Abschlusspräsentation

Einführung in das Web Engineering

von Lennart Nixdorf

Themen

- **HTML / CSS**
- RWD
- Javascript
- DOM
- **Async Javascript**
- Modulares Web
- SVG
- WebApps
- Vue.js / Router
- MEVN
- **PHP**
- **React**

HTML /CSS

HTML:

- Ermöglicht es statische Websites zu erstellen
- Dokumentenbasiert
- Simple Sprache zum formatieren des Textes

CSS:

- Gestaltung und Design durch CSS

HTML /CSS

```html
<html>

  <head>
    <link rel="stylesheet" href="styles.css">
    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=1.0, maximum-scale=1.0">

    <title> Web Engineering </title>
  </head>

  <body>
    <ul class="sidenav">
                Übungen
                <br>
                <li>
                        <a href="./U02/u.html">HTML / CSS</a>
                </li>
                <li>
                        <a href="./U03/u.html">RWD</a>
                </li>
                <li>
                        <a href="./U04/u.html">Javascript</a>
                </li>
                <li>
                        <a href="./U05/u.html">DOM</a>
                </li>
                <li>
                        <a href="./U06/u.html">Async</a>
                </li>
                <li>
                        <a href="./U07/u.html">Modulares Web</a>
                </li>
                <li>
                        <a href="./U08/u.html">SVG</a>
                </li>
                <li>
                        <a href="./U09/u.html">WebApps</a>
                </li>
                <li>
                        <a href="./U10/u.html">Vue</a>
                </li>
                <li>
                        <a href="">MEVN</a>
                </li>
                <li>
                        <a href="./U12/index.php">PHP</a>
                </li>
                <li>
                        <a href="./U13/build/index.html">React</a>
                </li>
        </ul>
        <div class="mainpage">
                <h1>Einführung in das Web Engineering</h1>
                <br>
                Präsentation:
                <br>
                <iframe src="./test.pdf" width="90%" height="90%"></iframe>
        </div>
        <div class="sidecontent">
                Anmerkung und Links:
                <br><br>
                <a href="https://github.com/lnixdo2s/Web-Engineering.git">Github</a>
                <br>

        </div>
  </body>

</html>
```

```css
@import url('https://fonts.googleapis.com/css2?family=Baumans&display=swap');

body {
  font-family: 'Baumans';
  background-image: url(https://media.giphy.com/media/RkDZq0dhhYHhxdFrJB/giphy.gif);
  background-size: cover;
  background-position-y: 30%;
  display: flex;
  flex-direction: row;
  justify-content: flex-start;
  color: aliceblue;
}

.mainpage {
  width: 66%;
  margin-left: 15px;
}

.sidecontent {
  width: 24%;
  margin-left: 5px;
  margin-top: 5%;
}
.sidecontent a {
  color: aliceblue;
}

.sidenav {
  list-style-type: none;
  height: auto;
  display: flex;
  flex-direction: column;
  align-items: flex-start;
  margin-left: 0px;
  margin-top: 5%;
  width: 10%;
}
.sidenav li {
  border: 3px solid aliceblue;
  border-radius: 10px;
  padding: 5px;
  margin: 2px 0px 2px 0px;
  width: 90%;
}
.sidenav a {
  text-decoration: none;
  color: aliceblue;
}
```

Javascript & Async Javascript

Javascript:

- Ermöglicht clientseitig dynamische Websites

Async:

- Ausführung im Hintergrund durch Async-Funktionen
- Promise API
- Async und Await Keywords

Javascript & Async Javascript

```
function fun1() {
  Promise.all([
    fetch("http://localhost:8080/Desktop/U06/A.txt"),
    fetch("http://localhost:8080/Desktop/U06/B.txt")
  ])

  .then(responses => Promise.all(responses.map(r => r.text())))
  .then(texts => {
    document.getElementById('A').innerText = texts[0];
    document.getElementById('B').innerText = texts[1];
    let concat = concatTexts(texts[0], texts[1]);
    document.getElementById('concat').innerHTML = concat;
  });
}
```

```
async function loadData() {
    const response = await fetch("data.json");
    const data = await response.json();
    return data;
}
```

React

- Ermöglicht die Erstellung von Web-Komponenten
- Vergleichbar mit Vue.js
- State-Basierend

React

```
import React from 'react';
import './Login.css';
import people from './people.json';

class Login extends React.Component {
    constructor(props) {
        super(props);
        this.state = {value: ''};
        this.handleChangeName = this.handleChangeName.bind(this);
        this.handleChangePassword = this.handleChangePassword.bind(this);
        this.handleSubmit = this.handleSubmit.bind(this);
    }

    handleChangeName(event) {
        this.setState({name: event.target.value});
    }
    handleChangePassword(event) {
        this.setState({password: event.target.value});
    }
    handleSubmit(event) {
        let check = false;
        people.forEach(person => {
            if(this.state.name === person.name && this.state.password === person.password) {
                /* Send them away! */
                this.props.onLogin();
                check = true;
            }
        });
        if(!check) {
            alert('Password was wrong! Try: "' + people[0].name + '" and "' + people[0].password + '" !');
        }
    }

        render() {
                return (
                        <div className="login">
                <h1> {this.props.name}</h1>
                <div className="form">
                    <label>Nutzername </label>
                    <input type="text" name="Nutzername" value={this.state.name} onChange={this.handleChangeName}></input>
                    <label>Passwort </label>
                    <input type="password" name="Passwort" value={this.state.password} onChange={this.handleChangePassword}></input>
                    <button name="Login" onClick={this.handleSubmit}>Login</button>
                </div>
            </div>
        );
    }
}
export default Login;
```

PHP

- Im Gegensatz zu Javascript ist PHP Serverseitig
- Anwendung z.B. für Logins und andere Serverseitigen Aufgaben
- Ermöglicht das erstellen von Sessions, welche Information seitenübergreifen speichern können

PHP

```php
<?php
    session_start();
    $userfile = "users.json";
    $arr_data = array();

    try
    {

        $n = $p = "";
        if($_SERVER["REQUEST_METHOD"] == "POST") {
            $n = test_input($_POST["name"]);
            $p = test_input($_POST["password"]);
        }
        $nuser = array(
            'name'=> $n,
            'password'=> $p
            );

        $jsondata = file_get_contents($userfile);
        $arr_data = json_decode($jsondata, true);
        if($arr_data == null) {
            $arr_data = array();
        }
        foreach($arr_data as $value) {
            if(strcmp($value['name'], $nuser['name']) == 0) {
                if(password_verify($nuser['password'], $value['password'])) {
                    $_SESSION["logged_in"] = true;
                    $_SESSION["username"] = $nuser["name"];
                    echo '<script type="text/javascript" language="Javascript">
                    alert("Erfolgreich eingeloggt!");
                    window.location.replace("index.php");
                    </script>';
                } else {
                    echo '<script type="text/javascript" language="Javascript">
                    alert("Passwort ist nicht korrekt!");
                    window.location.replace("index.php");
                    </script>';
                }
                die();
            }
        }
        echo '<script type="text/javascript" language="Javascript">
            alert("Nutzer nicht registriert!");
            window.location.replace("index.php");
            </script>';
    }
    catch (Exception $e) {
        echo 'Caught exception: ',  $e->getMessage(), "\n";
    }
    function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }
?>
```

ENDE

Github-Rep zur Lehrveranstaltung