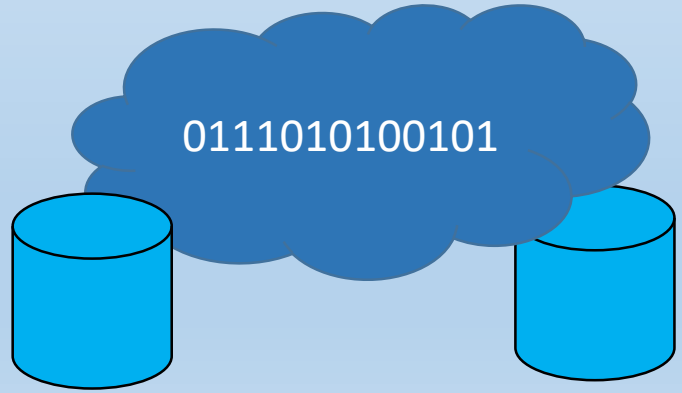
Data Exchange

COMP-3077 Web-Based Data Management

Dr. Saja AL-Mamoori



Standard format

- We have thousands of different Technology, OS, Web-services, APIs ,etc.
- Exchanging data between those technology needs to follow a standard, so all those different sources will speak the same language.
- Exposing data with simple format/paring techniques will help different entities to integrate with each other.
- Integration with other platforms without giving access to the back-end database.
- It started with XML, nowadays, newer technologies prefers JSON.

XML

- Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a standard format.
- It is both human readable and machine-readable.
- The main idea of XML to standardize the data that is transferred/ exchanged on the web.
- XML is HTML-alike format, where there are opening and closing tags for all nodes.
- RSS (+Rich Site Summary) is an application of XML.

XML - Example

```
<employees>
    <employee>
        <firstName>John</firstName> <lastName>Doe</lastName>
    </employee>
    <employee>
        <firstName>Anna</firstName> <lastName>Smith</lastName>
    </employee>
    <employee>
        <firstName>Peter</firstName><lastName>Jones</lastName>
    </employee>
</employees>
```

JSON

- JSON: **J**ava**S**cript **O**bject **N**otation.
- JSON is a syntax for storing and exchanging data. It is simpler than XML.
- When exchanging data between a browser and a server, the data can only be text.
- JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server.
- We can also convert any JSON received from the server into JavaScript objects.
- No complicated parsing and translations.
- It s not a markup language.

JSON – Object and arrays

```
//object in JavaScript
var myCat= {
      "name": "Whiskers",   //attribute:value
       "species" : "cat",
       "favfood": "tuna"}
//object as an array
{"employees":[
   { "firstName":"John", "lastName":"Doe" },
   { "firstName":"Anna", "lastName":"Smith" },
   { "firstName":"Peter", "lastName":"Jones" }
]}
```

Reading from animal.json example

- In the following Example we will be implementing:
- HTML file (loaddata.html) that reads and shows the JSON file, on specific event onclick.
- JavaScript file (getload.js) that contains a function to load and parse JSON Object, this function will be called in the onclick() event
- the source JSON file (animal.json) that contains JSON objects.
- Note: anything after(within the same line of) // in JavaSript is a comment, which means that the browser will not read it, it is just for the other developers to understand.

  e.g. // the following code is to read JSON object

animal.json array of objects

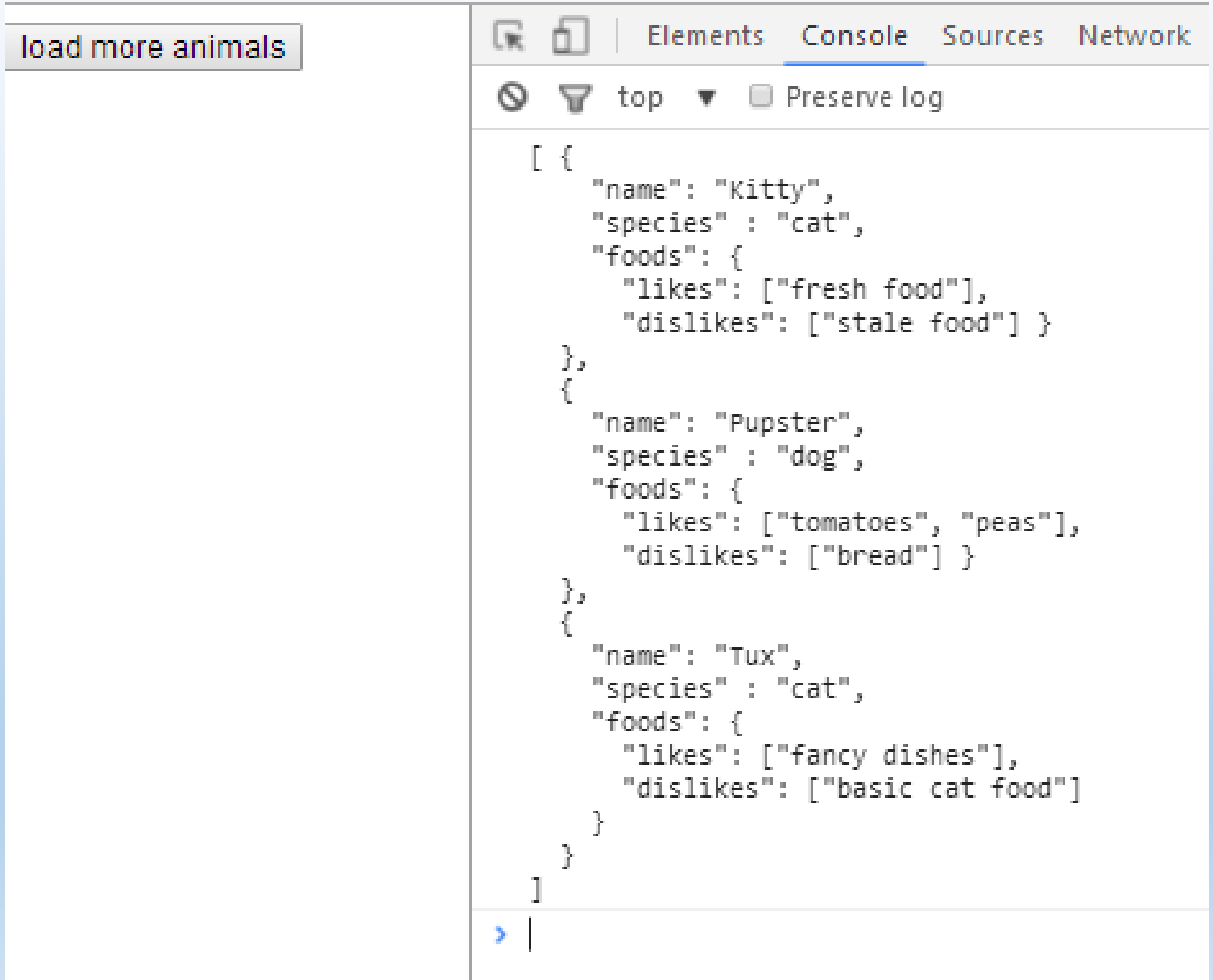
```
[ {
   "name": "Kitty",
   "species" : "cat",
   "foods": {
     "likes": ["fresh food"],
     "dislikes": ["stale food"] }
 },
 {
   "name": "Pupster",
   "species" : "dog",
   "foods": {
     "likes": ["tomatoes", "peas"],
     "dislikes": ["bread"] }
 },
 {
   "name": "Tux",
   "species" : "cat",
   "foods": {
     "likes": ["fancy dishes"],
     "dislikes": ["basic cat food"]
   }
  }
]
```

HTML File to load from JSON - loaddata.html

```
<html>
<head>
<script src="../js/exchangedata.js"> </script>
</head>
<body>
<button onclick="getload()"> load more animals </button>
<div id ="animal-info"> </div>
</body>
</html>
```

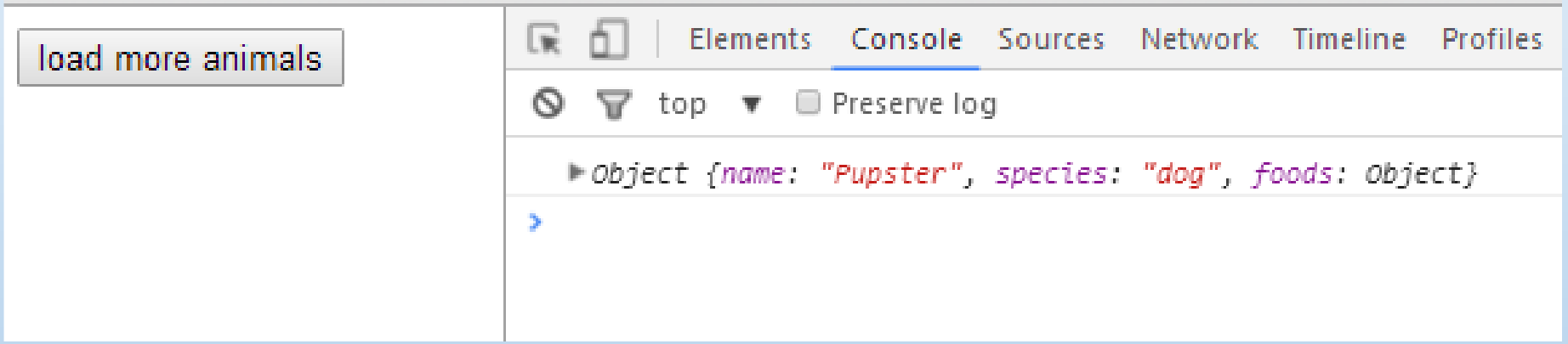
Reading JSON by JavaScript (exchangedata.js)

```
function getload(){   //use AJAX to send a request
var animalrequest = new XMLHttpRequest();
animalrequest.open('GET','animals.json');
animalrequest.onload = function() {
   var result = animalrequest.responseText;
   console.log(result);
};
animalrequest.send();
}
```


load more animals
Elements
Console
Sources
Network
top
Preserve log
[{
"name": "Kitty",
"species" : "cat",
"foods": {
"likes": ["fresh food"],
"dislikes": ["stale food"] }
},
{
"name": "Pupster",
"species" : "dog",
"foods": {
"likes": ["tomatoes", "peas"],
"dislikes": ["bread"] }
},
{
"name": "Tux",
"species" : "cat",
"foods": {
"likes": ["fancy dishes"],
"dislikes": ["basic cat food"]
}
}
]

Parsing the results

```
// we parse the result so we can process it ( save theresponse as an array of
//objects).
function getload(){
var animalrequest = new XMLHttpRequest();
animalrequest.open('GET','animals.json');
animalrequest.onload = function() {
   var result = JSON.parse(animalrequest.responseText); // an array of JSON objects
   console.log(result[1]);
};
animalrequest.send();
}
```


load more animals
Elements
Console
Sources
Network
Timeline
Profiles
top
Preserve log
Object {name: "Pupster", species: "dog", foods: Object}
>

```
function getload(){
var animalrequest = new XMLHttpRequest();
animalrequest.open('GET','animals.json');
animalrequest.onload = function() {
  var result = JSON.parse(animalrequest.responseText);
  renderHTML(result); };
animalrequest.send(); } //end of getload()
function renderHTML(data){
var animalContainer = document.getElementById("animal-info");
var htmlStr = "";
for (i =0;i< data.length;i++)
htmlStr += '<p>' + data[i].name + 'is a' + data[i].species +'.</p>';
animalContainer.insertAdjacentHTML('beforeend',htmlStr); } //end of renderHTML
```

load more animals

Kittyis acat.

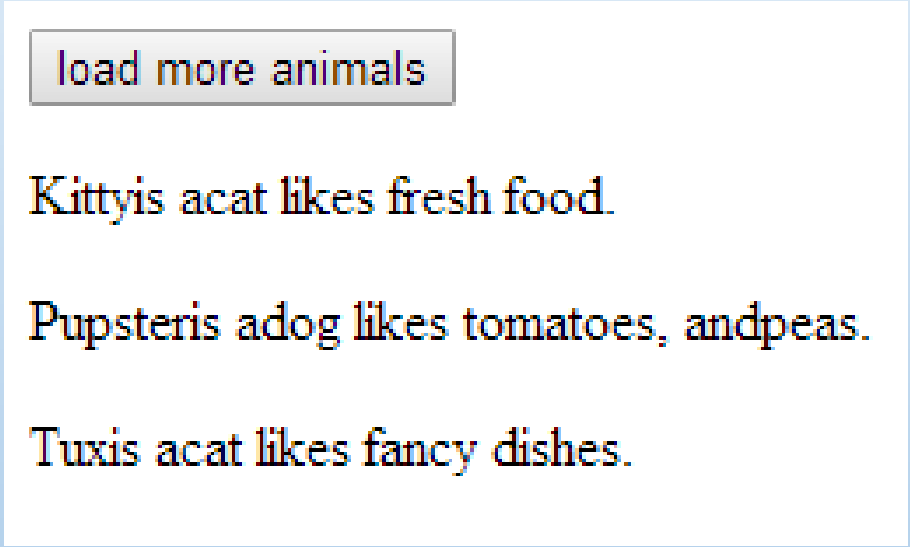
Pupsteris adog.

Tuxis acat.

SubElements/arrays

```
function renderHTML(data){
var animalContainer = document.getElementById("animal-info");
var htmlStr = "";
for (i =0;i< data.length;i++){
htmlStr += '<p>' + data[i].name + 'is a' + data[i].species + ' likes ' ;
        for(j =0;j<data[i].foods.likes.length;j++)
            if (j==0) htmlStr +=  data[i].foods.likes[j];
             else if (j== data[i].foods.likes.length-1)
                htmlStr += ',  and' +data[i].foods.likes[j];
             else htmlStr += ', ' +data[i].foods.likes[j];
htmlStr += '.</p>';
}
animalContainer.insertAdjacentHTML('beforeend',htmlStr); }
```

One click


load more animals
Kittyis acat likes fresh food.
Pupsteris adog likes tomatoes, andpeas.
Tuxis acat likes fancy dishes.

Two clicks

load more animals

Kittyis acat likes fresh food.

Pupsteris adog likes tomatoes, andpeas.

Tuxis acat likes fancy dishes.

Kittyis acat likes fresh food.

Pupsteris adog likes tomatoes, andpeas.

Tuxis acat likes fancy dishes.

Repeating, it reloads the same Data again!!!

Different data for each click

- Different APIs have different ways to expose their data.
- Assuming that I have 3 different JSON file:
  - animals-1.json
  - animals-2.json
  - animals-3.json
- At each click, we will load different file, e.g. at first click animals-1.json.

Different file for each click

```
var counter=0;  //counter for the clicks
function getload(){
var animalrequest = new XMLHttpRequest();
++counter; //counter will be increased by 1 for each click
animalrequest.open('GET','animals-'+counter+'.json');
animalrequest.onload = function() {
   var result = JSON.parse(animalrequest.responseText);
   renderHTML(result);
};//exchangedata.js
```

3 clicks

load more animals

Meowsyis acat likes tuna, andcatnip.

Barkyis adog likes bones, andcarrots.

Purrpawsis acat likes mice.

Whiskersis acat likes celery, andstrawberries.

Woofis adog likes dog food.

Fluffyis acat likes canned food.

Kittyis acat likes fresh food.

Pupsteris adog likes tomatoes, andpeas.

Tuxis acat likes fancy dishes.

Broadcasting JSON

- The simplest way for broadcasting data on the web is to create JSON dumps, which is read by the other users.
- In the following example, we will generate JSON file from news table, to mimic broadcasting data in the simplest form.
- Keeping in mind, most of the public website use advanced methods for broadcasting data such as web services and APIs, to register the reader. (get API key) .
- e.g.(weather-MAP API https://www.youtube.com/watch?v=53AoDB7vcbU).

generatejson.php

```
<?php
  require_once 'login.php';
  $conn = new mysqli($hn, $un, $pw, $db);
  if ($conn->connect_error) die($conn->connect_error);
  $query  = "SELECT * FROM news";
  $result = $conn->query($query);
  if (!$result) die($conn->error);
```

generatejson.php

```
$myfile = fopen("news.json", "w") or die("Unable to open file!");
 $txt ='[ ';
 for ($j = 0 ; $j < $rows ; ++$j){
   $result->data_seek($j);
 $txt.='{"title":"'. $result->fetch_assoc()['title']   . '",  ' ;
 $result->data_seek($j);
 $txt.= '"news":"'.$result->fetch_assoc()['news'].'"}' ;
 }
```

generatejson.php

```
$txt.=" ]";
echo $txt; // unnecessary
fwrite($myfile, $txt);
fclose($myfile);
?>
```