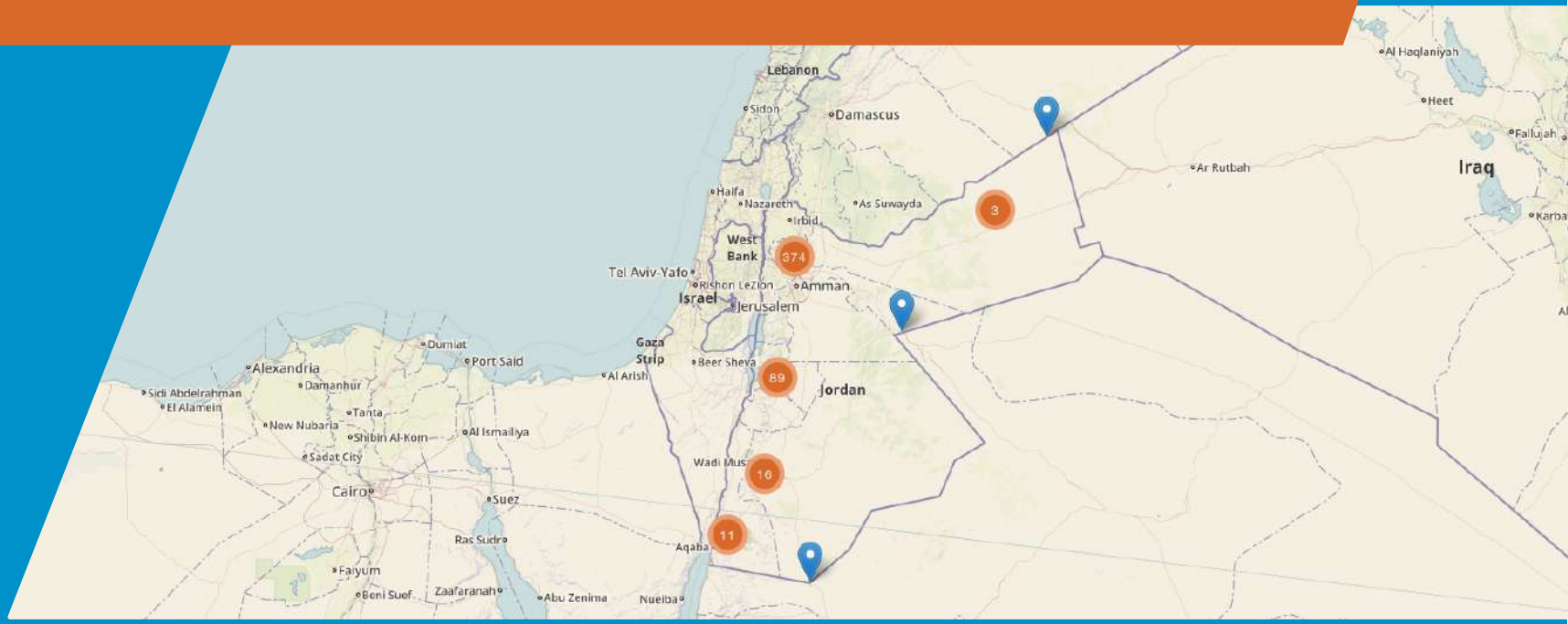
Jordan · Public Health Surveillance



Meerkat Software Training

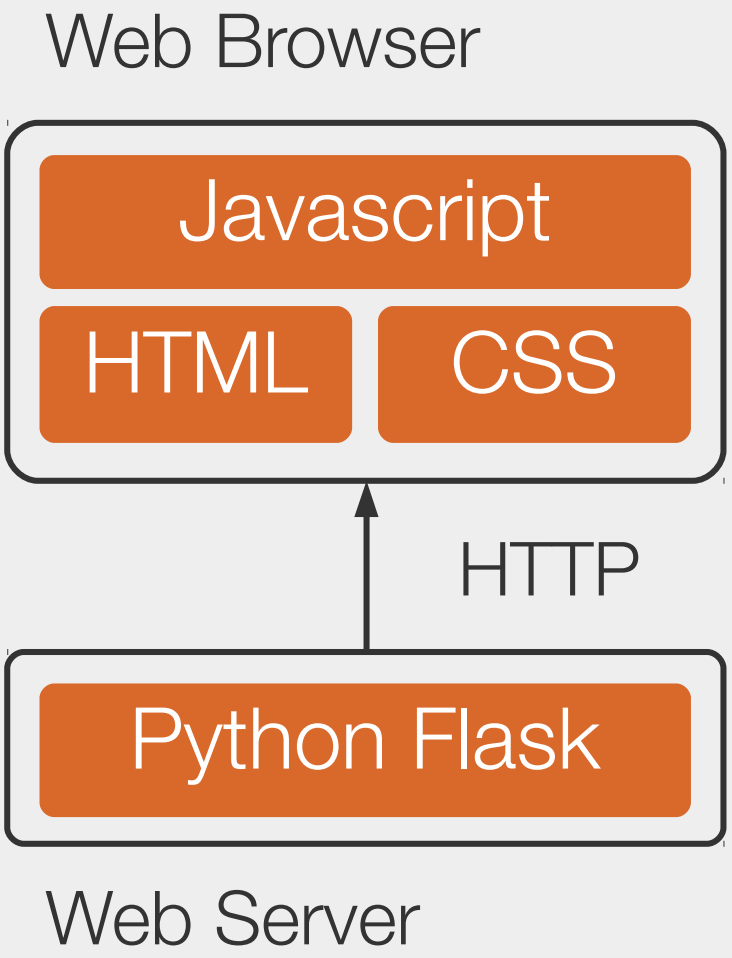
Frontend Javascript Programming

Overview

- **Talk:** Introducing Javascript [15]
- **Exercises:** Javascript syntax and concepts [25]
- **Talk:** Meerkat Frontend Javascript & “asynchronous” programming [10]
- **Exercise:** Getting and displaying data from Meerkat API [45]
- **Summary** [5]

[Total time: 90 mins]

A “client side” scripting language



Executed in Web Browser

```
var from_server = "Hello World";
```

Compiled on server

Sent to Web Browser

```
var from_server = "{{data.str}}";
```

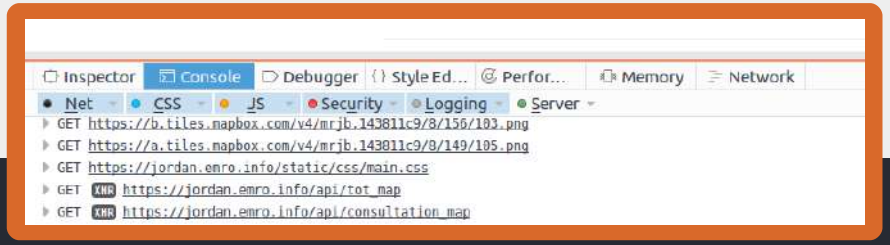
Reviewing a simple script

```
1   <html>
2     <head>
3       <script src="/static/js/jquery.js"></script>
4     </head>
5     <body>
6       <p id='demo1' class='box'></p>
7       <p id='demo2' class='box'></p>
8       <script>
9         document.getElementById('demo1').innerHTML = 'Hi!';
10        $('#demo2').html('Hello!');
11        $('.box').hide();
12      </script>
13    </body>
14  </html>
```

http://jquery.com/

The browser's Javascript console

[Right Click] > [Inspect] > [Console]

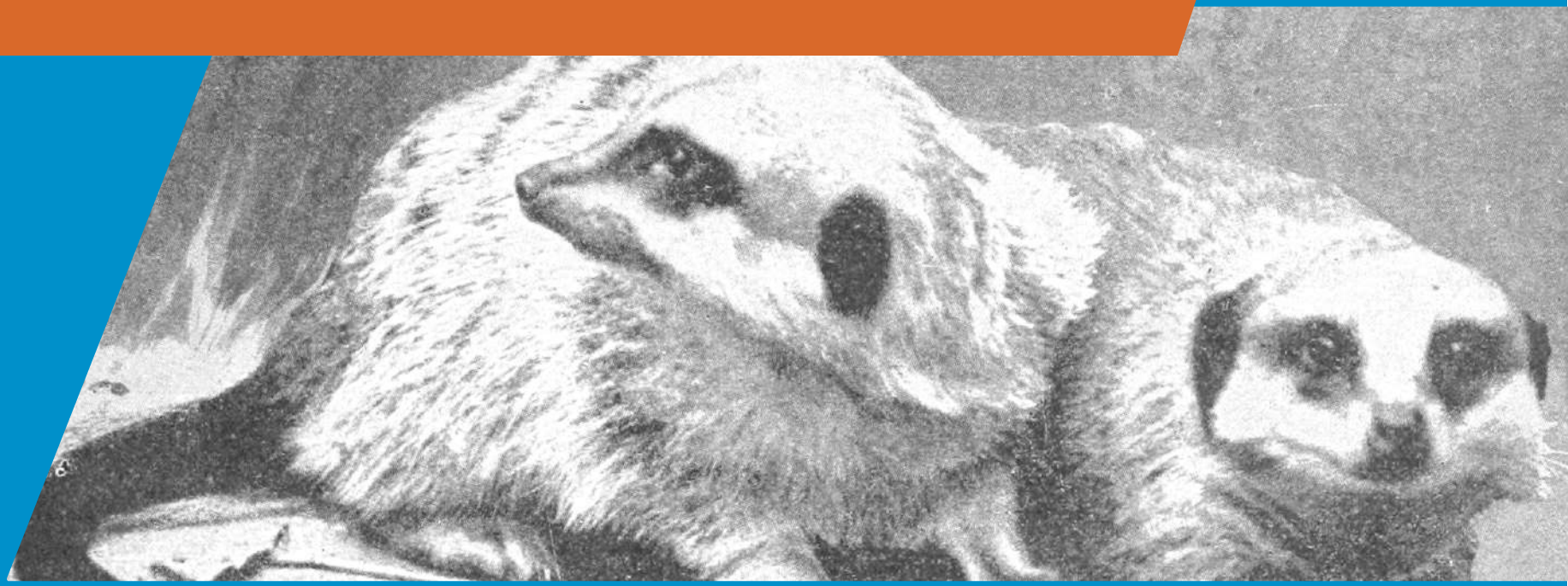



```
get_epi_week();

get_date();

$('.keyIndicator').html('Hello');

$('.keyIndicator').attr('style', 'background:red');
```

Exercise: Javascript Basics

Syntax and Concepts

Meerkat Frontend Javascript



Meerkat Frontend

Javascript can be found in BOTH:

- The static assets source folder **meerkat_frontend/src/js**
- The Jinja2 templates in **meerkat_frontend/templates**

Javascript is imported into our html files here:

meerkat_frontend/templates/includes/js.html

Gulp combines all meerkat javascript into a single file app.js to import into the the html files.

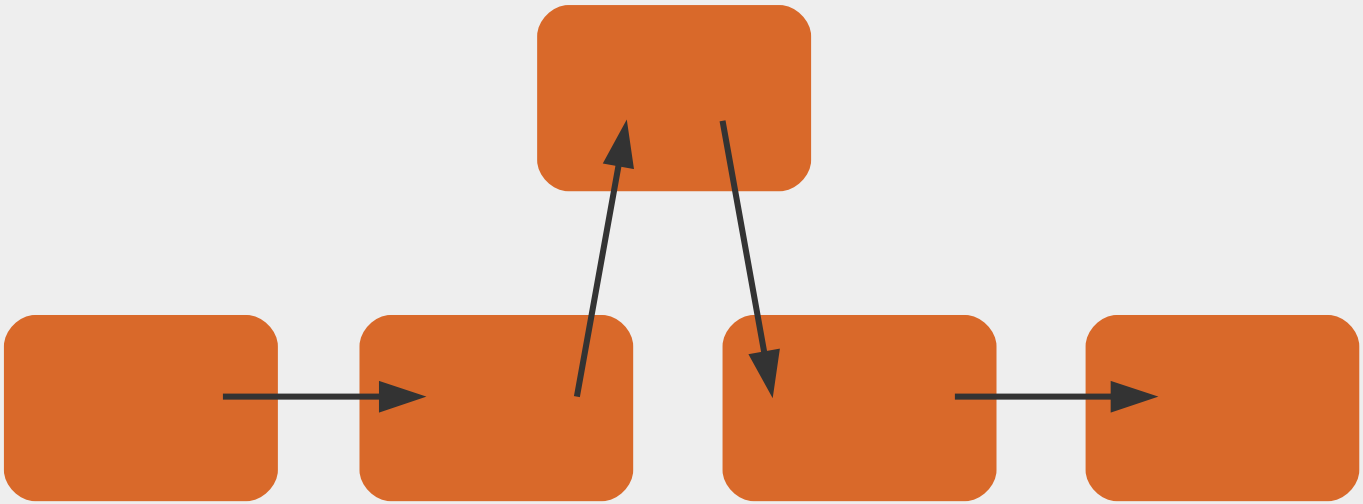
JSON Data from Meerkat API

```
"simple_object":{
    "string": "Hello",
    "number": 33,
    "boolean": true,
    "empty": null,
    "list":["Hello", "world!"],
    "nested":{
        "key": "value"
    }
}
```

Data types:
- string
- number
- boolean
- null
- list
- object

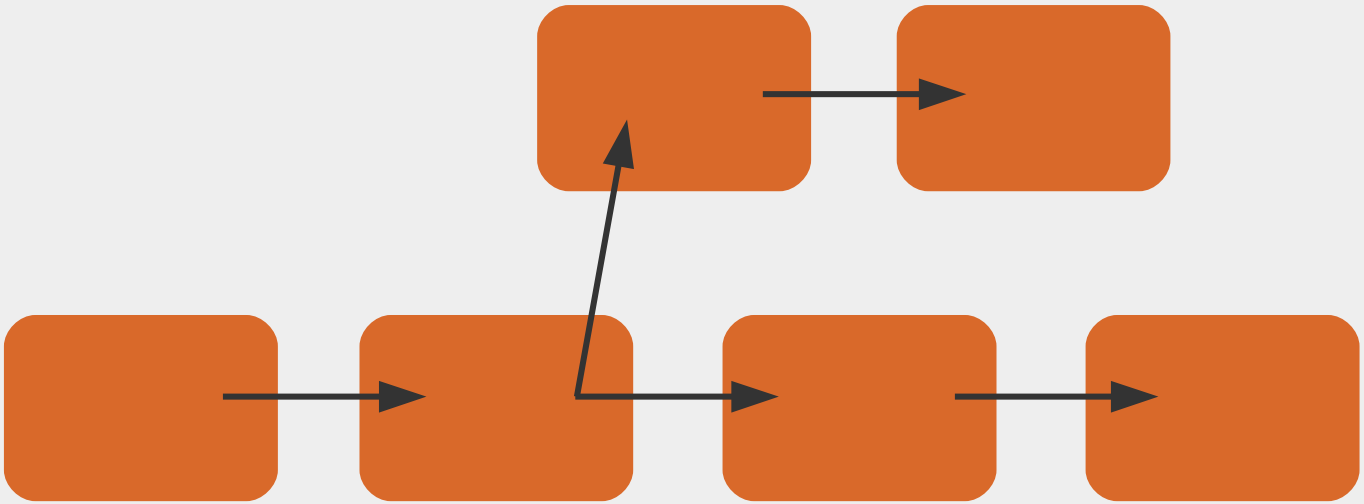
simple_object.string = “Hello”
simple_object.nested.key = “value”
simple_object.list[1] = “world!”

Asynchronous programming



```
$.getJSON( "/api/locationtree", function( data ){
    $('#demo').html(data);
});
```

Asynchronous programming



```
$.getJSON( "/api/locationtree", function( data ){
    $('#demo').html(data);
});
```

Exercise: Editing Meerkat Frontend

Using AJAX to display API data

Summary

A “client side” scripting language

Javascript code is a static resource that can be linked or inserted into HTML files

Code that is dependant on HTTP requests require asynchronous programming structures

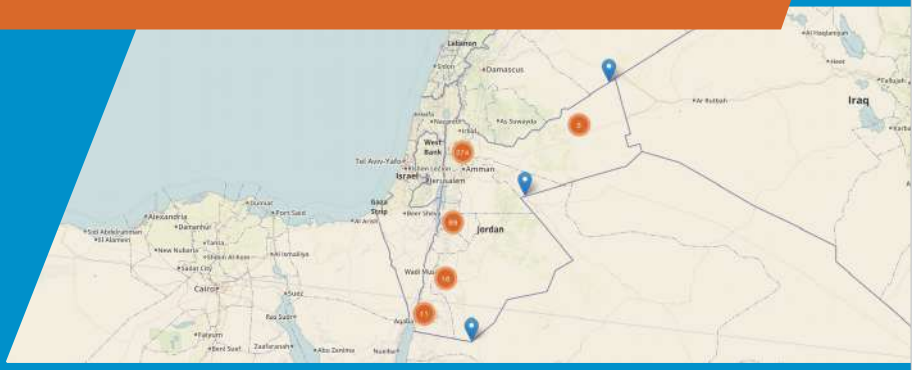
Countless tutorials/docs online to help you e.g...

www.w3schools.com

jquery.com

and of course: google.com

Jordan · Public Health Surveillance



Meerkat Software Training
Frontend Javascript Programming

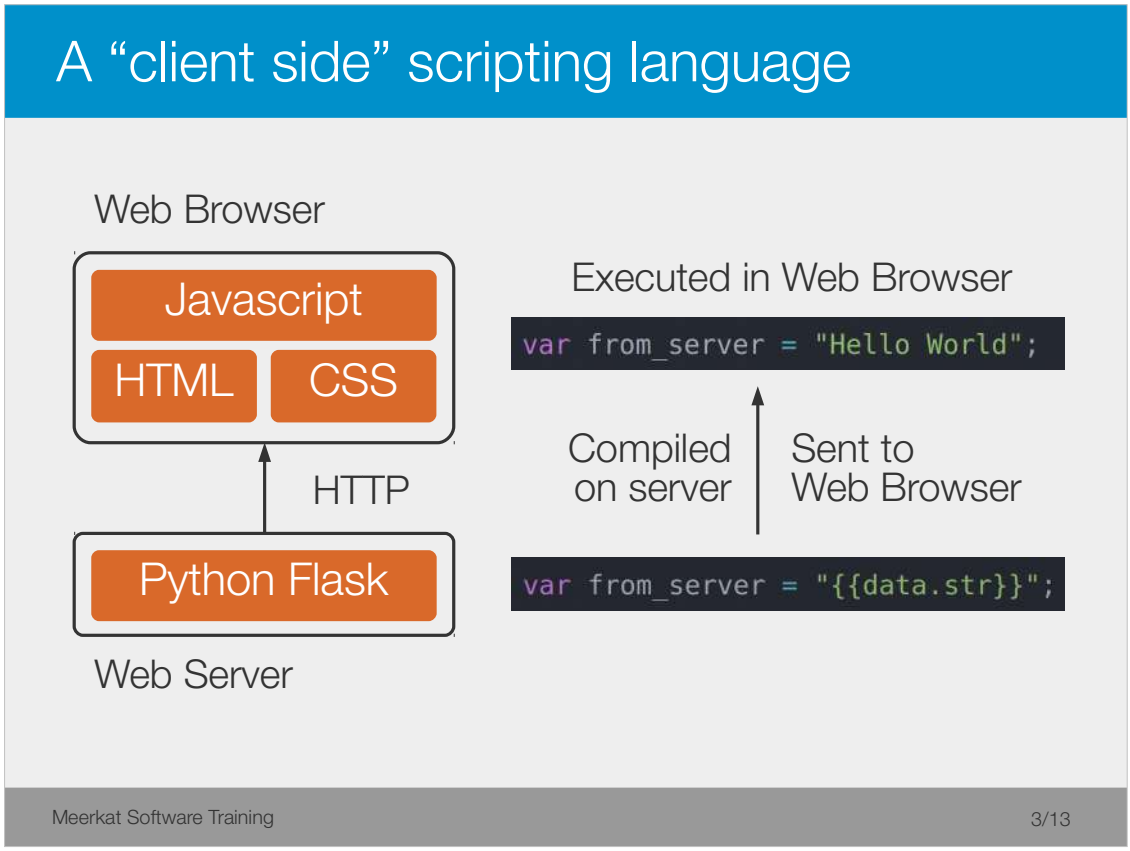
Overview

- **Talk:** Introducing Javascript [15]
- **Exercises:** Javascript syntax and concepts [25]
- **Talk:** Meerkat Frontend Javascript & "asynchronous" programming [10]
- **Exercise:** Getting and displaying data from Meerkat API [45]
- Summary [5]

[Total time: 90 mins]

Key aims:

- Understand the limitations of Javascript as a static resource executed on the client.
- Become familiar with the basics of the Javascript language.
- Introduction to the JSON data format.
- Understand the power of asynchronous programming with AJAX and getJSON
- Become familiar with the structure of Javascript in meerkat_frontend.
- Make some simple Javascript edits to meerkat_frontend.
- Know where to look for further tutorials docs and resources.



Javascript is primarily used as a “client side” scripting language in Meerkat Frontend. This means that the code is executed IN THE BROWSER. If you recall our web technology stack from earlier, Javascript sat right at the very top...

Javascript allows us to create interactive dynamic pages without having to resend a request to the web server for every little change. In meerkat Frontend we use javascript to draw charts...

It's important to understand that this limits what Javascript can and can't do. Javascript can't have access to dynamic variables on the server.

If Javascript wants new information from the server it has to request it in the same way the browser does.

So remember: Javascript is “Client Side” it's run in the browser after all the HTML and CSS is loaded.

Oh and... Javascript is VERY different to Java.

Reviewing a simple script

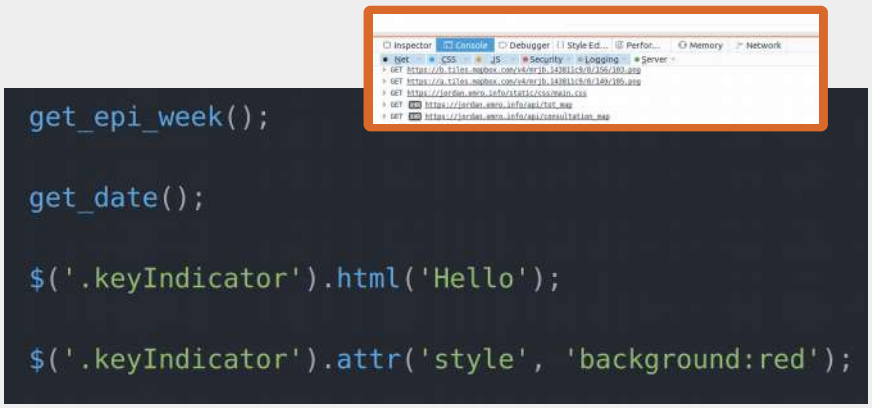
```
<html>
  <head>
    <script src="/static/js/jquery.js"></script>
  </head>
  <body>
    <p id='demo1' class='box'></p>
    <p id='demo2' class='box'></p>
    <script>
      document.getElementById('demo1').innerHTML = 'Hi!';
      $('#demo2').html('Hello!');
      $('.box').hide();
    </script>
  </body>
</html>
```

http://jquery.com/

Here is a very simple script... Let's step through it line by line and see what it does...

The browser's Javascript console

[Right Click] > [Inspect] > [Console]



One of the developers tools available in most modern web browsers is the ability to execute javascript commands in real time within a particular web site's environment.

We do this using the Javascript console.

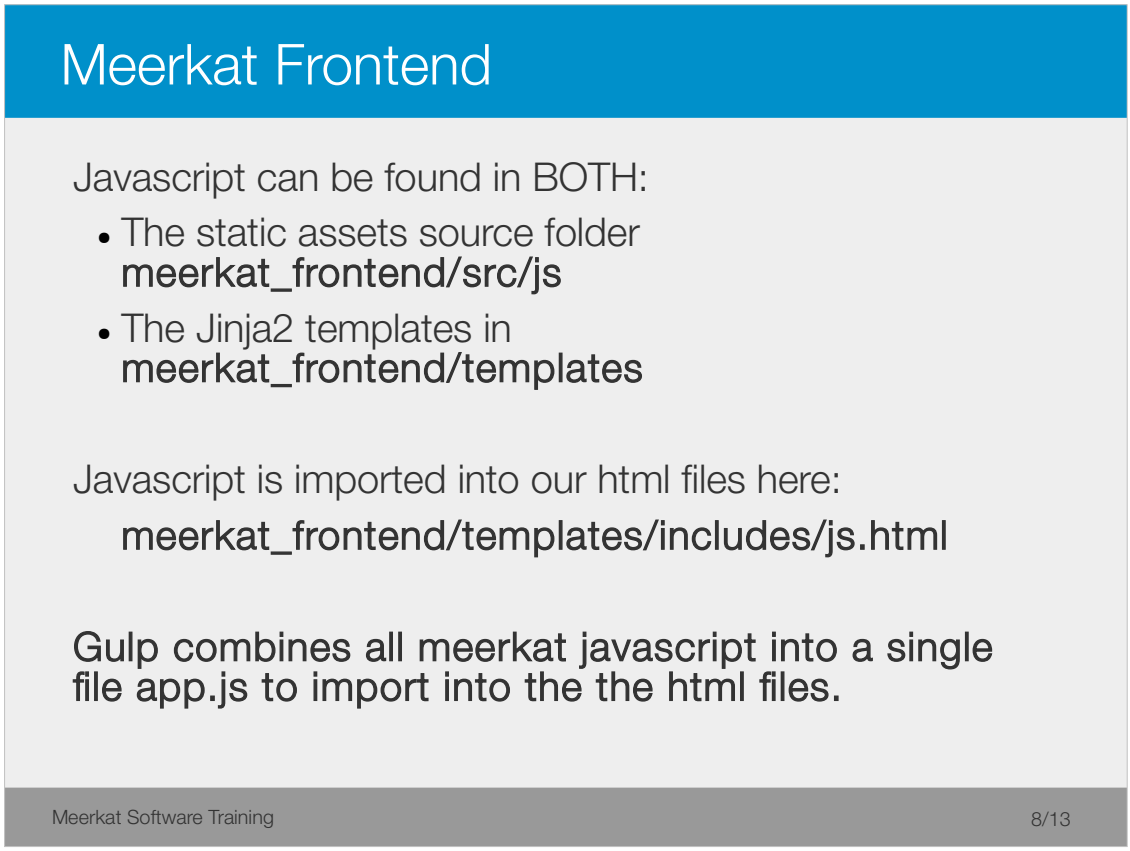
Examples: iers.moh.gov.jo

- get_epi_week()
- get_date()
- Jquery: $('.keyIndicator').html('hello');
- $('footer').attr('style', 'background:red');

Exercise: Javascript Basics

Syntax and Concepts

Meerkat Frontend Javascript



Javascript is both included in the Jinja2 templates and in the static assets source folder.

All the Javascript in the src folder is combined by the Gulp build process into a single file called app.js that is found in the static assets folder. This helps reduce the number of separate requests made for Javascript files.

Javascript from installed packages are also copied into the static folder by Gulp. All this Javascript is then imported into the HTML files using Jinja2.

The debug/development trail begins in a Jinja2 template, and will make reference to functions defined in the imported Javascript.

JSON Data from Meerkat API

Data types:
- string
- number
- boolean
- null
- list
- object

simple_object.string = “Hello”
simple_object.nested.key = “value”
simple_object.list[1] = “world!”

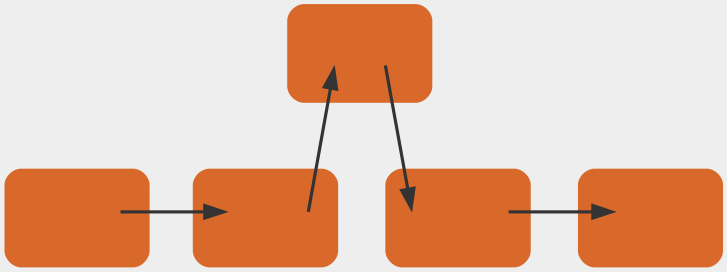
JSON stands for Javascript Object Notation, and is a widely used data format. It's both very simple and very flexible, and quite similar to the concept of a Python dictionary.

When exchanging data between a browser and a server, the data can only be text. JSON is a textual representation of a data structure.

A JSON object is a collection of key, value pairs enclosed in curly braces. Each value can be either:

number/string/boolean/null, OR, a list of these value types enclosed in square brackets OR another nested JSON object.

Asynchronous programming



```
$.getJSON( "/api/locationtree", function( data ){
    $('#demo').html(data);
});
```


When we want new data from the server we have to request it.

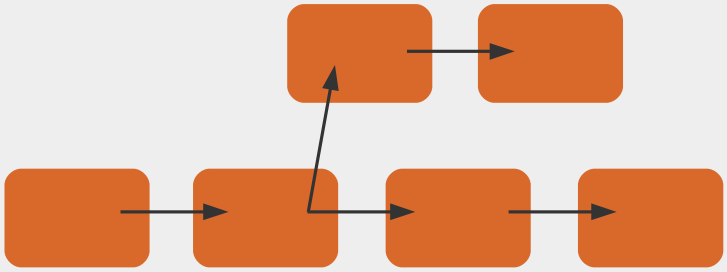
AJAX stands for Asynchronous Javascript and XML. It is a procedure that is implemented in JQuery and allows us to request data from the server and then execute Javascript using it.

We do this all the time in Meerkat Frontend, when we get data from the Meerkat API.

The trouble is that requesting data over the internet is unreliable – it can fail, it can take a long time. Generally we don't want our entire Javascript codebase to freeze because of one AJAX request.

Our code is therefore structured into “callback functions” that are executed upon an AJAX request completing. Changes within the the callback shouldn't impact changes outside the callback.

Asynchronous programming



```
$.getJSON( "/api/locationtree", function( data ){
    $('#demo').html(data);
});
```


When we want new data from the server we have to request it.

AJAX stands for Asynchronous Javascript and XML. It is a procedure that is implemented in JQuery and allows us to request data from the server and then execute Javascript using it.

We do this all the time in Meerkat Frontend, when we get data from the Meerkat API.

The trouble is that requesting data over the internet is unreliable – it can fail, it can take a long time. Generally we don't want our entire Javascript codebase to freeze because of one AJAX request.

Our code is therefore structured into “callback functions” that are executed upon an AJAX request completing. Changes within the the callback shouldn't impact changes outside the callback.

Exercise: Editing Meerkat Frontend

Using AJAX to display API data

Summary

A “client side” scripting language

Javascript code is a static resource that can be linked or inserted into HTML files

Code that is dependant on HTTP requests require asynchronous programming structures

Countless tutorials/docs online to help you e.g...

www.w3schools.com
jquery.com
and of course: google.com