

# WebApp

## Group 46

Group 46

StartseiteVerwaltung

Verwaltung: Übersicht

Eine einfache Web Applikation zum verwalten und Überwachen von Smart Meters

Zurück zur Startseite

Übersicht

Gerätetyp	Geräteerkennung	Optionen
	AB12345678	<a href="#">Zur Detailansicht</a>

# HTML

- Insgesamt 5 Seiten
  - Startseite index.html
  - Verwaltung (Übersicht) verwalten.html
  - 3 Detailansichten detail1.html, detail2.html, detail.html
- Navigation
  - Zwischen den Seiten mit Hilfe von Links frei hin und her gewechselt werden.

# Header (menu)

Das <div> Element, das  
den Header markiert

Unsortierte Liste der  
Menu-Tabs

Die Zwei Menu-Tabs

```
<div class="headerBg">
  <div class="menuBg">
    <div class="menu">
      <div class="logoFont">
        Group 46
      </div>
      <ul>
        <li><a href="./verwalten.html">Verwaltung</a></li>
        <li><a class="activ" href="#">Startseite</a></li>
      </ul>
    </div>
  </div>
</div>
```

# Header (heading)

Das „style“ Attribut zum  
Stylen direkt in der HTML

Der Link zur  
Verwaltung <a>

```
<div class="heading">
  <p style="font-size:44px">Anwendungssysteme Übung 1 SS 2017</p>
  <div class="subHeading">
    <div class="subHLeft">
      Eine einfache Web Applikation zum verwalten und überwachen von Smart Meters
    </div>
    <div class="subHRight">
      <a class="button" href="./verwalten.html">Zur Verwaltung</a>
    </div>
  </div>
</div>
```

# Footer

Fette Hervorhebung  
von „Group 46“ **<b>**

Der Link, um die  
Seite neuzuladen  
(Attribut href=„#“)

```
<div class="footerBg">
  <div class="footer">
    <div class="copyright">
      © Copyright 2017 by <b class="footerFont">Group 46</b>
    </div>
    <div class="toTop">
      <a class="ttop" href="#">Go to top</a>
    </div>
  </div>
</div>
```

# Startseite (index.html) [1]

`<center>` Element zentriert alle Inhalte, die zwischen den Tags stehen (not supported in HTML 5)

Tabelle mit zwei Zeilen `<tr>` und zwei Spalten `<td>` zum ordentlichen Darstellen der Text/Bild-Inhalte

Das `<img>` Element mit Attribut `src`, dass den Pfad zum Bild angibt

```
<div class="main">
  <div class="content">
    <center><h1>POWERFUL TOOLS TO HELP YOU MANAGE YOUR INTERESTS</h1></center>
    <table class="landing">
      <tr>
        <td>
          </img>
        </td>
        <td class="landing">
          Ihre Webapplikation soll sowohl über eine Gesamtübersicht aller eingetragenen Smart
          Meter, als auch individuelle Detailsichten für jeden Smart Meter verfügen. (...)
        </td>
      </tr>
      <tr>
        <td>
          </img>
        </td>
        <td class="landing">
          Beim Ablesen trägt der Nutzer Ihrer Webapplikation dabei seine Nutzerkennung und einen
          Verbrauchswert (in kWh) in ein Formular ein. (...)
        </td>
      </tr>
    </table>
  </div>
</div>
```

# Startseite (index.html) [2]

<p> Element mit mehreren <b> Elementen, die durch Zeilenumbrüche <br/> getrennt sind.

Dient zum Darstellen der einzelnen Bearbeitungsschritte.

```
<br />
<h2>Vorgehensweise</h2>
<hr />
<p>
  <b>Schritt 1</b><br />
  (...)
  <br /><br />
  <b>Schritt 2</b><br />
  (...)
  <br /><br />
  <b>Schritt 3</b><br />
  (...)
  <br /><br />
  <b>Schritt 4</b><br />
  (...)
  <br /><br />
  <b>Schritt 5</b><br />
  (...)
  <br /><br />
  <b>Schritt 6</b><br />
  (...)
</p>
</div>
</div>
```

# Verwaltung (verwalten.html)

Die Tabelle stellt die Übersicht aller Messgeräte dar.

Die drei `<th>` Elemente sind die header Zellen. Hier stehen die Spaltennamen: Gerätetyp, Gerätekenung, Optionen

Die drei `<td>` Elemente sind die Datenzellen. Sie können mit allen Elementen befüllt werden. Hier stehen die entsprechenden Inhalte: `<img>`, text, `<a>`.

Die Klasse „overview“ dient zum Stylen mit Hilfe von CSS. Auf die kann sowohl direkt als auch über Elemente zugegriffen werden. So kann man verschiedene Styles auf verschiedene Elemente mit derselben Klasse anwenden.

```
<div class="main">
  <div class="content">
    <h2>Übersicht</h2>
    <hr />
    <table class="overview">
      <tr>
        <th>Gerätetyp</th>
        <th>Gerätekenung</th>
        <th>Optionen</th>
      </tr>
      <tr class="overview">
        <td></img></td>
        <td>AB12345678</td>
        <td><a class="detailButton" href="./detail1.html">Zur Detailansicht</a></td>
      </tr>
      <tr class="overview">
        <td></img></td>
        <td>AB32942628</td>
        <td><a class="detailButton" href="./detail2.html">Zur Detailansicht</a></td>
      </tr>
      <tr class="overview">
        <td></img></td>
        <td>AB78988293</td>
        <td><a class="detailButton" href="./detail3.html">Zur Detailansicht</a></td>
      </tr>
    </table>
  </div>
</div>
```



# Detailansicht (detail1.html) [1]

Hier ist ein Tabellenelement in ein anderes verschachtelt im `<td>` Element. So kann das Bild links von der Tabelle positioniert werden.

Die Datenzellen `<td>` werden mit einer `id` ausgezeichnet, damit sie später mit Hilfe von JQuery ausgewählt und dynamisch mit Inhalt (Zahlen) befüllt werden können

Die `<div>`, die die Warnung (`<h3>`, `<p>`) enthält kriegt auch eine `id` zugewiesen, um später dynamisch angezeigt zu werden.

```
<div class="main">
  <div class="content">
    <table class="detail">
      <tr class="detail">
        <td></td>
        <td>
          <h2>Gerätename: AB12345678 </h2>
          <table class="status">
            <tr>
              <td>Maximale Belastung:</td>
              <td><p id="maxstrom">80</p></td>
            </tr>
            <tr>
              <td>Spannung:</td>
              <td><p id="spannung"></p></td>
            </tr>
            <tr>
              <td>Stromstärke:</td>
              <td><p id="strom"></p></td>
            </tr>
            <tr>
              <td>
                <div id="warnung">
                  <h3>WARNUNG!</h3>
                  <p>Die Stromstärke liegt oberhalb der zulässigen Maximalbelastung</p>
                </div>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </div>
</div>
```

# Detailansicht (detail1.html) [2]

Dieses Formelement <form> soll von dem Nutzer ausgefüllt und abgeschickt werden können

Die <input> Elemente können vom Nutzer z.B. mit Text (wird mit dem Attribut „type“ festgelegt) befüllt werden.

Das <button> Element kann vom Nutzer geklickt werden. Die zugewiesene Id dient zum Eventhandling durch JQuery.

Die Tabelle kann mit Elementen (Spalten, Zeilen) dynamisch befüllt werden. Der Zugriff erfolgt, wie davor über das Id-Attribut.

```
<br />
<br />
<h2>Ablesen</h2>
<hr />
<form>
  Nutzererkennung:
  <br />
  <input type="text" name="nutzerkennung" maxlength="4"><br>
  Verbrauchswert (in kWh):
  <br />
  <input type="text" name="verbrauchswert">
  <br />
  <br />
  <button class="inButton" type="button" id="btnAblesen">Ablesen</button>
</form>
<br />
<br />
<h2>Alle Ablesungen</h2>
<hr />
<table class="event" id="tableAblesen">
  <tr>
    <th class="event">Nutzerkennung</th>
    <th class="event">Verbrauchswert </th>
    <th class="event">Datum</th>
  </tr>
</table>
</div>
</div>
```

JavaScript

# Deklaration/Initialisierung von Variablen

Die Ober- und  
Untergrenze für die  
Spannung

Das Objekt Date wird mit  
Hilfe des Schlüsselwortes  
„new“ instanziiert.

```
var minNumber = 220;  
var maxNumber = 240;  
  
var datum = new Date();  
var heute = datum.getDate() + "." + (datum.getMonth() + 1) + "." + datum.getFullYear();  
  
var spannung = randomNumberFromRange(minNumber, maxNumber);  
var strom = randomNumberFromRange(0, parseInt(document.getElementById("maxstrom").innerHTML) + 5);
```

Das heutige Datum im  
passenden Format in der  
Variable „heute“  
gespeichert.

Strom und Spannung werden mit Hilfe der Funktion  
randomNumberFromRange (minNumber, maxNumber) und in  
den entsprechenden Variablen abgespeichert.

# Funktionen

Die Funktion liefert eine randomisierte Zahl im vorgegebenen Intervall mit der Genauigkeit: 0.1.

```
function randomNumberFromRange(min,max) {  
    return (Math.random()*(max-min+1)+min).toFixed(1);  
}  
  
function validateInput(val) {  
    return /[A-Z]{2}\d{2}/i.test(val);  
}
```

Die toFixed() Methode konvertiert eine number in einen string, wobei der übergebene Parameter die Anzahl der zu bleiben Nachkommastellen angibt.

Die Funktion testet den übergebenen Wert auf die Übereinstimmung mit dem Pattern und gibt einen booleschen Wert zurück.

Die test() Methode testet auf eine Übereinstimmung in einer Zeichenketten.

# JQuery [1]

Die Elemente werden mit den randomisierten Werten befüllt

Der Ausdruck „Stromwert > Maximalbelastung“ wird ausgewertet und falls wahr, wird die <div> mit der Warnung angezeigt

Wenn der Nutzer auf „Ablese“ drückt, wird, wenn das Format stimmt, der Ablesetabelle eine Zeile mit drei Spaltwerten hinzugefügt, sonst wird eine Fehlermeldung angezeigt.

Der Zugriff erfolgt über das Id-Attribut des Elements

Hier wird auf das CSS Attribut „display“ zugegriffen und von „none“ auf „block“ geändert

```
$( document ).ready(function() {  
    $("#spannung").html(spannung + "V");  
    $("#strom").html(strom + "A");  
  
    if(strom > parseInt(document.getElementById("maxstrom").innerHTML)){  
        $("#warnung").css("display", "block");  
    }  
  
    $("#btnAblese").click(function(){  
        var val = $("input[name=nutzerkennung]").val();  
        if (validateInput(val)) {  
            $('#tableAblese').append('<tr><td>' + $("input[name=nutzerkennung]").val() + '</td><td>' +  
                $("input[name=verbrauchswert]").val() + '</td><td>' + heute + '</td></tr>');  
        } else {  
            $("#btnAblese").after('<p id="fehlermeldung">' + "Falsche Nutzerkennung! Die Nutzerkennung muss"  
                + " aus den Initialen Ihres Namens und den letzten beiden Ziffern ihrer Matrikelnummer bestehen" + '</p>');  
        }  
    });  
});
```



# JQuery [2]

Dem Element wird ein event listener click() hinzugefügt.

Dem event listener wird eine anonyme Callback-Funktion übergeben

Der Wert des Inputfeldes wird ausgelesen und in der Variable „val“ gespeichert

Der Tabelle wird mit append() direkt ein <tr> Element mit zwei <td> Elementen hinzugefügt

Sonst wird nach dem Knopf mit after() ein <p> Element mit der Warnung hinzugefügt.

```
$("#btnAblese").click(function(){  
  var val = $("input[name=nutzerkennung]").val();  
  if (validateInput(val)) {  
    $('#tableAblese').append('<tr><td>' + $("input[name=nutzerkennung]").val() + '</td><td>' +  
      $("input[name=verbrauchswert]").val() + '</td><td>' + heute + '</td></tr>');  
  } else {  
    $("#btnAblese").after('<p id="fehlermeldung">' + "Falsche Nutzerkennung! Die Nutzerkennung muss"  
      + " aus den Initialen Ihres Namens und den letzten beiden Ziffern ihrer Matrikelnummer bestehen" + '</p>');  
  }  
});
```