Functions

A function is a reusable block of code that is used to perform a specific task when something invokes it.

A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().The code to be executed, by the function, is placed inside curly brackets: {}

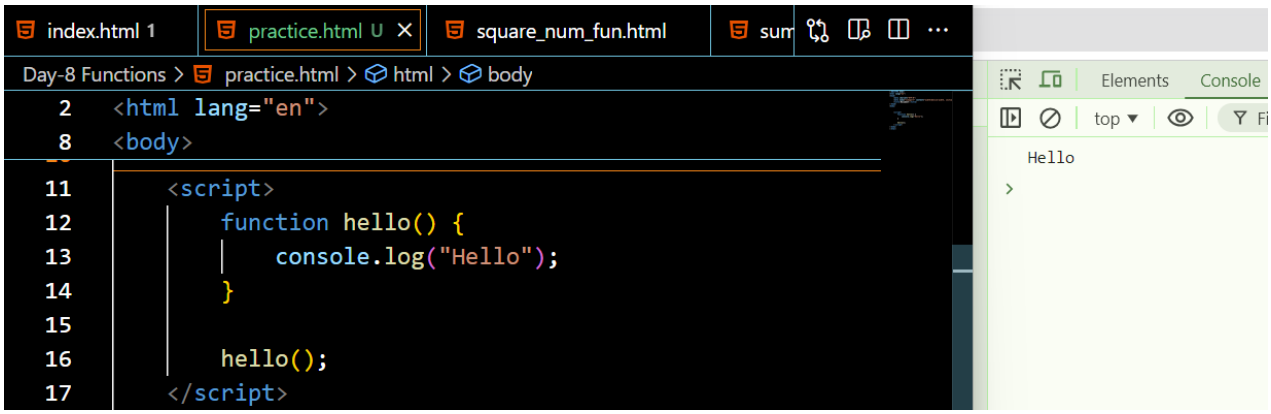
Function parameters are listed inside the parentheses () in the function definition. Function arguments are the values received by the function when it is invoked. Inside the function, the arguments (the parameters) behave as local variables.

Named functions can be hoisted

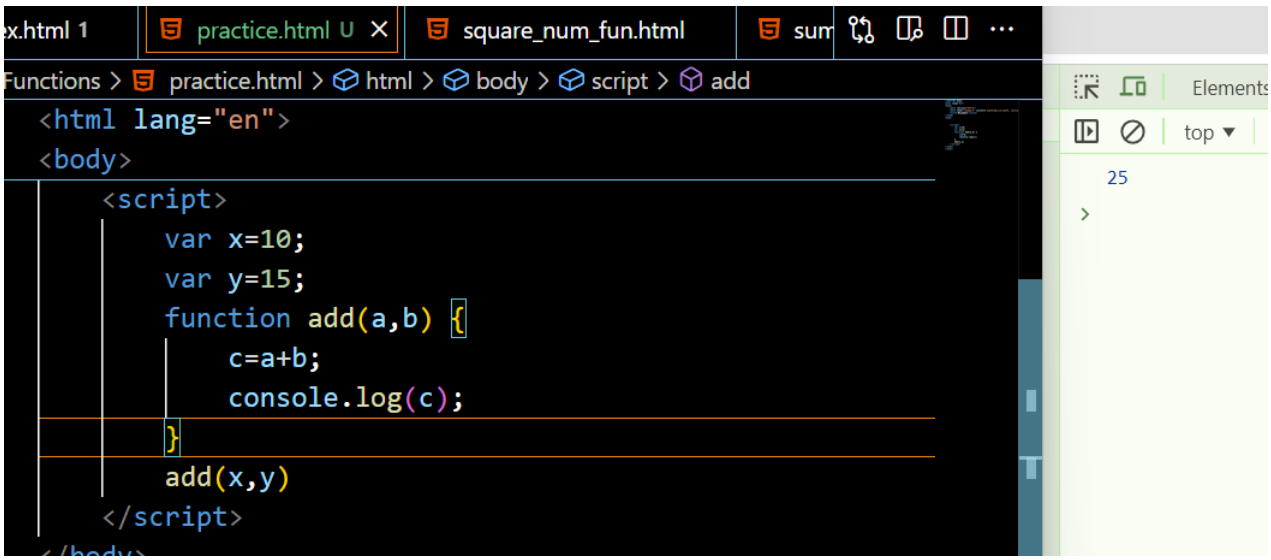
Syntax:

```
function name(params) {
        //code to be executed
    }
```

Example:



Example: adding two numbers



Example: even numbers in range

practice.html U | square_num_fun.html

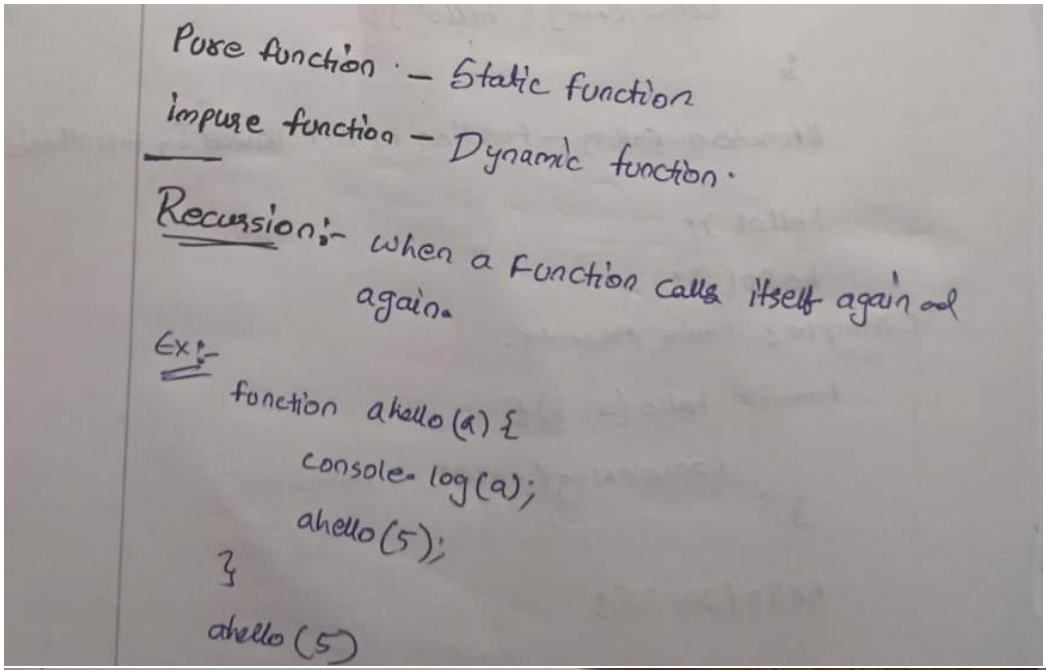
nctions > practice.html > html > body > script

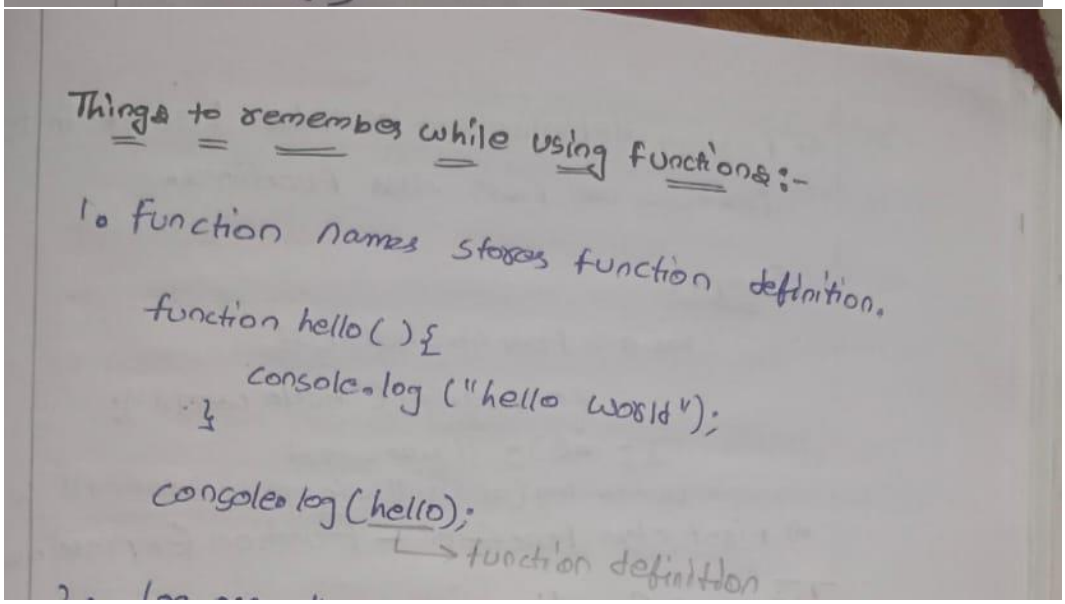
```
<html lang="en">
<body>
    <script>
        function even(a) {
            for(i=1;i<=a;i++){
                if(i%2==0){
                    console.log(i+"is even number");
                }
            }
        }
        even(10);
    </script>
```

Elements | Console

top

```
2is even number
4is even number
6is even number
8is even number
10is even number
```





2. Log off the function calling stores return value. or, return value stores in function calling.

Example:-

```
function hello(){
    console.log("hello world");
    return "hi world";
}
console.log(hello());
```

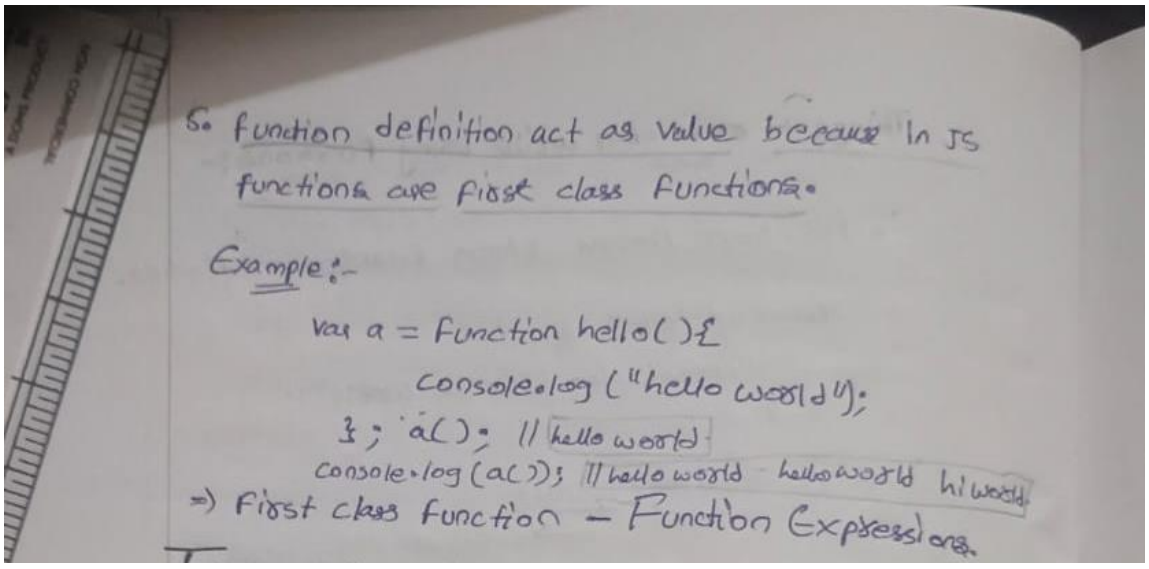
stores return value after execution

3. Statement after return will not execute because it was in void.

Example:-

```
function hello(){
    return "hello"; // it terminates from here
    console.log("This line will not print");
}
```

4. Named functions can only hoisted.



5. function definition act as value because in JS functions are first class functions.

Example:-

```
var a = function hello(){
        console.log("hello world");
};
a(); // hello world
console.log(a()); // hello world  hello world  hi world
```

=> First class function - Function Expressions.

Types of Functions

Annonymous function :-

Annonymous function is a function defined without a name.

Example :-

```
var a = function(){
        console.log("hello");
};
a(); //hello
```

Arrow function (ES-6)

Simple

Arrow function is a concise way of writing function is shorter way.

Example :-

```
var a = () => {
        console.log("hello");
};
a();
```

(or)

```
var a = () => console.log("hello");
a();
```

Named function :- Normal function.

Immediately Invoked Function Expression (IIFE) {}()

An IIFE is a JS function that runs as soon as it is defined.

Example:-

```
(
function (){
    console.log ("hello world");
}) ()
```

Default parameters (ES-6) :- when calling function with out arguments values are set undefined. Some times it is acceptable, but sometimes it is better to assign a default value to the parameters.

Example:-

```
function greet(a = "dear"){
    console.log("hello " + a);
}
greet("John");  // hello John
great ("");  // hello Dear
```

Example:

re assign a value

```
function greet(a, a, b){
    console.log(a, a, b);  // 3 3 5
}
greet(2, 3, 5)
```

Example:- to access nested function

```
function greet(){
    console.log("first function");
    return function greet2(){
        console.log("seconde function");
    }
}
greet()();
```

Research

1. Currying

Call back Function:-

call back function is a function deffinition passed into another function as a argument which is the invoked inside the outer function to complete some kind of task.

Example:-

```
function hi(){
    console.log("hi");
}

function hello(a, b, c){
    consdle.log(a, b, c);  // John 23 functiondefination
    c();                   // hi
}
hello("John", 23, hi);
```

Example:-

```
function payment(){
    Console.log("Payment Recived");
}
function order(a){
    Console.log("Order Placed");
    a();
}

order(payment);
```

Higher Order function (hof):-

hof is function which takes one or more functions as arguments is called hof.

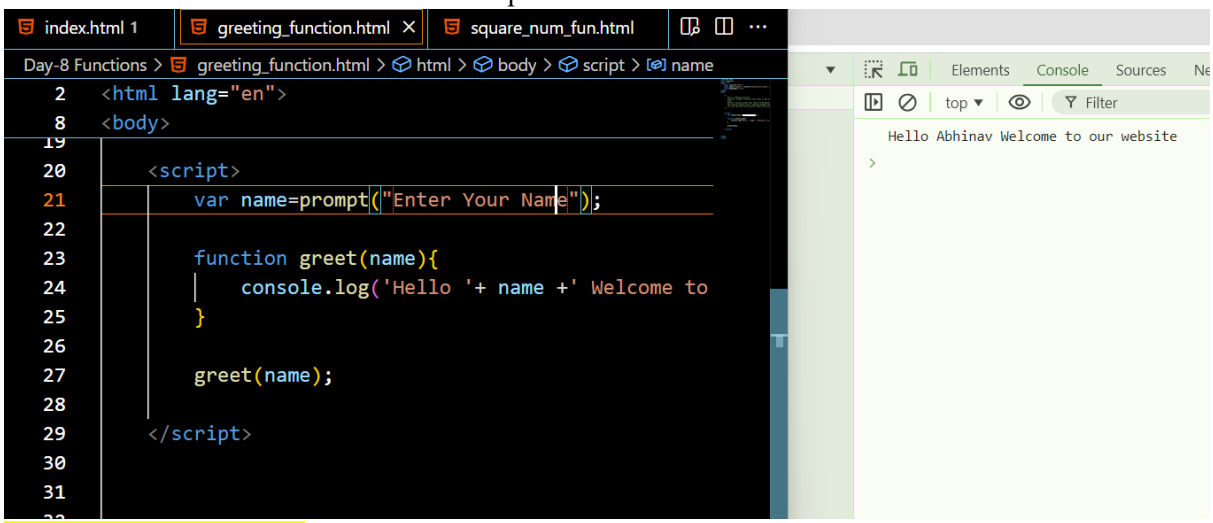
order(Payment);

↓ hof ↓ call back

Tasks

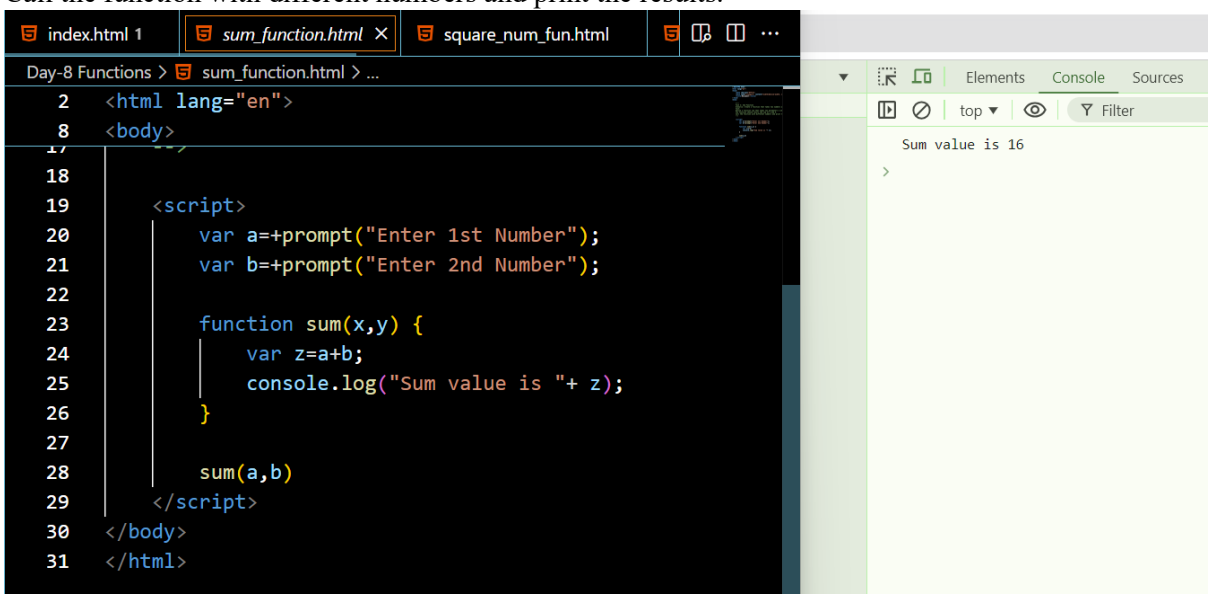
Task 1: Greeting Function

Scenario: Create a function that takes a name as an argument and returns a greeting message.
Task:
Define a function greet that takes one parameter name.
The function should print a greeting message like “Hello, [name]!”.
Call the function with different names and print the results.



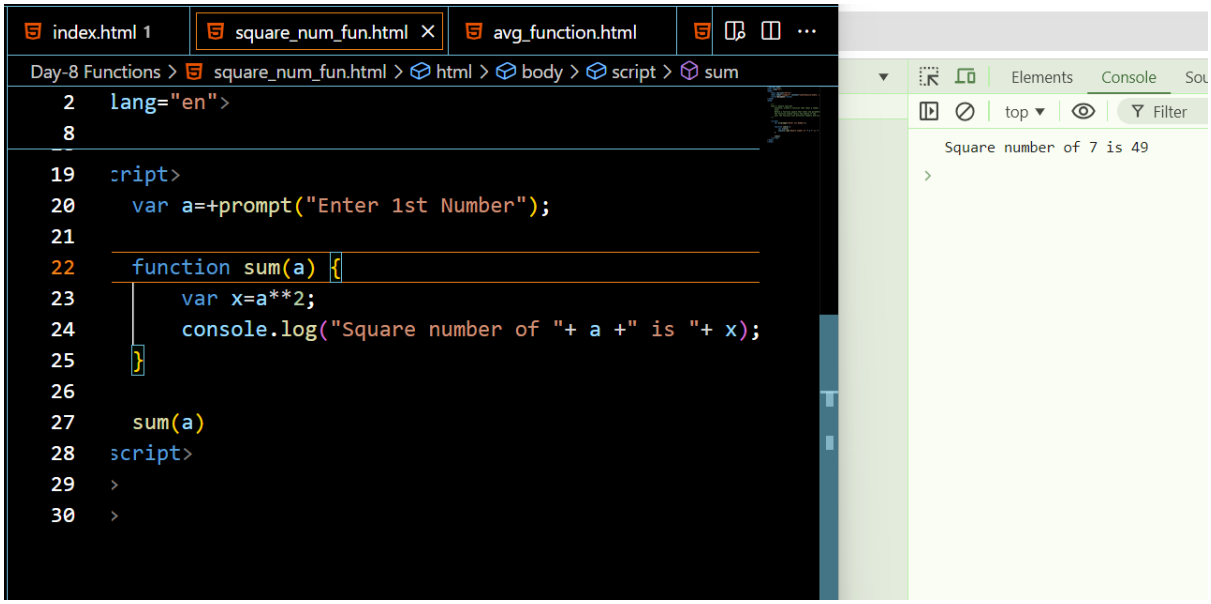
Task 2: Sum Function

Scenario: Create a function that takes two numbers as arguments and returns their sum.
Task:
Define a function sum that takes two parameters a and b.
The function should return the sum of a and b.
Call the function with different numbers and print the results.



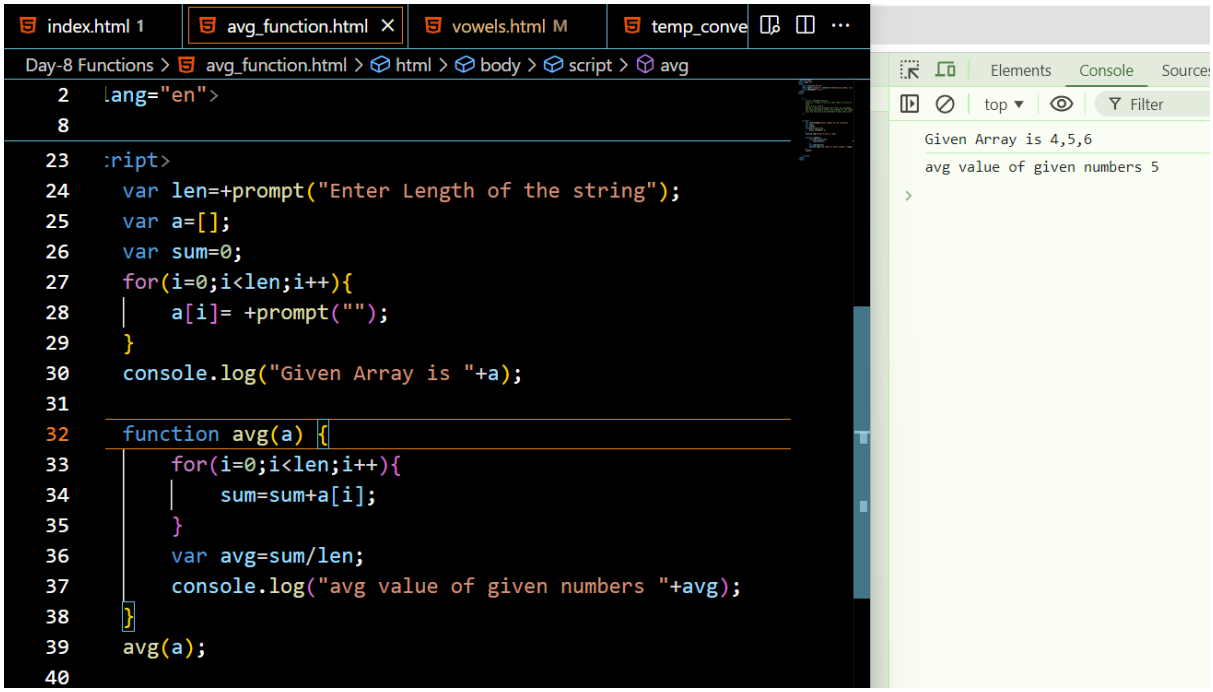
Task 3: Square Function

Scenario: Create a function that takes a number as an argument and returns its square.
Task:
Define a function square that takes one parameter num.
The function should return the square of num.
Call the function with different numbers and print the results.



Task 4: Average Function

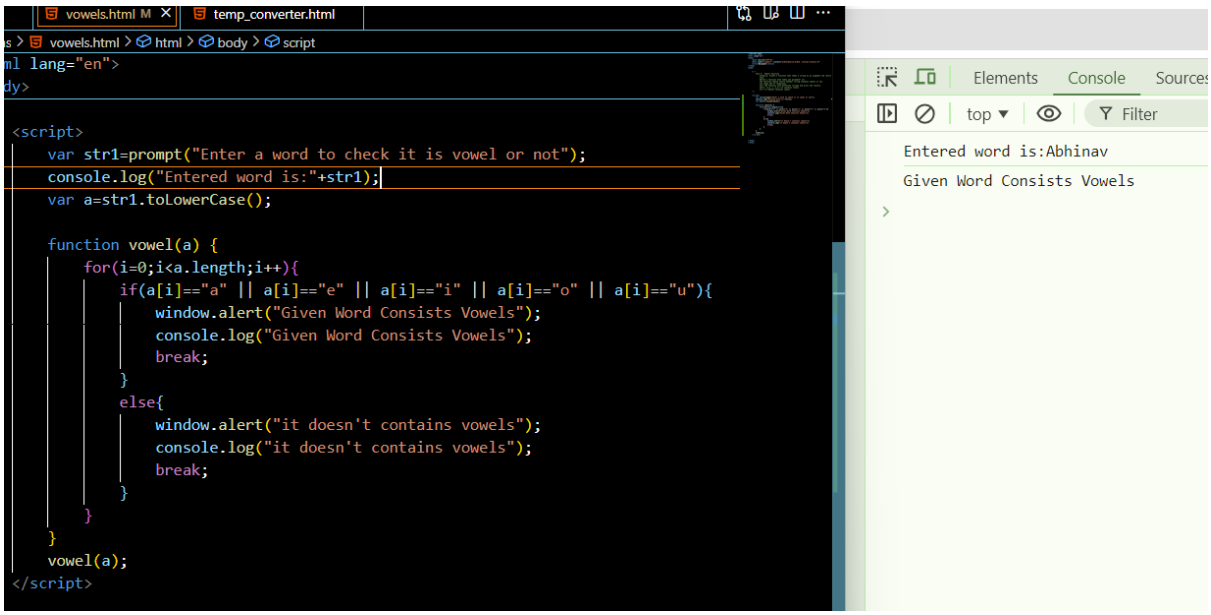
Scenario: Create a function that takes an array of numbers as an argument and returns the average.
Task:
hint:[2,5,2] =9/3=3
Define a function average that takes one parameter arr.
The function should return the average of the numbers in arr.
Call the function with different arrays and print the results.

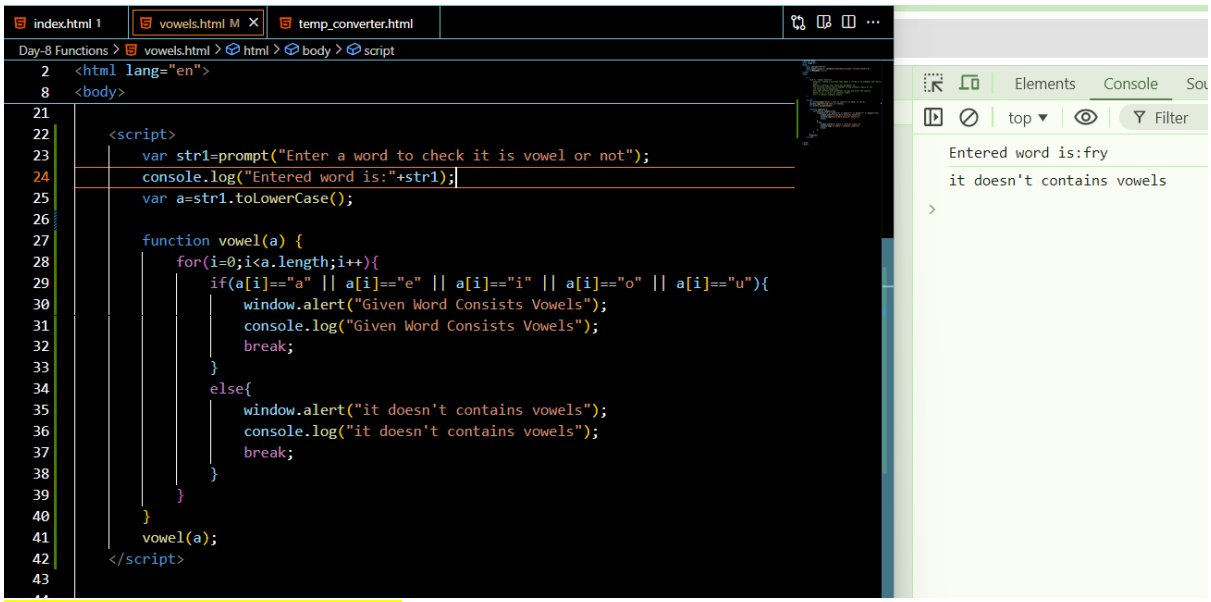


Task 5: Vowels Function

Scenario: Create a function that takes a string as an argument and returns weather the strings contains vowels or not .
Task:
Define a function that takes one parameter str.
The function should return whether string contains vowels or not.

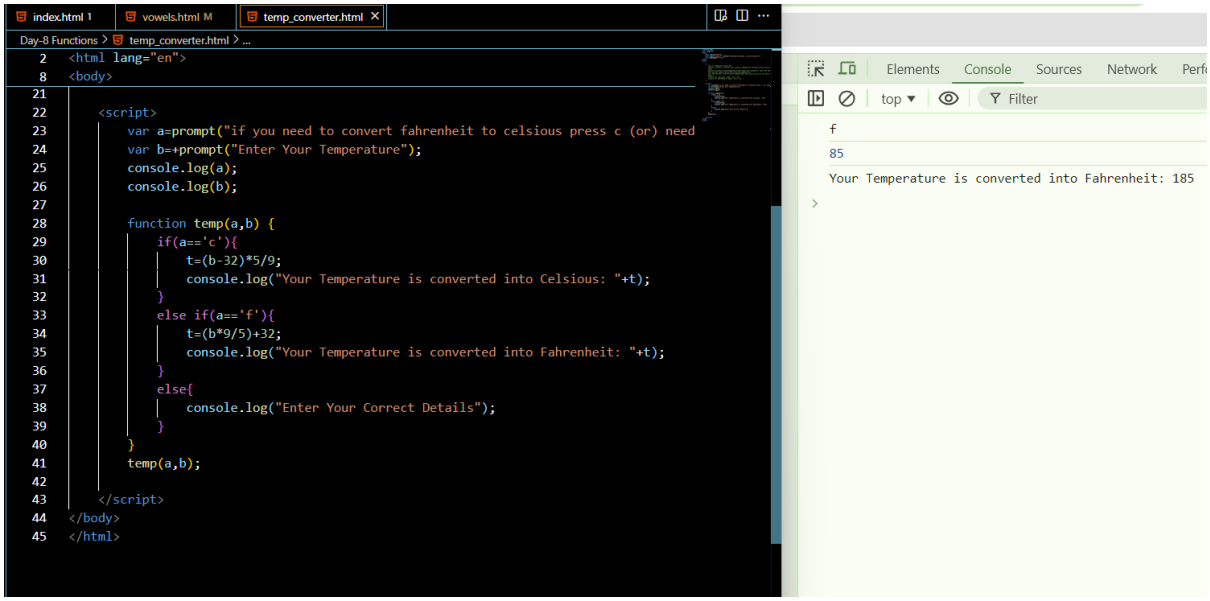
use loops and if conditions
Call the function with different strings and print the results.
hello --a,e ,i ,o,u---it contains vowels
hll---it doesnt contains vowels

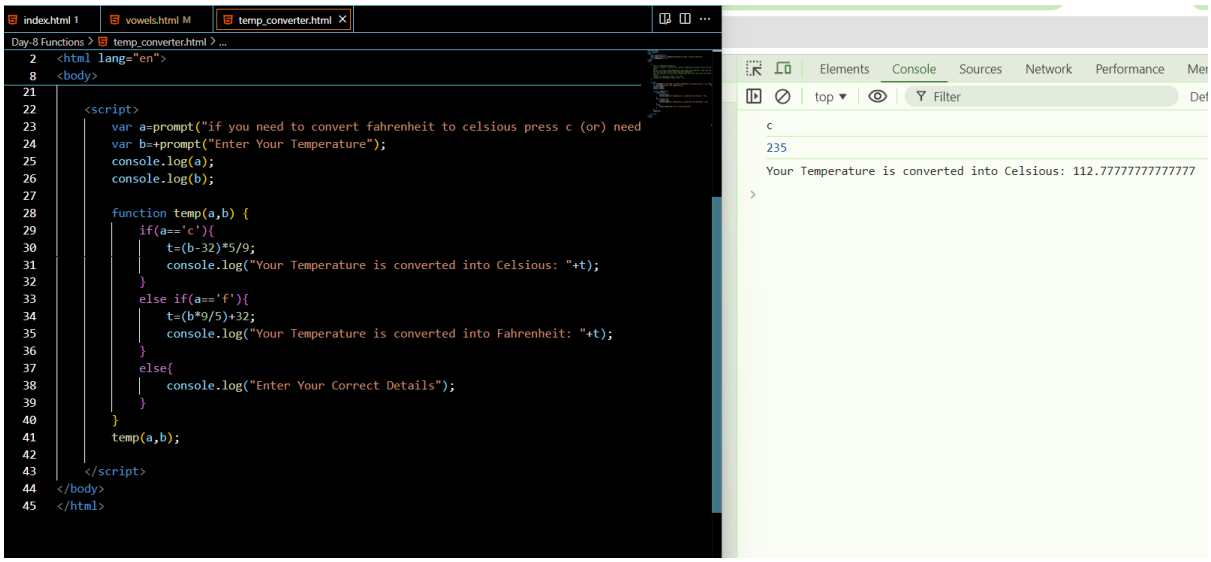




Task 6: Temperature Converter

Scenario: Create a function that converts temperatures between Celsius and Fahrenheit.
Task:
Define a function convertTemperature that takes two parameters: temp (the temperature) and scale (the scale to convert to, either “C” or “F”).
The function should return the converted temperature.
Test the function with different temperatures and scales and print the results.
Hints:-
formula for celscious (temp - 32) * 5/9
formula for fahrehnheit (temp * 9/5) + 32


index.html 1
vowels.html M
temp_converter.html
Day-8 Functions > temp_converter.html > ...
2 <html lang="en">
8 <body>
21
22 <script>
23 var a=prompt("if you need to convert fahrenheit to celsious press c (or) need
24 var b=+prompt("Enter Your Temperature");
25 console.log(a);
26 console.log(b);
27
28 function temp(a,b) {
29 if(a=='c'){
30 t=(b-32)*5/9;
31 console.log("Your Temperature is converted into Celsious: "+t);
32 }
33 else if(a=='f'){
34 t=(b*9/5)+32;
35 console.log("Your Temperature is converted into Fahrenheit: "+t);
36 }
37 else{
38 console.log("Enter Your Correct Details");
39 }
40 }
41 temp(a,b);
42
43 </script>
44 </body>
45 </html>
Elements
Console
Sources
Network
top
Filter
f
85
Your Temperature is converted into Fahrenheit: 185


index.html 1
vowels.html M
temp_converter.html
Day-8 Functions > temp_converter.html > ...
2 <html lang="en">
8 <body>
21
22 <script>
23 var a=prompt("if you need to convert fahrenheit to celsious press c (or) need
24 var b=+prompt("Enter Your Temperature");
25 console.log(a);
26 console.log(b);
27
28 function temp(a,b) {
29 if(a=='c'){
30 t=(b-32)*5/9;
31 console.log("Your Temperature is converted into Celsious: "+t);
32 }
33 else if(a=='f'){
34 t=(b*9/5)+32;
35 console.log("Your Temperature is converted into Fahrenheit: "+t);
36 }
37 else{
38 console.log("Enter Your Correct Details");
39 }
40 }
41 temp(a,b);
42
43 </script>
44 </body>
45 </html>
Elements
Console
Sources
Network
Performance
top
Filter
c
235
Your Temperature is converted into Celsious: 112.77777777777777