Code structure:

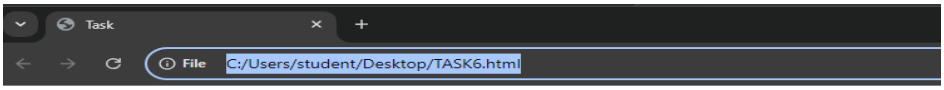
Task 6:

Write a multi-line JavaScript comment and a single-line comment. Explain the difference.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      //declare name and age to variable a and b respectively
      let a = "PAVYA";
      let b = 18;
      document.writeln(a + b);
      /*document.writeln(a);
      document.writeln(b);*/
    </script>
  </body>
</html>
```

OUTPUT:



PAVYA18

DIFFERENCE:

1. Single-line Comment

- **Syntax**: Starts with `//` and comments out the rest of the line.
- **Usage**: Best for brief comments or notes that fit on one line.

2. Multi-line Comment

- **Syntax**: Starts with `/*` and ends with `*/`. It can span multiple lines.
- **Usage**: Ideal for longer comments, detailed explanations, or commenting out blocks of code.

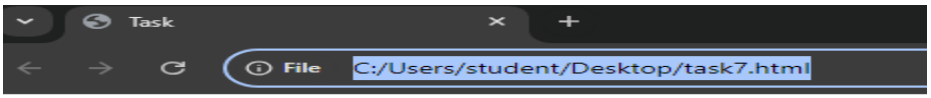
Task 7:

Create a script with both semicolon-separated and not separated lines. Note any differences in behavior.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      let a = "PAVYA";//with semicolon
      let b = "MANJU"//without semicolon
      document.writeln(a + b + "<br>");
      document.writeln(a + "<br>");
      document.writeln(b + "<br>");
    </script>
  </body>
</html>
```

OUTPUT:



PAVYAMANJU
PAVYA
MANJU

DIFFERNCE:

- **Semicolon-Separated Lines**:
  - Each statement is explicitly terminated with a semicolon (`;`).
  - This is the standard, explicit way to end statements in JavaScript.
  - It avoids ambiguity and ensures that the code behaves predictably across different environments.

- **Non-Semicolon-Separated Lines**:
  - Statements are written without semicolons.
  - JavaScript has **Automatic Semicolon Insertion (ASI)**, meaning it will try to insert semicolons at the end of statements where necessary.
  - ASI works in most cases, but there are specific scenarios where it might not behave as expected, leading to potential errors or unexpected behavior.

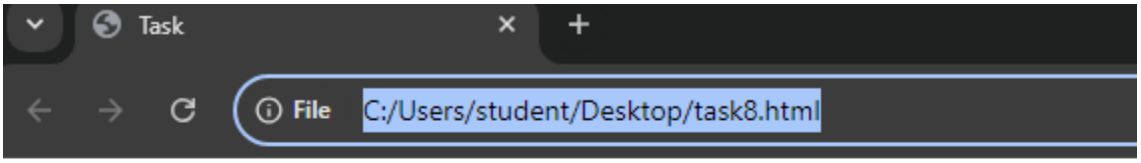
Task 8

Use proper indentation to format a nested loop.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      for (let i = 0; i < 2; i++) {
        for (let j = 0; j < 2; j++) {
          if (i === j) {
            document.writeln("i equals j " + i + "<br>");
          } else {
            document.writeln("i does not equal " + j + "<br>");
          }
        }
      }
    </script>
  </body>
</html>
```

OUTPUT:



i equals j 0
i does not equal 1
i does not equal 0
i equals j 1

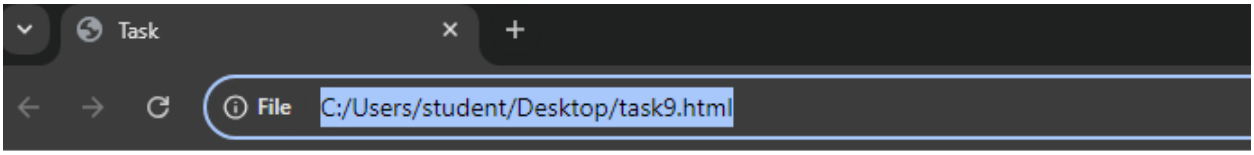
Task 9:

Declare multiple variables in a single line.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      //declare name and age to variable a and b respectively in single line
      let a = "PAVYA",
        b = 18;
      document.writeln(a + b);
    </script>
  </body>
</html>
```

OUTPUT:



PAVYA18

Task 10:

Place a script tag at the top and bottom of an HTML document. Note any differences in behavior.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <h1>SCRIPT IN BOTTOM</h1>
    <script>
      let a = "PAVYA",
        b = 18;
      document.writeln(a + "<br>");
```

```
      document.writeln(b);
    </script>
  </body>
</html>
```

OUTPUT:

SCRIPT IN BOTTOM

PAVYA
18

CODE:

```
<html>
  <head>
    <title>Task</title>
    <script>
      let a = "PAVYA",
        b = 18;
      document.writeln(a + "<br>");
      document.writeln(b);
    </script>
  </head>
  <body>
    <h1>SCRIPT IN TOP</h1>
  </body>
</html>
```

OUTPUT: