

# Das HTML-Control und das „Styling“

## Einleitung



In diesem Beispiel wird gezeigt, dass die Gestaltung der Bedienoberfläche einer TD-Mobile-Anwendung nicht auf die Container und Controls beschränkt ist, die zum Standardlieferungsumfang von TD Mobile gehören.

Genauer gesagt: Es geht um die Erläuterung der Frage, was man mit dem generischen HTML-Control, das in TD Mobile angeboten wird, anfangen kann.

In diesem Fall soll eine „aktive“ Visitenkarte auf einer Seite angezeigt werden. Aktiv ist die Visitenkarte insofern, als nach Antippen der Karte eine weitere, beliebige Aktion (in diesem Fall die Alert-Box) ausgeführt werden kann.

## Design

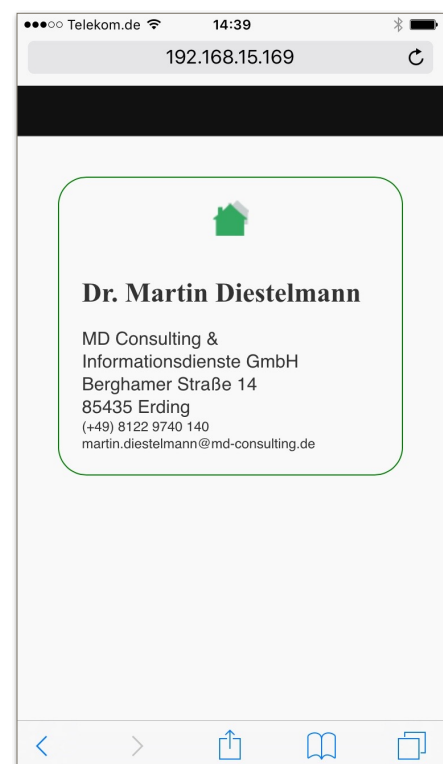
Auf der Seite `pageVisitenkarte` wird lediglich ein HTML-Control mit dem Namen `html1` angelegt. Davor wurde ein Textfeld angelegt, indem lediglich ein erläuternder Text angezeigt wird. Das Textfeld wird im Weiteren nicht betrachtet.

Wie, so stellt sich die Frage, werden nun in diesem HTML-Control die Daten dargestellt und, die zweite Frage, wie kommt es, dass die Seite so gestaltet ist, wie sie gestaltet ist.

## Codierung

Die Realisierung der Oberfläche erfolgt durch folgende Schritte:

- Anlegen einer Klasse `clsVisitenkarte` mit den (Instanz-)Variablen Name, Firma, Strasse, PLZ\_Ort, Land, Image, Telefon und EMail. Alle Instanzvariablen sind vom Datentyp `String`
- Instanziierung dieser Klasse auf der Seite `pageVisitenkarte` als einzelnes Objekt mit dem Namen `EINEVISITENKARTE`
- Anlegen einer serverseitigen Operation mit dem Namen `KarteEinlesen`
- Aufruf dieser Operation in der `Create`-Sektion der Seite `pageVisitenkarte`. Danach – also asynchron – wird die JavaScript-Funktion `visitenkarteAnzeigen` ausgeführt.



Es bleibt nur noch zu klären, welche Funktionalität in der serverseitigen Operation `KarteEinlesen` und welche Funktionalität in der clientseitigen JavaScript-Funktion `visitenkarteAnzeigen` ausgeführt wird.

## Die Operation KarteEinlesen

Die Operation `KarteEinlesen` ist folgendermaßen aufgebaut:

```

Operation: KarteEinlesen
...
Returns
  clsVisitenkarte
  Bindung: EINEKARTE
Lokale Variablen
  clsVisitenkarte: oKarte
Aktionen
  
```

```
Set oKarte.Name = "Dr. Martin Diestelmann"  
Set oKarte.Firma = "MD Consulting & Informationsdienste GmbH"  
Set oKarte.Strasse = "Berghamer Straße 14"  
Set oKarte.PLZ_Ort = "85435 Erding"  
Set oKarte.Image = "images/Kunden_Turquoise.png"  
Set oKarte.Telefon = "(+49) 8122 9740 140"  
Set oKarte.EMail = "martin.diestelmann@md-consulting.de"  
Return oKarte
```

Als Rückgabewert der Operation wird der komplexe Datentyp `clsVisitenkarte` definiert, der an die Seitenbindung `EINEKARTE` gebunden wird.

Als lokale Variable wird ebenfalls die Klasse `clsVisitenkarte` instanziiert. Auf diese Art und Weise werden die einzelnen Variablen der Klasse mit (Dummy-)Daten versorgt. In der letzten Zeile wird das Objekt `oKarte` als Rückgabewert an die aufrufende Funktion auf der Seite `pageVisitenkarte` zurückgegeben. Hiermit ist die Bindung `EINEKARTE` initialisiert (also: true) und die clientseitige Verarbeitung kann durch den Aufruf der JavaScript-Funktion `visitenkarteAnzeigen` fortgesetzt werden.

Noch ein Hinweis: der Variablen `image` wird ein String zugewiesen. Dieser String enthält den relativen Pfad zu einer PNG-Datei, die sich im Unterverzeichnis `images` befindet.

Wie gesagt, diese Operation wird aufgerufen, wenn die Seite `pageVisitenkarte` für die Anzeige erstellt wird. Im Ergebnis ist die Bindung `EINEKARTE` initialisiert. Mit der entsprechenden JavaScript-Funktion kann nun die Anzeige der Daten in der Bindung `EINEKARTE` vorgenommen werden.

## Die JavaScript-Funktion `visitenkarteAnzeigen`

Die JavaScript-Funktion `visitenkarteAnzeigen` hat folgenden Aufbau:

```
var karte = Td.Data.Binding.get("EINEKARTE").getValue();  
var anzeige = "";  
anzeige += htmlHinzufuegen("<div class='Person'>", karte.Name, "</div>");  
anzeige += htmlHinzufuegen("<div class='Firma'>", karte.Firma, "</div>");  
anzeige += htmlHinzufuegen("<div class='Adresse'>", karte.Strasse, "</div>");  
anzeige += htmlHinzufuegen("<div class='Ort'>", karte.PLZ_Ort, "</div>");  
anzeige += htmlHinzufuegen("<div class='erreichbar'>", karte.Telefon, "</div>");  
anzeige += htmlHinzufuegen("<div class='erreichbar'>", karte.EMail, "</div>");  
$("#html1").html(anzeige);  
$("#html1").click(function(){  
    alert("Hier könnte eine andere Seite für Firma " + karte.Firma + " aufgerufen werden");  
})  
  
function htmlHinzufuegen(vorher, data, nachher){  
    var zeile;  
    zeile = vorher + data + nachher;  
    return zeile;  
}
```

Es wird die Bindung `EINEKARTE` über die TD Mobile API ermittelt und das Ergebnis wird der JavaScript-Variablen `karte` zugewiesen.

Danach wird die JavaScript-Variable `anzeige` aufgebaut, indem zunächst das Image-Tag mit dem Pfad zur Abbildung der Variablen `anzeige` zugewiesen wird. Danach wird die (weiter unten stehende) JavaScript-Funktion `htmlHinzufuegen` mehrfach aufgerufen und der Rückgabewert wird mit der Variablen `anzeige` konkateniert.

Schließlich wird die jQuery Mobile-Notation `$` verwendet, um dem Objekt `html1` zum einen die Variable `anzeige` zuzuordnen. Das heißt, dass dem HTML-Objekt `html1` der HTML-Code `anzeige` zugewiesen wird. Zum

anderen wird in der Zeile darunter festgelegt, dass das HTML-Objekt `html1` auf das Antippen mit der Anzeige einer Alert-Box reagieren soll. Es wird also ein Event-Handler beim HTML-Objekt `html1` hinzugefügt.

Am Ende des JavaScripts `visitenkarteAnzeigen` wird die mehrfach aufgerufene Funktion `htmlHinzufuegen` definiert: es werden jeweils drei Parameter übergeben, um das `<div>`-Tag, einen Wert und das schließende `</div>`-Tag zusammensetzen zu können. Das Ergebnis dieser JavaScript-Funktion wird als Rückgabewert zurückgegeben (und damit letztlich die komplexe Layoutinformation aufgebaut, die an das HTML-Control `html1` gebunden wird).

Wenn die Anwendung nun gestartet wird, erscheinen die Daten an der Oberfläche der Seite `pageVisitenkarte`. Allerdings ist die Darstellung noch „unschön“. Eine Veränderung der Darstellung kann über eine entsprechende CSS-Datei vorgenommen werden.

## Gestaltung

Um die Ausgabe zu gestalten, wird in TD-Mobile eine CSS-Datei erzeugt und in die Datei werden die nachfolgend dargestellten CSS-Anweisungen eingetragen.

Wichtig ist zunächst, dass mit der Deklaration `#` der Name eines Objekts angesprochen wird. In diesem Fall bedeutet das, dass alle Objekte in einer TD-Mobile-Anwendung, die den Namen `html1` haben, auf die folgende Art und Weise formatiert werden. (Es muss also aufgepasst werden, dass nicht beispielsweise auf anderen Seiten aus Versehen noch ein Objekt mit dem Namen `html1` angelegt wird).

Demgegenüber bedeutet die Deklaration `.` in einer CSS-Datei, dass eine CSS-Klasse angesprochen werden soll. In diesem Fall sind eine Reihe von Klassen (`.Person`, `.erreichbar`, `.Adresse`, `.Logo`, `.Ort`, `.Firma`) angelegt worden.

```
#html1{
  border-color: black;
  border-width: 1px;
  border-radius: 25px;
  border-style: solid;
  padding: 10px;
  font-family:sans-serif;
  width: 80%;
}
.Person{
  font-weight: bold;
  font-size: 20px;
  line-height: 20px;
}
.erreichbar{
  font-size: 12px;
}
.Adresse{
  font-size: 12px;
  margin-top:5px;
}
.Logo{
  margin: auto;
}
.Ort{
  font-weight: semibold;
  margin-top:10px;
  margin-bottom:10px;
}
.Firma{
  margin-top: 20px;
  color: #16738F;
  font-weight: semibold;
}
```

```
font-size: 18px;  
width: auto;  
}
```

Gezeigt werden sollte exemplarisch, dass die Gestaltung einer Oberfläche vollkommen unabhängig von der Generierung der Oberfläche vorgenommen werden kann. CSS ermöglicht eine sehr umfangreiche „Umgestaltung“ von HTML-Objekten, die hier noch nicht einmal ansatzweise erklärt werden kann. Hier kann nur auf die umfangreichen Darstellungen in Büchern und im Internet verwiesen werden.

## Zusammenfassung

Es konnte gezeigt werden, dass das HTML-Objekt (unter anderem) in TD Mobile die „offene“ Schnittstelle zur Oberflächengestaltung einer mobilen Anwendung darstellt.

In diesem Fall wurden die Daten, die in der Oberfläche dargestellt werden sollen, über eine serverseitige Operation an die Bindung EINEKARTE übergeben. Mit der JavaScript-Funktion `visitenkarteAnzeigen` wurden die Daten in HTML umformatiert und dieser HTML-Text wurde dem Oberflächenobjekt `html1` zugewiesen. Damit werden die Daten in der Oberfläche angezeigt. Das Objekt ist bereits auch antippbar.

Allerdings gefiel die Gestaltung des (komplexen) HTML-Objekts noch nicht, deshalb wurden der `<div>`-Tag mit unterschiedlichen Klassen durch eine entsprechenden CSS-Definition geändert. Jetzt sieht die Anzeige zwar immer noch nicht „überzeugend“ aus, aber die Daten werden anders dargestellt, als sie ursprünglich der Oberfläche zur Anzeige übergeben wurden.

### MD Consulting & Informationsdienste GmbH

Michaelisstraße 13a  
99084 Erfurt

Berghamer Straße 14  
85435 Erding

phone (+49) 8122 97400  
email [info@md-consulting.de](mailto:info@md-consulting.de)