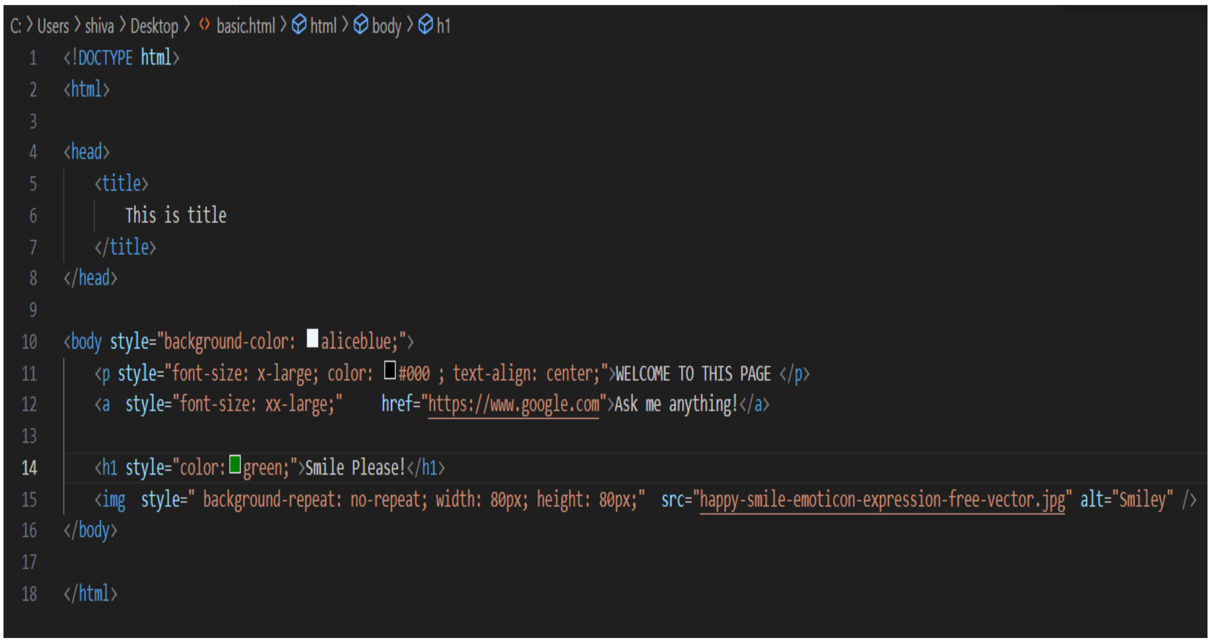
Q1.. Create a basic HTML document. (All basic tags along with required attributes, also mention the tags that

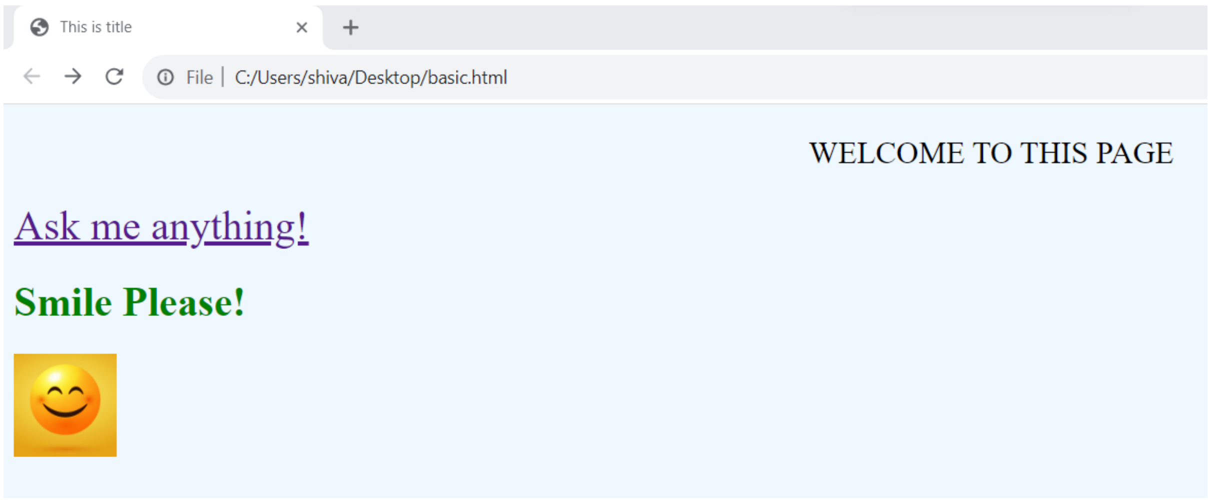
goes in head element).

Ans-



```
C: > Users > shiva > Desktop > basic.html > html > body > h1
<!DOCTYPE html>
<html>

<head>
    <title>
        This is title
    </title>
</head>

<body style="background-color: aliceblue;">
    <p style="font-size: x-large; color: #000 ; text-align: center;">WELCOME TO THIS PAGE </p>
    <a  style="font-size: xx-large;"     href="https://www.google.com">Ask me anything!</a>

    <h1 style="color:green;">Smile Please!</h1>
    <img  style=" background-repeat: no-repeat; width: 80px; height: 80px;"  src="happy-smile-emoticon-expression-free-vector.jpg" alt="Smiley" />
</body>

</html>
```

OUTPUT



```
This is title
File | C:/Users/shiva/Desktop/basic.html

                                        WELCOME TO THIS PAGE

Ask me anything!

Smile Please!
```

The following tags can go inside head element.

<title>

<style>

<base>

<link>

<meta>

<script>

<noscript>

Q2. Mention 3 ways of styling a document, along with cascading order (mention priority using comments).

Ans-Documents can be styled using css (Cascading style sheet).

1.inline css

-In this type of css styling is applied to the particular element

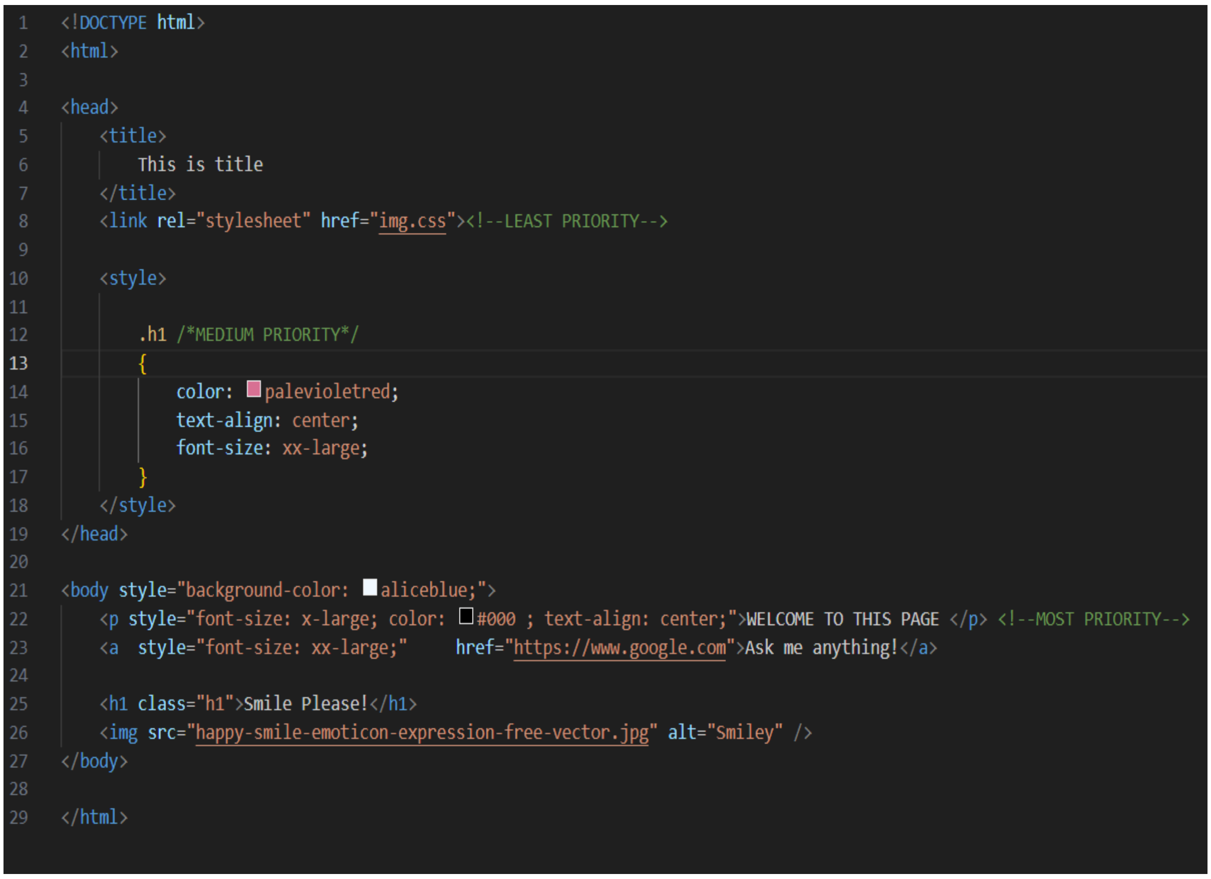
2.internal css

-In this type of css styling is predefined in <style></style> tag, this is used when single elemnt needs to be signed uniquely.

3.external css

-In this type of css separate css file is styled and used in html using link.

Separate file has extention with .css



```html
<!DOCTYPE html>
<html>

<head>
    <title>
        This is title
    </title>
    <link rel="stylesheet" href="img.css"><!--LEAST PRIORITY-->

    <style>

        .h1 /*MEDIUM PRIORITY*/
        {
            color: palevioletred;
            text-align: center;
            font-size: xx-large;
        }
    </style>
</head>

<body style="background-color: aliceblue;">
    <p style="font-size: x-large; color: #000 ; text-align: center;">WELCOME TO THIS PAGE </p> <!--MOST PRIORITY-->
    <a  style="font-size: xx-large;"     href="https://www.google.com">Ask me anything!</a>

    <h1 class="h1">Smile Please!</h1>
    <img src="happy-smile-emoticon-expression-free-vector.jpg" alt="Smiley" />
</body>

</html>
```

Q3. Explain CSS selectors: simple, combinator, pseudo-class(links/form/table), pseudo-elements and attribute

selectors.

Ans-CSS selectors are used to find the HTML elements you want to style.

1.Simple Selector

-Simple selectors are selectors which selects elements based upon class id name etc.

2.Combinator Selector

-Combinator selectors are selectors which select elements based on a specific relationship between them.

3.Pseudo class selector

-Pseudo-class selectors are selectors which select elements based on a certain state.

3a.Links

-Links can be displayed in various methods

Link, Visited, Hover

They allow you to apply styles based on different conditions and interactions, enhancing the visual and interactive experience of your website

3b.Form

-Pseudo-class selectors can be used to style form elements. By utilizing these selectors, you can enhance the usability and visual appeal of your forms and make the user experience more engaging.

Focus,Hover,Checked.

3c.Table

Pseudo-class selectors can be used to style different parts of tables and table elements. By utilizing these selectors, you can create visually appealing and user-friendly table layouts on your web pages

4.Pseudo Elements Selectors

-pseudo-element is used to style specified parts of an element.Eg Style the first letter, or line, of an element

Insert content before, or after, the content of an element

5.Attribute Selector

Q4. Mention ways to create an array, access and manipulate it. How to recognise if a variable is an array?

Ans-An array is a special variable, which can hold more than one value.

-ways to create an array

1.Using an array literal is the easiest way to create a JavaScript Array

```
const fruits = ["Mango", "Apple", "Banana"];
```

2.You can also create an array, and then provide the elements.

```
const fruits = [];
fruits[0]= "Mango";
fruits[1]= "Apple";
fruits[2]= "Banana";
```

3.Using the JavaScript Keyword new also creates an Array, and assigns values to it.

```
const fruits = new Array("Mango", "Apple", "Banana");
```

-Ways to access array

1.You access an array element by referring to the index number

```
const fruits = ["Mango", "Apple", "Banana"];
let fruits = fruits[0];
```

2.With JavaScript, the full array can be accessed by referring to the array name

```
const fruits = ["Mango", "Apple", "Banana"];
document.getElementById("demo").innerHTML = fruits;
```

-How to recognise if a variable is an array?

1.Method 1: Using JavaScript isArray() method

This method checks whether the passed variable is an Array object or not.

2.Method 2: Using JavaScript instanceof operator

It is used to test whether the prototype property of a constructor appears anywhere in the prototype chain of an object.

This can be used to evaluate if the given variable has a prototype of ‘Array’.

```
  variable instanceof Array
```

3.Method 3: Checking the constructor property of the variable

Another method to check a variable is an array by checking its constructor with Array.

```
  variable.constructor === Array
```

Q5. Find the size of an array, access the first and last element of array and add new element using appropriate

array property.

Ans-The length property returns the length (size) of an array

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];

let size = fruits.length;
```

Using Array.pop() and Array.shift() Method

The Array.pop() method removes the last element of the array and returns it and the Array.shift() method removes the first element from the array and returns it.

```
let s = [3, 2, 3, 4, 5];

function Gfg() {


    let f = s.shift(0);

    let l = s.pop();
```

}

first element is 3

Last element is 5

Q6. Mention different methods to add new element to an array.

Ans-1.The unshift() method adds a new element to an array (at the beginning), and "unshifts" older elements

```
  -const fruits = ["Banana", "Orange", "Apple", "Mango"];
   fruits.unshift("Lemon")
   o/p-Lemon,Banana,Orange,Apple,Mango
```

2.The push() method adds a new element to an array (at the end)

```
   -const fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.push("Strawberry");
    o/p-Banana,Orange,Apple,Mango,Strawberry
```

3.The length property provides an easy way to append a new element to an array.

```
   -const fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits[fruits.length] = "Chickoo";
  o/p-Banana,Orange,Apple,Mango,Chickoo
```

4.The splice() method can be used to add new items to an array.

```
  -const fruits = ["Banana", "Orange", "Apple", "Mango"];
  document.getElementById("demo1").innerHTML = fruits;

   fruits.splice(3, 1, "Chikoo", "Jackfruit");
```

```
    document.getElementById("demo2").innerHTML = fruits;
```

o/p-Banana,Orange,Apple,Chikoo,Jackfruit

Q7. Mention different methods to delete element from an array.

Ans-1.Array elements can be deleted using the JavaScript operator delete

```
    -const place = ["Panjim", "Mapusa", "Margao", "Vacso"];

document.getElementById("demo1").innerHTML =
"The first place is: " + place[0];

delete place[0];

document.getElementById("demo2").innerHTML =
"The first place is: " + place[0];
```

o/p-The first place is: Panjim

The first place is: undefined

2.

The shift() method removes the first array element and "shifts" all other elements to a lower index.

```
-const animals = ["Cat", "Rabbit", "Lion", "Monkey"];
document.getElementById("demo1").innerHTML = animals;
animals.shift();
document.getElementById("demo2").innerHTML = animals;
```

```
o/p-Cat,Rabbit,Lion,Monkey

Rabbit,Lion,Monkey
```

Q8. Mention iteration methods that loop through every elements in an array along with their return value.

Ans-

1.The forEach() method calls a function (a callback function) once for each array element.

```
const numbers = [1, 2, 3, 4, 5];

let txt = "";
numbers.forEach(myFunction);
document.getElementById("demo").innerHTML = txt;

function myFunction(value, index, array) {
  txt += value + "<br>";
```

```
o/p-
1
2
3
4
5
```

2.Using while loop method A While Loop in JavaScript is a control flow statement that allows the code to be executed repeatedly based on the given boolean condition.

```
let index = 0;
let array = [1, 2, 3, 4, 5, 6];

while (index < array.length) {
   console.log(array[index]);
   index++;
}
```

```
o/p-
1
2
3
4
5
6
```

3.Using map() Method A map applies a function over every element and then returns the new array.

```
let x = 0;
let array = [2, 3, 4, 5];

let square = x => Math.pow(x, 2);

square = array.map(square);

console.log(array);
console.log(square);
```

```
o/p-[2, 3, 4, 5]
[ 4, 9, 16, 25,]
```

Q9. Mention methods that return index(es) of array element(s).

Ans-The indexOf() method returns the index (position) the first occurrence of a string in a string.

The lastIndexOf() method returns the index of the last occurrence of a specified text in a string.

Q10. Mention methods used to check if element is present in array along with their return value.

Ans-

1.The match() method returns an array containing the results of matching a string against a string (or a regular expression)

```
let text = "The rain in spain stays mainly in the plain";

const myArr = text.match(/ain/g);

document.getElementById("demo").innerHTML = myArr.length + " " + myArr;
```

```
o/p-4 ain,ain,ain,ain
```

Q11. Mention methods used to access last element, part of array along with their return value.

Ans-

1.The pop() method returns the value that was "popped out"

```
const cars = ["Suzuki", "Hyundai", "Tesla", "Mercedez"];
```

```
document.getElementById("demo1").innerHTML = cars.pop();
document.getElementById("demo2").innerHTML = cars;
```

o/p-

```
Mercedez

Suzuki,Hyundai,Tesla
```

2.Using the array length propert

```
let arry = [2,3,4,5];
let lastElement = arry.slice(-1);

console.log(lastElement);
```

o/p-5

3.Using Slice method

```
let arry = [2, 4, 6, 8, 10, 12, 14, 16];
let lastElement = arry.slice(-1);

console.log(lastElement);
```

o/p-16

Q12. Mention methods used to combine arrays

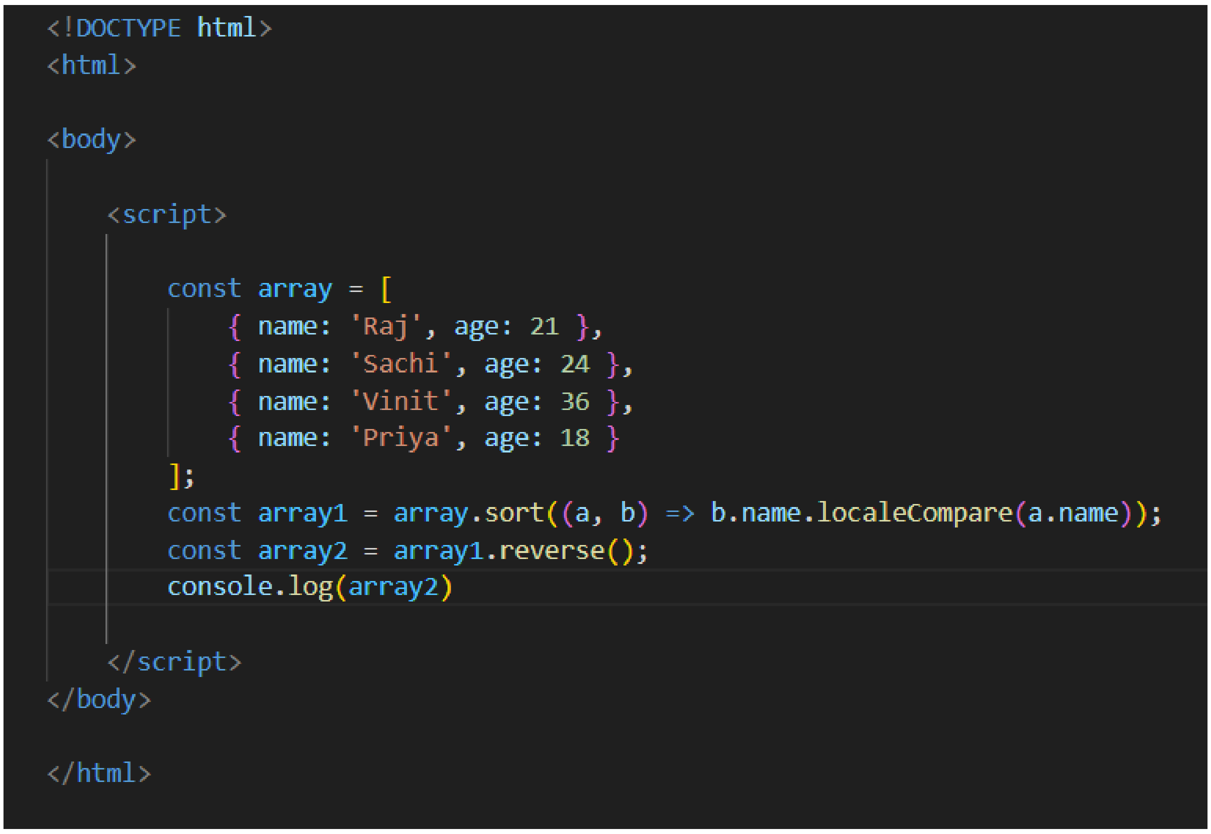
Ans-1.The concat() method creates a new array by merging (concatenating) existing arrays

```
  const arr1 = ["My", "Name"];
const arr2 = ["is", "Swara", "Shetye"];
const arr3 = arr1.concat(arr2);

document.getElementById("demo").innerHTML = arr3;
```

o/p

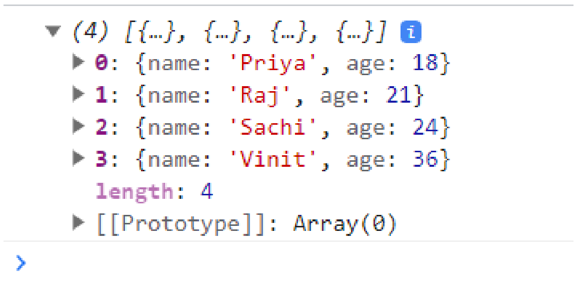
```
-My,Name,is,Swara,Shetye
```

13. Sort a number array, array of objects based on string (eg: [{name: 'Zayn', age: 23 }, {name: 'Ana', age: 35 },...]) and reverse it.



```
<!DOCTYPE html>
<html>

<body>

    <script>

        const array = [
            { name: 'Raj', age: 21 },
            { name: 'Sachi', age: 24 },
            { name: 'Vinit', age: 36 },
            { name: 'Priya', age: 18 }
        ];
        const array1 = array.sort((a, b) => b.name.localeCompare(a.name));
        const array2 = array1.reverse();
        console.log(array2)

    </script>
</body>

</html>
```

OP

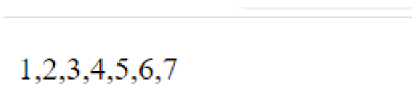


```
▼ (4) [{…}, {…}, {…}, {…}] i
  ▶ 0: {name: 'Priya', age: 18}
  ▶ 1: {name: 'Raj', age: 21}
  ▶ 2: {name: 'Sachi', age: 24}
  ▶ 3: {name: 'Vinit', age: 36}
    length: 4
  ▶ [[Prototype]]: Array(0)
>
```

Q14. In array: [1, 2, [3, 4], [5, [6, 7]]], mention methods used to get following output: [1, 2, 3, 4, 5, 6, 7].

```html
<!DOCTYPE html>
<html>

<body>
    <p id="demo"></p>
    <script>

const array = [1, 2, [3, 4], [5, [6, 7]]];
const array1 = array.flat();
document.getElementById("demo").innerHTML =array1;


    </script>
</body>

</html>
```

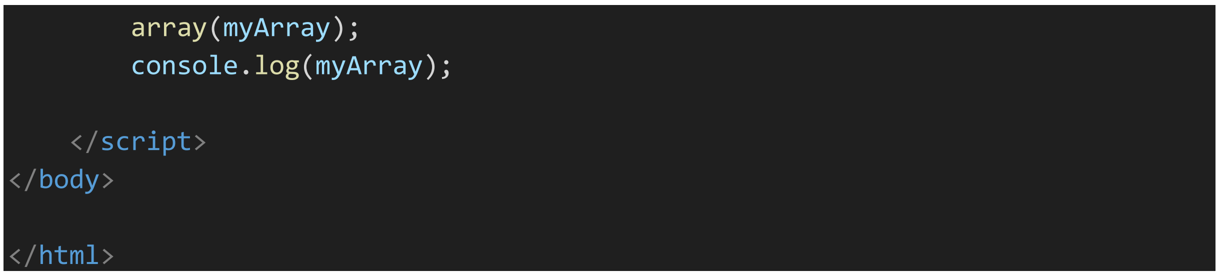
OP



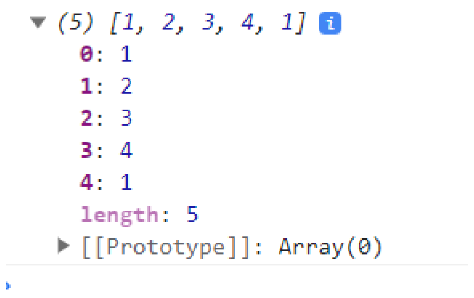
```
1,2,3,4,5,6,7
```

Q15. In array: [1, 2, 3, 4, 5], mention method to replace the last element with the first element.

```html
<!DOCTYPE html>
<html>

<body>
    <script>

        let myArray = [1, 2, 3, 4, 5];
        function array(arr) {
            if (arr.length > 0) {
                arr[arr.length - 1] = arr[0];
            }
        }
```

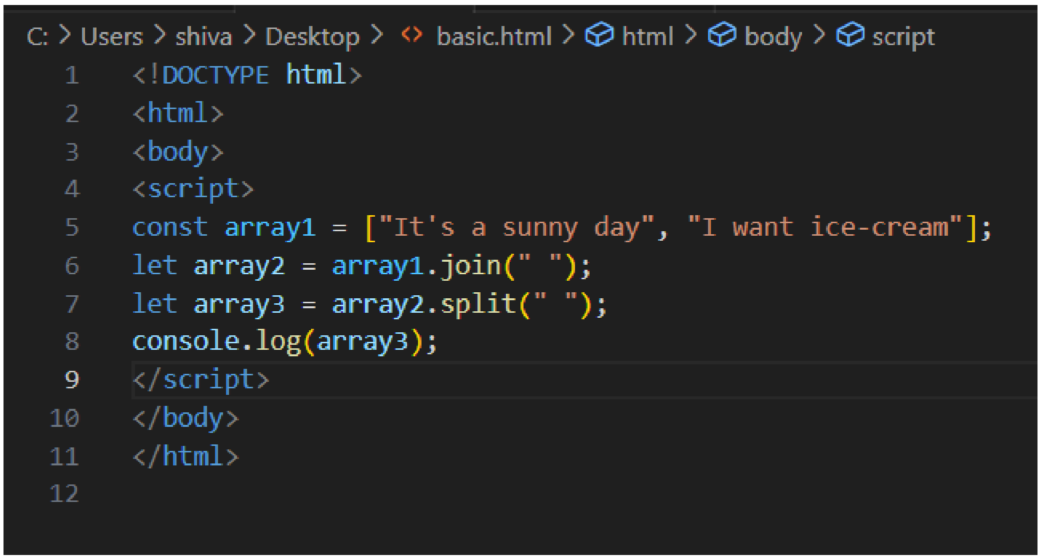

```
        array(myArray);
        console.log(myArray);

    </script>
</body>

</html>
```

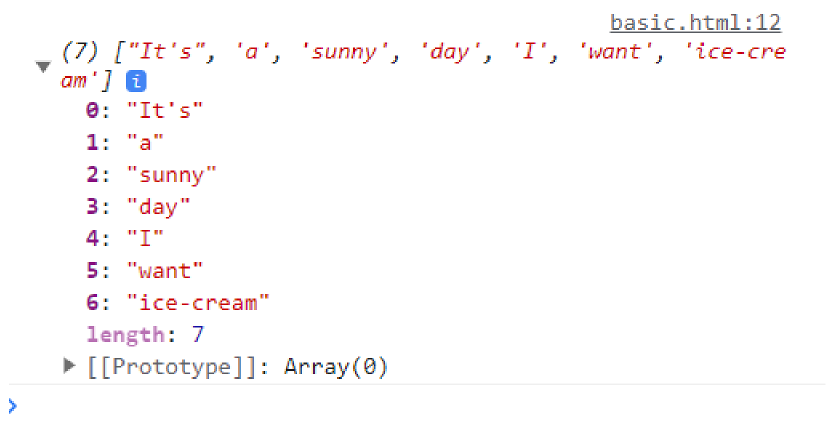
OP


```
▼ (5) [1, 2, 3, 4, 1] i
    0: 1
    1: 2
    2: 3
    3: 4
    4: 1
    length: 5
  ▶ [[Prototype]]: Array(0)
›
```

Q16. In array: ["It's a sunny day", "I want ice-cream"] using appropriate method get this output: ["it's", "a", "sunny", "day", "I", "want", "ice-cream"] (Hint: "A boy".split(" ") => ["a", "boy"])

Ans


```
C: > Users > shiva > Desktop > <> basic.html > html > body > script
<!DOCTYPE html>
<html>
<body>
<script>
const array1 = ["It's a sunny day", "I want ice-cream"];
let array2 = array1.join(" ");
let array3 = array2.split(" ");
console.log(array3);
</script>
</body>
</html>
```



```
                                              basic.html:12
▼ (7) ["It's", 'a', 'sunny', 'day', 'I', 'want', 'ice-cre
  am'] i
    0: "It's"
    1: "a"
    2: "sunny"
    3: "day"
    4: "I"
    5: "want"
    6: "ice-cream"
    length: 7
  ▶ [[Prototype]]: Array(0)
>
```

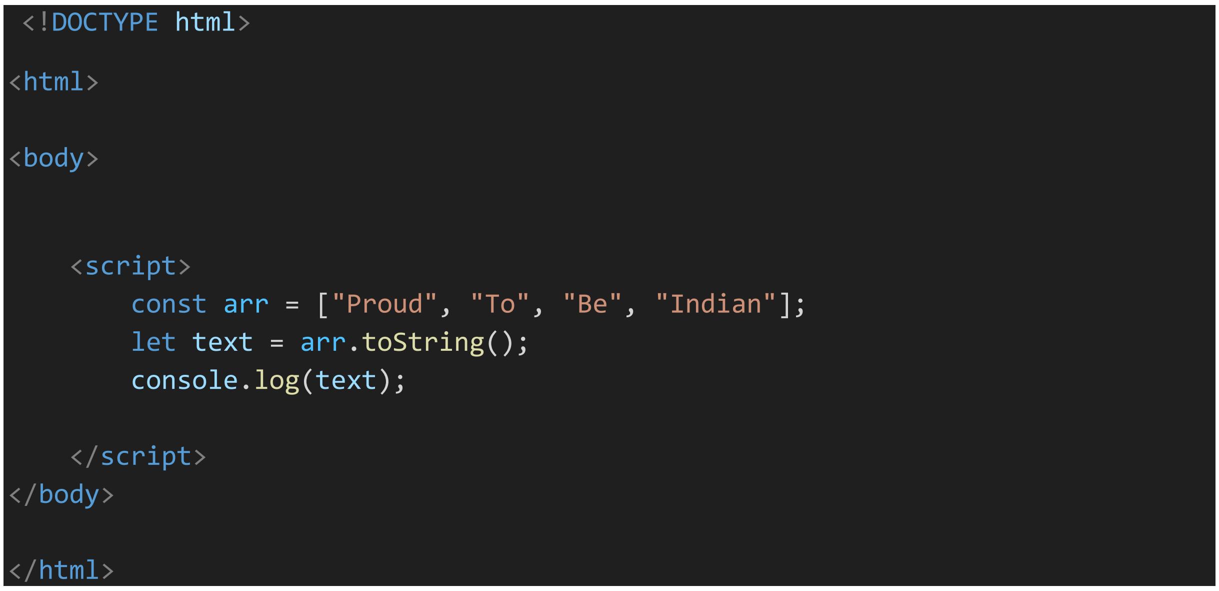
Q.17 Use appropriate methods to convert string to array and vice versa.

```
<!DOCTYPE html>

<html>

<body>

<p id="demo"></p>

<script>

let text = "Chandrayaan will land today.";

const myArray = text.split(" ");

document.getElementById("demo").innerHTML = myArray;

</script>

</body>

</html>
```

o/p

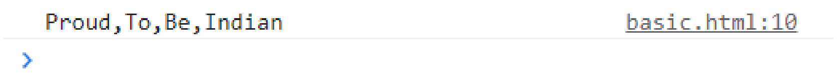
```
Chandrayaan,will,land,today.
```

Array to string



```
 <!DOCTYPE html>

<html>

<body>


    <script>
        const arr = ["Proud", "To", "Be", "Indian"];
        let text = arr.toString();
        console.log(text);

    </script>
</body>

</html>
```

op



```
  Proud,To,Be,Indian                              basic.html:10
>
```