

Task-1

Aim:

Declare a variable using var, let, and const. Assign different data types to each variable and print their values.

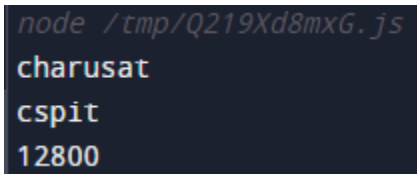
Description: Variables are Containers for Storing Data.

JavaScript Variables can be declared in 4 ways:

- Automatically
- Using var
- Using let
- Using const

Source Code:

```
var x='charusat';  
console.log(x);  
let y='cspit';  
console.log(y);  
const z=12800  
console.log(z);
```

Output:

```
node /tmp/Q219Xd8mxG.js  
charusat  
cspit  
12800
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-2

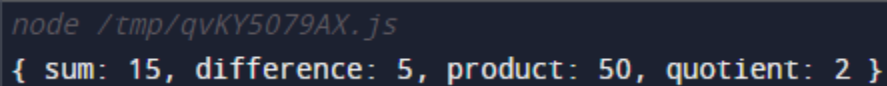
Aim:

Write a function that takes two numbers as arguments and returns their sum, difference, product, and quotient using arithmetic operators.

Description : The outcome of adding two or more numbers gives the sum. The outcome of subtracting the two numbers gives the difference. The outcome of multiplying the two or more numbers gives the product. The result of the division of one number by another is the quotient.

Source Code:

```
function calculateOperations(a, b) {  
  var sum = a + b;  
  var difference = a - b;  
  var product = a * b;  
  var quotient = a / b;  
  return {  
    sum: sum,  
    difference: difference,  
    product: product,  
    quotient: quotient  
  };  
}  
// Usage example  
var result = calculateOperations(10, 5);  
console.log(result);
```

Output:

```
node /tmp/qvKY5079AX.js  
{ sum: 15, difference: 5, product: 50, quotient: 2 }
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-3

Aim:

Write a program that prompts the user to enter their age. Based on their age, display different messages:

- If the age is less than 18, display "You are a minor."
- If the age is between 18 and 65, display "You are an adult."
- If the age is 65 or older, display "You are a senior citizen."

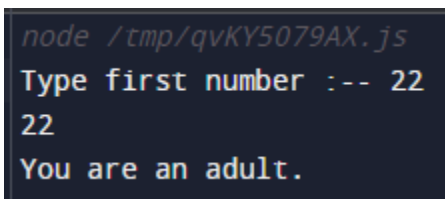
Description: Understanding of basic Python syntax, including if-elif-else statements.

Input Handling and conversion of user input from string to integer. Function implementation and usage.

Source Code:

```
function Age() {  
  var x=prompt("Type first number");  
  if(x<18)  
    return("You are a minor.");  
  else if(x>18 && x<65)  
    return("You are an adult.");  
  else if(x>65)  
    return("You are a senior citizen.");  
}  
var result = Age();  
console.log(result);
```

Output:



```
node /tmp/qvKY5079AX.js  
Type first number :-- 22  
22  
You are an adult.
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-4

Aim:

Write a function that takes an array of salary as an argument and returns the min/max salary in the array.

Description:

- Use of conditional statements to track the minimum and maximum values.
- Proper initialization and updating of variables within a loop.
- Returning multiple values from a function using a tuple.

Source Code:

```
function findMinMaxSalary(salaries) {  
  var minSalary = Math.min(...salaries);  
  var maxSalary = Math.max(...salaries);  
  return {  
    minSalary: minSalary,  
    maxSalary: maxSalary  
  };  
}  
  
// Usage example  
var salaries = [3000, 5000, 2000, 8000, 4000];  
var result = findMinMaxSalary(salaries);  
console.log(result);
```

Output:

```
node /tmp/qvKY5079AX.js  
{ minSalary: 2000, maxSalary: 8000 }
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

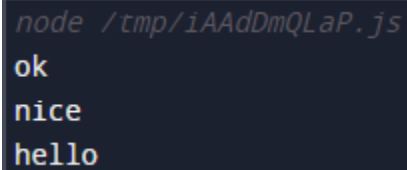
Task-5

Aim:

Create an array of your favorite books. Write a function that takes the array as an argument and displays each book title on a separate line.

Source Code:

```
// Array of favorite books
const favoriteBooks = [
  'ok',
  'nice',
  'hello',
];
function displayBooks(books) {
  for (let i = 0; i < books.length; i++) {
    console.log(books[i]);
  }
}
displayBooks(favoriteBooks);
```

Output:

```
node /tmp/iAAdDmQLaP.js
ok
nice
hello
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-6

Aim:

Declare a variable inside a function and try to access it outside the function. Observe the scope behavior and explain the results. [var vs let vs const].

Description:

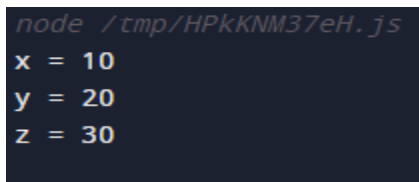
Scope in JavaScript refers to the current context of code, which determines the accessibility of variables to JavaScript. The two types of scope are local and global:

- Global variables are those declared outside of a block
- Local variables are those declared inside of a block

Source Code:

```
var x;  
let y;  
const z = 30;  
function testScope() {  
  x = 10;  
  y = 20;  
}  
testScope();  
console.log("x =",x);  
console.log("y =",y);  
console.log("z =",z);
```

Output:



```
node /tmp/HPkKNM37eH.js  
x = 10  
y = 20  
z = 30
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-7

Aim:

Create an HTML page with a button. Write JavaScript code that adds an event listener to the button and changes its text when clicked.

Description:

The `addEventListener()` method attaches an event handler to an element without overwriting existing event handlers.

When using the `addEventListener()` method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.

Source Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Button Text Change Example</title>
</head>
<body>
  <button id="myButton">Click me!</button>
  <script>
    // Get the button element by its ID
    var button = document.getElementById("myButton");
    // Add event listener to the button
    button.addEventListener("click", function() {
      // Change the button's text
      button.innerText = "Button clicked!";
    });
  </script>
</body>
</html>
```

Output:**Learning Outcome:**

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-8

Aim:

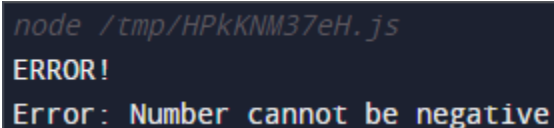
Write a function that takes a number as an argument and throws an error if the number is negative. Handle the error and display a custom error message.

Description:

The try statement defines a code block to run (to try). The catch statement defines a code block to handle any error.

Source Code:

```
function checkNumber(number) {  
  if (number < 0) {  
    throw new Error("Number cannot be negative");  
  }  
  console.log("Number is valid:", number);  
}  
try {  
  checkNumber(-5);  
} catch (error) {  
  console.log("Error:", error.message);  
}
```

Output:

```
node /tmp/HPkKNM37eH.js  
ERROR!  
Error: Number cannot be negative
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Task-9

Aim:

Write a function that uses `setTimeout` to simulate an asynchronous operation. Use a callback function to handle the result.

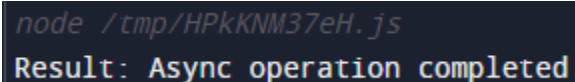
Description:

In order to create the timeout, one can use an asynchronous function that takes in a Promise, as well as time limit. Since JavaScript's native `setTimeout` function uses milliseconds as a parameter, one can use the same to keep things simple.

Source Code:

```
function simulateAsyncOperation(callback) {  
  setTimeout(function() {  
    const result = "Async operation completed";  
    callback(result);  
  }, 2000); // Delay of 2000 milliseconds (2 seconds)  
}  
function handleResult(result) {  
  console.log("Result:", result);  
}  
simulateAsyncOperation(handleResult);
```

Output:



```
node /tmp/HPkKNM37eH.js  
Result: Async operation completed
```

Learning Outcome:

1. CO1- Understand various technologies and trends impacting single page web applications.

Conclusion: In this practical , I learned about Javascript.