

Declaring Objects in JS

In JavaScript, an object is a collection of properties and methods that are used to represent a real-world object or concept. Objects can be used to store and manipulate data, and to organize and structure code. There are several ways to declare objects in JavaScript:

1. Using object literal notation:

```
var myObject1 = {  
  prop1: "value1",  
  prop2: "value2",  
  meth1: function() {  
    // code  
  }  
};
```

2. Using the object constructor:

```
var myObject2 = new Object();  
myObject2.property1 = "value1";  
myObject2.property2 = "value2";  
myObject2.method1 = function() {  
  // code  
};
```

3. Using object.create():

```
var myObject3 = Object.create(Object.prototype);  
myObject3.property1 = "value1";  
myObject3.property2 = "value2";  
myObject3.method1 = function() {  
  // code  
};
```

JavaScript object keys are given in double quotes if it has space or some special character or if the keys start with a digit.

```
var myObject1 = {  
  "1 prop": "value1",  
  "2 prop": "value2",  
  meth1: function() {  
    // code  
  }  
};
```

Access the Properties

1. Using the dot notation:

```
var myObj = {  
  property1: "value1",  
  property2: "value2"  
};  
console.log(myObj.property1); // logs "value1"  
console.log(myObj.property2); // logs "value2"
```

2. Using the bracket notation:

```
var myObject1 = {  
  property1: "value1",  
  property2: "value2"  
};  
console.log(myObject1["property1"]); // logs "value1"  
console.log(myObject1["property2"]); // logs "value2"
```

Add and Delete Properties

In JavaScript, you can add and delete properties of an object using the following methods:

1. Adding a property:

```
var myObj = {};  
myObj.property1 = "value1";  
console.log(myObj); // logs { property1: "value1" }
```

You can also use the bracket notation to add a property:

```
var myObj = {};  
myObj["property1"] = "value1";  
console.log(myObj); // logs { property1: "value1" }
```

2. Deleting a property:

```
var myObject = {  
  property1: "value1",  
  property2: "value2"  
};  
delete myObject.property1;  
console.log(myObject); // logs { property2: "value2" }
```

You can also use the bracket notation to delete a property:

```
var myObject = {  
  property1: "value1",  
  property2: "value2"  
};  
delete myObject["property1"];
```

```
    property2: "value2"  
};  
delete myObject["property1"];  
console.log(myObject); // logs { property2: "value2" }
```