All Syllabus Me

**C language**

**Bit and Bytes**

1] 1-8 = Bit

2] 8 Bits = 1 Bytes

3] 1024 Bytes = 1 Kilo Byte

4] 1024 KB = 1 Mega Byte

5] 1024 MB = 1 Giga Byte

6] 1024 GB = 1 Tera Byte

7] 1024 TB = 1 Peta Byte

8] 1024 PB = 1 Exa Byte

9] 1024 EB = 1 Zeta Byte

10] 1024 ZB = 1 Yotta Byte

11] 1024 YB = 1 Bronto Byte

12] 1024 BB = 1 Geop Byte

**Decimal** **Binary** **Octal**

0 0 0

1 1 1

2 2

3 3

4 4

5 5

6 6

7 7

8

9

**Decimal to Binary**

1] 2 1832

2 916 0

2 458 0 Decimal Number

2 229 0 1832

2 114 1 To

2 57 0 Binary Number

2 28 1 11100101000

2 14 0 (1832)10 =(?)2

2 7 0

2 3 1

2 1 1

2 0 1

**Binary to Decimal**

1] (10001101)2 = (?)2

128 64 32 16 8 4 2 1

1 0 0 0 1 1 0 1

128+8+4+1

**(141)10**

2] (010011010)

256 128 64 32 16 8 4 2 1

0 1 0 0 1 1 0 1 0

128+16+8+2

**(128)10**

**Decimal to Octal**

1] (540)10 =(?)8

540/8 67 4

67/8 8 3

8/8 1 0

1/8 0 1

**(1034)8**

2] (128)10=(?)8

128/8 15 4

15/8 1 7

1/8 0 1

**(174)8**

**Divisibility Rule -12**

2] The number ends with even 0,2,4,6 divisible by 2.

**3] Sum of digit and divisible by 3.**

4] Last two digits divided by 4 and then number is divisible by 4.

**5] It last two digits 0 and 5 then the number is divisible by 5.**

6] The number divided by 2 and 3 then the number is divisible by 6.

**7] Calculate last digit and then multiply by two minus the multiplication by remaning number then if the number is divisible by 7.**

8] Calculate the last two digits and add no 4 in that if the addition then the number is divisible by 8.

**9] If number is sum of digits is then the number is divisible by 9.**

10] The number end with 0 the the number is divisible by 10.

**11] If the number difference of altimate sum of digits is divisible by 11.**

12] If the number is divisible by 3 and 4 then the number of divisible by 12.

**Number Pattern**

**1]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number ");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

for(j=1;j<=i;j++)

{

printf(" %d",j);

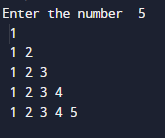
}

printf("\n");

}

}

Output:-



**2]**

**#include<stdio.h>**

**void main()**

**{**

**int i,j,num;**

**printf("Enter the number ");**

**scanf("%d",&num);**

**for(i=num;i>=1;i--)**

**{**

**for(j=num;j>=i;j--)**

**{**

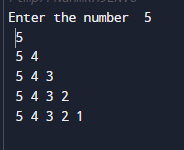
**printf(" %d",j);**

**}**

**printf("\n");**

**}**

**}**



**3]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number ");

scanf("%d",&num);

for(i=num;i>=1;i--)

{

for(j=num;j>=1;j--)

{

if(i>=j)

printf(" %d",i);

else

printf(" ");

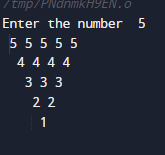
}

printf("\n");

}

}

Output:-



**4]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number ");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

for(j=num;j>=1;j--)

{

if(i<=j)

printf(" %d",i);

else

printf(" ");

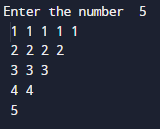
}

printf("\n");

}

}

Output:-



**5]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number ");

scanf("%d",&num);

for(i=1;i<=num;i++)

{

for(j=num;j>=1;j--) output:-

{ 1

if(j<=i) 2 2

printf(" %d",i); 3 3 3

else 4 4 4 4

printf(" "); 5 5 5 5 5

}

printf("\n");

}

}

**6]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number "); output:-

scanf("%d",&num); 1 2 3 4 5

for(i=1;i<=num;i++) 2 3 4 5

{ 3 4 5

for(j=1;j<=num;j++) 4 5

{ 5

if(j>=i)

printf(" %d",j);

}

printf("\n");

}

}

**7]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number "); output:-

scanf("%d",&num); 5

for(i=num;i>=1;i--) 4 5

{ 3 4 5

for(j=1;j<=num;j++) 2 3 4 5

{ 1 2 3 4 5

if(j>=i)

printf(" %d",j);

}

printf("\n");

}

}

**8]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number ");

scanf("%d",&num);

for(i=num;i>=1;i--) output:-

{ 5

for(j=1;j<=num;j++) 4 4

{ 3 3 3

if(j>=i) 2 2 2 2

printf(" %d",i); 1 1 1 1 1

}

printf("\n");

}

}

**9]**

#include<stdio.h>

void main()

{

int i,j,num;

printf("Enter the number "); output:-

scanf("%d",&num); 1 1 1 1 1

for(i=1;i<=num;i++) 2 2 2 2

{ 3 3 3

for(j=num;j>=1;j--) 4 4

{ 5

if(i<=j)

printf(" %d",i);

}

printf("\n");

}

}

**10]**

#include<stdio.h>

void main()

{

for(int i=4;i>=1;i--) outpt:-

{ 1 2 3 4 5 6 7

for(int j=1;j<=i\*2-1;j++) 1 2 3 4 5

{ 1 2 3

printf(" %d",j); 1

}

printf("\n");

}

}

**11]**

#include<stdio.h>

void main()

{

for(int i=1;i<=4;i++)

{

if(i%2==1)

{

for(int j=1;j<=i;j++) output:-

{ 1

if(j%2==0) 0 1

1 0 1

printf(" 0"); 0 1 0 1

else{

printf(" 1");

}

}

}

else

{

for(int j=1;j<=i;j++)

{

if(j%2==1)

printf(" 0");

else{

printf(" 1");

}

}

}

printf("\n");

}

}

**12]**

#include<stdio.h>

void main()

{

for(int i=5;i>=1;i--)

{

if(i%2==1)

{

for(int j=1;j<=i;j++)

{

printf(" %d",j); output:-

1 2 3 4 5

} 4 3 2 1

1 2 3

} 1

else

{

for(int l=i;l>=1;l--)

{

printf(" %d",l);

}

}

printf(" \n");

}

}

**13]**

#include<stdio.h>

void main()

{

for(int i=1;i<=5;i++) output:-

{ 1 2 3 5 7 9

for(int j=i;j<=5;j++) 3 5 7 9

{ 5 7 9

printf(" %d",j\*2-1); 7 9

} 9

printf(" \n");

}

}

**14]**

#include<stdio.h>

void main()

{

for(int i=1;i<=10;i++)

{

for(int j=1;j<=i;j++)

{

printf("%d",j\*i);

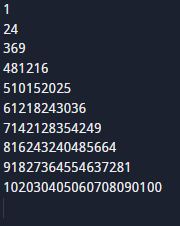
}

printf("\n");

}

}

Output:-



* **Html**

**List Tag**

There are two type of list tag

1] Ordered list

2] Unordered list

**Ordered list**

**<ol>**

<li>HTML & CSS</li>

<li>JavaScript</li>

<li>Pythan</li>

<li>C++</li>

**</ol>**

**Answer:** 1 HTML & CSS

2 JavaScript

3 Pythan

4 C++

**<ol type=”I”>**

<li>Coffee</li>

<li>Cold Coffee</li>

<li>Tea</li>

<li>Black Tea</li>

**</ol>**

**Answer:** I Coffee

II Cold Coffee

III Tea

IV Black Tea

**<ol type=”A”>**

<li>Apple</li>

<li>Mango</li>

<li>Grapes</li>

<li>Orange</li>

**</ol>**

**Answer:** A Apple

B Mango

C Grapes

D Orange

**Unordered list**

**<ul>**

<li>Computer</li>

<li>CPU</li>

<li>Mouse</li>

**</ul>**

**Answer:**

* Computer
* CPU
* Mouse

**<ul type=”square”>**

<li>Pink</li>

<li>Red</li>

<li>Blue</li>

<li>Green</li>

**</ul>**

**Answer:**

* Pink
* Red
* Blue
* Green

**<ul type=”circle”>**

<li>Shri</li>

<li>Shantuli</li>

<li>Pratiksha</li>

<li>Kavita</li>

<li>Komal</li>

**</ul>**

**Answer:**

* Shri
* Shantuli
* Pratiksha
* Kavita
* Komal

**Table tag**

**<table>**

Define:- tad is used to define takes in HTML .

It is used to format and display tutorial data.

**<tr>** difine table row.

**<th>** difine table heading.

**<td>** difine table data.

**<th> example**

Table heading

|  |  |  |
| --- | --- | --- |
| **Year** | **Mont** | **Day** |
|  |  |  |
|  |  |  |

**<td> example**

|  |  |  |
| --- | --- | --- |
| **Year** | **Month** | **Day** |
| 2022 | July | Monday |
| 2023 | August | Saturday |

Table data

**Table tag attributes**

<table **border**=”1px” **width**=”200px” **height**=”100px” **style**=”color: black” style=”background color: red”>

**Table**

|  |  |  |
| --- | --- | --- |
| **Name** | **Subject** | **Marks** |
| Kanchan | Math | 50 |
| Kavita | English | 90 |
| Komal | Marathi | 55 |
| Pratiksha | Biology | 60 |
| Shree | Physics | 70 |
| Shantuli | Chemistry | 80 |
| Percentage | Total = | 405 |

**CSS Syllabus**

**\*Display property\***

***What is display ?***

The display property is the most important CSS property for controlling layout.

**1]display: contents;**

display: contents causes an element's children to appear as if they were direct children of the element's parent, ignoring the element itself.

**2]display:None;**

CSS Display None helps developer to hide the element with display property set to none.

**3]display:block;**

set to block starts on a new line and takes up the available screen width.

**4]display:inline-block;**

 inline-block allows to set a width and height on the element.

**5]display:flex;**

By using display flex in parent element, child elements automatically align like column or row with auto width and auto height.

**6]display:linline;**

Displays an element as an inline element (like <span>). Any height and width properties will have no effect.

**7]display:grid;**

Displays an element as a block-level grid container.

***\*Box Model\****

**What is Box model?**

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model.

1] What is margins?

Clears an area outside the border. The margin is transparent.

2] what is borders?

 A border that goes around the padding and content.

3]what is padding?

 Clears an area around the content. The padding is transparent.

4]what is content.?

The content of the box, where text and images appear.

***Ex:-***

<!DOCTYPE html>

<html>

<head>

<style>

div {

  background-color: lightgrey;

  width: 300px;

  border: 15px solid green;

  padding: 50px;

  margin: 20px;

}

</style>

</head>

<body>

<h2>Demonstrating the Box Model</h2>

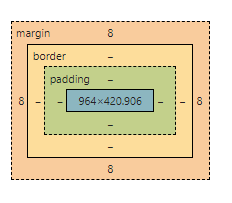
<p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p>

<div>This text is the content of the box. We have added a 50px padding, 20px margin and a 15px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div>

</body>

</html>

Output:-



* **JavaScript**
* **What is JavaScript?**

**Ans :-** JavaScript is the programming languge for the web. JavaScript is a client side scripting language. Which is understood by the browser.

Uses :- create dynamically updating cotatent control multi media animation images and pretty much everything else.

Developed by:-  Brendan Eich in 1995.

* **What is compiler?**

**Ans :-** a compile is translate the language source code into machine code.

Compiler check all type of erros limits and rang it is more intelligent.

The run time of its program is longer and it occupies more memory.

* **What is interpreter?**

**Ans** :- interpreter is converting high-level program statements into machine code.

An interpreter translate only one statements at a time of the program.

An interpreter takes single lines of a code.

The simple rote of an interpreter is to translate the material into a target languge

.

* **Client side scripting?**

**Ans** :- client side scripting means running scripts such as javascript on the client device, usually withan a browser.

Web browsers execute client side scripting.

Source code is visible to the user.

**Usese:-** html,css,javascript are used.

* **server side scripting?**

**Ans** :- server-side scripting is a programing technique for creacting code that may run software on the server side in other words.

Server side scripting is any scripting method that may operate on a web server.

* **What is dynamic?**

**Ans** :- An action that changes the view of the page or adds information to the page without making a sarver request.

* **What is database?**

**Ans** :- Data base is an organized collection of structured information or data typacilly stored electronically in a computer system.

**\*Set program in js\***

* **Common number in single array ?**

let arr=[10,50,20,58,35,58 ,42,10,20];

let i,j;

for(i=0;i<arr.length;i++)

{

    for(j=i+1;j<arr.length;j++)

    {

         if(arr[i]==arr[j])

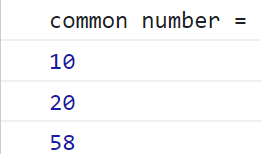
         {

            console.log(arr[i]);

         }

    }

}

Output :-

* **Common number in two dimasitional array ?**

let arr=[10,50,20,58,35,51,42,11,22];

let arr1=[10,54,65,42,87,96,43,15,35];

let i,j;

console.log('common number =')

for(i=0;i<arr.length;i++)

{

    for(j=0;j<arr1.length;j++)

    {

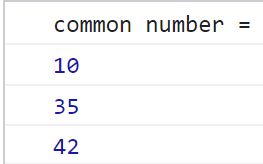
         if(arr[i]==arr1[j])

         {

            console.log(arr[i]);

         }

    }

}Output :- 

* **unique number in single array ?**

let a = [10, 20, 34, 11, 56, 10, 11, 67, 20, 10, 67];

    let flag = 0;

    let c = [];

    let d = [];

    let count = 0;

    for (let i = 0; i < a.length; i++) {

        for (let j = i + 1; j < a.length; j++) {

            if (a[i] == a[j]) {

                c[count] = a[i];

                count++;

            }

        }

    }

    console.log('Unique Elements = ');

    for (let i = 0; i < a.length; i++) {

        flag = 0;

        for (let j = 0; j < c.length; j++) {

            if (a[i] == c[j]) {

                flag = 1;

            }

        }

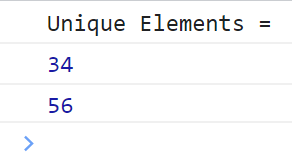
        if (flag == 0) {

             console.log(a[i]);

        }

    }

Output :-



**\*matrix program in js\***

* **b-a ?**

let d = [2, 4, 6, 8, 11, 13, 17];

    let dd = [1, 3, 5, 7, 11, 13, 17];

    let c = 0;

    console.log("B-A =");

    for (let i = 0; i < dd.length; i++) {

        c = 0;

        for (let j = 0; j < d.length; j++) {

            if (dd[i] == d[j]) {

                c = 1;

            }

        }

        if (c == 0) {

            console.log(dd[i]);

        }

    }

Output :-



* **a-b ?**

let e = [2, 4, 6, 8, 11, 12, 17];

    let ee = [1, 3, 5, 7, 11, 12, 17];

    let flag1 = 0;

    console.log("A-B = ");

    for (let i = 0; i < e.length; i++) {

        flag1 = 0;

        for (let j = 0; j < ee.length; j++) {

            if (e[i] == ee[j]) {

                flag1 = 1;

            }

        }

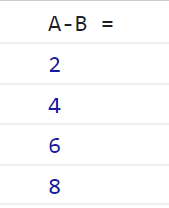
        if (flag1 == 0) {

            console.log(e[i]);

        }

    }

Output :-



* **following elements are compleentary a =?**

  console.log("Following Elements Are Complementary A=");

    let a = [51, 3, 3, 4, 5];

    let b = [6, 5, 3, 4, 3];

    let count = 0;

    let flag = 0;

    let c = [count];

    for (let i = 0; i < 5; i++) {

        for (let j = 0; j < 5; j++) {

            if (a[i] == b[j]) {

                c[count] = a[i];

                count++;

            }

        }

    }

    for (let i = 0; i < count; i++) {

        flag = 0;

        //printf("%d  ",c[i]);

        for (let j = i + 1; j < count; j++) {

            if (c[i] == c[j]) {

                flag = 1;

            }

        }

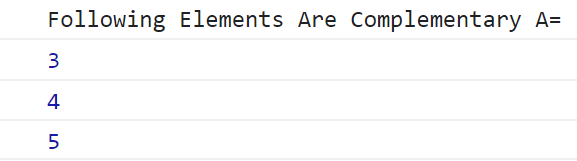
        if (flag == 0) {

            console.log(c[i]);

        }

    }

Output :-



**Bootsrap**

* **What is Model ?**

The modal plugin toggles your hidden content on demand, via data attributes or JavaScript.

**Uses :-** The modal plugin toggles your hidden content on demand, via data attributes or JavaScript. It also adds .modal-open to the <body> to override default scrolling behavior and generates a .modal-backdrop to provide a click area for dismissing shown modals when clicking outside the modal.

**Example:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Model</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmfTnH7ois6mU1QK+m14rQ1l2bGBq41eYeM/fS" crossorigin="anonymous"></script>

  <style>

    img{

        width: 100%;

        height: 250px;

    }

    body{

        text-align: center;

        margin-top:23px;

    }

  </style>

</head>

<body>

    <!-- Button trigger modal -->

    <button type="button" class="btn btn-warning" data-bs-toggle="modal" data-bs-target="#exampleModal">

        Click the modal Button

    </button>

    <!-- Modal -->

    <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

        <div class="modal-dialog">

            <div class="modal-content">

                <div class="modal-header">

                    <h1 class="modal-title fs-5" id="exampleModalLabel">Rose Information</h1>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body">

                    <div class="img1">

                    <img src="https://images.fineartamerica.com/images/artworkimages/mediumlarge/1/red-rose-on-black-background-aleksandr-tkach.jpg" alt="">

                    </div>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>

                   <a href=" https://www.britannica.com/plant/rose-plant"> <button type="button" class="btn btn-primary">Save changes</button></a>

                </div>

            </div>

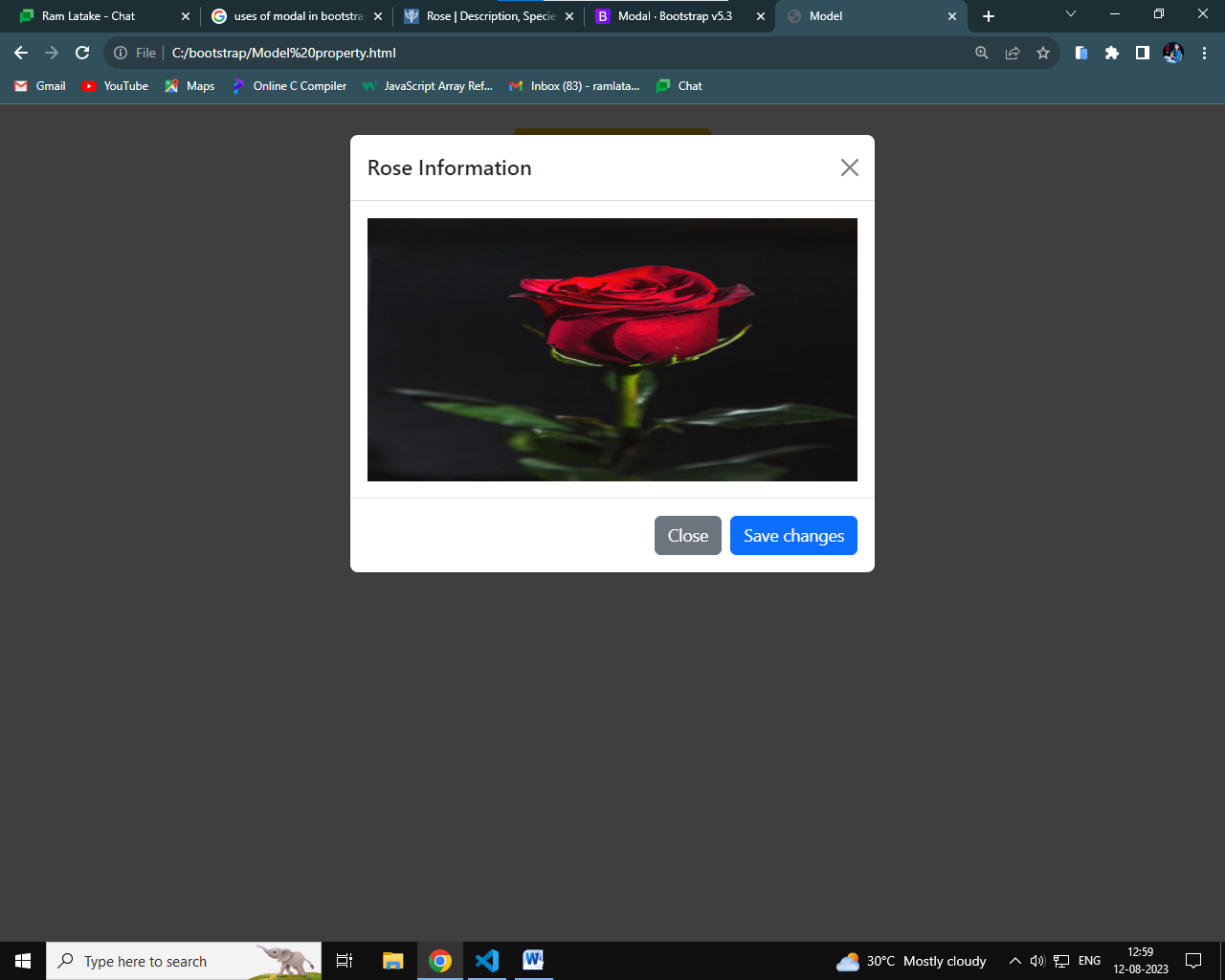
        </div>

    </div>

</body>

</html>

**output:-**

****

* **What is navbar ?**

 a navigation bar can extend or collapse, depending on the screen size.

* **uses :**

Navbar navigation links build on our . nav options with their own modifier class and require the use of toggler classes for proper responsive styling.

Example:-

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Navbar</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>**

**<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmfTnH7ois6mU1QK+m14rQ1l2bGBq41eYeM/fS" crossorigin="anonymous"></script>**

**</head>**

**<body>**

**<nav class="navbar navbar-expand-lg navbar-light bg-info">**

**<a class="navbar-brand" href="#">Navbar</a>**

**<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">**

**<span class="navbar-toggler-icon"></span>**

**</button>**

**<div class="collapse navbar-collapse" id="navbarNavAltMarkup">**

**<div class="navbar-nav">**

**<a class="nav-item nav-link active" href="#">Home <span class="sr-only">(current)</span></a>**

**<a class="nav-item nav-link" href="#">Features</a>**

**<a class="nav-item nav-link" href="#">Pricing</a>**

**<a class="nav-item nav-link disabled" href="#">Disabled</a>**

**</div>**

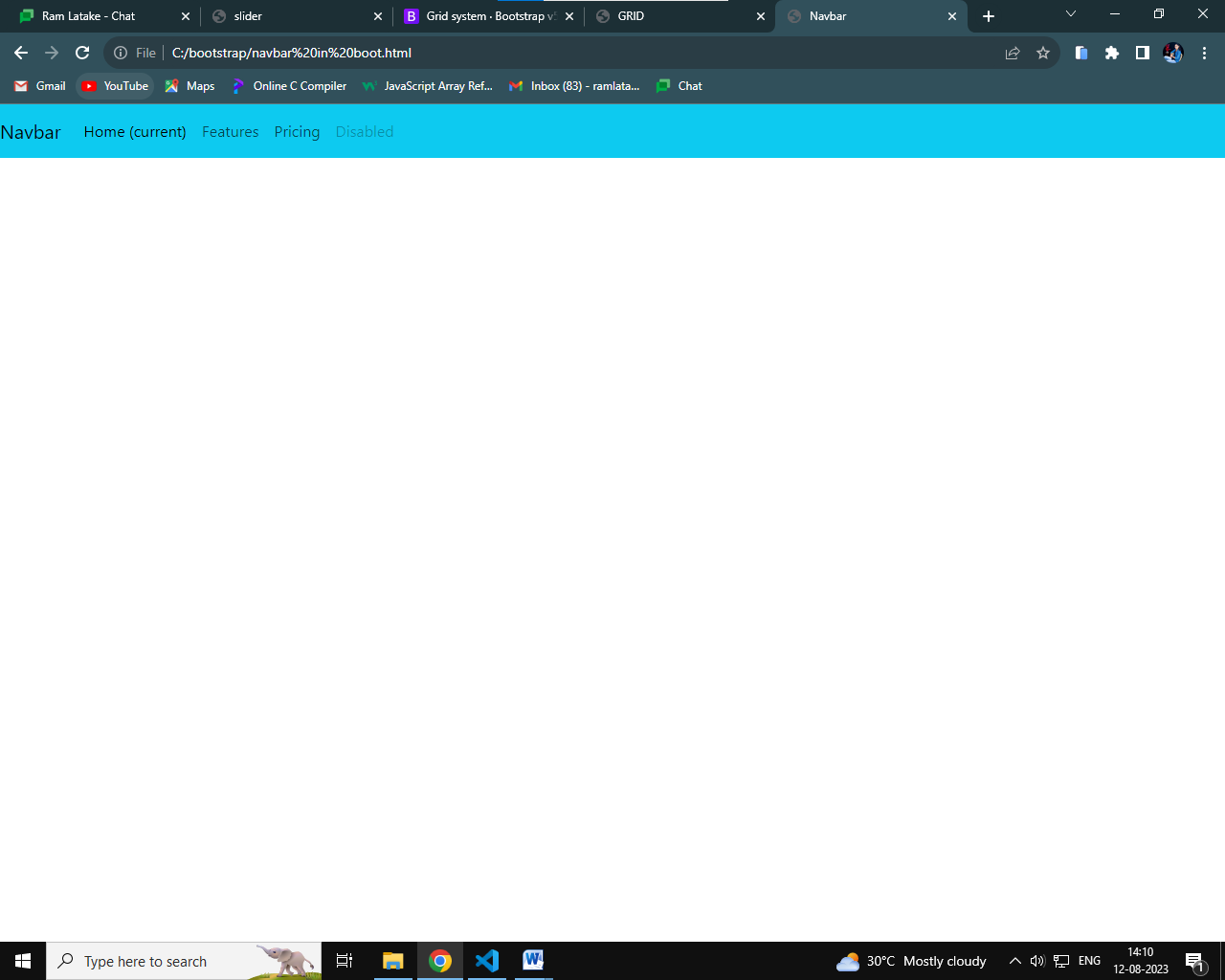
**</div>**

**</nav>**

**</body>**

**</html>**

Output:-



* **What is slider/carousoe?**

A slideshow component for cycling through elements images or slides of text like a carousel.

**uses:-**

1. Add image elements inside the parent div element. ...
2. Wrap each image in a div element. ...
3. Wrap all these inner divs in another div element. ...
4. Add controls with ARIA attributes.
5. Create a parent div element with an id and class attribute. ...

**Example:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>slider</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmfTnH7ois6mU1QK+m14rQ1l2bGBq41eYeM/fS" crossorigin="anonymous"></script>

    <style>

        img{

            width: 100%;

            height: 400px;

        }

    </style>

</head>

<body>

      <div id="carouselExampleInterval" class="carousel slide" data-bs-ride="carousel">

        <div class="carousel-inner">

          <div class="carousel-item active" data-bs-interval="10000">

            <img src="https://cdn.pixabay.com/photo/2017/05/08/13/15/spring-bird-2295434\_1280.jpg" class="d-block w-100" alt="...">

          </div>

          <div class="carousel-item" data-bs-interval="2000">

            <img src="https://cdn.pixabay.com/photo/2017/05/08/13/15/bird-2295431\_1280.jpg" class="d-block w-100" alt="...">

          </div>

          <div class="carousel-item">

            <img src="https://cdn.pixabay.com/photo/2015/11/16/16/28/bird-1045954\_1280.jpg" class="d-block w-100" alt="...">

          </div>

        </div>

        <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleInterval" data-bs-slide="prev">

          <span class="carousel-control-prev-icon" aria-hidden="true"></span>

          <span class="visually-hidden">Previous</span>

        </button>

        <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleInterval" data-bs-slide="next">

          <span class="carousel-control-next-icon" aria-hidden="true"></span>

          <span class="visually-hidden">Next</span>

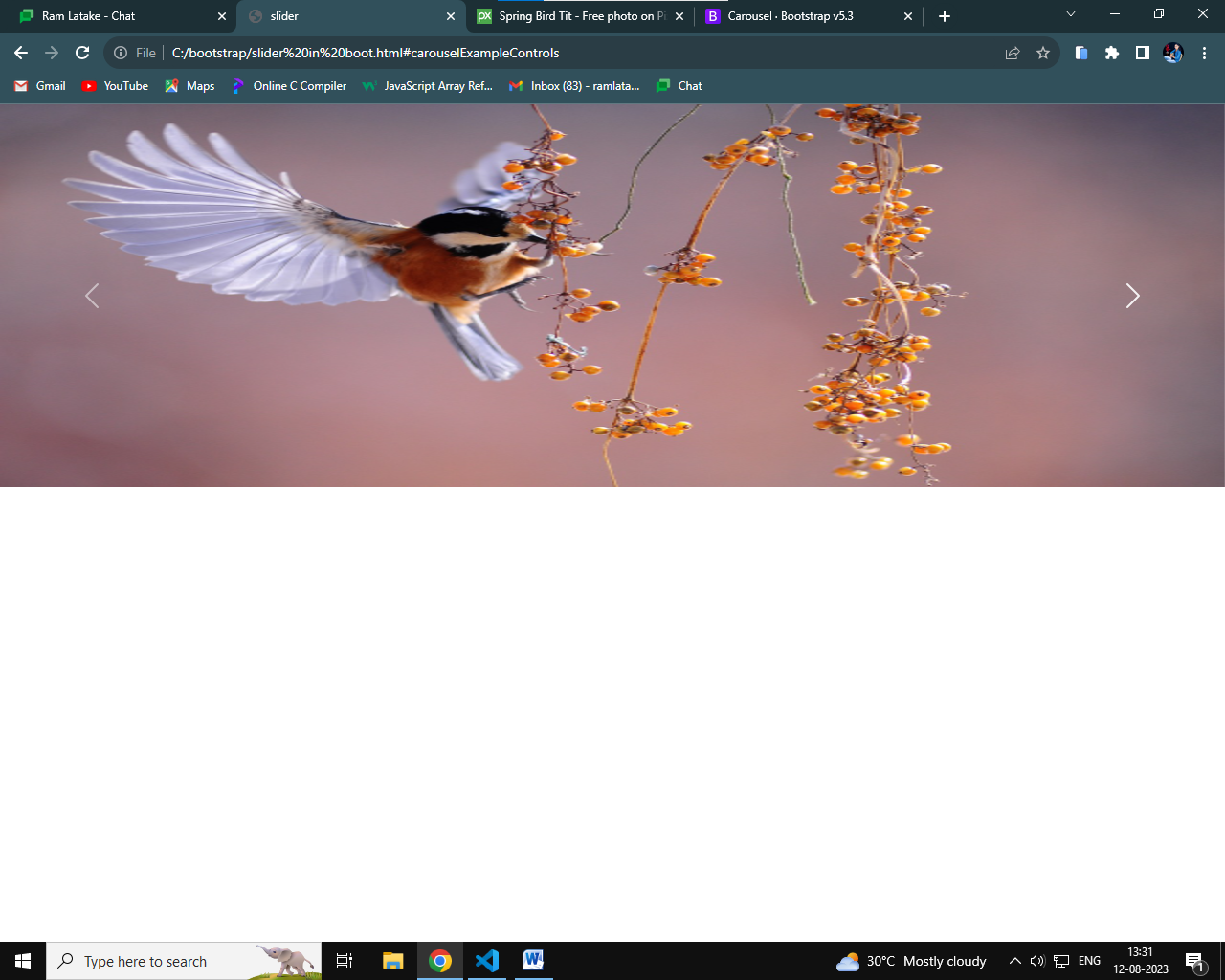
        </button>

      </div>

</body>

</html>

**output:-**

****

* **What is containers ?**

Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.

**uses:-**

used to contain, pad, and (sometimes) center the content within them.

* **What is grid?**

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content.

**use:-**

The grid system helps create columns with the required size using the "col-size" class for a single column in a row.

**Exmaple:-**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>GRID</title>**

**<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSmO7SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM" crossorigin="anonymous">**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz" crossorigin="anonymous"></script>**

**<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM8ODewa9r" crossorigin="anonymous"></script>**

**<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.min.js" integrity="sha384-fbbOQedDUMZZ5KreZpsbe1LCZPVmfTnH7ois6mU1QK+m14rQ1l2bGBq41eYeM/fS" crossorigin="anonymous"></script>**

**<style>**

**div,.row{**

**border: 1px solid black;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container text-center">**

**<!-- Stack the columns on mobile by making one full-width and the other half-width -->**

**<div class="row">**

**<div class="col-md-8">.col-md-8</div>**

**<div class="col-6 col-md-4">.col-6 .col-md-4</div>**

**</div>**

**<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->**

**<div class="row">**

**<div class="col-6 col-md-4">.col-6 .col-md-4</div>**

**<div class="col-6 col-md-4">.col-6 .col-md-4</div>**

**<div class="col-6 col-md-4">.col-6 .col-md-4</div>**

**</div>**

**<!-- Columns are always 50% wide, on mobile and desktop -->**

**<div class="row">**

**<div class="col-6">.col-6</div>**

**<div class="col-6">.col-6</div>**

**</div>**

**</div>**

**</body>**

**</html>**

output:-

