

ICFAI University, Dehradun

Assignment – 2

Web Technology (CST-3202)

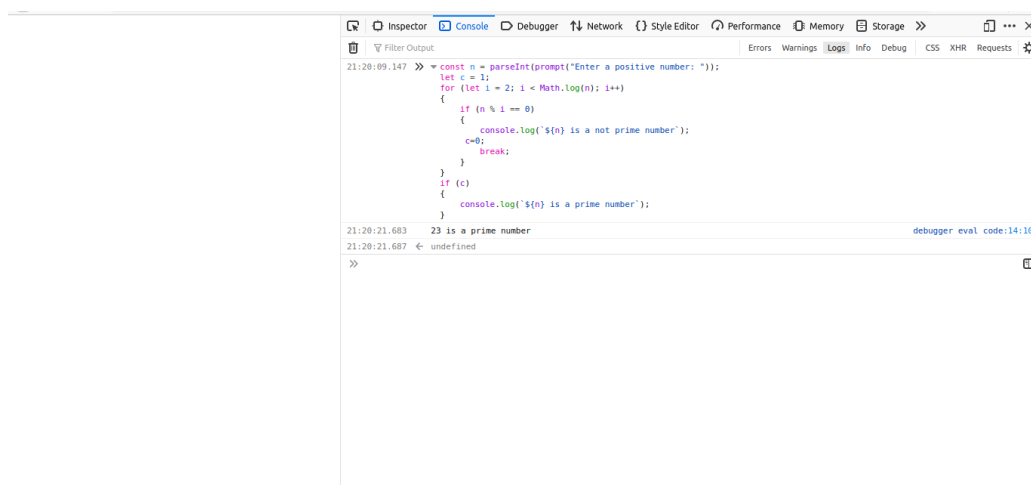
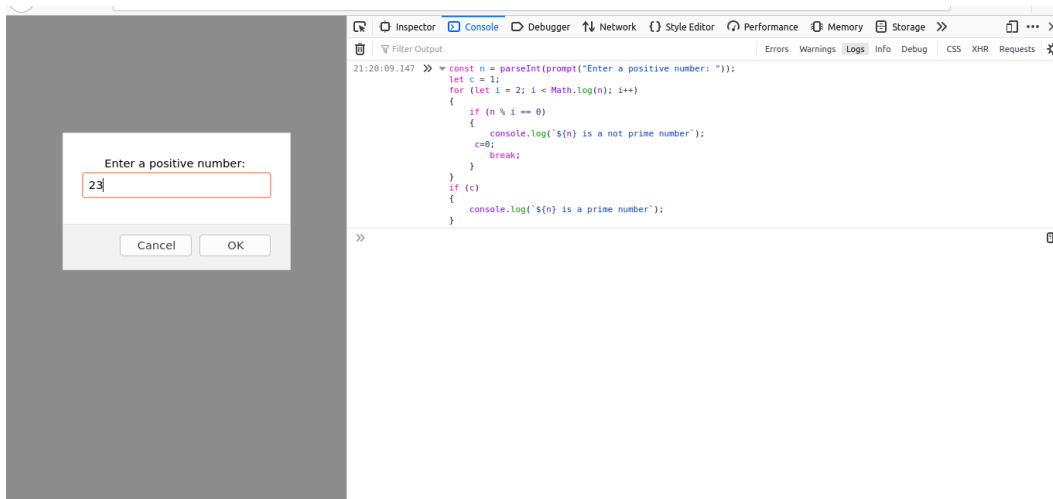
By : -

Ashirbad Sarangi

18STUCDDN01008

Prime Number :

```
const n = parseInt(prompt("Enter a positive number: "));
let c = 1;
for (let i = 2; i < Math.log(n); i++)
{
    if (n % i == 0)
    {
        console.log(`${n} is a not prime number`);
        c=0;
        break;
    }
}
if (c)
{
    console.log(`${n} is a prime number`);
}
```



Enter a positive number:

Cancel OK

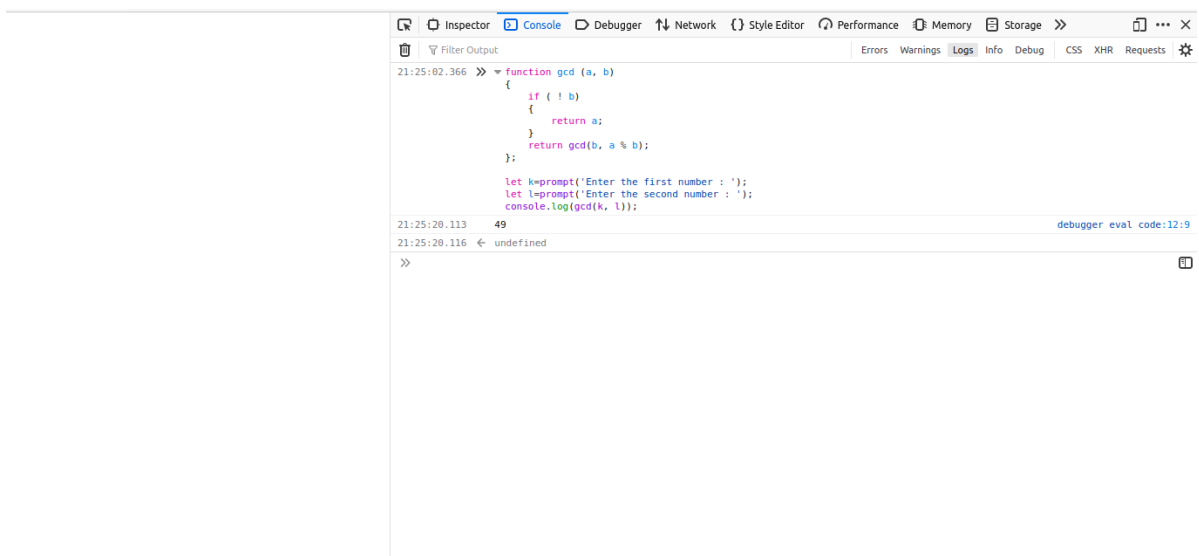
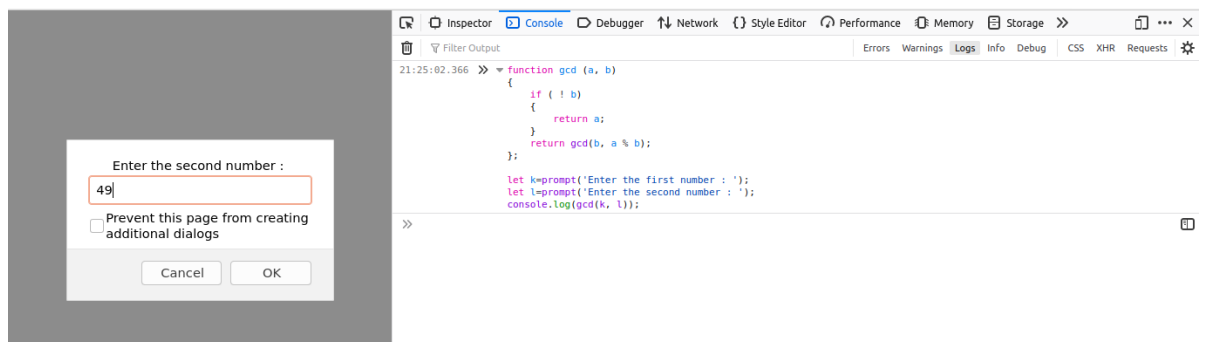
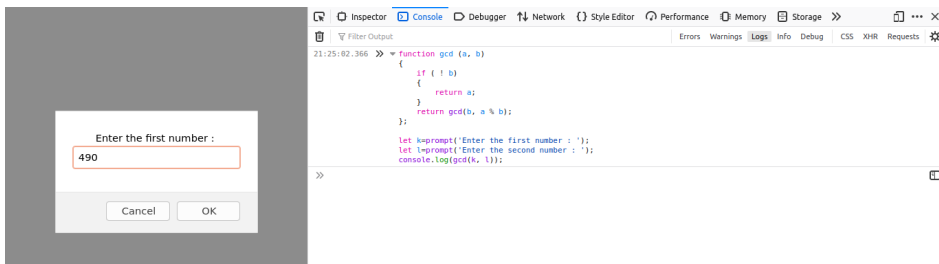
```
Inspector Console Debugger Network Style Editor Performance Memory Storage
Filter Output
21:20:46.774 >> const n = parseInt(prompt("Enter a positive number: "));
let c = 1;
for (let i = 2; i < Math.log(n); i++)
{
  if (n % i == 0)
  {
    console.log(`${n} is a not prime number`);
    c=0;
    break;
  }
}
if (c)
{
  console.log(`${n} is a prime number`);
}
```

```
Inspector Console Debugger Network Style Editor Performance Memory Storage
Filter Output
21:20:46.774 >> const n = parseInt(prompt("Enter a positive number: "));
let c = 1;
for (let i = 2; i < Math.log(n); i++)
{
  if (n % i == 0)
  {
    console.log(`${n} is a not prime number`);
    c=0;
    break;
  }
}
if (c)
{
  console.log(`${n} is a prime number`);
}
21:20:54.748 12 is a not prime number debugger eval code:7:11
21:20:54.763 < 0
>>
```

GCD :

```
function gcd (a, b)
{
    if ( ! b)
    {
        return a;
    }
    return gcd(b, a % b);
};
```

```
let k=prompt('Enter the first number : ');
let l=prompt('Enter the second number : ');
console.log(gcd(k, l));
```



Calculator :

HTML

```
<!--Desgined and Developed by Ashirbad-->
<html>
  <head>
    <title>Calculator</title>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <div class="calc">
      <input type="text" class="calculator-screen" value="" disabled />
      <div class="calculator-keys">
        <button type="button" class="operator" value="+"></button>
        <button type="button" class="operator" value="-"></button>
        <button type="button" class="operator" value="*>&times;</button>
        <button type="button" class="operator" value="/">&divide;</button>
        <button type="button" value="7">7</button>
        <button type="button" value="8">8</button>
        <button type="button" value="9">9</button>
        <button type="button" value="4">4</button>
        <button type="button" value="5">5</button>
        <button type="button" value="6">6</button>
        <button type="button" value="1">1</button><button type="button"
value="2">2</button>
        <button type="button" value="3">3</button>
        <button type="button" value="0">0</button>
        <button type="button" class="decimal" value=".">.</button>
        <button type="button" class="all-clear" value="all-clear">AC</button>
        <button type="button" class="equal-sign operator" value="=">=</button>
      </div>
    </div>
    <script src="app.js"></script>
  </body>
</html>
```

CSS

```
*
{
  font-size:70%;
}

body
{
  display: flex;
  justify-content: center;
```

```
        align-items: center;
    }

    .calc
    {
        border: 1px solid black;
        width:75%;
    }

    .calculator-screen
    {
        width:100%;
        font-size:10em;
        height:2em;
        border:12px solid white;;
        color: black;
        text-align:left;
    }

    button
    {
        height: 60px;
        border-radius: 30px;
        border: 2px solid teal;
        background:linear-gradient(to top, ivory,white);
        font-size: 2rem;
        color: black;
        box-shadow: 0 0 0 1px black;
    }

    button:hover
    {
        background-color: #eaeaea;
    }

    .operator
    {
        color: black;
    }

    .all-clear
    {
        background-color: #f0595f;
        border-color: #b0353a;
        color:black;
    }

    .all-clear:hover
    {
        background-color: #f17377;
    }
```

```

.equal-sign
{
    background-color: #2e86c0;
    border-color: #337cac;
    color:black;
    height: 100%;
    grid-area: 2 / 4 / 6 / 4; /*clm/row/height/width*/
}

.equal-sign:hover
{
    background-color: #4e9ed4;
}

.calculator-keys
{
    display: grid;
    grid-template-columns: repeat(4, 1fr);
    grid-gap: 20px;
    padding: 20px;
}

```

JS:

```

const calculator =
{
    displayValue: '0',
    firstOperand: null,
    waitingForSecondOperand: false,
    operator: null,
};

function inputDigit(digit)
{
    const { displayValue, waitingForSecondOperand } = calculator;
    if (waitingForSecondOperand === true)
    {
        calculator.displayValue = digit;
        calculator.waitingForSecondOperand = false;
    }

    else
    {
        calculator.displayValue =
            displayValue === '0' ? digit : displayValue + digit;
    }
}

function inputDecimal(dot)
{
    if (calculator.waitingForSecondOperand === true)

```

```

    {
        calculator.displayValue = '0.';
        calculator.waitingForSecondOperand = false;
        return;
    }

    if (!calculator.displayValue.includes(dot))
    {
        calculator.displayValue += dot;
    }
}

function handleOperator(nextOperator)
{
    const { firstOperand, displayValue, operator } = calculator;
    const inputValue = parseFloat(displayValue);
    if (operator && calculator.waitingForSecondOperand)
    {
        calculator.operator = nextOperator;
        return;
    }

    if (firstOperand == null && !isNaN(inputValue))
    {
        calculator.firstOperand = inputValue;
    }

    else if (operator)
    {
        const currentValue = firstOperand || 0;
        const result = calculate(currentValue, inputValue, operator);
        calculator.displayValue = `${parseFloat(result.toFixed(7))}`;
        calculator.firstOperand = result;
    }

    calculator.waitingForSecondOperand = true;
    calculator.operator = nextOperator;
}

function calculate(firstOperand, secondOperand, operator)
{
    if (operator === '+')
    {
        return firstOperand + secondOperand;
    }

    else if (operator === '-')
    {
        return firstOperand - secondOperand;
    }

```



```

    }

    else if (operator === '*')
    {
        return firstOperand * secondOperand;
    }

    else if (operator === '/')
    {
        return firstOperand / secondOperand;
    }
    return secondOperand;
}

```

```

function resetCalculator()
{
    calculator.displayValue = '0';
    calculator.firstOperand = null;
    calculator.waitingForSecondOperand = false;
    calculator.operator = null;
}

```

```

function updateDisplay()
{
    const display = document.querySelector('.calculator-screen');
    display.value = calculator.displayValue;
}

```

```

updateDisplay();
const keys = document.querySelector('.calculator-keys');
keys.addEventListener('click', event =>
{
    const { target } = event;
    const { value } = target;
    if (!target.matches('button')) {
        return;
    }
}

```

```

switch (value)
{
    case '+':
    case '-':
    case '*':
    case '/':
    case '=':
        handleOperator(value);
        break;
    case '.':
        inputDecimal(value);
        break;
}

```

```
    case 'all-clear':
        resetCalculator();
        break;
    default:
        if (Number.isInteger(parseFloat(value)))
        {
            inputDigit(value);
        }
}

updateDisplay();
});
```

The file is online [here](#)

Event Handling : Form Validation

```
<html>
  <head>
    <title>Form Validation</title>
  </head>

  <body style="color:white">
    <form>
      <table cellspacing = "2" cellpadding = "2" border = "1" style="background: linear-gradient(to
top ,crimson,red) fixed" >
        <tr>
          <td align = "right">First Name</td>
          <td><input type = "text"></td>
        </tr>
        <tr>
          <td align = "right">Second Name</td>
          <td><input type = "text"></td>
        </tr>
        <tr>
          <td align = "right">Final Name</td>
          <td><input type = "text"></td>
        </tr>
        <tr>
          <td align = "right">Email</td>
          <td><input type = "text"></td>
        </tr>
        <tr>
          <td align = "right">Phone No</td>
          <td><input type = "int"></td>
        </tr>
        <tr>
          <td align = "right">Country</td>
          <td>
            <select name = "Country">
              <option value = "-1"selected>Choose your Country
              <option value = "1">India</option>
              <option value = "2">INDIA</option>
              <option value = "3">Bharat</option>
              <option value = "4">Hindustan</option>
            </select>
          </td>
        </tr>
        <tr>
          <td align = "right"></td>
          <td><input type = "submit" value = "Submit"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

The file is online [here](#)

Write a program to demonstrate event handling - change color of background at click of button?

```
<!--Designed by Ashirbad Sarangi-->
<html>
  <head>
    <title>Change Background Colour </title>
  </head>

  <body style = "text-align:center;">
    <button onclick = "gfg_Run()">Change</button>
    <script>
      var el_up = document.getElementById("GFG_UP");
      var el_down = document.getElementById("GFG_DOWN"); var str = "Click on
button to change the background color";
      el_up.innerHTML = str;
      function changeColor(color)
      {
        document.body.style.background = color;
      }
      function gfg_Run()
      {
        changeColor('linear-gradient(to top,cyan,black) fixed');
        el_down.innerHTML = "Background Color Changed !!!";
      }
    </script>
  </body>
</html>
```

The file is online [here](#)