

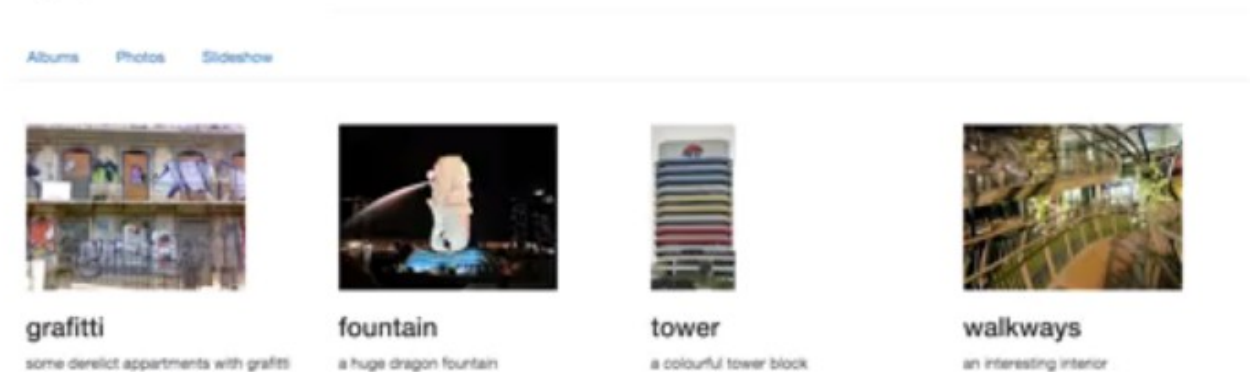
### 2.4.1.1: JavaScript arrays

#### JavaScript arrays:

[https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/enUS/docs/Web/JavaScript/Reference/Global_Objects/Array)

Last week, we looked at the separation of the structure from the webpage and the content via data objects. Now, that's a really powerful technique but there's still a drawback, which we only ever worked with one individual data object, or maybe two. Last week, we talked about how we wanted to create really massive websites that sort of scale up to thousands of images across sort of hundreds of users.

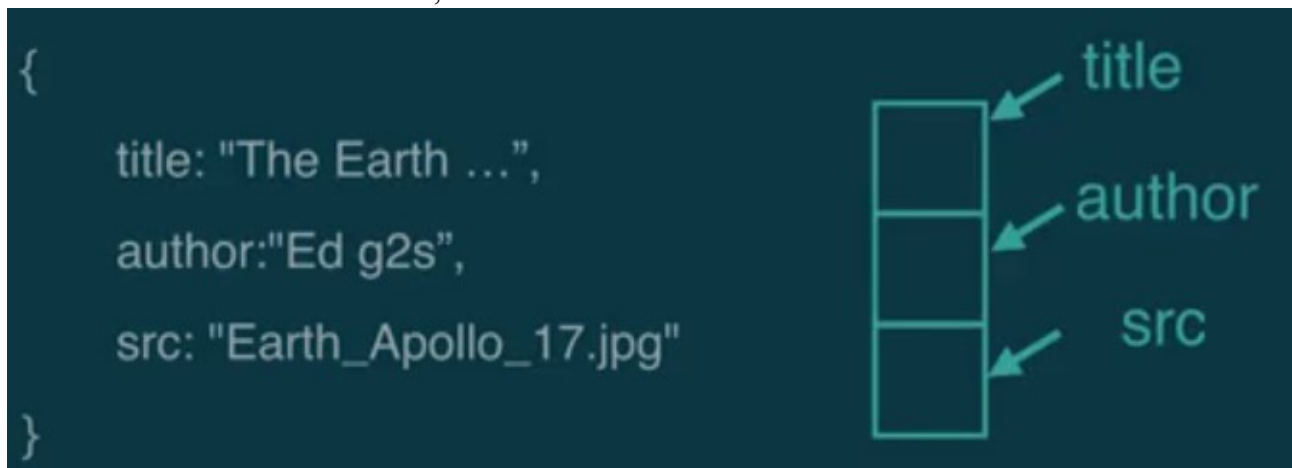
#### My photo albums



If we look at this kind of photo album example, then you can see that we've really got several albums. Each album contains several photos and if we look at a realistic one, you know we are looking at tens, if not more, photos in each album and possibly hundreds of albums. To have a really modern site, we need to scale this up quite a lot and part of that scaling up is being able to handle lots and lots of individual data items. We need to go from our single object, single template approach that we've been looking at into large amounts of data. I'm gonna start talking about how we do that and how we represent large sets of data.

We know a bit about data already. We've talked about variables, which is a box where you can put a bit of data and give it a name and that's essentially, one piece of data.

We know a little bit about how to get more than one piece of data because we've looked at objects where we can have several boxes, each of them with a name.



But, once we get bigger and bigger, we don't want to sort of have to invent a name for every single item in our database. In fact, if you think about a gallery, we're not going to know until the website's running and people start uploading photos, exactly which albums there are and exactly which photos there are.

- We can't give a name to each individual one in our original code. We need to be able to sort of add them as we go. We need a slightly different structure from the objects we've been looking at so far. This structure is something called an **array**.

If we look at the layout we want, we want a bunch of images and we want a corresponding layout in terms of memory, in terms of variables, which is just add a bunch of boxes like this. Each image corresponds to one of these boxes in memory. The way an array works is that it's based not on names but on numbers. These boxes have an order. There's a first, second, third. Now one slightly strange thing you may notice, I started at zero, that's just an age old convention in computer programming. I won't particularly go into the reasons. They're not necessarily relevant anymore in the world of JavaScript. We still can use it because most other languages start at zero.

- You just have to remember an array starts at zero, and then one, two, three, four, etc. Each of these little boxes can contain any kind of data we like. They can contain numbers, text, objects, even other arrays.

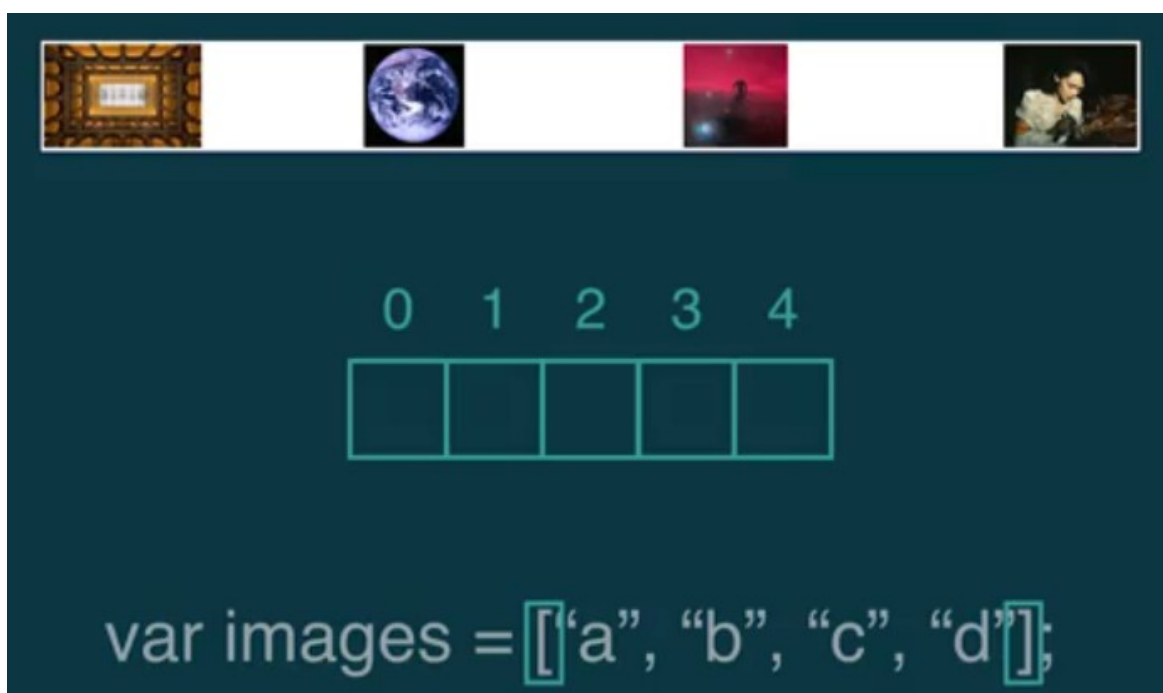
An array, when you create one, looks like

```
var images = ["a", "b", "c", "d"];
```

The key thing is that they've got these square brackets. Up to now, we're working with curly brackets, an object.

- The start and end of an object is defined by these curly brackets. To distinguish an array from an object, you use square brackets instead.

There's a bunch of elements separated by these commas, and then the elements are whatever they are. Each of the elements, you can grab hold of using a number.



What we're going to want to do is get hold of individual images out of the array. To do that, if we want to get hold of image number 1, which is actually the second one, we do this.



All we need to do is give it a number and it gives us back an image, an element to that array. Again, we're using square brackets to contain that number. It looks a little bit like how we call a function with curly brackets but instead, we're using square brackets and again, that tells us this is an array. We can do things like putting values in the array, each element to the array is just a variable, so we can put a value in there, just as we can with a variable. We can actually add another element. If there's no element five, if I put an element five in there, it just adds on to the end of the array and we can put a value in there. For the rest of this week, we're gonna use arrays to scale out our sites, so that we can have galleries and albums with many, many photos in.

The next thing I need to tell you is how to take it away and put this into a template.