

wawiwa

OBJETO

Objeto

1. JS está diseñado en un paradigma simple basado en objetos.
2. Un objeto es una colección de propiedades, y una propiedad es una asociación entre un nombre (o *key*) y un valor.
3. El valor de una propiedad puede ser una función, en ese caso la propiedad se conoce como método.
4. Además de los objetos predefinidos en el navegador, puedes definir tus propios objetos.
5. Este capítulo describe como usar objetos, propiedades, funciones y métodos, y de cómo crear tus propios objetos.

Objetos y propiedades

Un objeto de js tiene propiedades asociados con el. Una propiedad de un objeto puede ser explicada con una variable que está adjunta a un objeto. Las propiedades de los objetos son básicamente lo mismo que variables ordinarias de js. Las propiedades de un objeto define las características de un objeto. Accedes a las propiedades de un objeto con una notación de punto simple:

`objectName.propertyName`

Objeto

Como todas las variables de js, también tanto el nombre del objeto (que podría ser una variable normal) como el nombre de la propiedad distinguen entre mayúsculas y minúsculas. Puedes definir una propiedad asignándole un valor. Por ejemplo, creemos un objeto llamado `player` y dale propiedades llamadas `make`, `name`, `level` y `points` así:

```
<html>
  <body>
    <script>
      var player = new Object();
      player.name = "Mike";
      player.level = 3;
      player.points = "8";
      console.log(player);
    </script>
  </body>
</html>
```

Objeto

El objeto arriba también pudo haber sido escrito usando un **inicializador de objeto**, que es una lista separada por comas de cero o más pares de nombres de propiedades y valores asociados de un objeto, encerrados en llaves ({}):

Las propiedades sin asignar son undefined (y no null).

player.Color; // undefined

```
<html>
<body>
  <script>
    var player = {
      name: "Mike",
      level: 3,
      points: "8",
    };
    console.log(player);
  </script>
</body>
</html>
```

Objeto

Agrupar todos los atributos bajo una entidad.

Las propiedades de los objetos de js pueden también ser accedidos o establecidos usando notación de corchetes (para más detalles ver [property accessors](#)).

A los objetos a veces se les llaman “arreglos asociativos” porque cada propiedad está asociada con un un valor string que puede ser usada para acceder a ella. Por ejemplo, puedes acceder las propiedades del objeto myCar así:

```
player["name"] = "Mike";  
player["level"] = 3;  
player["points"] = "8";
```

Creando objetos nuevos

JS tiene unos objetos predeterminados. Adicionalmente, puedes crear tus propios objetos. Puedes crear un objeto usando un inicializador de objeto. Alternativamente, puedes primero crear una función constructora y después instar un objeto invocando esa función en conjunción con el nuevo operador.

La sintaxis para un objeto usando un inicializador de objeto es:

```
var obj = {  
  property 1: value 1,  // property_# puede ser un inicializador...  
  2: value 2,  // o un número...  
  'property n': value_n // o un string  
};
```

```
var myHonda = {  
  color: 'red',  
  wheels: 4,  
  engine: {  
    cylinders: 4,  
    size: 2.2  
  }  
};
```

7

Creando objetos nuevos

El siguiente ejemplo crea myHonda con tres propiedades. Ten en cuenta que el la propiedad **engine** también es un **objeto** con sus propias propiedades.

```
var myHonda = {  
  color: 'red',  
  wheels: 4,  
  engine: {  
    cylinders: 4,  
    size: 2.2  
  }  
};
```


Creando objetos nuevos - usando constructor

Alternativamente, puedes crear un objeto siguiendo estos 2 pasos:

1. Define el objeto escribiendo una función constructora. Existe una convención fuerte, que con una buena razón, inicia con mayúscula.
2. Crea una instancia del objeto con “new”.

Para definir un objeto, crea una función para el objeto que especifique su nombre, propiedades y métodos. Por ejemplo, supón que quieres crear un objeto para autos. Quieres que este objeto se llame Car y quieres que tenga propiedades para make, model y year. Para hacer esto, escribirías la siguiente función:

```
function Car(make, model, year) {  
  this.make = make;  
  this.model = model;  
  this.year = year;  
}
```

Ten en cuenta que el uso de **this** asigna valores a las propiedades el objeto basándose en los valores que se pasan a la función.

Creando objetos nuevos - usando constructor

```
function Car(make, model, year) {  
    this.make = make;  
    this.model = model;  
    this.year = year;  
}
```

Ten en cuenta que el uso de **this** asigna valores a las propiedades el objeto basándose en los valores que se pasan a la función.

Ahora puedes crear un objeto llamado myCar así:

```
var myCar = new Car('Eagle', 'Talon TSi', 1993);
```

Esta declaración crea un objeto y le asigna a sus propiedades los valores especificados. Después el valor myCar.make es el "Eagle", myCar.year es el número 1993 y así sucesivamente.

Creando objetos nuevos - usando constructor

Puedes crear cualquier número de objetos Car usando una nueva clave. Por ejemplo:

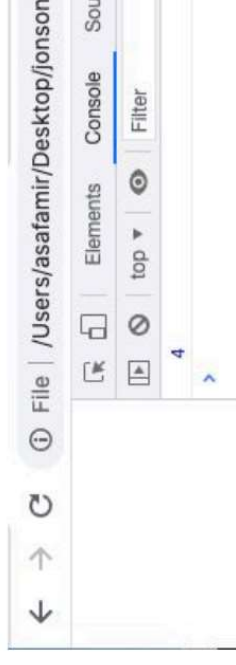
```
var kenscar = new Car('Nissan', '300ZX', 1992);
```

```
var vpgscar = new Car('Mazda', 'Miata', 1990);
```

Objeto + función

```
<html>
<body>
  <script>
    var player = {
      name: "Asaf",
      level: 3,
      points: 8,
      updateLevel: function() {
        this.level++;
        return this.level;
      }
    }
  </script>
</body>
</html>
```

```
console.log(player.updateLevel());
</script>
</body>
</html>
```



Hazlo tú mismo 1

Crea un objeto con las propiedades: precio, color y memoria.

Asígnale el objeto a una variable llamada computadora.

Hazlo tú mismo 2

1. Crea un objeto llamado Persona usando un constructor y dale las siguientes propiedades: nombre, apellido y edad.
2. Crea dos instancias del objeto
 - a. Primera instancia p1 : “Michael”, “Jordan”, 49.
 - b. Segunda instancia p2 : “Lara”, “bili”, 74.

después de los siguientes varios capítulos, cubriremos más acerca de los temas: objetos, funciones y arreglos. En el próximo capítulo hablaremos sobre el DOM.

A nighttime photograph of a city skyline. In the foreground, a large, curved skyscraper with a grid-like facade is illuminated with blue and white lights. Behind it, another tall building with a similar grid pattern is visible. To the right, a street with traffic lights and other buildings is shown. The sky is dark, and the city lights create a vibrant scene.

wawiwa

¿Preguntas?