

Context: A Movie List

In this chapter, we'll start with an example program and improve it little by little, without adding any new functionality. This important programming task is called refactoring.

Our initial program is about recent Batman movies. The data comes under the form of an array of objects, with each object describing a movie.

```
1  const movieList = [  
2    {  
3      title: "Batman",  
4      year: 1989,  
5      director: "Tim Burton",  
6      imdbRating: 7.6  
7    },  
8    {  
9      title: "Batman Returns",  
10     year: 1992,  
11     director: "Tim Burton",  
12     imdbRating: 7.0  
13   },  
14   {  
15     title: "Batman Forever",  
16     year: 1995,  
17     director: "Joel Schumacher",  
18     imdbRating: 5.4  
19   },  
20   {  
21     title: "Batman & Robin",  
22     year: 1997,  
23     director: "Joel Schumacher",  
24     imdbRating: 3.7  
25   },  
26   {  
27     title: "Batman Begins",  
28     year: 2005,  
29     director: "Christopher Nolan",  
30     imdbRating: 8.3  
31   },  
32 ]
```



And here is the rest of the program that uses this data to show some results about the movies. Check it out, it should be pretty self-explanatory.

```
1  // Get movie titles  
2  const titles = [];  
3  for (const movie of movieList)  
4    titles.push(movie.title);
```



```
5 }
6 console.log(titles);
7
8 // Count movies by Christopher
9 const nolanMovieList = [];
10 for (const movie of movieList)
11   if (movie.director === "Chris
12     nolanMovieList.push(movie)
13   }
14 }
15 console.log(nolanMovieList.leng
16
17 // Get titles of movies with an
18 const bestTitles = [];
19 for (const movie of movieList)
20   if (movie.imdbRating >= 7.5)
21     bestTitles.push(movie.title
22   }
23 }
24 console.log(bestTitles);
25
26 // Compute average movie rating
27 let ratingSum = 0;
28 let averageRating = 0;
29 for (const movie of nolanMovie
30   ratingSum += movie.imdbRating
31 }
```

