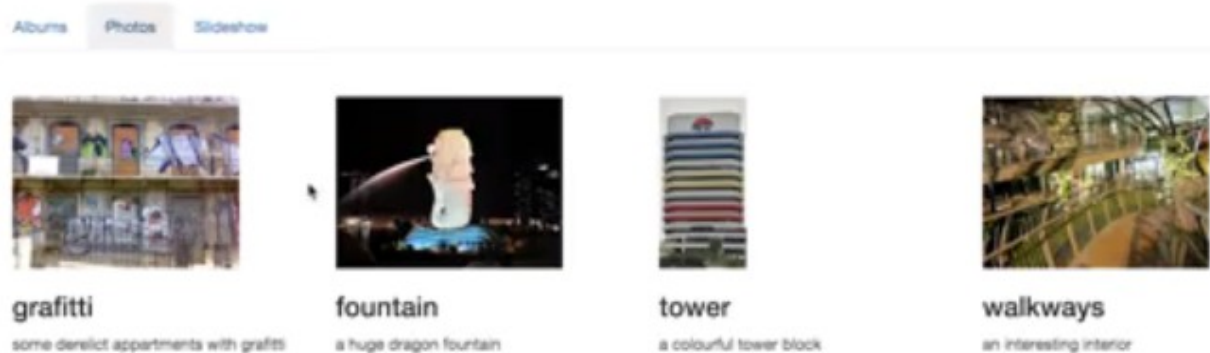


### 2.4.1.3: Dynamically displaying single images from an array

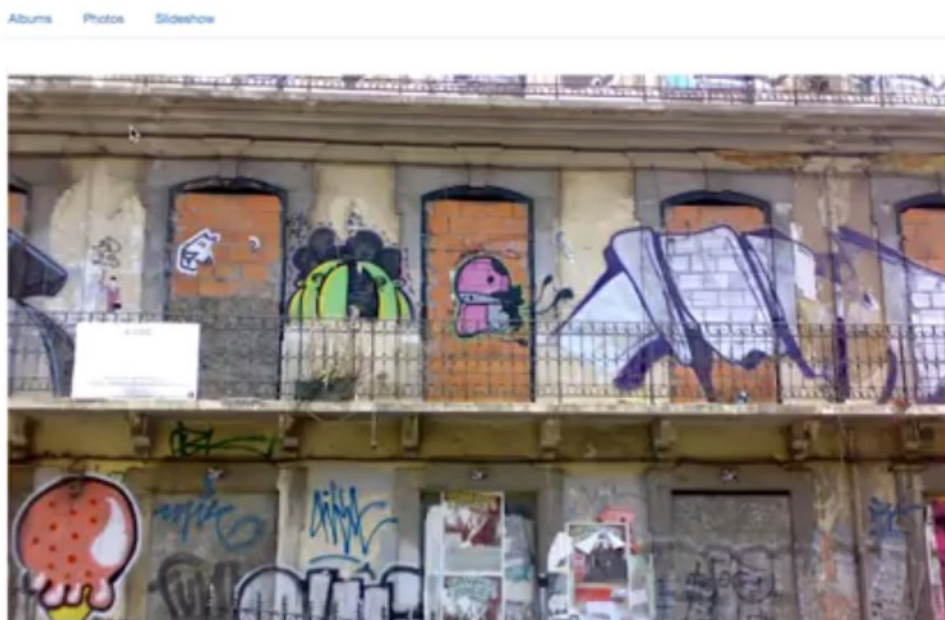
In the last video we looked at how we could store lots of images in an array and put them in a template. Now I'm gonna look at some of the details of how you work with a site like that.

#### My photo albums



So if we have a look at it here we can see a little photo album with a number of photos. Now, one of the first things you want to do with this kind of site is be able to see one of the photos bigger. And the obvious thing to do is click on it. So we want to click on a photo and get a bigger view of that. We've already seen how to do that.

#### My photo albums



This is kind of an example we saw a while ago. We've got, we're showing a modal, where putting the data into some templates, generate some html, putting it into a container, and then sort of displaying that in a modal.

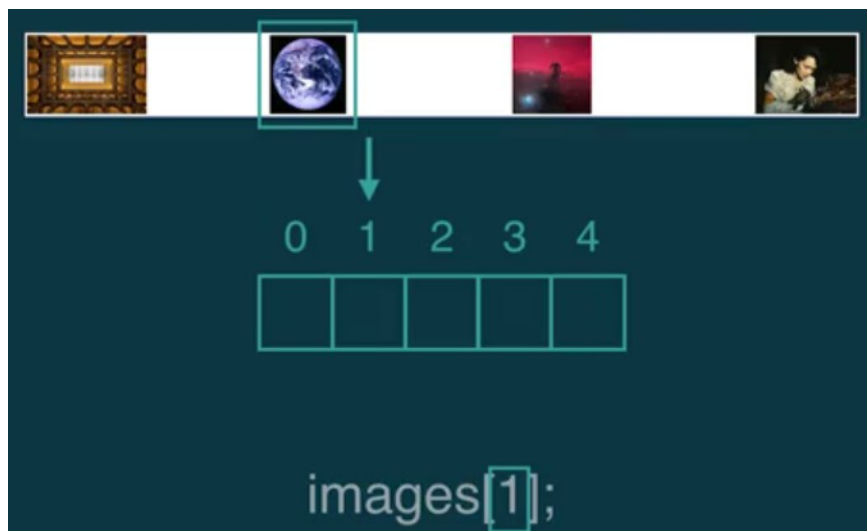
- but now there's a problem. We've got to grab some data to put in. And we don't know, how do we know which image data we're going to put in when we click this function?

```
function showModal () {
    var html = template(data);
    how do we know which image?
    $('#container').html(html);

    $("#image").modal('show');
};
```

So before we've always had a single image object that we're using and applying here. But from now on we've got an entire array of them and we can't create a special show function for every single objects in the array. We need to be able to know already from the template regenerating, which object we're talking about, so that when we click on it, it'll show the correct image. And I'm gonna take you through how we do that.

Just to remind you the key thing we need to do here is something I showed you in the last video. If we want to guess a particular object out of the array, I would say this image here, we grab it by number.



So we need to know which number it is and we use this code, images, square brackets one. So the name of the array is Square Brackets and a number inside there. And that's the kind of element that allows us to do that.

- But in order to do that we need to be able to have a number that corresponds to our image. So the key problem here is how from our template and our jQuery call back code do we actually get the number of the image we've just clicked on.

```
<div class="thumbnail" data-id="{{@index}}">

$('.thumbnail').click(function (event) {
  var imgNum = $(this).data("id");

  var image = data.images[imgNum];
```

.thumbnail selects all images with class thumbnail.

\$(this) refers to the thing they clicked on.

And this is how you do it. So here I'm showing you a couple of things. First I'm at the top, I'm showing you some HTML that can go inside a template. And at the bottom here I'm showing you an on-click function that is able to, that can respond that setup. So we're starting off with our div, which is called Class Thumbnail, and I'm using jQuery to get hold of all the divs with class thumbnail. And call in this click function on. And the first thing we do is then we grab that div, and we use the this variable. So if you remember from the first MOOC the variable this corresponds to the object we've just clicked on. So the HTML element that we just clicked on. So, when we call jQuery this, it's giving us the specific div that we've clicked on. And that already gives us sort of power. It rather than just getting a generic div, or the div of all images, it's giving the div of the specific image you've clicked on. So, does taken us half the way.

We kind of know which div we've clicked on but we still know that's an HTML element. It doesn't tell us which actual object in the array we've clicked on.

- So we need the next stage and that is this data-id. So, there's a special type of ID, of attributes called Data Attributes. And there's nothing particularly strange about them. All the are is the name of the attributes starts with the data-and can be anything else. These are kind of special attributes that don't have any meaning for any standard HTML when doing anything like that. They just used to add extra data onto our HTML elements. So that we can then use it later in our JavaScript.

```
<div class="thumbnail" data-id="{{@index}}">

$('.thumbnail').click(function (event) {
  var imgNum = $(this).data("id");

  var image = data.images[imgNum];
```

And that's what we're doing here. We're adding this extra attribute data-id which is hold the identifier, the number which identifies which image we've got. And down in the jQuery code, we're using jquery.data to get hold of the value of that attribute. So if you call .data, it looks for all the data attributes attached to the elements. In this case I'm asking for the data attribute called ID, so data-id. And it will us back the value of that, which is a number. And what is the value of that? How are we working out what it is? Well, it's this bit of code.

```
<div class="thumbnail" data-id="{{@index}}">
```

The diagram shows a teal arrow pointing from the `data-id="{{@index}}"` attribute in the HTML snippet to the `id` argument in the `$(this).data("id")` call within the jQuery click function. Another teal arrow points from the `imgNum` variable to its use as an index in the `data.images[imgNum]` array access.

So this is another bit of, sort of special templates expression that is returning not an actual attribute of the object we're looking at. So we've got this object attached to the array, which is a particular image. It doesn't have a variable core index. But if you say `@index` it will tell you as long as you're in some sites and template code which is going through an array, so that each template expression that I looked at earlier. It will give you the index, which means the number of the current element in the array. So if we're currently displaying element four in the array, `@index` will be number four. And that's exactly what we need.

We save that value into a variable image number. And so we know now with that variable image number, we now know how to get hold of the actual object for the array because as I've shown you in the previous slides, you just pass that number into the array, square brackets. And that will get you out of that object of that array, which is our image. And from then on in we can just use that value to pass that into a template, create some HTML and then add it to the dom.

Just as we've been doing in the past and I can show you a slightly fuller example.

```
$('.thumbnail').click(function (event) {  
    var imageNumber = $(this).data("id");  
    var html = modal_template(data.images[  
        imageNumber]);  
    $('#modal-container').html(html);  
    $("#imageModal").modal('show');  
});
```

This complete example is available on our course site to download. But we've got what we had here, this click function, we're getting the data-id as I just showed you. And then we're, from there we've got, we're using our data.images array. We're passing the image number into there to get out the object that corresponds to the image you clicked on. We're passing that through the template, creating some HTML. And we're just adding that to the container in the HTML. And that gives us the ability to pop-up some new HTML codes, just as we've been doing in the past with templates and even before that. But this time we don't have to have the actual variable stored in our program. We can grab it half the way. So it allows us to do exactly the same code, exactly the same functionality for every single image in the array, without having to know which particular image it is. And in fact without having to know even how many images. So that we can add images later down the line as people use and upload stuff to our site.