JavaScript OBJECT

Object.create() method creates a new object with the specified prototype object and properties.

Syntax

```
Object.create(proto[, propertiesObject])
```

```
function Car (desc) {
   this.desc = desc;
   this.color = "red";
}

Car.prototype = {
   getInfo: function() {
     return 'A ' + this.color + ' ' + this.desc + '.';
   }
};
//instantiate object using the constructor function
var car =  Object.create(Car.prototype);
car.color = "blue";
alert(car.getInfo());
```

Object.defineProperties() method defines new or modifies existing properties directly on an object, returning the object.

Syntax

```
Object.defineProperties(obj, props)
```

```
var obj = {};
Object.defineProperties(obj, {
  'property1': {
    value: true,
    writable: true
  },
  'property2': {
    value: 'Hello',
    writable: false
  }
  // etc. etc.
});
```

Object.defineProperty() method defines a new property directly on an object, or modifies an existing property on an object, and returns the object.

Syntax

```
Object.defineProperty(obj, prop, descriptor)
```

Parameters

obj

The object on which to define the property.

prop

The name of the property to be defined or modified.

descriptor

The descriptor for the property being defined or modified.

```
var o = {}; // Creates a new object

// Example of an object property added
// with defineProperty with a data property descriptor
Object.defineProperty(o, 'a', {
  value: 37,
  writable: true,
  enumerable: true,
 configurable: true
});
```

Object.entries() method returns an array of a given object's own enumerable property [key, value] pairs, in the same order as that provided by a for...in loop (the difference being that a for-in loop enumerates properties in the prototype chain as well).

Syntax

```
Object.entries(obj)
```

```
var obj = { foo: 'bar', baz: 42 };
console.log(Object.entries(obj)); // [ ['foo', 'bar'], ['baz', 42] ]
```

Object.keys() method returns an array of a given object's own enumerable properties, in the same order as that provided by a for...in loop (the difference being that a for-in loop enumerates properties in the prototype chain as well).

Syntax

```
Object.keys(obj)
```

Parameters

obj

The object of which the enumerable own properties are to be returned.

```
var arr = ['a', 'b', 'c'];
console.log(Object.keys(arr)); // console: ['0', '1', '2']
```

Object.assign() method is used to copy the values of all enumerable own properties from one or more source objects to a target object. It will return the target object.

Syntax

```
Object.assign(target, ...sources)
```

Parameters

target

The target object.

sources

The source object(s).

```
var obj = { a: 1 };
var copy = Object.assign({}, obj);
console.log(copy); // { a: 1 }
```

Object.isSealed() method determines if an object is sealed.

Syntax

```
Object.isSealed(obj)
```

Parameters

obj

The object which should be checked.

```
var empty = {};
Object.isSealed(empty); // === false

// If you make an empty object non-extensible,
// it is vacuously sealed.
Object.preventExtensions(empty);
Object.isSealed(empty); // === true
```