

An Introduction to JavaScript, Code structure

1. An Introduction to JavaScript:

TASK 1

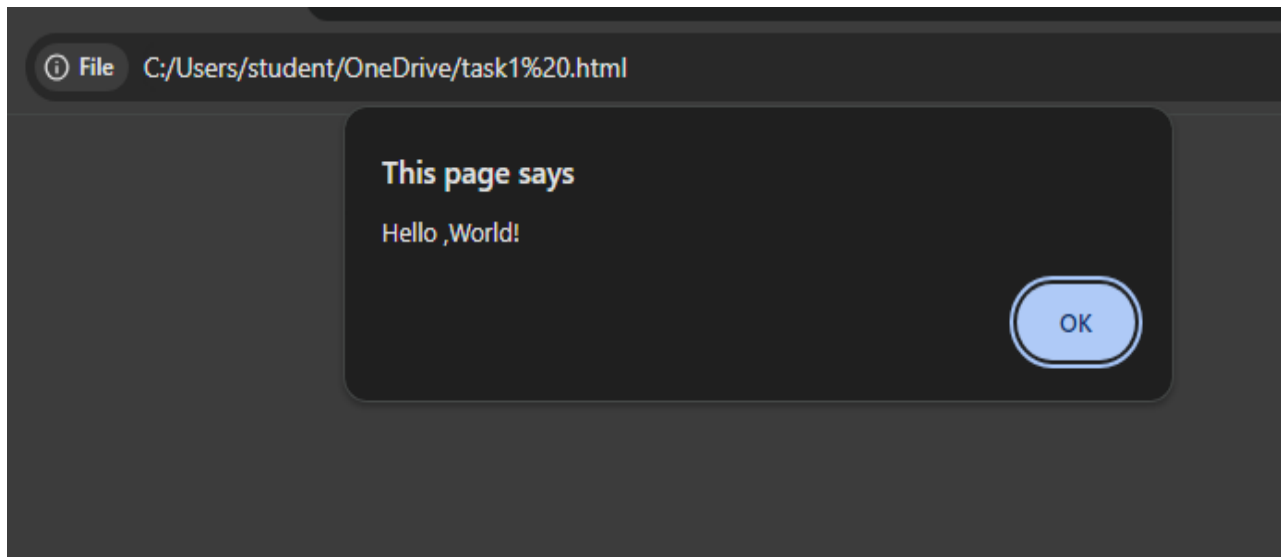
1.

Write a simple script that displays “Hello, World!” on the web page using an alert box.

CODE:

```
<> task1 .html X <> TASK 2.html
C: > Users > student > OneDrive > <> task1 .html > html
1  <html>
2    <head>
3      <title>task 1</title>
4    </head>
5    <body>
6      <script>
7        alert("Hello ,World!");
8      </script>
9    </body>
10 </html>
```

output:



TASK 2

2.

Experiment with different data types in JavaScript (e.g., string, number, boolean) by declaring and logging them in the console

CODE:

```
Users > student > OneDrive > TASK 2.html > html
<html>
  <head>
    <title>
      task 2
    </title>
  </head>
  <body>
    <script>
      var age= 20;
      var name="john";
      var x=true;
      console.log(age);
      console.log(name);
      console.log(x);
    </script>
  </body>
</html>
```

output:

```
20 TASK 2.html:12
john TASK 2.html:13
true TASK 2.html:14
> |
```

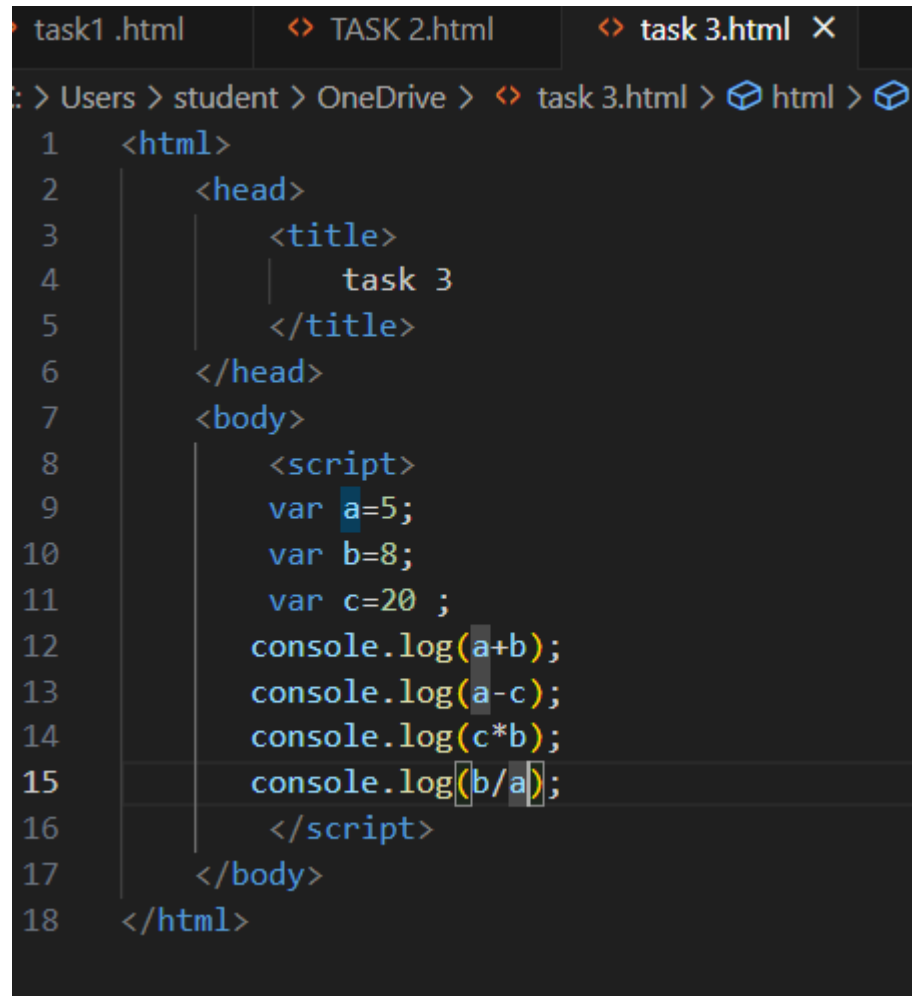
TASK 3

3.

Use the console to perform basic math operations like addition,

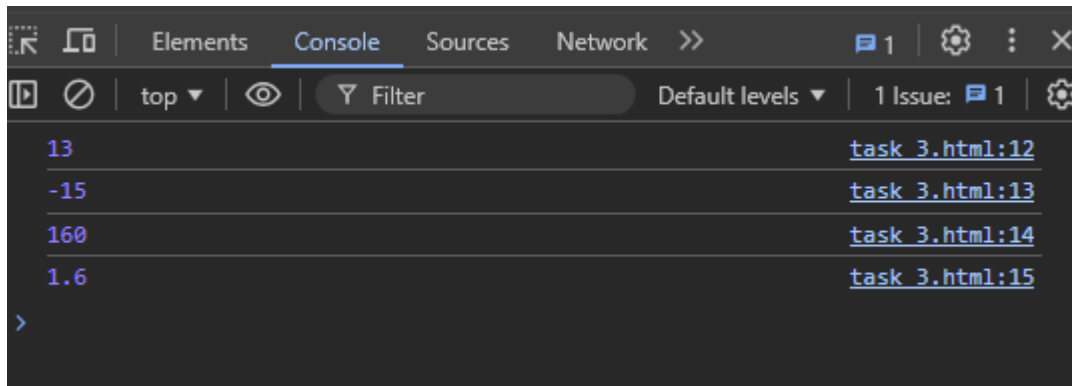
subtraction, multiplication, and division.

code:



```
task1 .html TASK 2.html task 3.html X
: > Users > student > OneDrive > task 3.html > html >
1  <html>
2      <head>
3          <title>
4              task 3
5          </title>
6      </head>
7      <body>
8          <script>
9              var a=5;
10             var b=8;
11             var c=20 ;
12             console.log(a+b);
13             console.log(a-c);
14             console.log(c*b);
15             console.log(b/a);
16         </script>
17     </body>
18 </html>
```

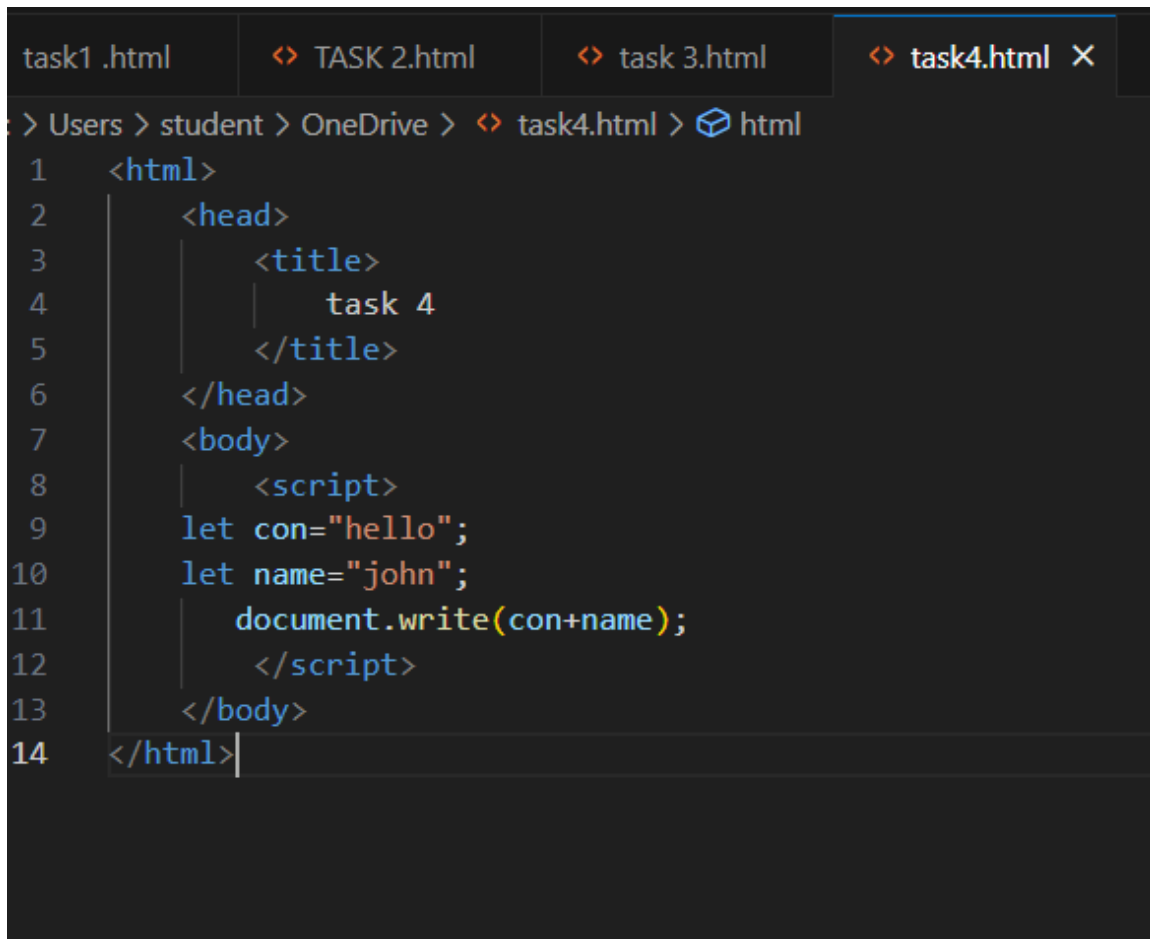
OUTPUT:



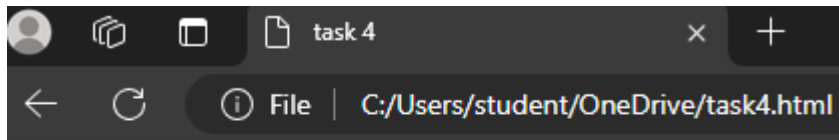
TASK 4

4. Declare two strings and concatenate them using the + operator

CODE:



output:



hellojohn

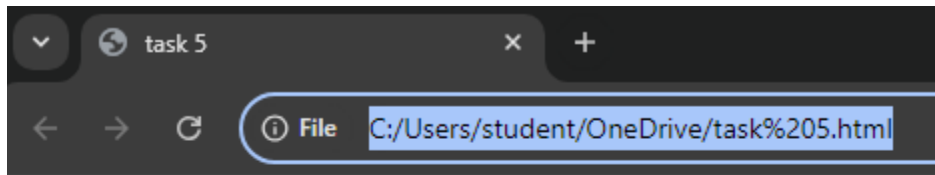
TASK 5

5. Use the typeof operator to check the data type of various variables.

CODE:

```
task1 .html  TASK 2.html  task 3.html  task
: > Users > student > OneDrive > task 5.html > html > body
1  <html>
2    <head>
3      <title>
4      </title>
5    </head>
6    <body>
7      <script>
8        let con="hello";
9        let age= 18;
10       let doc=null;
11       let per=85.67;
12       let num=true;
13       document.write(typeof(con)+"<br>");
14       document.write(typeof(age)+"<br>");
15       document.write(typeof(doc)+"<br>");
16       document.write(typeof(per)+"<br>");
17       document.write(typeof(num)+"<br>");
18     </script>
19   </body>
20 </html>
```

output:



string
number
object
number
boolean

2. Code structure

TASK 6

6. Write a multi-line JavaScript comment and a single-line comment.

Explain the difference.

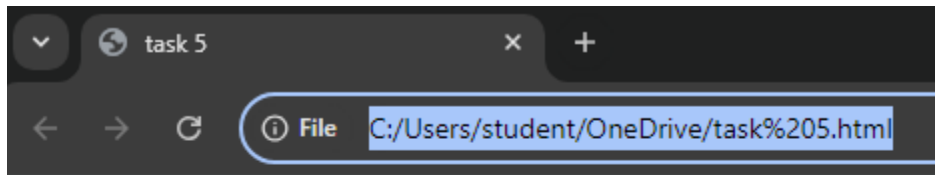
CODE:


```

<html>
  <head>
    <title>
      task 5
    </title>
  </head>
  <body>
    <script>
let con="hello";
let age= 18;
let doc=null;
let per=85.67;
let num=true;
    // THIS TYPEOF GIVES STRING EXPRESSION
    document.write(typeof(con)+"<br>");
    /*
    typeof() function returns the type of the variable
    the below examples are shows the types like number
    object, double and boolean.
    */
    document.write(typeof(age)+"<br>");
    document.write(typeof(doc)+"<br>");
    document.write(typeof(per)+"<br>");
    document.write(typeof(num)+"<br>");
    </script>
  </body>
</html>

```

output:



string
number
object
number
boolean

Difference:

multi line:

i) The multi-line text variable creates a field that lets you enter multiple lines of text.

ii) use the symbol `/* */`

single line:

i) The single-line text variable creates a field to enter a single line of text.

ii) use the symbol `//`

TASK 7

7. Create a script with both semicolon-separated and not separated lines.

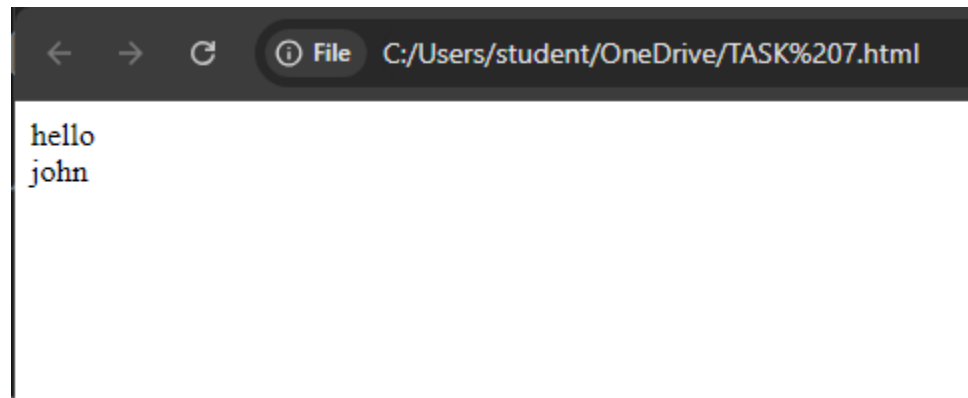
Note any differences in behavior

CODE :

C: > Users > student > OneDrive > TASK 7.html > html

```
1  <html>
2    <head>
3      <title>
4        task 4
5      </title>
6    </head>
7    <body>
8      <script>
9        let con="hello"
10       let name="john";
11       document.write(con+"<br>");
12       document.write(name+"<br>");
13     </script>
14   </body>
15 </html>
```

output:



hello
john

behaviour:

All of the line terminator characters are equivalent to each other. It doesn't matter which one is the separator for those two lines of code, any separator will have the same effect.

TASK 8

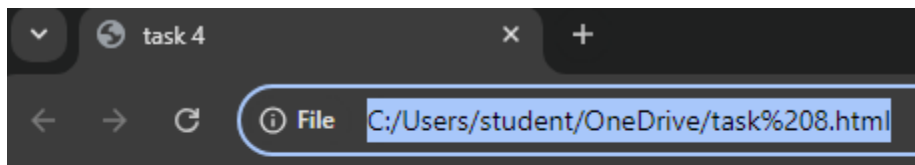
8. Use proper indentation to format a nested loop

code:

```
Users > student > OneDrive > task 8.html > html > body > script
<html>
  <head>
    <title>
      task 4
    </title>
  </head>
  <body>
    <script>
      let age=18;

      if(isNaN(age)){
        document.write("invalid input");
      }
      else{
        if(age>=18){
          document.write("you are eligible to vote");
        }
        else{
          document.write("you are not eligible to vote");
        }
      }
    </script>
  </body>
</html>
```

output:



you are eligible to vote

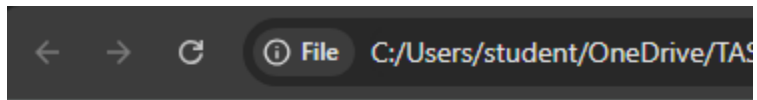
TASK 9

9. Declare multiple variables in a single line.

CODE:

```
> TASK 2.html    <> task 3.html    <> task4.html
C: > Users > student > OneDrive > <> TASK 9.html >
1  <html>
2    <head>
3      <title>
4        task 4
5      </title>
6    </head>
7    <body>
8      <script>
9        let con="hellJMI2W7*&2234";
10
11       document.write(con);
12     </script>
13   </body>
14 </html>
```

OUTPUT:



hellJMI2W7*&2234

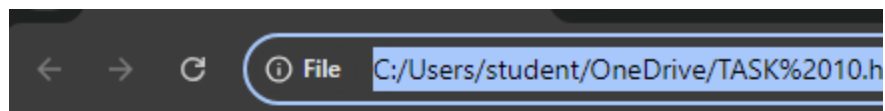
TASK 10

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

CODE:

```
TASK 2.html    <> task 3.html    <> task4.html
: > Users > student > OneDrive > <> TASK 10.html > C
1  <script>
2  <html>
3      <head>
4          <title>
5              task 4
6          </title>
7      </head>
8      <body>
9
10         let con="hello";
11         document.write(con+name);
12
13     </body>
14 </html>
15 </script>
```

OUTPUT:



BEHAVIOUR:

It seems to recall that you are not supposed to place these in the `<head>` section, but placing at the beginning of the `<body>` section is bad, too, since the JavaScript will have to be parsed before the page is rendered completely (or something like that). This seems to leave the end of the `<body>` section as a logical place for `<script>` tags.