# Lab 3: XML Lists

# Setup

Digital Skills Academy

This lab builds on the JQM list you created in lab 2 and extends it by generating the list items from XML data.

You should complete lab 2 before attempting this lab.

Download and save the supplemental xml file for this lab on Moodle now (WE3.0_A_MWA_3_Supplemental_LOLCatz.xml). You should rename the file to lolcatz.xml to keep things simple.

# Setup

Digital Skills Academy

To save time, you can remove any list item elements from your lab 2 file and resave it for this lab.

Remember to remove the data attribute **data-filter-reveal="true"** so that you can actually see what you're doing first!

You'll also need to temporarily remove any **data-role="page"** attributes.

We will be using ajax to read the supplemental LolCats xml file and use its contents to make a new list.

This technique relies on HTTP calls across the web and some browsers will have difficulty with local files. You can either:

- Work via your localhost.
- Use IE for this session.

# Steps for Part 1

Digital Skills Academy

1. Open the xml file to familiarize yourself with the structure.
2. Add an ID to your UL tag so that you can use jQuery to find it.

```
<div data-role="content">
<div class="content-primary">
    <div data-demo-html="true">
        <ul id="myList">

        </ul>
    </div>
</div>
</div>
```

# Step 2: Load and Parse XML

```
12  <script type="text/javascript">
13
14      //is this working at all?
15      console.log("starting...");
16
17      //standard jQuery ajax technique to load an xml file
18      var xml;
19      $(document).ready(function(){
20          $.ajax({
21              type: "GET",
22              url: "lolcatz.xml",
23              dataType: "xml",
24              success: xmlParser
25          });
26      });
27
28      //loading XML file and parsing ....
29      //this function is the callback specified in line 24 above.
30
31      function xmlParser(data) {
32          xml = data;
33
34          $(xml).find("kitteh").each(function () {
35              kitteh = $(this);
36              var name = $(kitteh).find("name").text();
37              var src =  $(kitteh).find("url").text();
38
39              //console.log(name);
40
41              $("#myList").append('<li>' + name + '</li>');
42
43              });
44          }
45
46  </script>
```

Use jQuery.ajax() to load the xml.

I've started the script for you here but it only adds simple list items now. You will need to improve it.

See the next slide for the requirements.

# Requirements

Digital Skills Academy

Improve the script in the last slide to do the following.

Change the list items (hint, line 41) so that they wrap the LolCat names in anchor tags.

Point the href of each link to some ID named after the ID node in the XML (e.g. href="#1"). Hint, the ID is not being retrieved yet, look at the name and src variables to see how.

Create a parent DIV to contain your JQM "page" DIVs.

Using the same technique as line 41 in the last slide, append a JQM "page" DIV into your DIV container. Give each an ID corresponding to the ID node above.

Add an image tag into each "page" and set the src attribute accordingly.

# Optional Extras

Digital Skills Academy

You should now have a list of links, each pointing to a DIV "page" which contains an image.

Try to reintegrate the pieces you removed earlier to make the list a "page" and to hide the list until it gets filtered.