

Урок по цепочке прототипов

Пошаговый разбор кода из урока:

1. Создание функции-конструктора `Book`:

```
const Book = function (title, author) {  
  this.author = author;  
  this.title = title;  
};
```

В этой части кода определяется функция-конструктор `Book`, которая принимает два параметра: `title` и `author`. Когда объект будет создан с использованием этой функции-конструктора, эти параметры будут присвоены свойствам `author` и `title` создаваемого объекта.

2. Добавление свойства `isRead` к прототипу `Book`:

```
Book.prototype.isRead = false;
```

Здесь мы добавляем свойство `isRead` к прототипу `Book`. Это означает, что все объекты, созданные с помощью функции-конструктора `Book`, будут иметь доступ к этому свойству, и изначально оно установлено в `false`.

3. Создание объекта `lordOftheRings`:

```
const lordOftheRings = new Book('1', '1');
```

Мы создаем объект `lordOftheRings`, используя функцию-конструктор `Book`. Этот объект будет иметь свойства `author` и `title`, установленные в `'1'`, и унаследует свойство `isRead` из прототипа `Book`.

4. Вывод в консоль различных прототипов и объектов:

- `Book.prototype` :
Здесь мы выводим прототип функции-конструктора `Book`, который содержит свойство `isRead`. Этот прототип является общим для всех объектов, созданных с помощью `Book`.
- `Book.__proto__` :
`Book` - это функция, поэтому она имеет свойство `__proto__`, которое указывает на прототип функции. Это свойство связано с прототипом `Function`.
- `lordOftheRings.__proto__` :
`lordOftheRings` - это объект, созданный с использованием функции-конструктора `Book`, поэтому его прототип (`__proto__`) указывает на прототип `Book`.
- `Book.prototype.__proto__` :
Прототип `Book` сам по себе является объектом, поэтому у него также есть прототип. В данном случае, он указывает на прототип всех объектов в JavaScript, который является `Object.prototype`.
- `Object.prototype.__proto__` :
Прототип всех объектов в JavaScript указывает на `null`. Это означает, что `Object.prototype` не имеет своего собственного прототипа, и это является вершиной цепочки прототипов.