

Prototype Property

This lesson explains the prototype property of objects and what `__proto__` is used for by using an example.

We'll cover the following

- `__proto__` Property
- Syntax
 - Example
 - Explanation

`__proto__` Property

As discussed, in JavaScript, objects contain a property `[[Prototype]]` that is hidden. This property either points to another object or is `null`.

The `[[Prototype]]` property of objects, i.e., `anObject.[[Prototype]]` defines the prototype of `anObject`. It is defined through the `__proto__` property which is used as a setter/getter for the `[[Prototype]]` property; i.e., `__proto__` is used in order to access/set the `[[Prototype]]` property of an object.

Syntax

Let's take a look at the syntax for accessing and setting the `[[Prototype]]` property of an object.

```
1 //using __proto__ to access and
2 anObject.__proto__ = someother
```



Example

Let's take a look at an example to make this concept more clear.

```
1 //Shape object
```

```

2 var Shape={
3   name: 'Rectangle',
4   sides: 4
5 }
6
7 //Rectangle object
8 var Rectangle = {
9   length: 3,
10  width: 5
11 }
12
13 //setting [[Prototype]] of Rectangle equal to Shape
14 Rectangle.__proto__ = Shape
15
16 //displaying Rectangle object's properties
17 console.log("Name of shape is:",Rectangle.name)
18 console.log("Number of sides are",Rectangle.sides)
19 console.log("Length is:",Rectangle.length)
20 console.log("Width is:",Rectangle.width)

```



Shape

name: Rectangle
sides: 4



[[Prototype]]

Rectangle

length:3
width: 5

Rectangle's prototype property set to Shape

Explanation

As seen in the code above:

- Two objects **Shape** and **Rectangle** are created.
- **Shape** contains the properties:

- **name**

- `sides`

- At the start, `Rectangle` contains the properties:
 - `length`
 - `width`
- In **line 14**, the `__proto__` property sets `Rectangle`'s `[[Prototype]]` property equal to `Shape`.

Line 14 can translate to:

`Shape` is the prototype of `Rectangle`

or

`Rectangle` inherits prototypically from `Shape`

What do both of the above lines mean? Why does the code above display the `name` and `sides` value when accessed from `Rectangle`? Let's discuss the answers to these questions in the next lesson.