

Arrays and objects in JavaScript are like stories and newspapers!

Why say arrays and objects in JavaScript are like stories and newspapers. Let's TipsMake.com review each type in this article offline!

1. Do you know the 15 hottest programming languages ??on this GitHub?
2. 12 extremely useful tricks for JavaScript programmers
3. What is JavaScript? Can the Internet exist without JavaScript?

If you've ever read books and newspapers, you can understand the difference between arrays and objects in JavaScript.

When you first started with JavaScript, you were easily mistaken about choosing the best way to organize and store data.

To some extent, you're probably familiar with the array since you started learning about the 'for' loop. However, when there is too much data to be put into the array, you will face a mess that is hard to follow and understand when reviewing your own code.



The choice between the array and the object becomes much easier when you quickly determine the target of each type. Arrays are similar to how books store information, while similar objects are a newspaper that stores information. Let's TipsMake.com review each type in this article offline!

Arrays: The order of data is the most important.

Here are each chapter of a short storybook presented in an array form.

```
var book = ['foreword', 'boyWhoLived', 'vanishingGlass', 'lettersFromNoOne', 'af
```

Our book consists of the first three chapters in the Harry Potter book. This array is displayed in the form of images.



You want to use arrays when order is the most important thing in sorting information data. I hope no one reads the Harry Potter chapter title and thinks: "*Hmmm, this part looks good, skip the previous paragraph and read this paragraph!*". The order of the chapters in the story tells you which one to read first, which one is later.

When retrieving information from an array, you use the **index** of each element in the array. The array in JavaScript is **0-indexed**, meaning that the elements in the array are marked with 0 instead of 1.

That means that when you want to get data at the book's index 0, you should use the command:

```
book[0]
```

Results returned:

```
'foreword'
```

If you want to name the third chapter in the book, you use:

```
book[2]
```

You choose which chapter to read based on the order of the index, not based on the chapter title.

Objects (Objects): The title of information (Data Label) is the most important.

Below is a newspaper that presents the audience:

```
var newspaper= {  
  sports: 'ARod Hits Home Run',  
  business: 'GE Stock Dips Again',  
  movies: 'Superman Is A Flop'  
}
```

This is the same data but the newspaper is presented in the form of images:

Objects are suitable for data sorted **by title** . When reading a newspaper, you usually don't read from the first to the last page. You can easily jump between sections or move to certain parts of your interest. No matter where the item is located, the reader quickly overcomes this section and reads the necessary information. This is the difference between a newspaper and a book, story lines and order of chapters that are very important.



Objects store information in each pair of **properties / values** (key / value pairs).

```
key: value
```

If you want to see the information in the Business section of the newspaper, you will use the "business" attribute to see the value:

```
newspaper['business']
```

or

```
newspaper.business
```

The above statement returns the value information contained in this ' *GE Stock Dips Again* ' attribute . So, the easiest way to get data is based on the title (or attribute), you should save the data as an object.

Combine arrays and objects

In the previous section, we talked about saving information in arrays or objects. In addition, you can also store other basic data types such as numbers and booleans objects in the following ways:

1. Array in the object;
2. Object in array;
3. Array in array;
4. Objects in the object;

This can become a bit complicated. However, it is almost certain that you will need this combination to save data to help extend data later. You will definitely want to store it in a way that you can understand your code at a later week.

Let's review the example of the previous book. For example, if we need to save the page number of each chapter. We can save in the following ways:

```
var book =[
  ['foreword', 14],
  ['boywholived', 18]
]

var book = [
  {name:'foreword', pageCount: 14},
  {name:'boyWhoLived', pageCount: 18},
  {name:'vanishingGlass', pageCount: 13},
  {name:'lettersFromNoOne', pageCount: 17},
  {name:'afterword', pageCount: 19}
];
```

We have maintained the order of chapters and now we have the ability to name specific features of each chapter. Therefore, if we want to know the page number of the second chapter, we can use:

```
book[1]['pageCount']
```

The result returns 18.

Now let's say you want to see the top writers' rankings for each section of the newspaper, based on your work time. You can express that in an array used in the newspaper object like:

```
var newspaper= {
  sports: 'ARod Hits Home Run',

  sportsWriters: ['Miramon Nuevo', 'Rick Reilly', 'Woddy Paige'], //nhà báo m
?c th? thao
  business: 'GE Stock Dips Again',
  businessWriters: ['Adam Smith', 'Albert Humphrey', 'Charles Handy'],
  movies: 'Superman Is A Flop',
  moviesWriters: ['Rogert Ebert', 'Andrew Sarris', 'Wesley Morris']
}
```

Arrays are a good choice to save journalists' names because we need to arrange them in order (work experience). Looking at it, you can understand that the previous journalist has more long-term work experience than the back press in each segment. Journalists arranged from **index 0** are the oldest journalists.

In addition, you can also optimize the above newspaper by creating an object inside the newspaper object. For example, the " *sports* " property now has the value of an object, inside the " *sports* " object there are two " *title* " attributes that store the " *sports* " heading and the " *writers* " attribute that stores the array of houses. newspaper of " *sports* " section.



JavaScript

Here are a few small challenges for you:

1. Suppose your web application has a quiz, where users fill in some questions and then get points after completing. You want to save all user answers and then test them again effectively. Which structure will you use to store all user answers before testing? Why?
2. Suppose the user creates a new profile on your site with your own name, last name, email and password. You want to store this information before sending it to the backend. What structure will you use to store all new user information? Why?
3. Suppose you are building a forum page, where you need to rate comments based on votes (votes). Which data structure is suitable if you need to track the comment content and the number of votes of each comment?**Hint** : combine 2 forms.

Author: Kevin Kononenko

Refer to some more articles:

1. Beginners of computer programming need to focus on what?
2. Journey from unknown to becoming software engineer in San Francisco for 12 months
3. Journey to change jobs from fashion models to software engineers within 1 year

Having fun!

You finished reading the article "**Arrays and objects in JavaScript are like stories and newspapers!**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.