; // 20% discount
16         } else if (amount >= 3000) {
17             discount = amount * 0.10; // 10% discount
18         } else {
19             discount = 0; // No discount
20         }
21         finalAmount = amount - discount;
22         console.log("Purchase Amount: " + amount);
23         console.log("Discount: " + discount);
24         console.log("Final Amount: " + finalAmount);
25     </script>
26 </body>
27 </html>

```

## Output :-



## Technical Requirements :-

The solution must store the purchase amount in a variable using primitive numeric data types. Arithmetic operators (+, -, \*, /, %) should be used to calculate the discount and final payable amount. Conditional statements using if–else must determine the applicable discount rate. No user input is required, and hardcoded values are allowed.

## Problem 4: Traffic Signal Simulator (Level-2)

### Scenario

A traffic control system needs a JavaScript program that displays instructions based on traffic signal color.

### Requirements

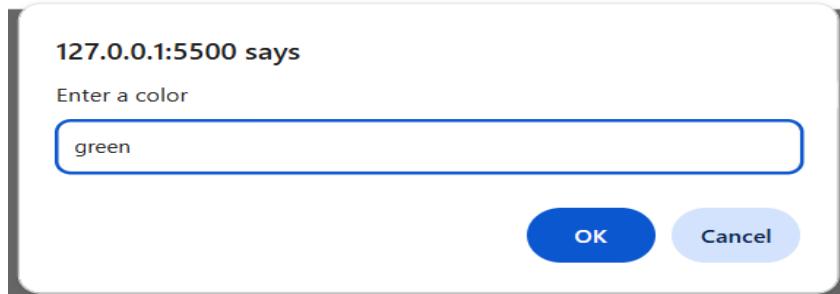
- Store signal color in a variable ("red", "yellow", "green")
- Use a **switch statement** to display:
  - Red → Stop
  - Yellow → Get Ready
  - Green → Go

Handle invalid signal input gracefully

### Code :-

```
Problem4 > index.html > html > body > script > signal
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document4</title>
7  </head>
8  <body>
9      <h1>Traffic Signal Simulator</h1>
10     <script>
11         let signal = prompt("Enter a color");
12         switch(signal) {
13             case "red":
14                 document.write("Stop");
15                 break;
16             case "yellow":
17                 document.write("Get Ready");
18                 break;
19             case "green":
20                 document.write("Go");
21                 break;
22             default:
23                 document.write("Invalid signal color");
24         }
25     </script>
26 </body>
27 </html>
```

## Output :-



# Traffic Signal Simulator

Go

## Technical Requirements :-

The signal color must be stored in a string variable. A switch-case statement must be used to compare the signal color and display the appropriate instruction. `console.log()` must be used for output, and if-else statements are not permitted. A default case should be included to handle invalid signal inputs gracefully. The program must use basic JavaScript control flow structures only.

## Problem 5: Number Analysis Tool (Level-2)

**Scenario:** A utility program is required to analyze numbers and provide insights such as positivity, parity, and range.

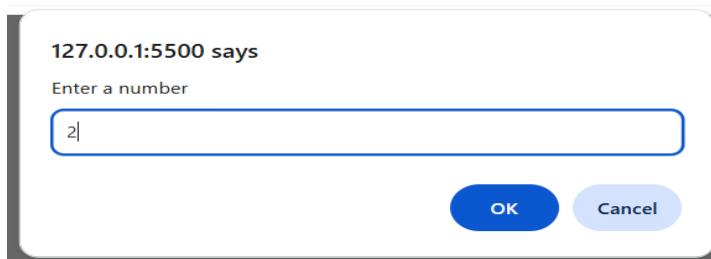
### Requirements

- Store a number in a variable
- Use **conditional (ternary) operator** to check:
- Positive or Negative
- Use **if-else** to check:
- Even or Odd
- Use a **loop** to print all numbers from 1 to the given number

## Code :-

```
problem5 > index.html > html > body > script
1   <!DOCTYPE html>
2   <html>
3   <head>
4       <meta charset="UTF-8">
5       <meta name="viewport" content="width=device-width, initial-scale=1.0">
6       <title>Number Analysis Tool</title>
7   </head>
8   <body>
9       <h1>Number Analysis Tool</h1>
10      <script>
11
12          // Store number in a variable
13          let num = prompt("Enter a number");
14
15          // Check Positive or Negative using ternary operator
16          let result = (num >= 0) ? "Positive" : "Negative";
17          console.log("Number is:", result);
18
19          // Check Even or Odd using if-else
20          if (num % 2 === 0) {
21              console.log("Number is Even");
22          } else {
23              console.log("Number is Odd");
24          }
25
26          // use loop to print numbers from 1 to given number
27          console.log("Numbers from 1 to", num, ":");
28
29          for (let i = 1; i <= num; i++) {
30              console.log(i);
31          }
32
33      </script>
34
35  </body>
36  </html>
37
```

## Output :-





## Technical Requirements :-

The program must store a number in a variable using a numeric data type. A ternary operator must be used to determine whether the number is positive or negative. An if–else statement must check whether the number is even or odd using the modulus (%) operator. A loop (such as for) must print numbers from 1 up to the given number. The program must combine multiple control flow techniques without using advanced features like functions or arrays.