

JavaScript Encapsulation

The JavaScript Encapsulation is a process of binding the data (i.e. variables) with the functions acting on that data. It allows us to control the data and validate it. To achieve an encapsulation in JavaScript:

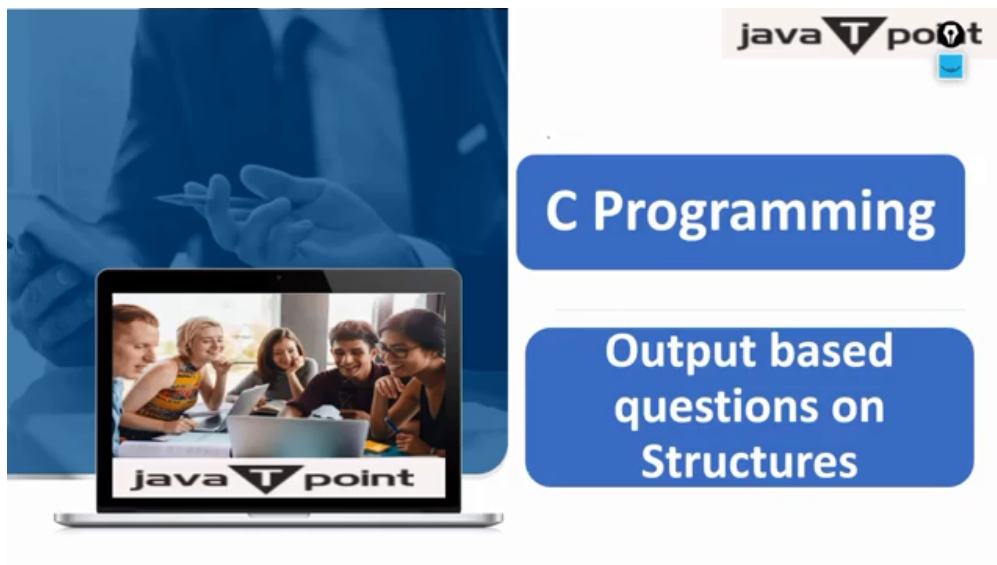
-

- Use var keyword to make data members private.
- Use setter methods to set the data and getter methods to get that data.

The encapsulation allows us to handle an object using the following properties:

Read/Write - Here, we use setter methods to write the data and getter methods read that data.

Read Only - In this case, we use getter methods only.



Write Only - In this case, we use setter methods only.

JavaScript Encapsulation Example

Let's see a simple example of encapsulation that contains two data members with its setter and getter methods.

<script>

```
class Student
{
  constructor()
  {
    var name;
    var marks;
```

↑ SCROLL TO TOP

```
    getName()
    {
        return this.name;
    }
    setName(name)
    {
        this.name=name;
    }

    getMarks()
    {
        return this.marks;
    }
    setMarks(marks)
    {
        this.marks=marks;
    }

}
var stud=new Student();
stud.setName("John");
stud.setMarks(80);
document.writeln(stud.getName()+" "+stud.getMarks());
</script>
```

Test it Now

Output:

John 80

JavaScript Encapsulation Example: Validate

In this example, we validate the marks of the student.

```
<script>
class Student
{
```

↑ SCROLL TO TOP

```
{
    var name;
    var marks;
}

getName()
{
    return this.name;
}

setName(name)
{
    this.name=name;
}

getMarks()
{
    return this.marks;
}

setMarks(marks)
{
    if(marks<0||marks>100)
    {
        alert("Invalid Marks");
    }
    else
    {
        this.marks=marks;
    }
}

var stud=new Student();
stud.setName("John");
stud.setMarks(110);//alert() invokes
document.writeln(stud.getName()+" "+stud.getMarks());
</script>
```

Test it Now

Output:

↑ SCROLL TO TOP

John undefined

JavaScript Encapsulation Example: Prototype-based approach

Here, we perform prototype-based encapsulation.

<script>

```
function Student(name,marks)
{
    var s_name=name;
    var s_marks=marks;
    Object.defineProperty(this,"name",{
        get:function()
        {
            return s_name;
        },
        set:function(s_name)
        {
            this.s_name=s_name;
        }
    });

    Object.defineProperty(this,"marks",{
        get:function()
        {
            return s_marks;
        },
        set:function(s_marks)
        {
            this.s_marks=s_marks;
        }
    });
}

stud = new Student("John",80);
console.log(stud.name+" "+stud.marks);
```

↑ SCROLL TO TOP

```
</script>
```

[Test it Now](#)**Output:**

John 80

[← Prev](#)[Next →](#) [For Videos Join Our Youtube Channel: Join Now](#)

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share

[↑ SCROLL TO TOP](#)