Load JSON file locally using pure Javascript

Having needed an pure javascript alternative to jQuery's **$.getJSON()** and **$.parseJSON()** recently I thought I would try and clear this up for future users. There is a lot of misinformation out there on the web. By far the main trouble maker is the belief that by including a .JSON file between the head tags of your HTML document you can access structured JSON.

#The wrong way

```
<script type="text/javascript"
src="mydata.json"></script>
```

Many examples will evidence that you can access the data with a simple function such as the one below. In fact, what this is not actually loading a JSON document but creating a Javascript object. *This technique will not work for true JSON files.*

```
 // 'JSON' data included as above
```

```
  data = '[{"blue" : "is ok", "red" : "is my fave
color"}]';

 // Function to 'load JSON' data
  function load() {
    var someData_notJSON = JSON.parse(data);
    console.log(someData_notJSON[0].red); // Will log
"is my fave color"
    }
```

If you're not fussy about using an actual JSON file then creating a Javascript object in a seperate .js file may be the way to go. If like me you do need to work with JSON in pure Javascript here's what you'll need to do.

#The correct method - create a new XMLHttpRequest

The clue here is the jQuery method **$.getJSON()** which is shorthand for **$.ajax()**. It may seem an odd approach requesting a local file in this way but it offers the most flexibility with minimum fuss.

```
function loadJSON(callback) {

    var xobj = new XMLHttpRequest();
        xobj.overrideMimeType("application/json");
    xobj.open('GET', 'my_data.json', true); //
Replace 'my_data' with the path to your file
    xobj.onreadystatechange = function () {
          if (xobj.readyState == 4 && xobj.status ==
"200") {
```

```
            // Required use of an anonymous callback
as .open will NOT return a value but simply returns
undefined in asynchronous mode
            callback(xobj.responseText);
          }
    };
    xobj.send(null);
}
```

The function above will create a new instance of a XMLHttpRequest and load asynchronously the contents of *my_data.json. I have gone with asynchronous but you can change the argument to false if you want a synchronous load. Thankfully all modern browsers support the native* ***JSON.parse*** *method. Remember our anonymous callback? here's how you use it.*

#Usage

```
function init() {
loadJSON(function(response) {
 // Parse JSON string into object
    var actual_JSON = JSON.parse(response);
});
}
```
