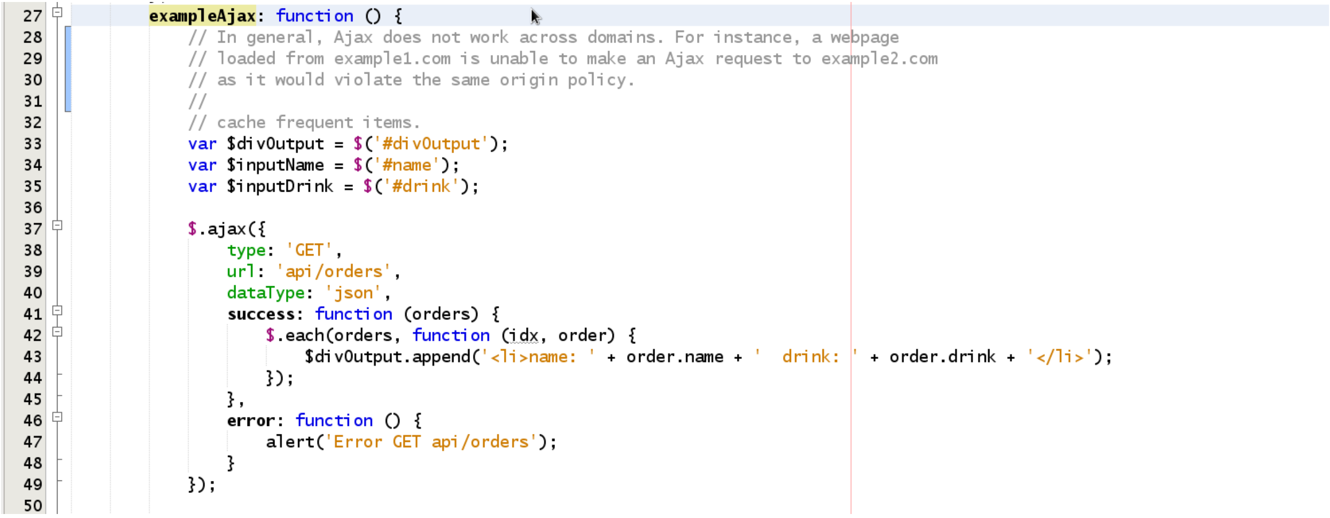
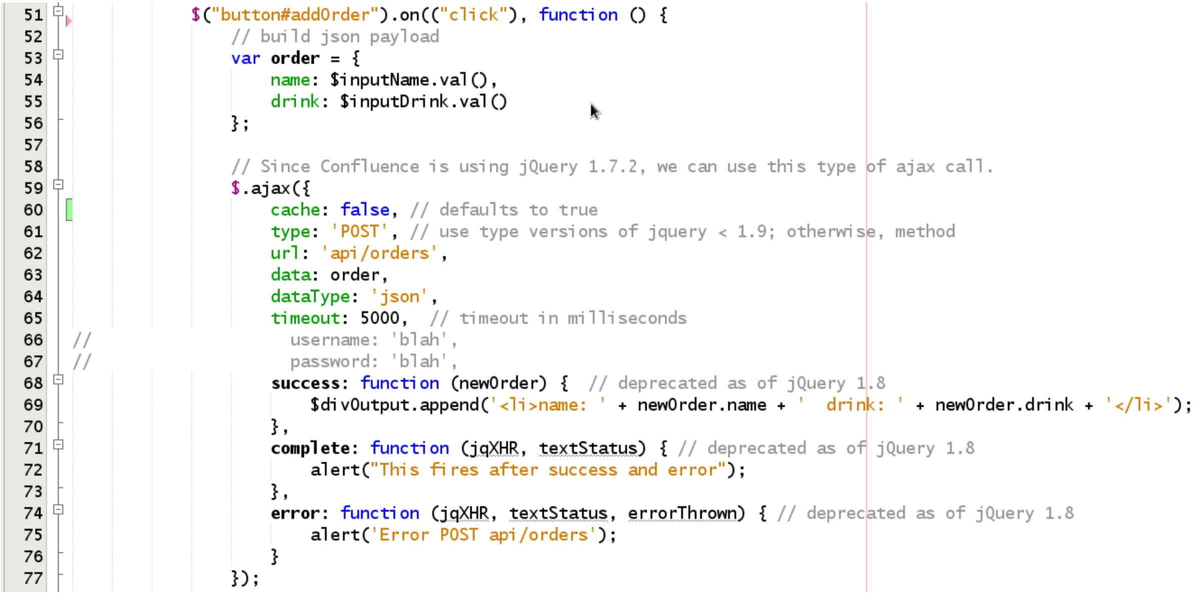
jQuery AJAX

Versions < jQuery 1.9



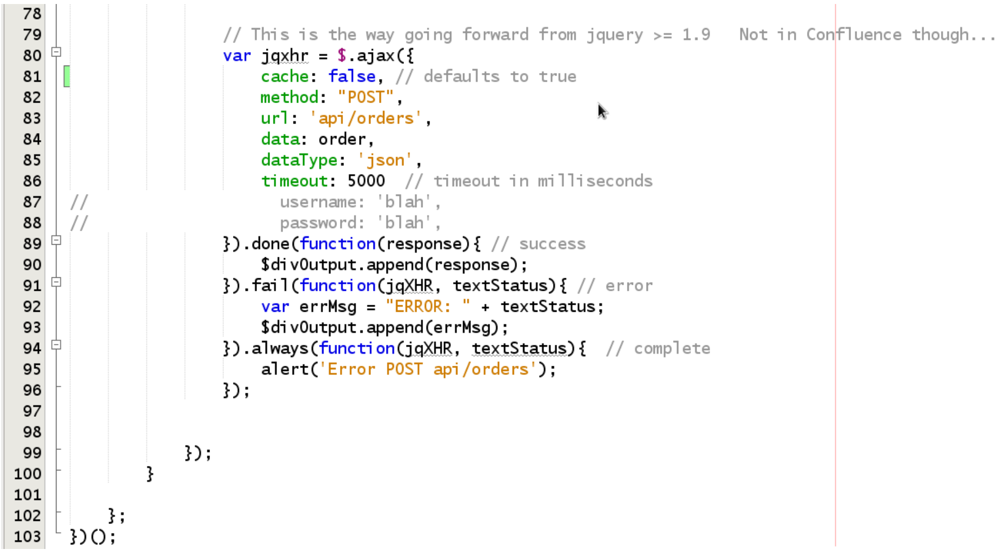
```javascript
        exampleAjax: function () {
            // In general, Ajax does not work across domains. For instance, a webpage
            // loaded from example1.com is unable to make an Ajax request to example2.com
            // as it would violate the same origin policy.
            //
            // cache frequent items.
            var $divOutput = $('#divOutput');
            var $inputName = $('#name');
            var $inputDrink = $('#drink');

            $.ajax({
                type: 'GET',
                url: 'api/orders',
                dataType: 'json',
                success: function (orders) {
                    $.each(orders, function (idx, order) {
                        $divOutput.append('<li>name: ' + order.name + '  drink: ' + order.drink + '</li>');
                    });
                },
                error: function () {
                    alert('Error GET api/orders');
                }
            });

```

Versions < jQuery 1.9



```javascript
            $("button#addOrder").on(("click"), function () {
                // build json payload
                var order = {
                    name: $inputName.val(),
                    drink: $inputDrink.val()
                };

                // Since Confluence is using jQuery 1.7.2, we can use this type of ajax call.
                $.ajax({
                    cache: false, // defaults to true
                    type: 'POST', // use type versions of jquery < 1.9; otherwise, method
                    url: 'api/orders',
                    data: order,
                    dataType: 'json',
                    timeout: 5000,  // timeout in milliseconds
//                    username: 'blah',
//                    password: 'blah',
                    success: function (newOrder) {  // deprecated as of jQuery 1.8
                        $divOutput.append('<li>name: ' + newOrder.name + '  drink: ' + newOrder.drink + '</li>');
                    },
                    complete: function (jqXHR, textStatus) { // deprecated as of jQuery 1.8
                        alert("This fires after success and error");
                    },
                    error: function (jqXHR, textStatus, errorThrown) { // deprecated as of jQuery 1.8
                        alert('Error POST api/orders');
                    }
                });
```

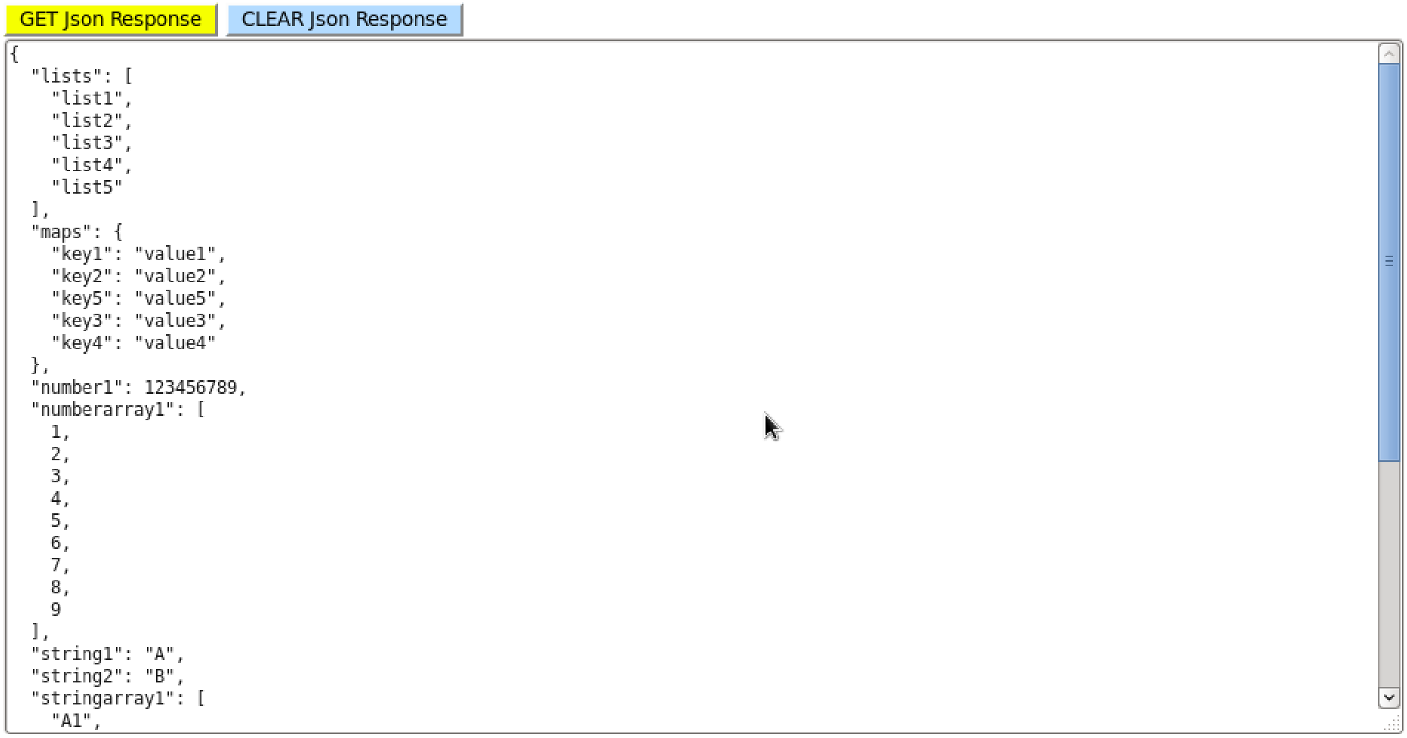
Versions >= 1.9



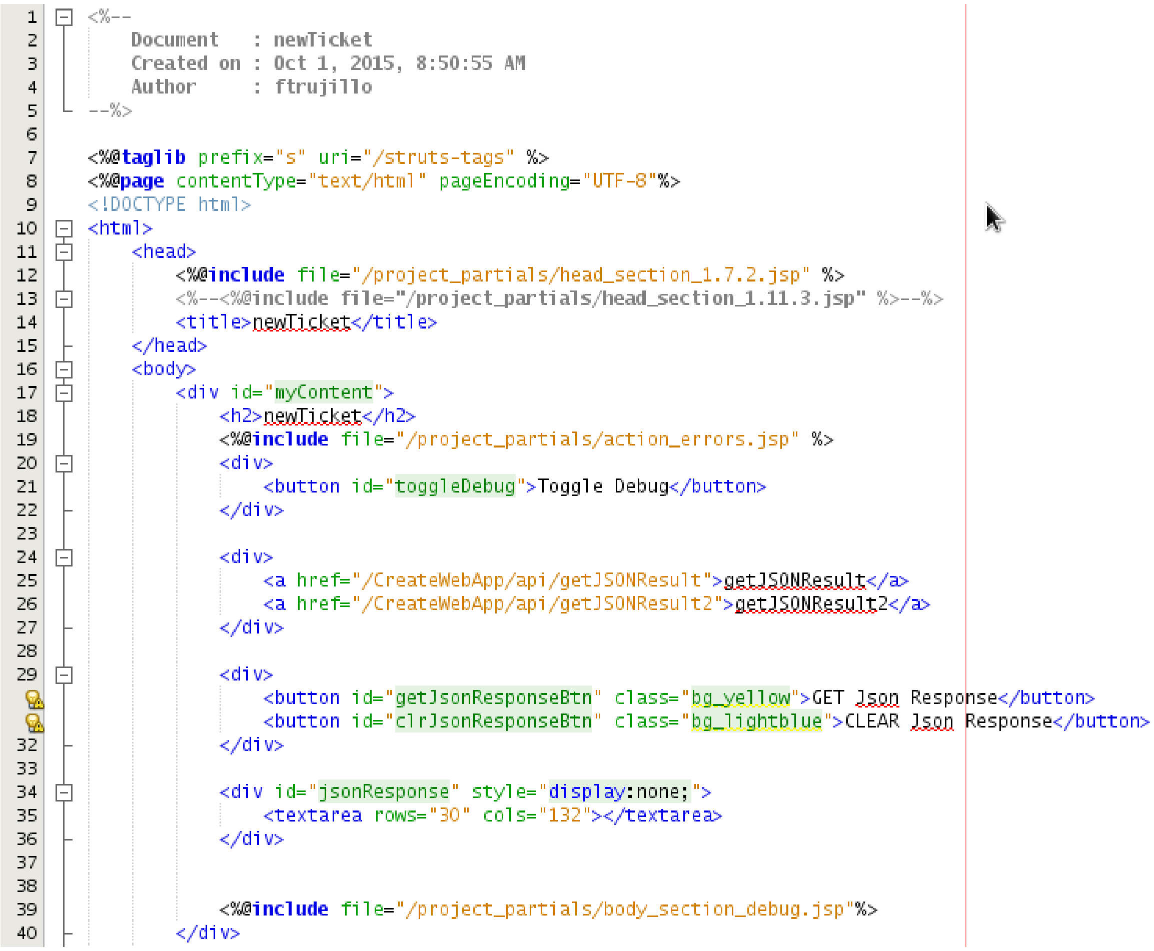
```javascript

                // This is the way going forward from jquery >= 1.9   Not in Confluence though...
                var jqxhr = $.ajax({
                    cache: false, // defaults to true
                    method: "POST",
                    url: 'api/orders',
                    data: order,
                    dataType: 'json',
                    timeout: 5000  // timeout in milliseconds
//                    username: 'blah',
//                    password: 'blah',
                }).done(function(response){ // success
                    $divOutput.append(response);
                }).fail(function(jqXHR, textStatus){ // error
                    var errMsg = "ERROR: " + textStatus;
                    $divOutput.append(errMsg);
                }).always(function(jqXHR, textStatus){  // complete
                    alert('Error POST api/orders');
                });


            });
        }

    };
})();
```

// If you need to pretty print json to an html page....

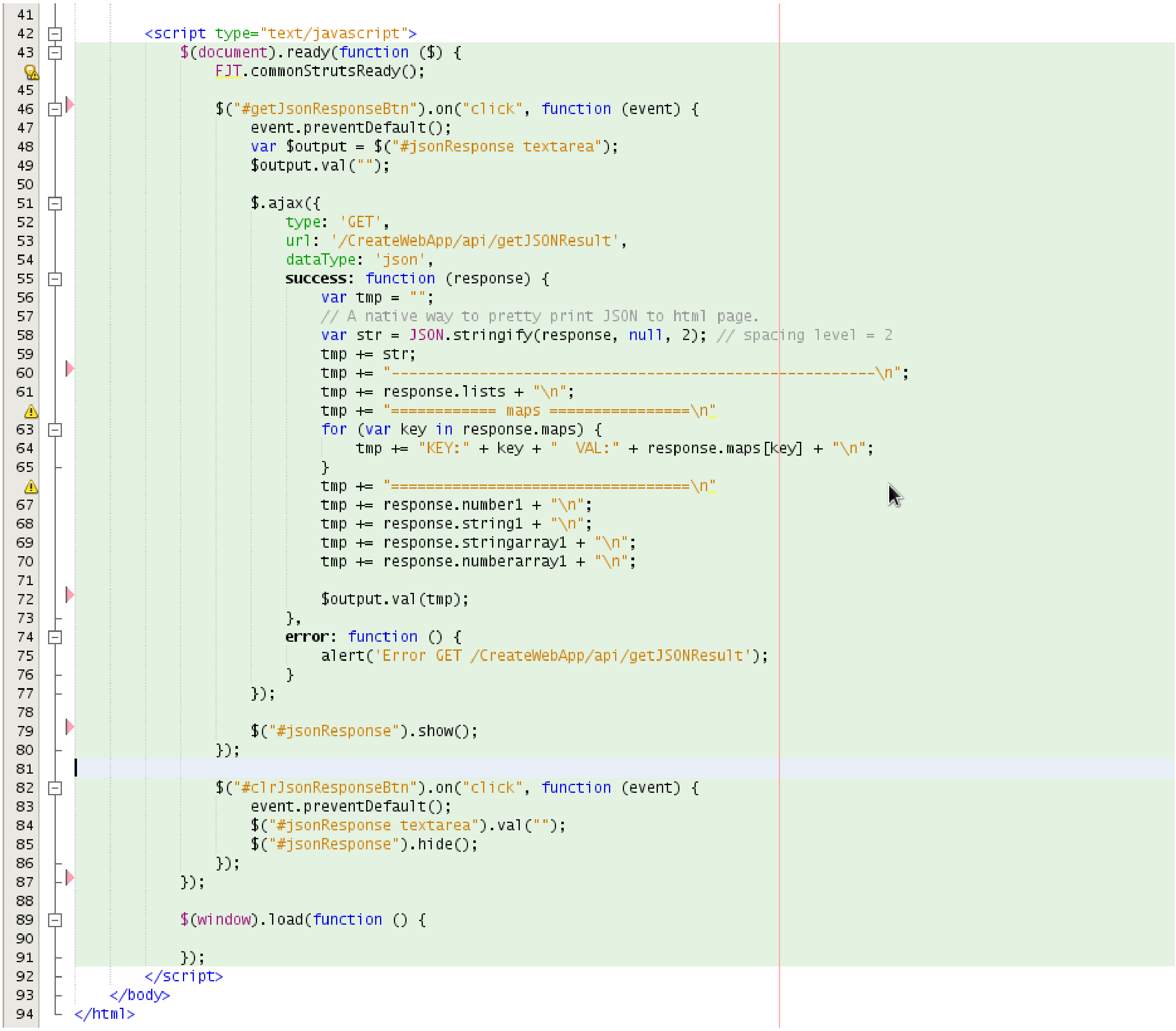
var str = JSON.stringify(newOrder, null, 2); // spacing level = 2



```
{
  "lists": [
    "list1",
    "list2",
    "list3",
    "list4",
    "list5"
  ],
  "maps": {
    "key1": "value1",
    "key2": "value2",
    "key5": "value5",
    "key3": "value3",
    "key4": "value4"
  },
  "number1": 123456789,
  "numberarray1": [
    1,
    2,
    3,
    4,
    5,
    6,
    7,
    8,
    9
  ],
  "string1": "A",
  "string2": "B",
  "stringarray1": [
    "A1",
```



```
<%--
    Document   : newTicket
    Created on : Oct 1, 2015, 8:50:55 AM
    Author     : ftrujillo
--%>

<%@taglib prefix="s" uri="/struts-tags" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <%@include file="/project_partials/head_section_1.7.2.jsp" %>
        <%--<%@include file="/project_partials/head_section_1.11.3.jsp" %>--%>
        <title>newTicket</title>
    </head>
    <body>
        <div id="myContent">
            <h2>newTicket</h2>
            <%@include file="/project_partials/action_errors.jsp" %>
            <div>
                <button id="toggleDebug">Toggle Debug</button>
            </div>

            <div>
                <a href="/CreateWebApp/api/getJSONResult">getJSONResult</a>
                <a href="/CreateWebApp/api/getJSONResult2">getJSONResult2</a>
            </div>

            <div>
                <button id="getJsonResponseBtn" class="bg_yellow">GET Json Response</button>
                <button id="clrJsonResponseBtn" class="bg_lightblue">CLEAR Json Response</button>
            </div>

            <div id="jsonResponse" style="display:none;">
                <textarea rows="30" cols="132"></textarea>
            </div>


            <%@include file="/project_partials/body_section_debug.jsp"%>
        </div>
```



```

        <script type="text/javascript">
            $(document).ready(function ($) {
                FJT.commonStrutsReady();

                $("#getJsonResponseBtn").on("click", function (event) {
                    event.preventDefault();
                    var $output = $("#jsonResponse textarea");
                    $output.val("");

                    $.ajax({
                        type: 'GET',
                        url: '/CreateWebApp/api/getJSONResult',
                        dataType: 'json',
                        success: function (response) {
                            var tmp = "";
                            // A native way to pretty print JSON to html page.
                            var str = JSON.stringify(response, null, 2); // spacing level = 2
                            tmp += str;
                            tmp += "-------------------------------------------------------\n";
                            tmp += response.lists + "\n";
                            tmp += "============ maps ================\n"
                            for (var key in response.maps) {
                                tmp += "KEY:" + key + "  VAL:" + response.maps[key] + "\n";
                            }
                            tmp += "==================================\n"
                            tmp += response.number1 + "\n";
                            tmp += response.string1 + "\n";
                            tmp += response.stringarray1 + "\n";
                            tmp += response.numberarray1 + "\n";

                            $output.val(tmp);
                        },
                        error: function () {
                            alert('Error GET /CreateWebApp/api/getJSONResult');
                        }
                    });

                    $("#jsonResponse").show();
                });

                $("#clrJsonResponseBtn").on("click", function (event) {
                    event.preventDefault();
                    $("#jsonResponse textarea").val("");
                    $("#jsonResponse").hide();
                });
            });

            $(window).load(function () {

            });
        </script>
    </body>
</html>
```