

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: {}

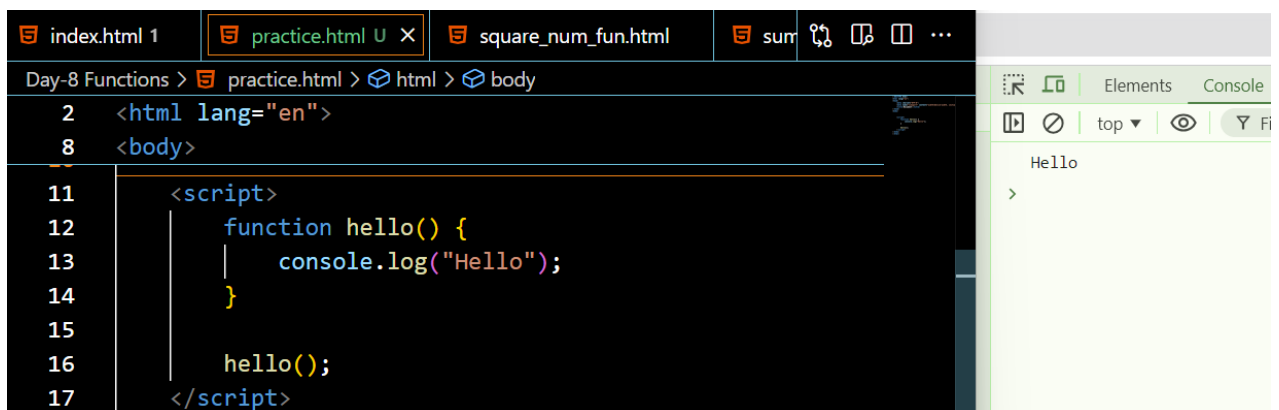
Function parameters are listed inside the parentheses () in the function definition. Function arguments are the values received by the function when it is invoked. Inside the function, the arguments (the parameters) behave as local variables.

Named functions can be hoisted

Syntax:

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

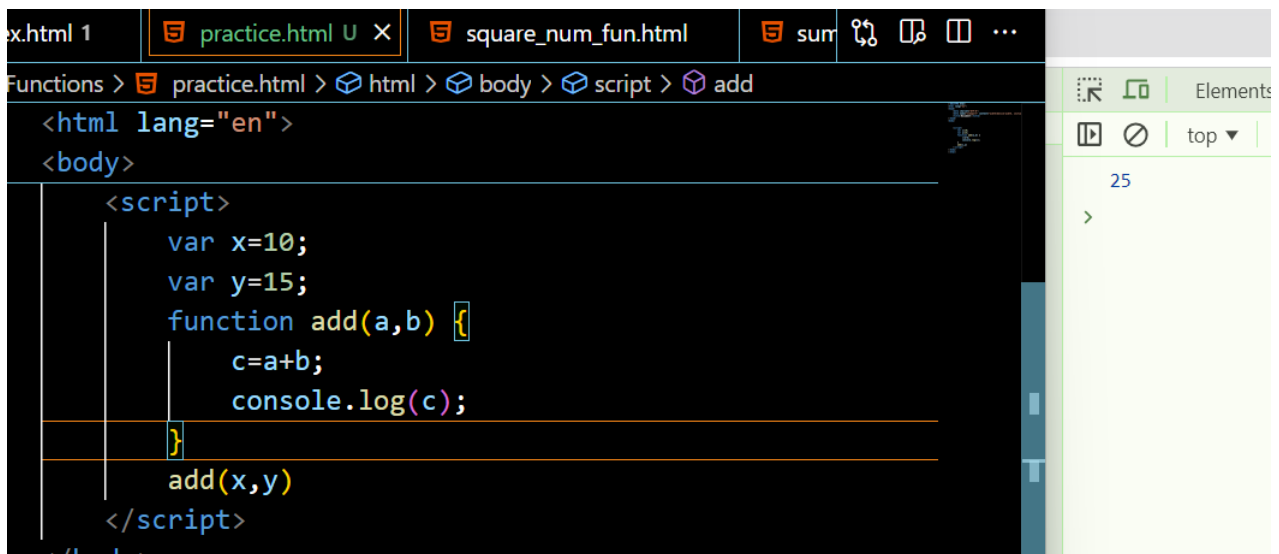
Example:



The screenshot shows a web browser with a single tab titled 'practice.html'. The browser's developer tools are open, showing the 'Elements' panel on the right. The 'Console' panel is also open, displaying the output 'Hello'. The code in the browser is as follows:

```
<html lang="en">  
<body>  
  
  <script>  
    function hello() {  
      console.log("Hello");  
    }  
  
    hello();  
  </script>  
</body>
```

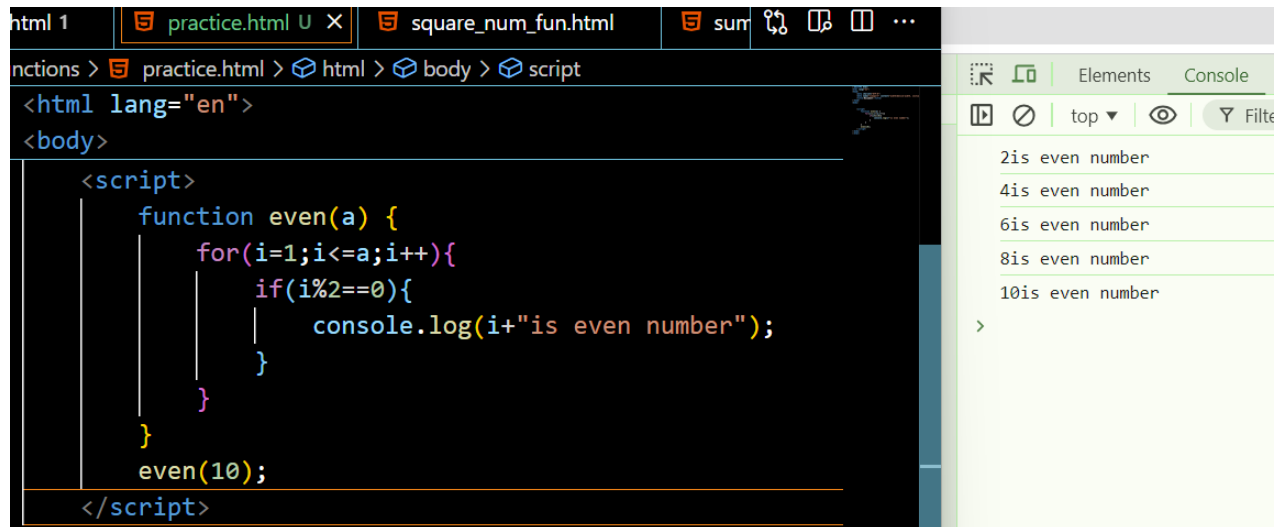
Example: adding two numbers



The screenshot shows a web browser with a single tab titled 'practice.html'. The browser's developer tools are open, showing the 'Elements' panel on the right. The 'Console' panel is also open, displaying the output '25'. The code in the browser is as follows:

```
<html lang="en">  
<body>  
  
  <script>  
    var x=10;  
    var y=15;  
    function add(a,b) {  
      c=a+b;  
      console.log(c);  
    }  
  
    add(x,y)  
  </script>  
</body>
```

Example: even numbers in range



```
<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>
</body>
</html>
```

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

Pure function - Static function
Impure function - Dynamic function.

Recursion:- when a function calls itself again and again.

Ex:-

```
function ahello(a) {
  console.log(a);
  ahello(5);
}
```

ahello(5)

Things to remember while using functions:-

1. function names stores function definition.

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

console.log(hello);

↳ function definition

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";
}

stores return value after execution

console.log(hello());

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

Anonymous function:-

Anonymous 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 parameter.

Example:-

```
function greet(a = "dear") {  
    console.log("hello " + a);  
    greet("John"); // hello John  
    greet(""); // 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("second function");  
    };  
    greet2()();  
}
```

Researcher c
1. Crying

Call back Function:-

call back function is a function definition 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) {  
  console.log (a, b, c);  
  c();  
}  
hello ("John", 23, hi);
```

// John 23 function definition
// 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

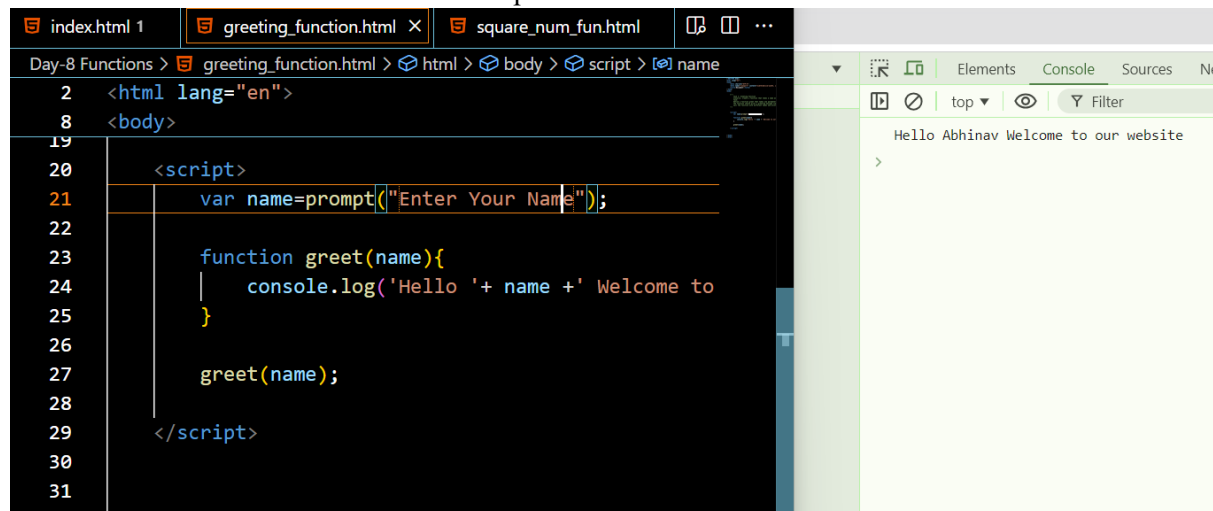
Scenario: Create a function that takes a name as an argument and returns a greeting message.

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.



```
index.html 1 | greeting_function.html x | square_num_fun.html | ...
Day-8 Functions > greeting_function.html > html > body > script > name
2 <html lang="en">
8 <body>
19
20 <script>
21   var name=prompt("Enter Your Name");
22
23   function greet(name){
24     console.log('Hello '+ name + ' Welcome to
25   }
26
27   greet(name);
28
29 </script>
30
31
```

Elements Console Sources Ne

top Filter

Hello Abhinav Welcome to our website

Task 2: Sum Function

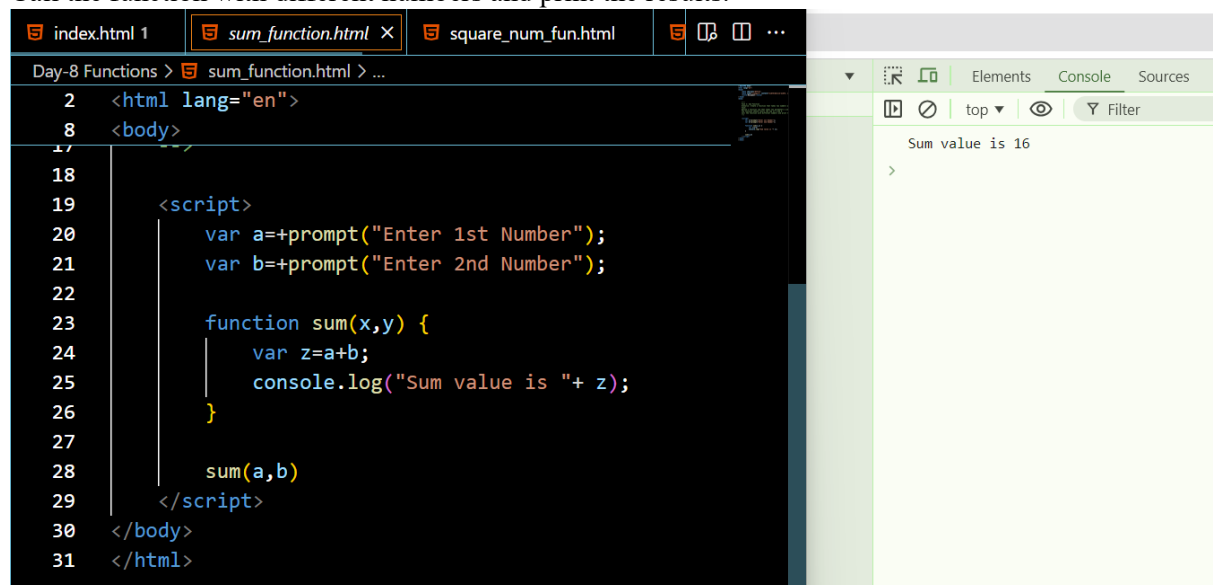
Scenario: Create a function that takes two numbers as arguments and returns their sum.

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.



```
index.html 1 | sum_function.html x | square_num_fun.html | ...
Day-8 Functions > sum_function.html > ...
2 <html lang="en">
8 <body>
17
18
19 <script>
20   var a+=prompt("Enter 1st Number");
21   var b+=prompt("Enter 2nd Number");
22
23   function sum(x,y) {
24     var z=a+b;
25     console.log("Sum value is "+ z);
26   }
27
28   sum(a,b)
29 </script>
30 </body>
31 </html>
```

Elements Console Sources

top Filter

Sum value is 16

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.

```
index.html 1 | square_num_fun.html X | avg_function.html | ...
Day-8 Functions > square_num_fun.html > html > body > script > sum
2  lang="en">
8
19  :ript>
20  var a+=prompt("Enter 1st Number");
21
22  function sum(a) {
23      var x=a**2;
24      console.log("Square number of "+ a +" is "+ x);
25  }
26
27  sum(a)
28  :script>
29  >
30  >
```

Elements Console Sou

top Filter

Square number of 7 is 49

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.

```
index.html 1 | avg_function.html X | vowels.html M | temp_conve | ...
Day-8 Functions > avg_function.html > html > body > script > avg
2  lang="en">
8
23  :ript>
24  var len+=prompt("Enter Length of the string");
25  var a=[];
26  var sum=0;
27  for(i=0;i<len;i++){
28      a[i]= +prompt("");
29  }
30  console.log("Given Array is "+a);
31
32  function avg(a) {
33      for(i=0;i<len;i++){
34          sum=sum+a[i];
35      }
36      var avg=sum/len;
37      console.log("avg value of given numbers "+avg);
38  }
39  avg(a);
40
```

Elements Console Source

top Filter

Given Array is 4,5,6

avg value of given numbers 5

Task 5: Vowels Function

Scenario: Create a function that takes a string as an argument and returns whether the string contains vowels or not.

Task:

Define a function that takes one parameter str.

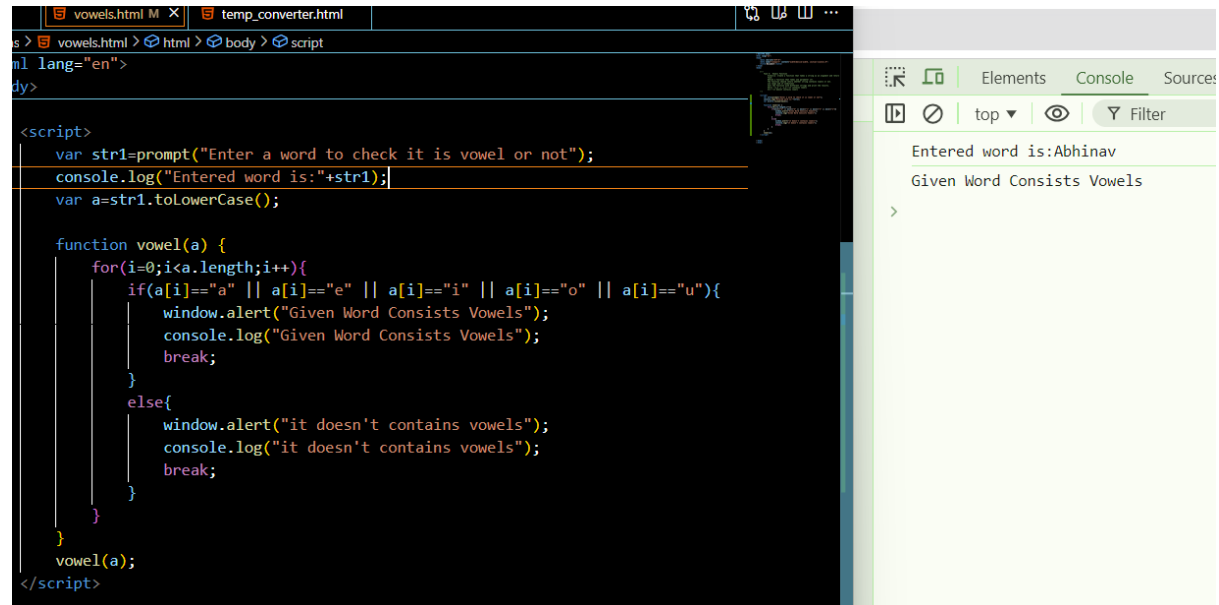
The function should return whether string contains vowels or not.

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 doesn't contain vowels

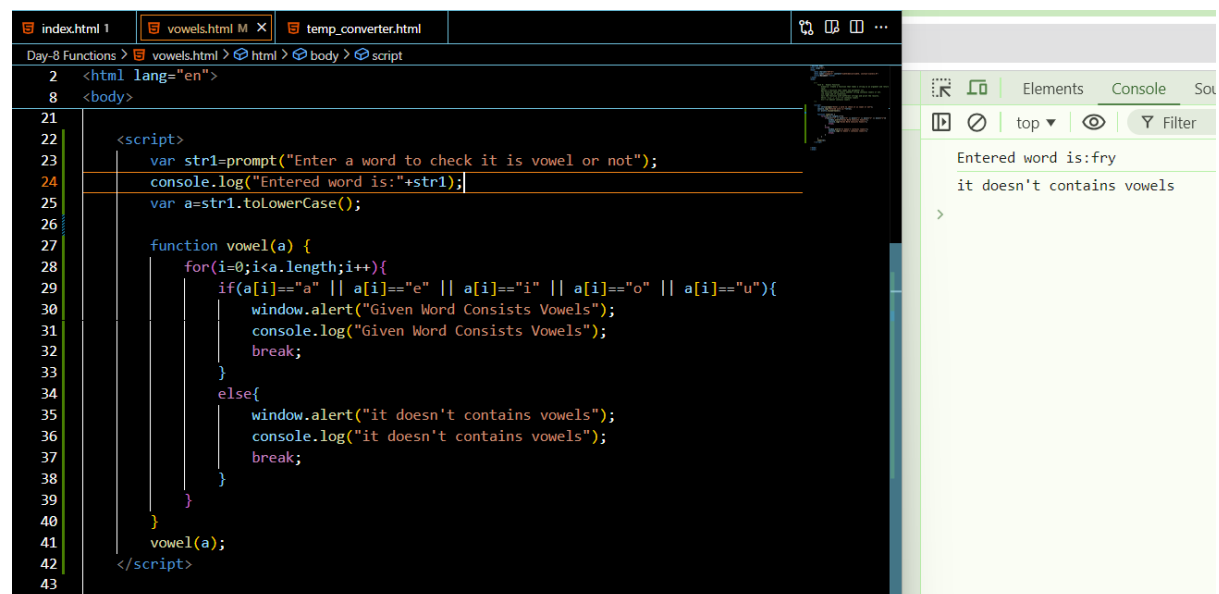


```
<script>
var str1=prompt("Enter a word to check it is vowel or not");
console.log("Entered word is:"+str1);
var a=str1.toLowerCase();

function vowel(a) {
    for(i=0;i<a.length;i++){
        if(a[i]=="a" || a[i]=="e" || a[i]=="i" || a[i]=="o" || a[i]=="u"){
            window.alert("Given Word Consists Vowels");
            console.log("Given Word Consists Vowels");
            break;
        }
        else{
            window.alert("it doesn't contains vowels");
            console.log("it doesn't contains vowels");
            break;
        }
    }
}

vowel(a);
</script>
```

Entered word is: Abhinav
Given Word Consists Vowels



```
<script>
var str1=prompt("Enter a word to check it is vowel or not");
console.log("Entered word is:"+str1);
var a=str1.toLowerCase();

function vowel(a) {
    for(i=0;i<a.length;i++){
        if(a[i]=="a" || a[i]=="e" || a[i]=="i" || a[i]=="o" || a[i]=="u"){
            window.alert("Given Word Consists Vowels");
            console.log("Given Word Consists Vowels");
            break;
        }
        else{
            window.alert("it doesn't contains vowels");
            console.log("it doesn't contains vowels");
            break;
        }
    }
}

vowel(a);
</script>
```

Entered word is: fry
it doesn't 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 celscius $(temp - 32) * 5/9$

formula for fahrehnheit $(temp * 9/5) + 32$

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

Elements Console Sources Network Perf

top top Filter

f

85

Your Temperature is converted into Fahrenheit: 185

>

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

Elements Console Sources Network Performance Met

top top Filter

c

235

Your Temperature is converted into Celsious: 112.77777777777777

>