JavaScript Abstraction:

The JavaScript abstraction is basically a process of hiding the implementation details and displaying only the functionality to all the users. In simple words we can say, JavaScript Abstraction ignores the irrelevant details and display only the necessary ones.

Note that:

- JavaScript Abstraction reduces the duplication of the code.
- An instance of Abstract Class cannot be created.

Examples:

1. This example depicts whether users can create an instance of Abstract class or not.

```
<script>
//Creating a constructor function
function Vehicle()
{
   this.vehicleName= vehicleName;
   throw new Error("An instance of Abstract class cannot be created");

}
Vehicle.prototype.display=function()
{
   return this.vehicleName;
}
var vehicle=new Vehicle();
</script>
```

- **Output** :An instance of Abstract class cannot be created

2. example to achieve abstraction

```
<script>
//Creating a constructor function
 function Vehicle()
{
   this.vehicleName="vehicleName";
   throw new Error("An instance of Abstract Class cannot be created");
}
Vehicle.prototype.display=function()
{
   return "Vehicle is: "+this.vehicleName;
}
//Creating a constructor function
function Bike(vehicleName)
{
   this.vehicleName=vehicleName;
}
//Creating object without using the function constructor
Bike.prototype=Object.create(Vehicle.prototype);
var bike=new Bike("Suzuki");
document.writeln(bike.display());
 </script>
```

- **Output** :Vehicle is: Suzuki

JavaScript Interface:

JavaScript doesn't support interfaces, it supports TypeScript instead.

TypeScript is an open-source programming language. It is a superset of JavaScript language. Type-Script is designed for the development of large applications. It is developed and maintained by Microsoft. TypeScript follows JavaScript syntactically but adds more features to it. It is a superset of JavaScript.

Interface: In general Interface is the structure or skeleton for object. Interface is programming syntax which enforce the syntax on the class. It is the definition of the object with only types of data it must have. Interface is the type for the object.

Interface in typescript is types for the object. In Type-Script interface is defined with the help of the interface keyword. JavaScript uses an interface for type checking. The interface is also known as structural subtyping.

Syntax:

```
interface InterfaceName{
   PropertyName: Type;
   methodName() => Type;
}
```

Example:

```
// Interface for Array
interface ForList {
   [index:number]: string
}
let newArray: ForList = ["Interface", "for", "Array"];
console.log(newArray);
```

Output:

```
[ 'Interface', 'for', 'Array' ]
```