Assignment 11 (HF) - Repetition

-- no due date, no submission --

This assignment is meant to help you prepare for the exam. It is not necessary to turn in your solutions. The solutions will be discussed in the Q&A sessions during the tutorial next week.

Task 1: PHP & Sessions

a) You want to implement a “guess-my-number” game with PHP.

```
<!DOCTYPE html><html>
<head lang="en">
    <title>Guess a Number</title>
</head>
<body>
<form method="post">
    <input type="number" name="guess" />
    <input type="submit" value="Check!" />
</form>
```

Extend the code fragment from the previous page to meet these requirements:

a. Create a variable that holds a secret number of your choosing.
b. If the form was submitted, check if the user entered the correct number. If the guess was correct, inform the user and reset the game. Otherwise, inform the user that the guessed number was incorrect.
c. The game counts how often the user submits the form. Display the number of guesses!

b) How are Sessions destroyed in PHP?

c) How does PHP identify users to keep track of individual sessions?

d) How do you transparently transmit form data to the server?

Task 2: HTML5

a) Extend the following code fragment to dynamically draw a rectangle on a canvas. Its size (width and height) depend on the number the users enter into the input field.

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Stop! Rectangle-Time.</title>
</head>
<body>
<div><input type="number" placeholder="20"
onkeyup="drawRectangle(event)"/></div>
<div><canvas id="myCanvas" width="600"
height="600"></canvas></div>
<script>
    var canvas, context;
    function clearCanvas(){
        context.clearRect(0,0,
                context.canvas.width,
                context.canvas.height);
    }




    function drawRectangle(e){




    }

</script>
</body>
</html>
```

b) Extend the following code fragment to enable drag and drop through HTML5 and JavaScript. After dropping the div (‘#dragme’) anywhere on the screen, it should stick where it is. You do not have to consider the offset between the mouse cursor and the top-left corner of the div.

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Drag and Drop</title>
    <style>
        #dragme{
            width: 200px; height: 200px;
            background-color: red;
            position: absolute;
            top: 0; left: 0;
        }
    </style>
</head>
<body>

<div id="dragme"
                                                       >
    DragMe!
</div>

<script>

    function handler(e){
        var target = e.toElement;











    }

</script>
</body>
</html>
```

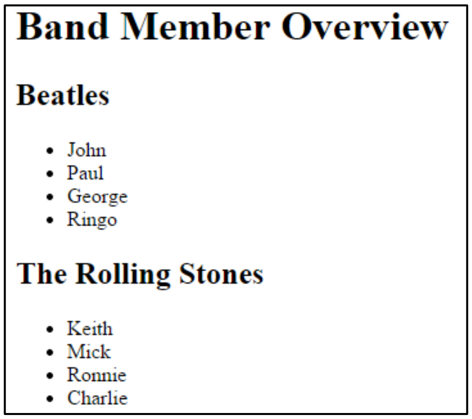
Task 3: jQuery & AJAX

```
var http = require('http');
http.createServer(function (req, res) {

    var bands = [
        {   name : 'Beatles',
            members : ['John', 'Paul', 'George', 'Ringo']
        },
        {   name : 'The Rolling Stones',
            members : ['Keith', 'Mick', 'Ronnie', 'Charlie']
        }
    ];

    var headers = {};
    headers["Access-Control-Allow-Origin"] = "*";
    headers["Access-Control-Allow-Methods"] = "POST, GET";
    headers["Content-Type"] = 'application/json';
    res.writeHead(200, headers);
    res.end(JSON.stringify({bands : bands}));
}).listen(1337, '127.0.0.1');
```

a) You want to retrieve a list of bands from a NodeJS back end in JSON format (just like the one in the code fragment above.



Band Member Overview

Beatles

- John
- Paul
- George
- Ringo

The Rolling Stones

- Keith
- Mick
- Ronnie
- Charlie

   a. Complete the function getBands(successCallback). It should request the information from the server and pass it to the successCallback, once it received a response.

   b. Complete the function bandDataAvailable(data).

      i. data is the response from the NodeJS back end.

      ii. Generate a separate <div / > for each band.

      iii. Each div should have a heading, that tells the band's name

      iv. Each div should include an un-numerated list of the band's members.

   c. The script loads in the <head>, so potentially the DOM is not yet complete, when the script is executed. Use a jQuery callback to make sure that the script is executed when the DOM is ready.

Put your code into the fragment on the next page!

b) Explain the advantages and disadvantages of AJAX!

c) Why does the NodeJS app respond with "`Access-Control-…`" headers?

```
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <title>Bands</title>
    <script src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
    <script>


            function getBands(successCallback){



            }

            function bandDataAvailable(data){















            }

            getBands(bandDataAvailable);



    </script>
</head>
<body>
<div id="container"> <h1>Band Member Overview</h1>  </div>
</body></html>
```

Task 4: NodeJS

a) The following code is a NodeJS implementation of a back end that implement the game from Task 1 a) asynchronously.

```
1   var express = require('express');
2   var bodyParser = require('body-parser');
3
4   var app = express();
5
6   var secretNumber = 42;
7
8   app.use(bodyParser.json());
9   app.use(bodyParser.urlencoded({extended: true}));
10
11  app.post('/',function(req,res){
12     var response = {};
13     if(req.body && req.body.guess){
14             response.correct = req.body.guess == secretNumber;
15     }
16     res.json(response);
17  });
18
19  app.listen(1337);
```

Explain the following blocks of code (by line numbers) in your own words:

a. Lines 1-2

b. Lines 8-9

c. Lines 11-16

b) What does a middleware do in NodeJS?

Task 5: Multimedia Content Description

a) In the lecture you have come to know the Resource Description Framework (RDF). It is used to represent information about resources in the web. Give three exemplary resources that can be described with the RDF.

b) One of the core components of RDF is the *Statement*. Which components are included in a statement? Explain each component!

c) Draw an RDF Graph for the following RDF Document:

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
    <contact:Person
            rdf:about="http://www.w3.org/People/EM/contact#me">
        <contact:fullName>Eric Miller</contact:fullName>
        <contact:mailbox rdf:resource="mailto:em@w3.org"/>
        <contact:personalTitle>Dr.</contact:personalTitle>
    </contact:Person>
</rdf:RDF>
```