



WebApp – Übung 1


Eine einfache Web Applikation zum verwalten und überwachen von Smart Meters

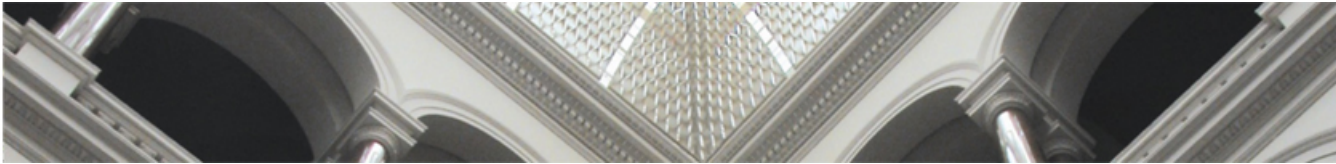
Verwaltung: Übersicht

Eine einfache Web Applikation zum verwalten und überwachen von Smart Meters

[Zurück zur Startseite](#)

Übersicht

Gerätetyp	Geräteerkennung	Optionen
	AB12345678	Zur Detailansicht



Content

Vorwort

HTML & CSS

Head

Body

Navigation Menu

Header

Footer

Startseite (index.html)

Verwaltung (verwalten.html)

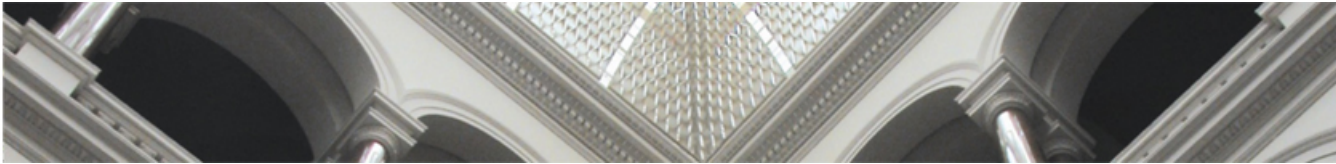
Detailansicht (detail1.html)

JavaScript

Deklaration/Initialisierung von Variablen

Funktionen

jQuery



Vorwort

- Insgesamt 5 Seiten
 - Startseite index.html
 - Verwaltung (Übersicht) verwalten.html
 - 3 Detailansichten detail1.html, detail2.html, detail.html
 - Entwickelt und getestet mit Google Chrome 58+
- Navigation
 - Zwischen den Seiten kann mit Hilfe der Navigationsleiste, sowie mit einem Button im Überschriftenbereich gewechselt werden
 - Ein zurück nach oben Button befindet sich immer am Ende einer Seite
- Cascading Style Sheets
 - Alle Div Elemente, sowie viele weitere Elemente, bei denen ein class Attribut vorkommt, wurden mithilfe von CSS gestaltet
 - Es werden nur gekürzte Auszüge präsentiert auf Grund des Umfangs
- script.js & style.css
 - JavaScript ist ganz ausgelagert und CSS größtenteils



HTML & CSS

Hypertext Markup Language & Cascading Style Sheets

Head



Body

`<body>`

HTML Bodyist in 4 Bereiche aufgeteilt:

```
<div class="menuBg">...</div>
```

`<div class="menuBg">`
Navigation Menu

```
<div class="heading">...</div>
```

`<div class="heading">`
Header

```
<div class="main">...</div>
```

`<div class="main">`
Content Box

```
<div class="footerBg">...</div>
```

`<div class="footerBg">`
Footer

`</body>`

Hinweis:

„menuBG“, „heading“, „footerBg“, sowie deren Kindklassen sind auf allen Seiten ähnlich aufgebaut

Body

Group 46
Startseite
Verwaltung

Verwaltung: Übersicht

Eine einfache Web Applikation zum verwalten und Überwachen von Smart Metern

Zurück zur Startseite

Übersicht

Gerätetyp	Gerätekennung	Optionen
	AB12345678	Zur Detailsicht
	AB32942628	Zur Detailsicht

Main Box

© Copyright 2017 by Group 46
Go to top

Box

Navigation Menu

Das Element, das die Navigationsleiste markiert

Unsortierte Liste der Menu-Tabs

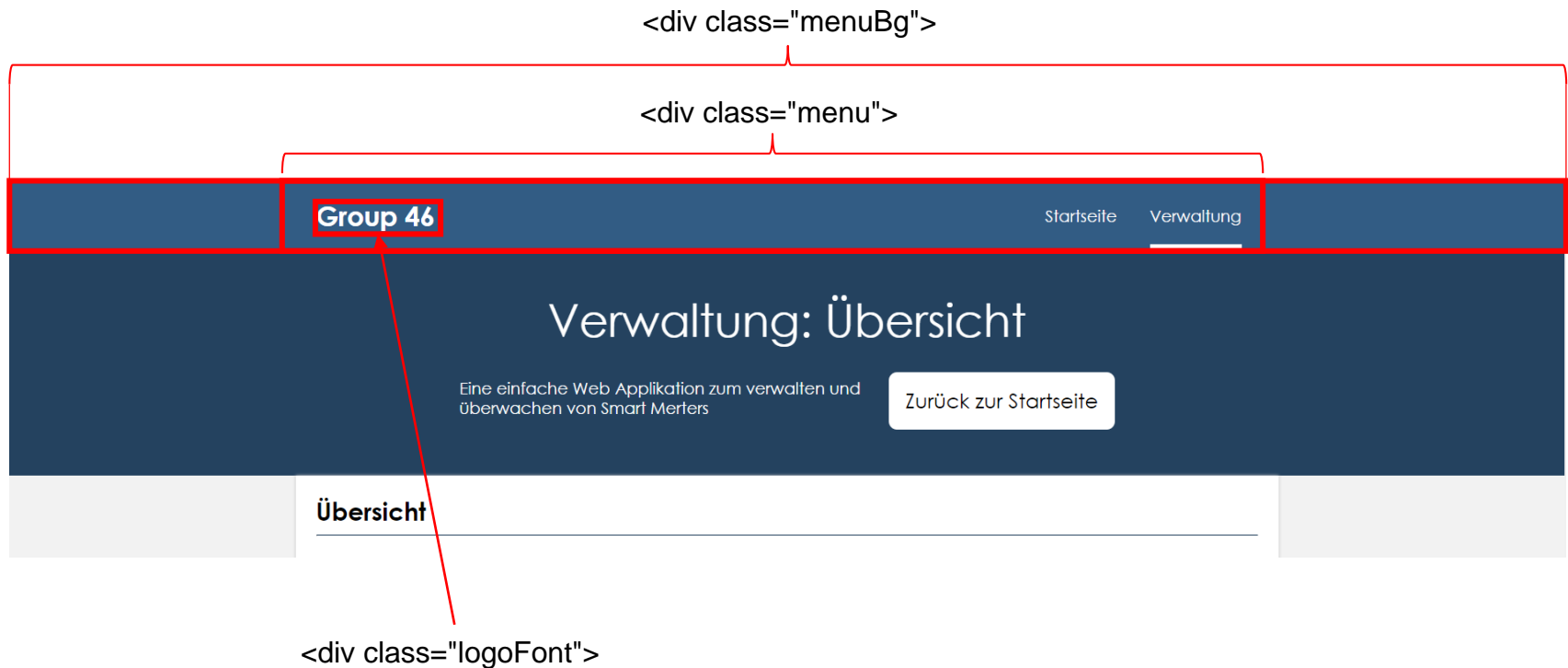
Die Zwei Menu-Tabs

```

<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>

```

Navigation Menu



Navigation Menu (CSS)

(menu) Liste
rechtbündig, statt
untereinander

```
div.menu ul li {
    float: right;
}
```

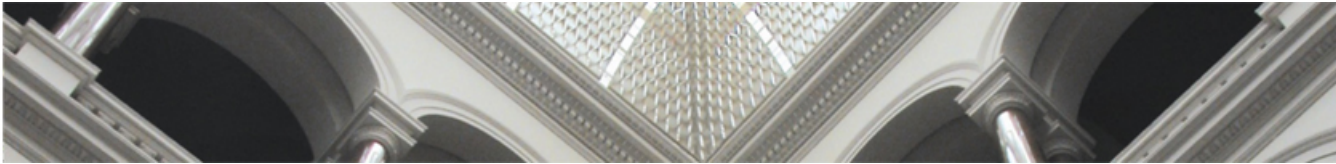
Innenabstand
nach oben, unten

Außenabstand
nach links, rechts

```
div.menu ul li a {
    padding-top: 6px;
    padding-bottom: 18px;
    margin-left: 16px;
    margin-right: 16px;
    color: white;
    text-decoration: none;
}
```

```
.menuBg {
    background-color: #325C84;
    position: fixed;
    min-width: 100%;
    color: white;
}
```

.menuBg ist fixiert, dh.
wird immer angezeigt



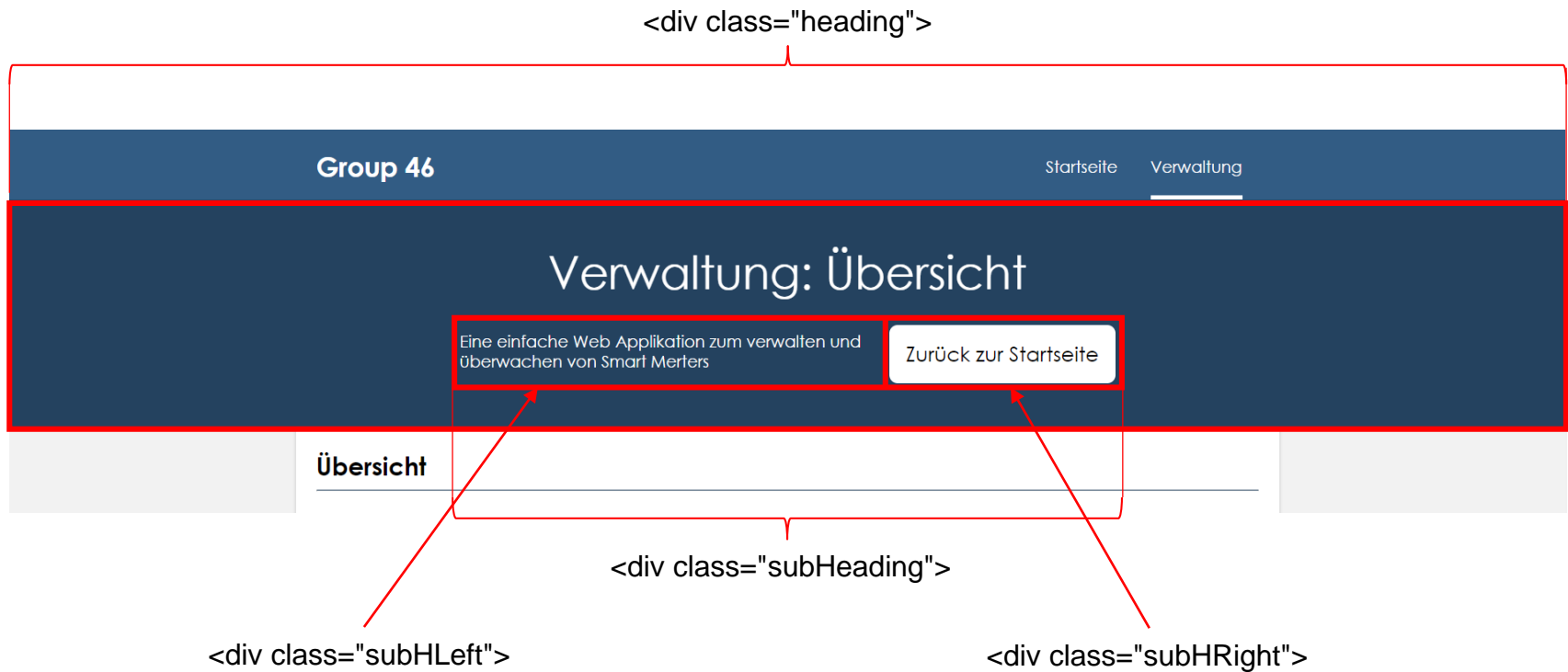
Header

```
<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>
```

Das „style“ Attribut zum
Stylen direkt in der HTML
(Schriftgröße: 44px)

Der Link zur
Verwaltung

Header



Header [CSS]

```
.heading {  
  text-align: center;  
  background-color: #24425F;  
}  
  
.subHeading {  
  margin: auto;  
  max-width: 640px;  
}  
  
.subHLeft {  
  max-width: 400px;  
  text-align: left;  
  float: left;  
}  
  
.subHRight {  
  text-align: right;  
}
```

Text Zentriert

Hintergrundfarbe

„margin: auto“ zentriert das Element

Maximale Breite von 400 px

„linksbündig“, damit .subHRight daneben gesetzt werden kann

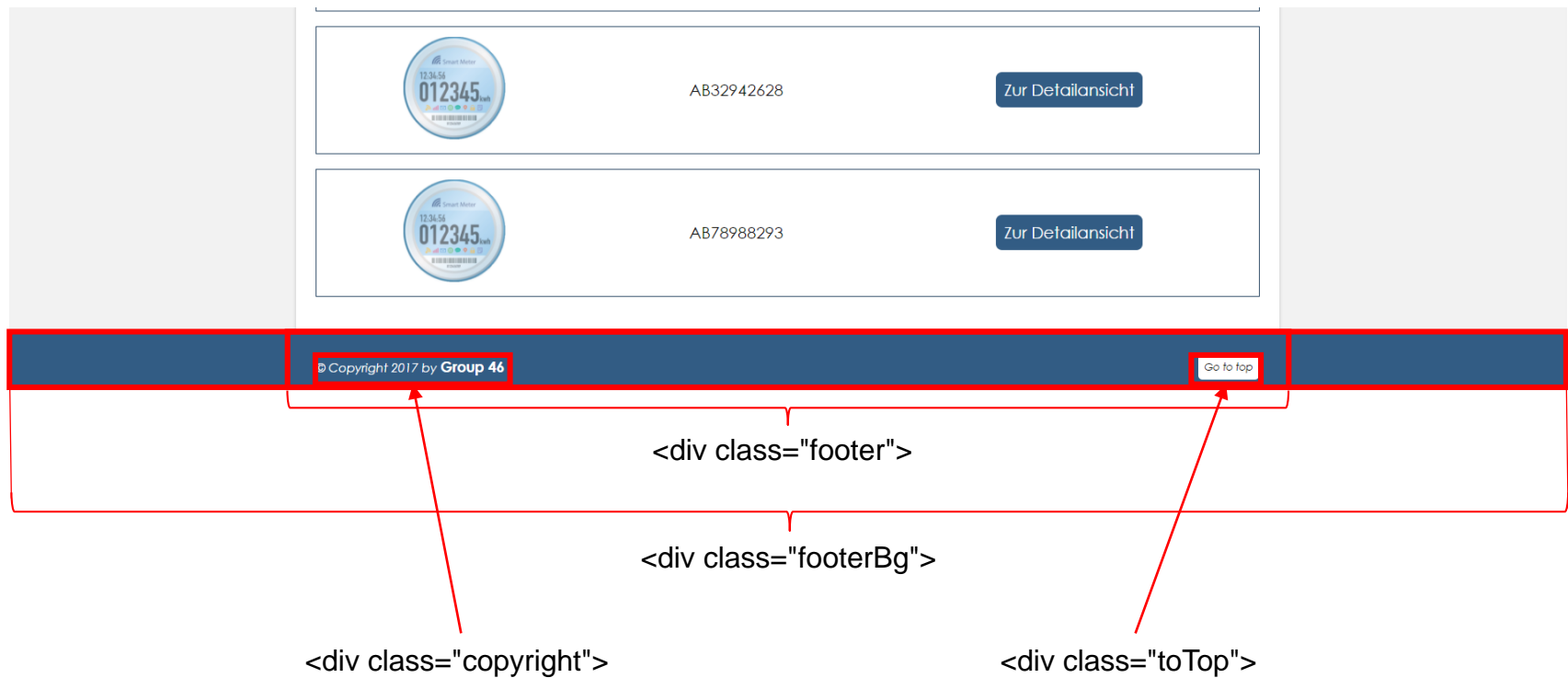
Footer

Fette Hervorhebung
(****) von „Group 46“

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>
```


Footer



Footer [CSS]

```
.footerBg {  
  background-color: #325C84;  
  min-width: 100%;  
  box-shadow: 0px 0px 4px 0px #325C84;  
}
```

4px breiter Schatten mit
der HEX-Farbe #325C84

Schriftfarbe

Kursive Schrift

Schriftgröße 12

```
.footer {  
  color: white;  
  font-style: italic;  
  font-size: 12px;  
}
```

```
.footerFont {  
  font-style: normal;  
  font-size: 14px;  
}
```

Überschreiben, der von
.footer festgelegten
Eigenschaften

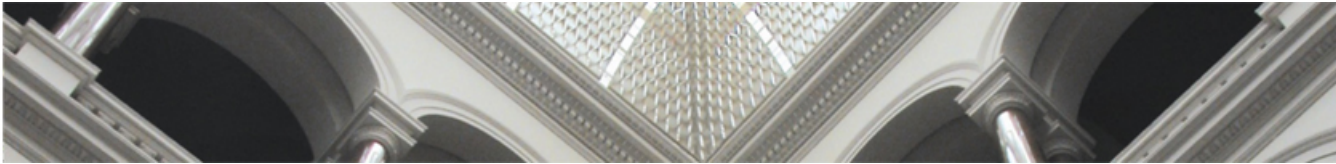
Startseite (index.html) [1]

Element zentriert alle Inhalte,
die zwischen den Tags stehen

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

Das 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>
          
        </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>
          
        </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]

`<h2>` Element als Überschrift → `<h2>Vorgehensweise</h2>`

`<hr>` Element als Trennstrich → `<hr />`

`` Element als fette Hervorhebung → `Schritt 1
`
(...)
`

`

`
` als Zeilenumbruch → `Schritt 2
`
(...)
`

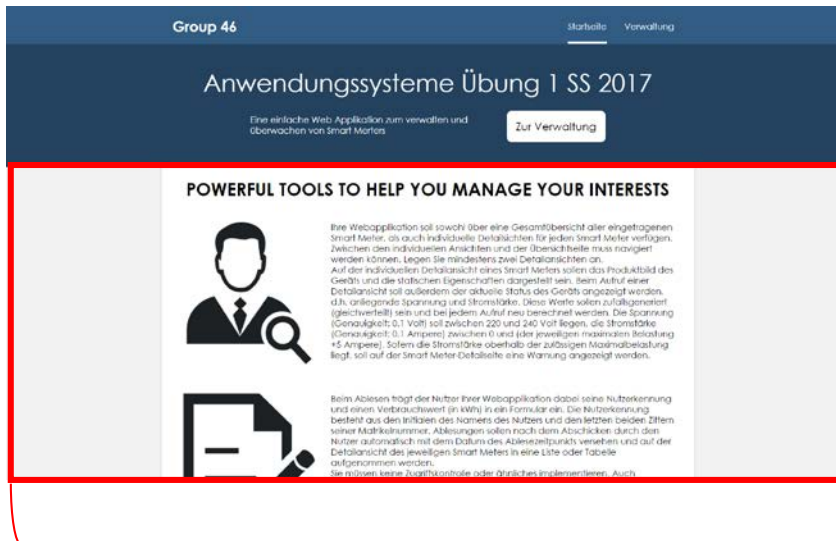
`
`Schritt 3
`
(...)
`

`
`Schritt 4
`
(...)
`

`
`Schritt 5
`
(...)
`

`
`Schritt 6
`
(...)
`</div>`
`</div>`

Startseite (index.html)

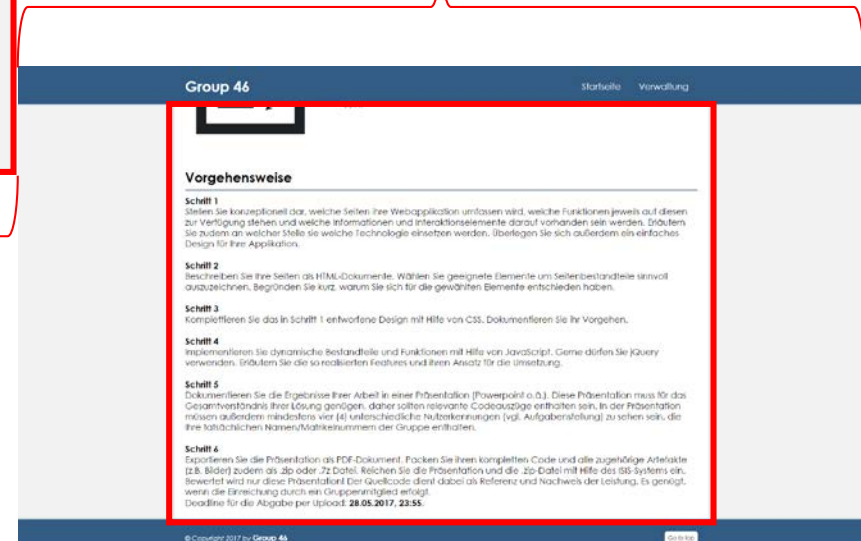


`<div class="main">`

Aufgabenstellung

Vorgehensweise

`<div class="content">`



Startseite [CSS]

```
.main {  
  background: #f2f2f2;  
}  
  
.content {  
  background: white;  
  margin: auto;  
  max-width: 920px;  
  color: black;  
  box-shadow: 0px 0px 4px 0px silver;  
}
```

Hintergrundfarbe

Zentrieren des Elements

Maximale Breite des Elements

„Silberner“ Schatten

Verwaltung (verwalten.html)

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

Die Tabelle stellt die Übersicht aller Messgeräte dar.

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

```
<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></td>
        <td>AB12345678</td>
        <td><a class="detailButton" href="./detail1.html">Zur Detailansicht</a></td>
      </tr>
      <tr class="overview">
        <td></td>
        <td>AB32942628</td>
        <td><a class="detailButton" href="./detail2.html">Zur Detailansicht</a></td>
      </tr>
      <tr class="overview">
        <td></td>
        <td>AB78988293</td>
        <td><a class="detailButton" href="./detail3.html">Zur Detailansicht</a></td>
      </tr>
    </table>
  </div>
</div>
```





Verwaltung (verwalten.html)

Group 46
Startseite
Verwaltung

Verwaltung: Übersicht

Eine einfache Web Applikation zum verwalten und überwachen von Smart Meters
Zurück zur Startseite

Übersicht

Gerätetyp	Gerätekennung	Optionen
	AB12345678	Zur Detailansicht
	AB32942628	Zur Detailansicht
	AB78988293	Zur Detailansicht

<table class=„overview">

Verwaltung (verwalten.html) [CSS]

```
table.overview {  
    border-spacing: 0px 16px;  
    width: 100%;  
}
```

Minimaler Abstand der
einzelnen Tabellenzeilen,
Tabellenspalten: 0px nach oben,
unten; 16px nach links, rechts

```
tr.overview {  
    outline: thin solid #24425F;  
}
```

Tabellenzeile dünner
Umrandung

```
a.detailButton {  
    background: #325C84;  
    padding: 8px;  
    border-radius: 6px;  
    color: white;  
    text-decoration: none;  
}
```

Abrunden des Außenrandes

Entfernen der Unterstreichung
bei <a> Elementen

Detailansicht (detail1.html) [1]

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

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

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>Geräteerkennung: AB12345678 </h2>
          <table class="status">
            <tr>
              <td>Maximale Belastung:</td>
              <td><p id="maxstrom">80.0</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]

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

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

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="number" 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 (in kWh)</th>
    <th class="event">Datum</th>
  </tr>
</table>
</div>
</div>
```

Detailansicht (detail1.html)

Group 46

Startseite Verwaltung



Gerätekennung: AB12345678

Maximale Belastung: 80.0

Spannung: 233.2V

Stromstärke: 83.3A

WARNUNG!
 Die Stromstärke liegt
 oberhalb der zulässigen
 Maximalbelastung

Ablesen

Nutzererkennung:

Verbrauchswert (in kWh):

Ablesen

Alle Ablesungen

Nutzerkennung	Verbrauchswert (in kWh)	Datum
---------------	-------------------------	-------

<div id="warnung">

<form>

<table class="event" id="tableAblesen">

Detailansicht (detail1.html) [CSS]

```
input {  
  border: 1px solid #24425F;  
  border-radius: 4px;  
  width: 100%;  
}
```

Standard-Rahmenfarbe überschreiben

Rahmen abrunden

Minimale Breite von 100%

```
#warnung {  
  display: none;  
  text-align: center;  
}
```

Element wird ausgeblendet

Text zentriert

```
#warnung h3 {  
  color: red;  
}
```

<h3> Schriftfarbe im Elements „warnung“



JavaScript

JavaScript & jQuery

Deklaration/Initialisierung von Variablen

Die Ober-
und
Untergrenze
für die
Spannung

```
[var minNumber = 220;  
var maxNumber = 240;
```

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

```
[var datum = new Date();  
var heute = datum.getDate() + "." + (datum.getMonth() + 1) + "." + datum.getFullYear();
```

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

```
[var spannung = randomNumberFromRange(minNumber, maxNumber);  
var strom = randomNumberFromRange(0, parseInt(document.getElementById("maxstrom").innerHTML) + 5);
```

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);  
}
```

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

```
function validateInput(val) {  
    return /[A-Z]{2}\d{2}/i.test(val);  
}
```

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]

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

Die Elemente werden mit den randomisierten Werten befüllt

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");
    }

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

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

jQuery [2]

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

Dem Element wird
ein event listener
click() hinzugefügt

Dem event listener wird eine
anonyme CallbackFunktion
übergeben

```
$("#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>');
    }
});
```

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

Sonst wird nach dem Knopf
mit after() ein Element mit der
Warnung hinzugefügt



Teammitglieder