

Os objetos JavaScript têm um conjunto de "próprias propriedades" e também herdam um conjunto de propriedades de seu objeto de protótipo. Para entender isso, nós deve considerar o acesso à propriedade com mais detalhes. Os exemplos nesta seção Use a função `object.create ()` para criar objetos com

Protótipos especificados. Veremos no capítulo 9, no entanto, que toda vez Você cria uma instância de uma classe com nova, você está criando um objeto Isso herda as propriedades de um objeto de protótipo.

Suponha que você consulte a propriedade `x` no objeto `o`. Se `O` não tiver uma propriedade própria com esse nome, o protótipo objeto de `O` é consultado para a propriedade `x`. Se o protótipo objeto não tiver uma propriedade própria por esse nome, mas tem um protótipo em si, a consulta é realizada no protótipo do protótipo. Isso continua até que a propriedade `X` seja encontrada ou até que um objeto com um protótipo nulo seja pesquisado. Como você pode ver, O atributo do protótipo de um objeto cria uma cadeia ou lista vinculada de onde as propriedades são herdadas:

```
Seja o = {};// o herda métodos de objeto de
```

```
Object.prototype
```

```
O.x = 1;// e agora tem uma propriedade própria
```

```
x.
```

```
Seja p = object.create (o);// p herda propriedades de O e
```

```
Object.prototype
```

```
p.y = 2;// e tem uma propriedade própria y.
```

```
Seja q = object.create (p);// q herda propriedades de p, o,
```

```
e...
```

```
q.z = 3;// ... object.prototype e tem um
```

```
Propriedade própria z.
```

```
Seja f = q.toString ();// ToString é herdado de
```

```
Object.prototype
```

```
q.x + q.y // => 3; X e Y são herdados de
```

```
o e p
```

```
1
```