

## JavaScript Abstraction

An abstraction is a way of hiding the implementation details and showing only the functionality to the users. In other words, it ignores the irrelevant details and shows only the required one.

### Points to remember

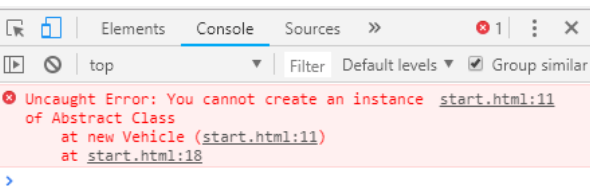
- We cannot create an instance of Abstract Class.
- It reduces the duplication of code.

## JavaScript Abstraction Example

### Example 1

Let's check whether we can create an instance of Abstract class or not.

```
<script>
//Creating a constructor function
function Vehicle()
{
  this.vehicleName= vehicleName;
  throw new Error("You cannot create an instance of Abstract class");
}
Vehicle.prototype.display=function()
{
  return this.vehicleName;
}
var vehicle=new Vehicle();
</script>
```



The screenshot shows a web browser's developer console with the 'Console' tab selected. It displays an 'Uncaught Error: You cannot create an instance of Abstract Class' at `start.html:11`. The stack trace indicates the error occurred at `new Vehicle (start.html:11)` and `at start.html:18`.

### Example 2

Let's see an example to achieve abstraction.

```
<script>
//Creating a constructor function
function Vehicle()
{
  this.vehicleName="vehicleName";
  throw new Error("You cannot create an instance of Abstract Class");
}
Vehicle.prototype.display=function()
{
  return "Vehicle is: "+this.vehicleName;
}

```

↑ SCROLL TO TOP

```
function Bike(vehicleName)
{
    this.vehicleName=vehicleName;
}
//Creating object without using the function constructor
Bike.prototype=Object.create(Vehicle.prototype);
var bike=new Bike("Honda");
document.writeln(bike.display());

</script>
```

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

```
Vehicle is: Honda
```

### Example 3

In this example, we use instanceof operator to test whether the object refers to the corresponding class.

```
<script>
//Creating a constructor function
function Vehicle()
{
    this.vehicleName=vehicleName;
    throw new Error("You cannot create an instance of Abstract class");
}
//Creating a constructor function
function Bike(vehicleName)
{
    this.vehicleName=vehicleName;
}
Bike.prototype=Object.create(Vehicle.prototype);
var bike=new Bike("Honda");
document.writeln(bike instanceof Vehicle);
document.writeln(bike instanceof Bike);

</script>
```

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

```
true true
```

[← Prev](#)[Next →](#)

For Videos Join Our Youtube Channel: [Join Now](#)

[↑ SCROLL TO TOP](#)