```css
#row-container{
    width: 100%;
    height: 450px;
    /*background: yellow;*/
    margin-top: 20px;
    text-align: center;
}

#section-1{
    width: 33%;
    height: inherit;/*height of parent will be given to the child element*/
    /*background: red;*/
    display: inline-block;
    padding-top: 20px;
}
#section-2{
    width: 33%;
    height: inherit;/*height of parent will be given to the child element*/
    /*background: blue;*/
    display: inline-block;
    padding-top: 20px;
}
#section-3{
    width: 33%;
    height: inherit;/*height of parent will be given to the child element*/
    /*background: green;*/
    display: inline-block;
    padding-top: 20px;
}
nav{
    box-shadow: 0px 10px 20px #eeeeee;/*adding shadow*/
}

body{
    margin: 0;
}
```

- display: none | block | inline | inline-block;
- display: none; <- element removed from screen
- display: block;<- new element will come to next line
- display: inline;<- new element will come in same line, (then width & height property will have no effect)
- display: inline-block;<- new element will come in same line, it will consist some amount of gap in
- if we want to persist the custom width and height so we have to display: inline-block;

- box-shadow: vertical | horizontal | depth | color;
- box-model height: 400px border: 2px padding-top: 20px padding-bottom: 20px

  400 + 2 + 2 + 20 + 20 = total height of element

  - is a pictorial representaion of any element, which depicts the amount of margins, paddings, borders and width&height occupied by any element
  - total width: original width+padding+margin+border

```
<div id="row-container">
    <div id="section-1">
        <img src="/pic.jpeg">
        <h3>Heading Text</h3>
        <p>Lorem12</p>
        <button>Click Me</button>
    </div>
    <div id="section-2"><img src="/pic2.jpeg"></div>
    <div id="section-3"><img src="/pic.jpeg"></div>
</div>
```

** JavaScript: * is to execute logic at the client side/web browser * With HTML & CSS page will be static one **Appications: * form validato=ion * animations * requst & response from a server

- Two write js code we need script tag

  Note : we should always place script tag at the end of the body ** LiveScript: orginal name of javascript
- Everything on the page is considered as an object in JavaScript.
  - e.g any value(numeric, string, boolean, array, etc.) html tags, browser tab/window
- JS is case-sensative
- document : is a predefined object in JS that represents the complete web page. Now if we wnat to perdorm any operation on the web page then document object will be used
- Syntax: obj.method() document.write("Hello JS");
- console: console is the area in the dev tools, that will log few debugging messages, JS errors To perform any operation on the browser console we have console objext in JS console.log("Hello message on console");
- window
- alert(String): will give a pop-up on page load or reload

** Variables:

- name=value;

- var, let, const are not data-type
- in js variables/data types are dynamic i.e. the data type will be assigned at runtime type for any variab;e is not fixed while development * in js there is not seperate type for characters we can wnclosw string wither in '' or "" * flag = false; //boolean * data = null; //
- typeof(): returns the datatype of the parameter
- improved string literal

```
        document.write("<h1>Hello World</h1>");
        console.log("Hello message on console");
        alert("This is a test");
        a=6;
        b=8.5;
        console.log(a);
        console.log(b);
        console.log(typeof(a));
        console.log("Sum is "+(a+b));
        //Sum of 5 and 4 is 9
        console.log(`Sum of ${a} and ${b} is ${a+b}`);
        console.log("4"*2);  //string is considered as numeric when - * / %
        console.log("4"+1-6*2);
```

***Operators: * arithematic: + * - / % **(power) * relational: < > <= >= == != === * logical: && || ! * conditional: z = condition ? x : y; * bitwise: & | ^ * assignment: = += -= *= /= %= * increment: ++ pre/post * decrement: -- pre/post * 5**3 = 125 // 5^3

***Control structures: * sequential * conditional: if else, if else if, switch case, nested if. * iteration: for, while, do-while, for-each

***Array: array is a collection of elements (Homogeneous or Hetrogeneous) each element of array will get an index no. that will start from 0 * storage is contigeous [10, 20, 30, 40, "Hello", 5.6, true]; ***2D-arrays: collection of 1D arrays arr=[[], [], []];

***jagged array: a = [[1, 2, 4], [1, 3], [2, 3, 4]]

```
    a = 5;
    arr=[10, 20, 30, 40, "Hello", 5.6, true];
    console.log(arr);
    console.log(arr.length);
    console.log(arr[2]);
    console.log(arr[13]);  //undefined

    for(i=0; i<arr.length; i++){
        console.log(arr[i]);
    }

    for(i=arr.length-1; i>=0; i--){
        console.log(arr[i]);
```

```
    }
    //even value
    arr1=[2, 10, 7, 3, 14, 15, 8, 6, 5];

    for(i=0; i<arr1.length; i++){
        if(arr[i]%2 == 0){
            console.log(arr[i]);
        }
    }
    //max value
    arr1=[2, 10, 7, 3, 14, 54, 8, 6, 5];
    //assume that 1st element is largest
    // get each value from array one by one and check if assumed max value is
greater than every value or not
    max=arr1[0];
    for(i=1 i<arr1.length; i++){
        if(max<arr1[i]){
            max = arr1[i];
        }
    }
    console.log(max);

    a = [[1, 2, 4], [1, 3], [2, 3, 4]];
    for(i=0; i<a.length; i++){
        console.log(a[i]);
        for(j=0; j<arr[i].length; j++){
            console.log(a[i][j]);
        }
    }
```