Interview question 3:

1.How to link css files

<link rel="stylesheet" href="style.css">

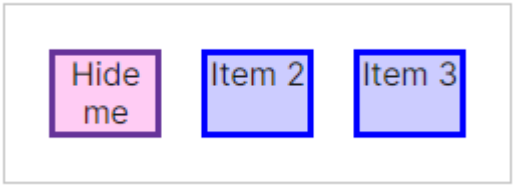
- Inline - by using the `style` attribute inside HTML elements
- Internal - by using a `<style>` element in the `<head>` section
- External - by using a `<link>` element to link to an external CSS file

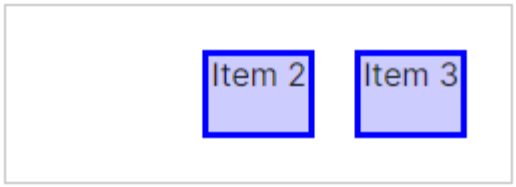
2.Display: visibility

visibility

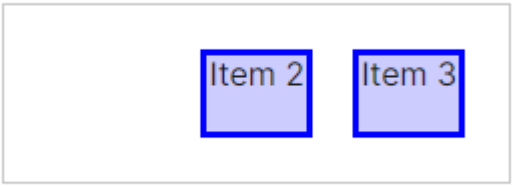
The visibility CSS property shows or hides an element without changing the layout of a document.

visibility: visible





Visibility:hidden;



Visibility:collapse

3.Anonymous function in javascript with example.

Anonymous Function is a function that does not have any name associated with it. Normally we use the *function* keyword before the function name to define a function in JavaScript.we use only the *function* keyword without the function name.

```
var greet = function () {
        console.log("Welcome to GeeksforGeeks!");
};

greet();
```

JavaScript supports Higher-Order Functions, we can also pass anonymous functions as parameters into another function.

```
setTimeout(function () {
        console.log("Welcome to GeeksforGeeks!");
}, 2000);
```

4. Splice and slice difference and what it will returns.

slice()	splice()
This method is used to get a new array by selecting a sub-array of a given array.	This method is used to add/remove an item from the given array.
The parameter **'s'** indicates the starting index and **'e'** indicates the ending index. They denote the index of the sub-array to be taken. By default, the value for start is '0' and end is 'n'.	The parameter **'i'** denotes the starting index, **'n'** denotes the number of items to be removed from the specified starting index.**'item 1, item 2, ...item n'** represents the list of new items to be added at the given index. If n=0, no item is removed, the new items are just added to the specified starting index.
The changes do not reflect in the original array.	The changes reflect in the original array

```
var cars=['Benz', 'Innova', 'Breeza', 'Etios', 'Dzire'];
var new_cars=cars.slice(2);
document.write("cars :", cars, "<br><br>");
document.write("new_cars :", new_cars);
```

```
cars :Benz, Innova, Breeza, Etios, Dzire

new_cars :Breeza, Etios, Dzire
```

```
<script>
     var cars=['Benz', 'Innova', 'Breeza', 'Etios', 'Dzire'];
     cars.splice(2, 0, 'ambassedor', 'BMW', 'Audi');
     document.write("cars :", cars, "<br><br>");
</script>
```

```
cars :Benz, Innova, ambassedor, BMW, Audi, Breeza, Etios, Dzire
```

5. Spread operator

Spread operator allows an iterable to expand in places where 0+ arguments are expected. It is mostly used in the variable array where there is more than 1 values are expected.

```
var variablename1 = [...value];
```

```
// spread operator doing the concat job
let arr = [1,2,3];
let arr2 = [4,5];

arr = [...arr,...arr2];
console.log(arr); // [ 1, 2, 3, 4, 5 ]
```

```
// changed the original array
let arr = ['a','b','c'];
let arr2 = arr;

arr2.push('d');

console.log(arr2);
console.log(arr); // even affected the original array(arr)



// spread operator for copying
let arr = ['a','b','c'];
let arr2 = [...arr];

console.log(arr); // [ 'a', 'b', 'c' ]

arr2.push('d'); //inserting an element at the end of arr2

console.log(arr2); // [ 'a', 'b', 'c', 'd' ]
console.log(arr); // [ 'a', 'b', 'c' ]
```

6. Difference Between local storage and session storage:

Local storage has 4 methods:

- **setItem() Method –** This method takes two parameters one is key and another one is value. It is used to store the value in a particular location with the name of the key.

```
localStorage.setItem(key, value)
```

- **getItem() Method –** This method takes one parameter that is key which is used to get the value stored with a particular key name.

```
localStorage.getItem(key)
```

- **removeItem() Method –** This is method is used to remove the value stored in the memory in reference to key.

```
localStorage.removeItem(key)
```

- **clear() Method –** This method is used to clear all the values stored in localstorage.

```
localStorage.clear()
```

Local Storage:

This read-only interface property provides access to the Document's local storage object, the stored data is stored across browser sessions. Similar to sessionStorage, except that localStorage data gets cleared when the page session ends – that is when the page is closed. It is cleared when the last "private" tab of a browser is closed (localStorage data for a document loaded in a private browsing or incognito session).

Session Storage:

The difference between sessionStorage and localStorage is that localStorage data does not expire, whereas sessionStorage data is cleared when the page session ends.

A unique page session gets created once a document is loaded in a browser tab. Page sessions are valid for only one tab at a time.

Local Storage	Session Storage	Cookies
The storage capacity of local storage is 5MB/10MB	The storage capacity of session storage is 5MB	The storage capacity of Cookies is 4KB
As it is not session-based, it must be deleted via javascript or manually	It's session-based and works per window or tab. This means that data is stored only for the duration of a session, i.e., until the browser (or tab) is closed	Cookies expire based on the setting and working per tab and window
The client can only read local storage	The client can only read local storage	Both clients and servers can read and write the cookies
There is no transfer of data to the server	There is no transfer of data to the server	Data transfer to the server is exist