



Programmierung mobiler Geräte

SoSe
2015

Mobile Webseiten und Webapps

Markus Berg

Hochschule Wismar

Fakultät für Ingenieurwissenschaften

Bereich Elektrotechnik und Informatik

<http://mberg.net>

Ziel für heute

- ▣ Nutzung von **Webtechnologien** zur Entwicklung von **Anwendungen für mobile Geräte** als Alternative zur nativen Programmierung

Entwicklungsmöglichkeiten

- Vor Entwicklung einer Applikation steht immer die Frage nach den gewünschten Funktionen und den damit verbundenen Entwicklungsmöglichkeiten
- Umsetzungsvarianten abhängig von:
 - Gewünschten Funktionen
 - Benötigte Hardware für deren Umsetzung
 - Nutzerkreis
 - Darüber hinaus:
 - Budget (Zeit und Geld)
 - Bereits Erfahrung (IDE, APIs,...)

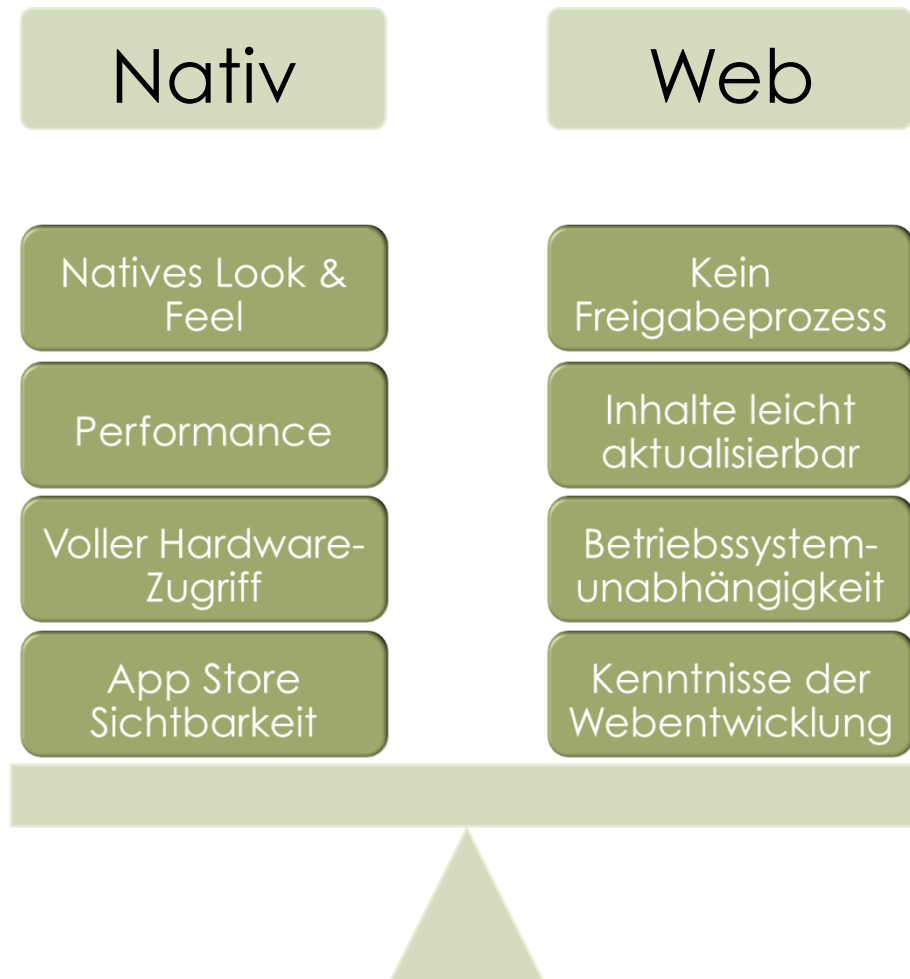
Native Entwicklung

- Entwicklung in der für das Gerät vorgesehenen Programmiersprache, die den Code in die Maschinenbefehle der entsprechenden Plattform („nativ“) übersetzt, mit der Möglichkeit das gesamte Potential der Plattform und des Gerätes auszunutzen
- z.B.
 - Android → Java
 - Windows Phone → C#
 - iOS → Objective C

Web: Mobile Webseite / Webapp

- Nutzung der Methoden und Kenntnisse der Webentwicklung (HTML, CSS, JS, AJAX) statt nativer Programmierung
- Realisiert als für kleine Bildschirme optimierte Webseite (mobile Webseite) oder als „optische Täuschung“ einer App als Webseite (Webapp)

Nativ vs. Web (Auswahl)



Nativ vs. Web

- Je nach Anwendungsfall ist nicht unbedingt eine native Applikation notwendig
 - z.B. rein informationsorientierte Anwendung
 - Kein Bedarf für Gyroskop, Mikrofon, Kamera,...
 - Performance nicht vorrangig
 - aber: keine Offline-Inhalte möglich
 - Marketingtechnisch
 - Nachteile durch fehlende AppStore-Integration
 - Vorteile, da keine App installiert werden muss
 - Entwicklung
 - Einheitliche Codebasis (nur eine einzige Anwendung) für alle Plattformen
- Aber nicht nur rein technisch beantwortbar: oft Philosophiefrage

Web: mobile Website

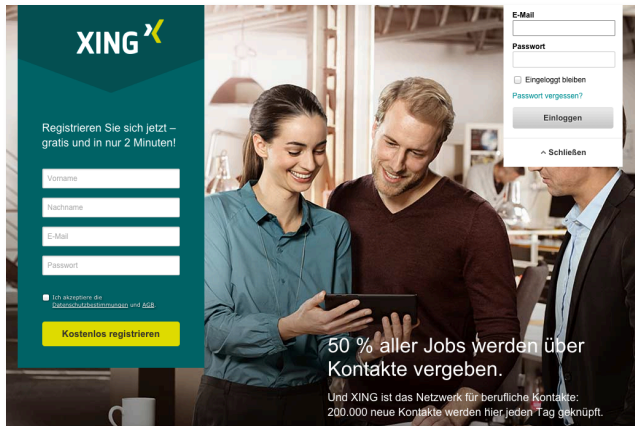
- (rein) **mobile Website**
 - Separater Webauftritt für mobile Geräte
 - Optimiert für kleine Auflösungen / Touchbedienung
 - Darstellung auf Desktop möglich, jedoch trotzdem in geringer Auflösung
 - Sieht nicht wie eine App aus (da es ja eine Website ist...)

Web: Webapp

- Eine Webapp ist ebenfalls eine herkömmliche Website
- Allerdings wird sie so designed, dass sie wie eine App aussieht
 - 2 Varianten:
 - Identisch auf jeder Plattform
 - Angepasst an native Darstellung der Plattform
 - Sieht aus wie App, jedoch keine Integration in App Store und Anzeige im Browser des mobilen Betriebssystems
- Hilfsmittel:
 - HTML (5)
 - CSS
 - JS

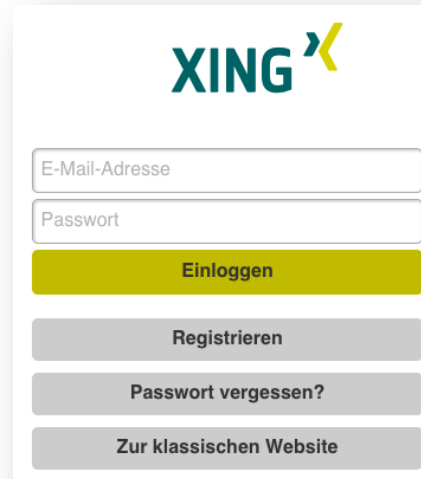
Beispiele (Vergleich)

Website (nicht mobil)



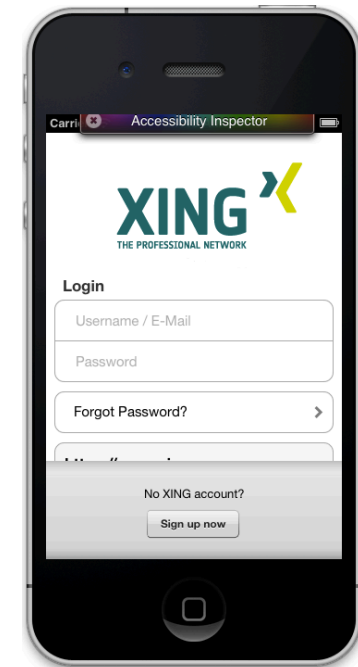
The screenshot shows the XING website registration form on a desktop browser. The form is overlaid on a background image of three people looking at a tablet. The registration form includes fields for 'Vorname', 'Nachname', 'E-Mail', and 'Passwort'. There are checkboxes for 'Eingeloggt bleiben' and 'Passwort vergessen?'. A 'Einloggen' button is visible. Below the form, there is a yellow button labeled 'Kostenlos registrieren'. Text on the page includes: 'Registrieren Sie sich jetzt! – gratis und in nur 2 Minuten!', '50 % aller Jobs werden über Kontakte vergeben.', and 'Und XING ist das Netzwerk für berufliche Kontakte: 200.000 neue Kontakte werden hier jeden Tag geknüpft.'

Webapp



The screenshot shows the XING webapp login screen. It features the XING logo at the top. Below the logo are two input fields: 'E-Mail-Adresse' and 'Passwort'. A yellow button labeled 'Einloggen' is positioned below the password field. Below the login button are three grey buttons: 'Registrieren', 'Passwort vergessen?', and 'Zur klassischen Website'.

Nativ



The screenshot shows the XING native mobile app login screen on an iPhone. The app interface includes the XING logo and the tagline 'THE PROFESSIONAL NETWORK'. Below the logo is a 'Login' section with input fields for 'Username / E-Mail' and 'Password'. There is a 'Forgot Password?' link with a right-pointing arrow. At the bottom, there is a 'No XING account?' section with a 'Sign up now' button. The status bar at the top shows 'Carrier' and 'Accessibility Inspector'.

Kombination: Responsive Website

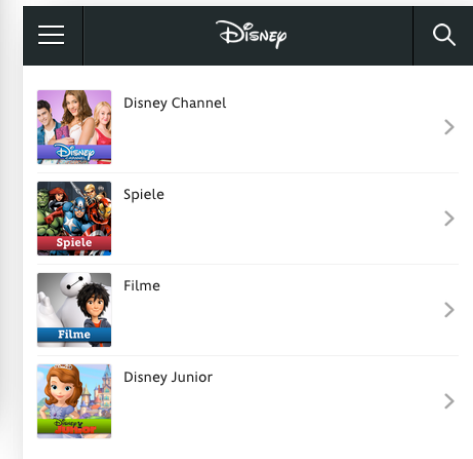
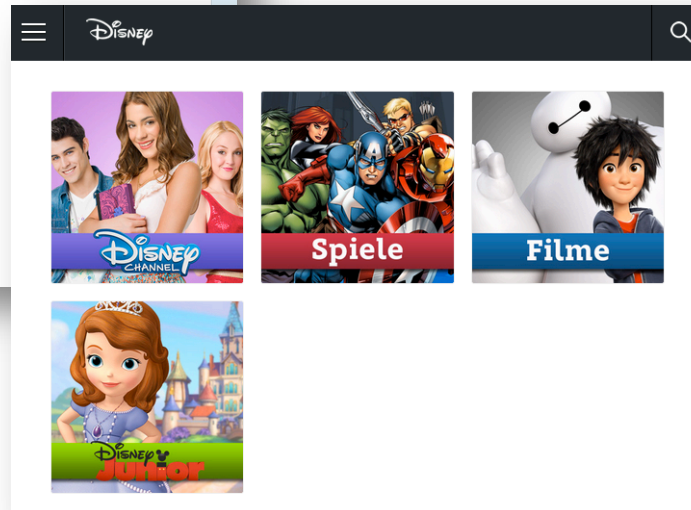
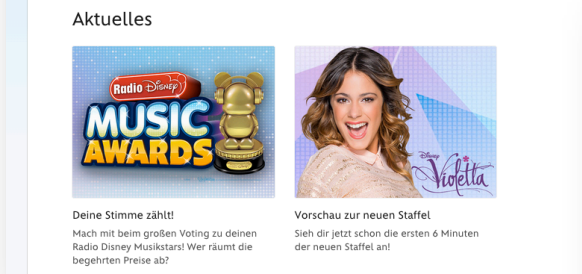
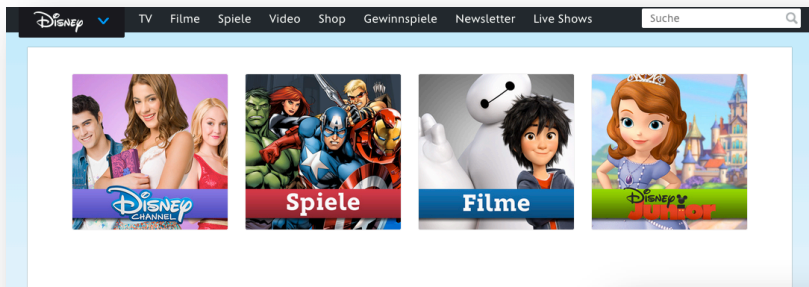
▣ Responsive Website

- ▣ Kombination aus herkömmlicher (Desktop-) Website und mobilem Pendant (oder sogar Webapp)
- ▣ Herkömmliche Website allerdings mit dynamischer Anpassung an Auflösung
 - ▣ Ist damit im Prinzip auch „mobil“
 - ▣ D.h. anderes Layout und größere Buttons bei kleinen Displays
- ▣ Gleiche Website wird somit auch zur Anzeige auf Desktop-Browser genutzt (z.B. mehrspaltiges Layout)
 - ▣ D.h. gleicher Inhalt!
- ▣ Oftmals „mobile-first“-Ansatz

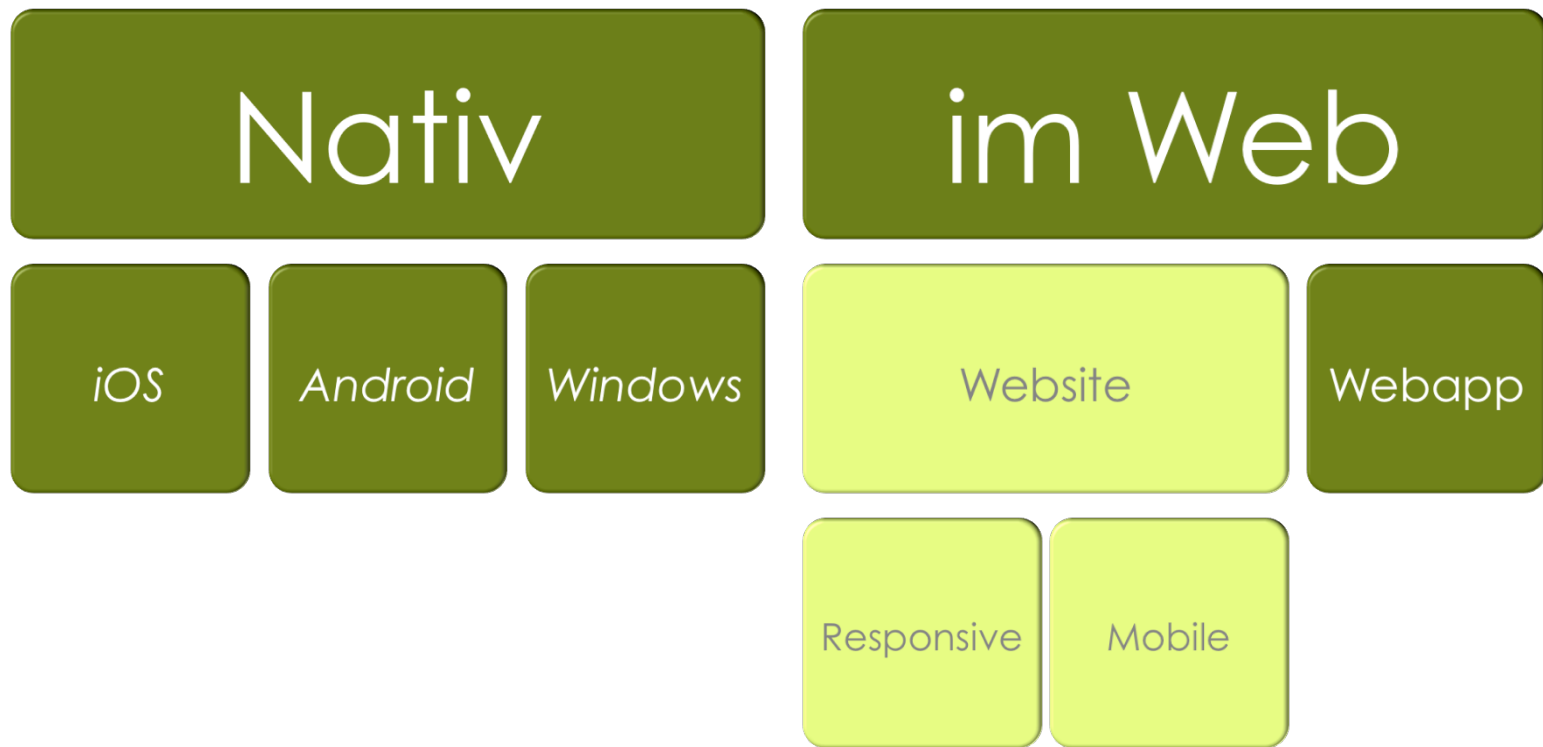
Die Grenzen sind fließend! Im Extremfall kann eine responsive Website bei kleiner Auflösung eine Webapp werden, d.h. das Aussehen einer App imitieren.

Responsive Website

Beispiel: www.disney.de



Übersicht der mobilen Entwicklungsmögl.



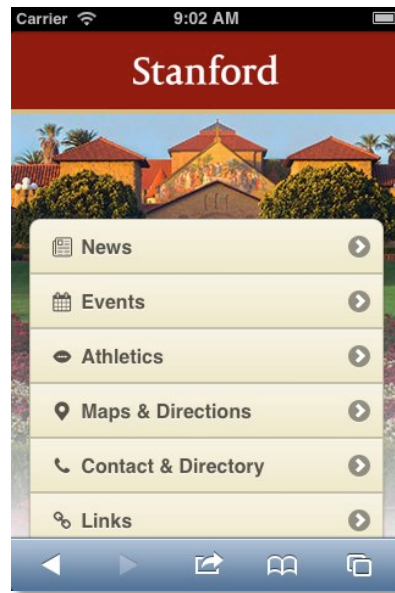
Übersicht der App-Entwicklungsmögl.

- Bisher kennengelernte Möglichkeiten zur Entwicklung von mobilen Anwendungen
 - Wir schließen Websites aus



Apps

- Engl. „**Mobile** App“, da jedes Programm eine App(lication) ist
- Jeder weiß was eine App ist. Einfache Frage:
 - **Ist das eine App?**



- Wie wurde sie realisiert? (Nativ, Mobile Website, Webapp)

Ab wann ist eine App eine App?

- These:
 - Eine Anwendung (Application) hat eine Funktion (Datentransformation, Steuerung, Berechnung, ...)
 - Taschenrechner, Kalender, Spiel,...
 - Eine Website ist und bleibt eine Website
 - Es werden hauptsächlich Informationen angezeigt
- Problem:
 - **Was ist nun mit Websites, die so aussehen wie eine mobile Anwendung?**
 - **Und was ist mit nativen mobilen Anwendungen, die nur Informationen anzeigen?**
 - Alles was aus einem AppStore geladen werden kann, ist im Volksmund eine App
 - Wikipedia, Tagesschau, ...
 - Was ist, wenn die Anwendung zwar Informationen darstellt, aber mir besondere Funktionen zum Filtern/Suche bietet? Ist das nicht auch eine Funktion?
 - Preisvergleich, Reiseportale, Fernsehzeitung, ...
 - Ist eine Website, die das bietet, dann auch eine Anwendung?
- Eine Antwort, die JEDEM gerecht wird, gibt es leider nicht
 - Grenzen sind fließend (im Zeitalter wo über den Browser als Betriebssystem nachgedacht wird erst recht)
 - Marketingbegriffe sind meist nicht klar definiert... (vgl. „Ist eine Netzwerkfreigabe schon eine Cloud?“)
 - **Wir erkennen eine App i.d.R. am „typischen“ Aussehen**

Apps

- Wir konzentrieren uns auf **Anwendungen** (im Gegensatz zu Webseiten), d.h. der Fokus liegt nicht (nur) auf dem reinen Anzeigen von Informationen, sondern vor allem auf einer Funktionalität (die darüber hinaus geht).

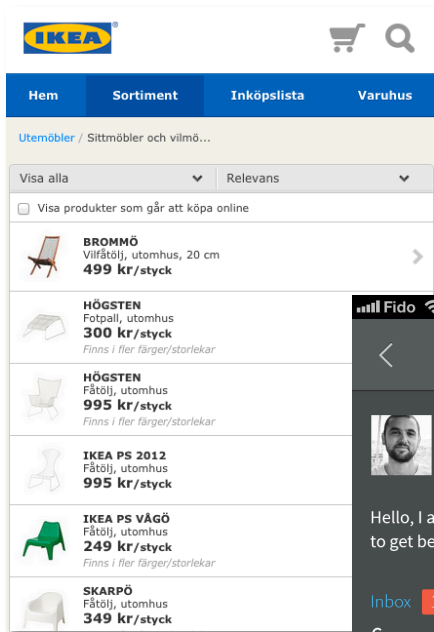
Heute: Webapps

- ▣ Webseiten, die so aussehen wie Apps
- ▣ Größter Vorteil für Entwickler
 - ▣ Einheitliche Codebasis
 - ▣ Eine Software für alle Plattformen
 - ▣ Keine Gebühren/Aufwand bzgl. App Stores
 - ▣ Nutzung vorhandener Web-Kenntnisse
- ▣ Aber nicht vergessen:
 - ▣ Nutzer erkennen Unterschied zu nativer App (Look & Feel)
 - ▣ Das alte Problem: evtl. unterschiedliche Darstellung in versch. Browsern
 - ▣ Anwendung ist nicht im App Store (geringere Sichtbarkeit am Markt)
 - ▣ Keine Nutzung von spezieller Hardware wie Kamera etc. möglich
 - ▣ Erfordert Internetverbindung

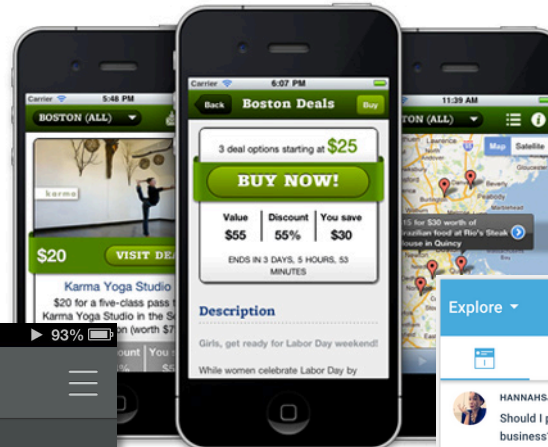
Webapp-Frameworks

- (UI-)Frameworks für Webapps erleichtern uns die Entwicklung und geben einheitliches Look & Feel vor
 - Homogenes Design auf allen Plattformen
 - *(Teilweise auch Themes, die natives Aussehen imitieren)*
 - Design ist vorgegeben: schnelle, gutaussehende Ergebnisse
 - Entwickler kann sich auf Funktionalität/Inhalte konzentrieren
 - z.B.
 - jQuery mobile
 - Sencha Touch
 - Ionic
 - Adobe Topcoat
 - Intel App Framework
 - und viele weitere mehr...

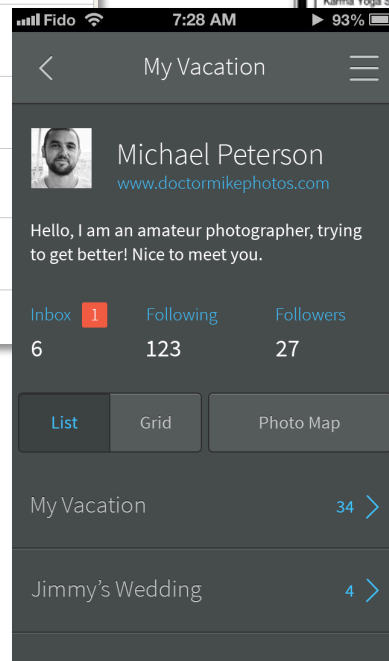
Frameworks: Ein Bild sagt mehr als...



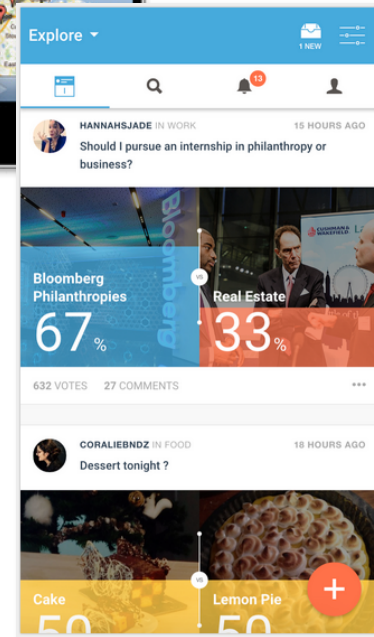
jQuery Mobile



Sencha Touch



Adobe Topcoat



Ionic

JQuery mobile



„jQuery Mobile is a HTML5-based user interface system designed to make responsive web sites and apps that are accessible on all smartphone, tablet and desktop devices.“ [<http://jquerymobile.com>]

- Mobile UI Framework für Webapps
 - Seit 2010
 - Open Source (MIT Lizenz)
 - „Plattform-agnostisch“
 - Setzt JQuery voraus
- Seitenstruktur über HTML 5
- Anwendung besteht aus **einer** HTML-Seite
 - Auszeichnungen markieren die einzelnen Screens der App
- Aussehen über CSS 3 gesteuert
- Dynamik (DOM-Manipulation) über Javascript bzw. JQuery

JQuery mobile: Vorbereitung

- Voraussetzungen: HTML/CSS/JS-Kenntnisse
- Anlegen eines neuen Projektverzeichnisses
- Herunterladen von JQuery...
 - <http://jquery.com/>
- ... und JQuery mobile
 - <http://jquerymobile.com/>
- Entpacken/kopieren in Unterordner des Projektes
- Anlegen einer HTML-Datei (z.B. index.html)
 - Zunächst einfach eine leere Textdatei
 - z.B. mit Notepad++
- Demos und Doku unter: <http://demos.jquerymobile.com/>

JQuery mobile: Moin Wismar!

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="../jquerymobile/jquery.mobile-1.4.5.min.css">
  <script src="../jquery/jquery-2.1.3.min.js"></script>
  <script src="../jquerymobile/jquery.mobile-1.4.5.min.js"></script>
  <title>Moin Wismar</title></head>
<body>
  <div data-role="page">

    <div data-role="header">
      <h1>Moin Wismar!</h1>
    </div>

    <div data-role="content">
      <p>"Urteilen kann als Fortschreiten von einem Gedanken zu seinem
        Wahrheitswerte gefasst werden" &mdash; <i>Gottlob Frege</i></p>
    </div>

    <div data-role="footer">
      <h4>Programmieren mobiler Anwendungen</h4>
    </div>

  </div>
</body>
</html>
```

JQuery (mobile) einbinden

Anwendung wird über HTML-divs strukturiert.

Anwendung besteht aus Pages.

Page besteht aus Header, Content und Footer.

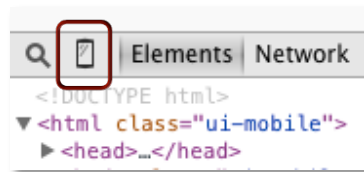
Moin Wismar!

"Urteilen kann als Fortschreiten von einem Gedanken zu seinem Wahrheitswerte gefasst werden" — *Gottlob Frege*

Programmieren mobiler Anwendungen

Darstellung

- Browserfenster verkleinern für app-ähnlichere Darstellung bzw. (in Chrome) Entwicklertools benutzen
- Anzeigen > Entwickler > Entwicklertools



- Oder auf Webserver kopieren und per mobilem Gerät abrufen



Liste

- Wir fügen eine einfache Liste hinzu
- Dies geschieht mit Standard HTML-Tags `` und ``
 - Zusätzlich erfolgt eine Auszeichnung mit `data-role="listview"` für jQuery mobile

```
<div data-role="content">  
  <p>"Urteilen kann als Fortschreiten von einem Gedanken zu seinem  
  Wahrheitswerte gefasst werden" &mdash; <i>Gottlob Frege</i></p>  
  <ul data-role="listview" data-inset="true">  
    <li><a href="#">Marienkirche</a></li>  
    <li><a href="#">Wonnemar</a></li>  
    <li><a href="#">Tierpark</a></li>  
    <li><a href="#">Hafen</a></li>  
    <li><a href="#">Bürgerpark</a></li>  
  </ul>  
</div>
```

- [Marienkirche](#)
- [Wonnemar](#)
- [Tierpark](#)
- [Hafen](#)
- [Bürgerpark](#)



Moin Wismar!

"Urteilen kann als Fortschreiten von einem Gedanken zu seinem Wahrheitswerte gefasst werden" — *Gottlob Frege*

- Marienkirche >
- Wonnemar >
- Tierpark >
- Hafen >
- Bürgerpark >

Programmieren mobiler Anwendungen

Button

- Buttons werden über einen Link erzeugt
 - `...`
 - Auszeichnung mit `data-role="button"`
 - Auswahl eines Icons: <http://demos.jquerymobile.com/1.4.5/icons/>
 - Star
 - Refresh
 - Forward
 - ...

```
<a href="#" data-role="button" data-  
icon="refresh">neues Zitat</a>
```



Themes

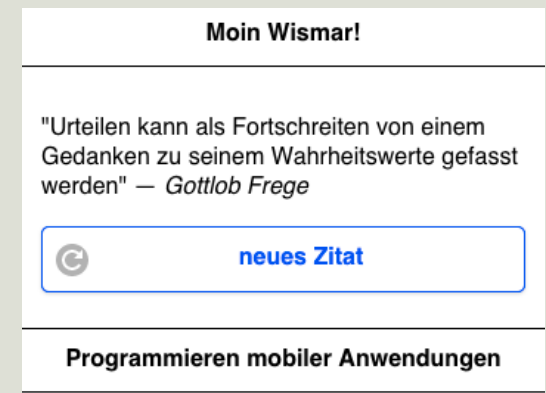
- Bzw. *Theme Swatch*
 - Variante eines Themes
 - mehrere Swatches pro Theme
- Identifiziert über Buchstaben a..e
- Vererben sich (gelten für alle Kindelemente)
- `<div data-role="page" data-theme="a">`



data-theme="a"



data-theme="b"



data-theme="c"

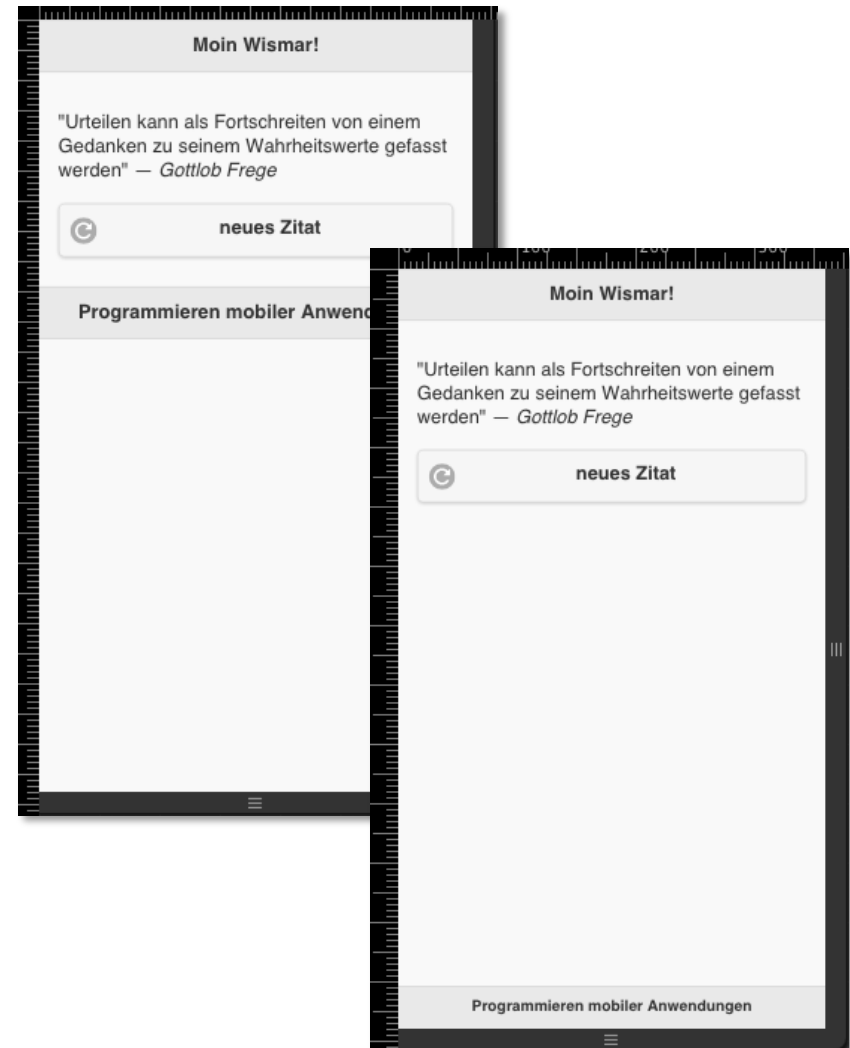
Footer

- Der Footer ist momentan nicht immer am unteren Rand

```
<div data-role="footer" data-position="fixed">
```

- Die Schrift ist zu groß
 - Neue CSS-Datei
 - Überschreiben des alten Wertes für h6

```
h4 {font-size:.8em !important;}
```



Mehrere Seiten

- Eine App besteht aus mehreren Seiten
 - Multipage-Template: alle Seiten in einer HTML-Datei
 - Wir haben bereits eine Seite angelegt
 - „Page“-div
 - Hinzufügen einer zweiten Seite, d.h. eines zweiten divs
 - Jede Page bekommt eine ID

```
<div data-role="page" data-theme="a" id="seite1">
  <div data-role="header">
    <h1>Moin Wismar!</h1>
  </div>

  <div data-role="content">
    ...
  </div>

  <div data-role="footer" data-position="fixed">
    <h4>Programmieren mobiler Anwendungen</h4>
  </div>
</div>

<div data-role="page" data-theme="a" id="seite2">
  <div data-role="header">
    <h1>Moin Wismar!</h1>
  </div>

  <div data-role="content">
    ...
  </div>

  <div data-role="footer" data-position="fixed">
    <h4>Programmieren mobiler Anwendungen</h4>
  </div>
</div>
```

Verlinken von Seiten

- Per Link (bzw. Anker, d.h. „#“) die Seiten-ID referenzieren
 - Setzt Standard-Ankerfunktionalität außer Kraft
- Statt „Back-Button“ fest zu verdrahten, besser:

```
<a href="#" data-rel="back">letzte Seite</a>
```

```
<div data-role="page" data-theme="a" id="seite1">
  <div data-role="header">
    <h1>Moin Wismar!</h1>
  </div>

  <div data-role="content">
    <p><a href="#seite2">nächste Seite</a></p>
  </div>

  <div data-role="footer" data-position="fixed">
    <h4>Programmieren mobiler Anwendungen</h4>
  </div>
</div>

<div data-role="page" data-theme="a" id="seite2">
  <div data-role="header">
    <h1>Moin Wismar!</h1>
  </div>

  <div data-role="content">
    <p><a href="#seite1">letzte Seite</a></p>
  </div>

  <div data-role="footer" data-position="fixed">
    <h4>Programmieren mobiler Anwendungen</h4>
  </div>
</div>
```

Seitenübergang: Transitions

- Animationen bei Seitenübergängen
 - Per CSS3 realisiert
 - Definition per data-transition-Attribut, z.B.
 - **fade** (default)
 - flip
 - pop
 - slideup

```
<a href="#seite2" data-transition="flip">nächste Seite</a>
```

Mehrere Single-Pages

- Bis jetzt: Multipage
 - Vorteil: einzelne Seiten werden schnell angezeigt, da gesamter DOM-Baum bereits geladen ist
 - Nachteil: längere Ladezeit
- Single-Page-Template + verlinkte Seiten vorladen, wenn Hauptseite fertig geladen wurde
 - Link annotieren mit Attribut: `data-prefetch`

Dialoge

- Seiten können auch als Dialog angezeigt werden
 - Anderes Design (mittig, dunkler Hintergrund)
 - Modal
 - Schließen-Button
- Separate HTML-Seite mit: `data-role="dialog"`
 - Achtung: *Cross-Origin-Problem* im Chrome
 - `--disable-web-security` (Sicherheitsrisiko)
 - Besser: Webserver nutzen
- Aufruf per Link mit Attribut:

```
<a href="dialog.html" data-rel="dialog">Dialog öffnen</a>
```

Toolbars

- Buttons im Header bzw. Footer
 - Normale Buttons (per Link `<a>` oder `<button>` anlegen)

```
<a href="#" data-rel="back"
class="ui-btn-left ui-btn ui-corner-
all ui-btn-icon-left ui-icon-carat-
l">Back</a>
```



- Position des Buttons und der Icons durch CSS-Klassen festlegen
- Icons: <http://demos.jquerymobile.com/1.4.5/icons/>
 - ui-icon-carat-l
 - ui-icon-carat-r
 - ui-icon-alert
 - ui-icon-check
 - ui-icon-home
 - ...

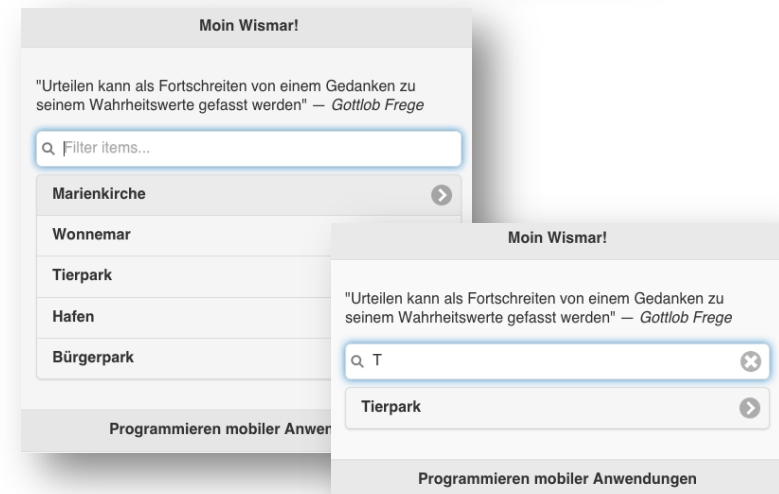
Filter Widget

□ Einfache Liste

```
<div data-role="content">
  <ul data-role="listview" data-inset="true">
    <li><a href="#">Marienkirche</a></li>
    <li><a href="#">Wonnemar</a></li>
    <li><a href="#">Tierpark</a></li>
    <li><a href="#">Hafen</a></li>
    <li><a href="#">Bürgerpark</a></li>
  </ul>
</div>
```

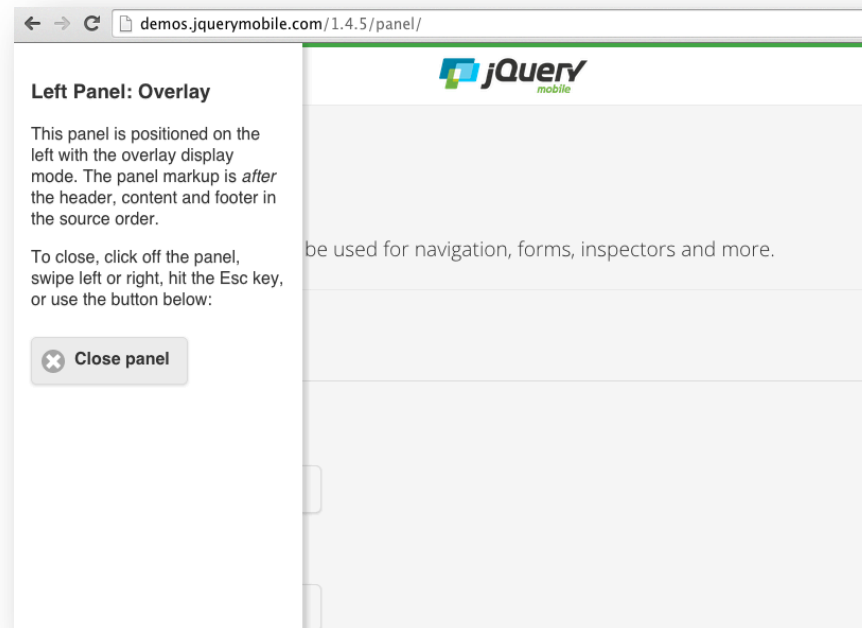


□ Ergänzen von `data-filter=„true“`



Panels

- Oft bei Tablets eingesetzt
 - Oftmals für Einstellungen genutzt (meist am linken Rand)
 - Jedoch flexibel einsetzbar
 - Overlay bzw. Reveal und Push (Inhalt wegschieben)



Weitere Elemente

■ Siehe <http://demos.jquerymobile.com/1.4.5/>

■ Datepicker

■ Select Menu

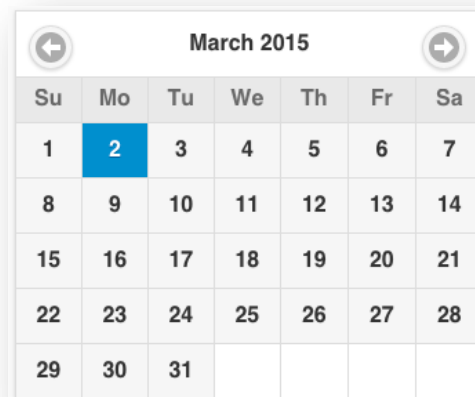
■ Slider

■ Text Input

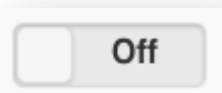
■ Flip Switch

■ Tabs

■ ...



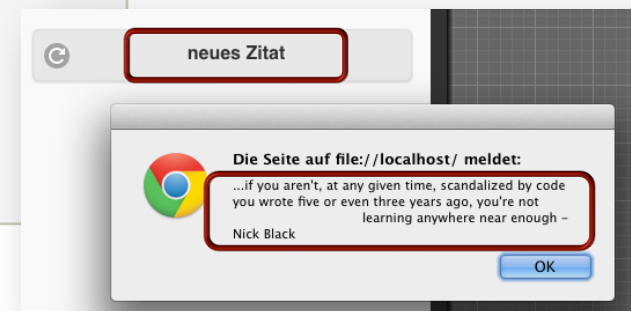
March 2015						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				



Interaktion

- Die Betätigung eines Buttons soll eine Aktion auslösen
 - In unserem Beispiel: Anzeigen eines neuen Zitates
 - Ansatz: JavaScript (bzw. JQuery)
 - Anzeigen einer Dialogbox (alert) mit Zitat bei Buttonklick
 - Dem Button die ID „newQuote“ zuweisen

```
<script>
  $(document).ready(function() {
    $( "#newQuote" ).click(function( event ) {
      alert( "...if you aren't, at any given time,
        scandalized by code you wrote five or even three
        years ago, you're not learning anywhere near
        enough - Nick Black" );
    });
  });
</script>
```



JQuery Kenntnisse?

- Wie gut können Sie sich noch an JQuery (Modul Webprogrammierung I) erinnern?



JQuery auf einen Blick

- Javascript-Bibliothek zur vereinfachten DOM-Manipulation bzw. –Navigation
 - Vereinfachte Handhabung von Ajax-Aufrufen, Event-Binding, Ändern der Inhalte von HTML-Elementen, Ein- und Ausblenden von Elementen bzw. dynamisches Ändern von CSS-Properties, ...
- JQuery-Funktion: `$`-Operator
 - Gefolgt von Selektor
- CSS-Selektoren
 - `$("selektor")`
 - `$("div")` → Selektieren per Element
 - `$("#id")` → Selektieren per ID
 - `$(".class")` → Selektieren per Klasse
- Event: Wenn Dokument fertig geladen ist (d.h. DOM-Tree vollständig)
 - `$(document).ready(function(){...});`
 - `$(function(){...});`
 - Startpunkt für jedes jQuery Skript, um sicherzustellen, dass der DOM-Tree, auf den man zugreifen möchte, geladen ist

Interaktion

- Statt *alert* nun Anzeige in der App
 - D.h. Manipulation des DOM-Trees
 - HTML-Element, dessen Text verändert werden soll mit ID „quote“ auszeichnen, um eindeutigen Zugriff zu ermöglichen
 - Zugriff per JQuery / CSS Selektor

```
<script>
  $(function() { //onDocumentReady
    $( "#newQuote").click(function( event ) { //bind click event, define handler
      $("#quote").text( "...if you aren't, at any given time, scandalized by
        code you wrote five or even three years ago, you're not learning
        anywhere near enough - Nick Black" );
    });
  });
</script>
```

- Statt `.text()` auch `.html()` möglich, um HTML-Code auszuwerten (z.B. `<i>` oder ``)

Interaktion

▣ Array mit Zitaten

```
quotes = [  
  {quote:"Urteilen kann als Fortschreiten von einem Gedanken zu seinem Wahrheitswerte gefasst  
werden",author:"Gottlob Frege"},  
  {quote:"...if you aren't, at any given time, scandalized by code you wrote five or even three years  
ago, you're not learning anywhere near enough",author:"Nick Black"},  
  {quote:"The internet could be a very positive step towards education, organisation and participation  
in a meaningful society.",author:"Noam Chomsky"},  
  {quote:"I think computer viruses should count as life. I think it says something about human nature  
that the only form of life we have created so far is purely destructive. We've created life in our  
own image.",author:"Stephen Hawking"},  
  {quote:"Computing is not about computers any more. It is about living.",author:"Nicholas  
Negroponte"},  
];
```

▣ Nächstes Zitat aus Array laden

```
function getNextQuote(){  
  if(i<quotes.length) return quotes[i++];  
  else{  
    i=1;  
    return quotes[0];  
  }  
}
```

Interaktion

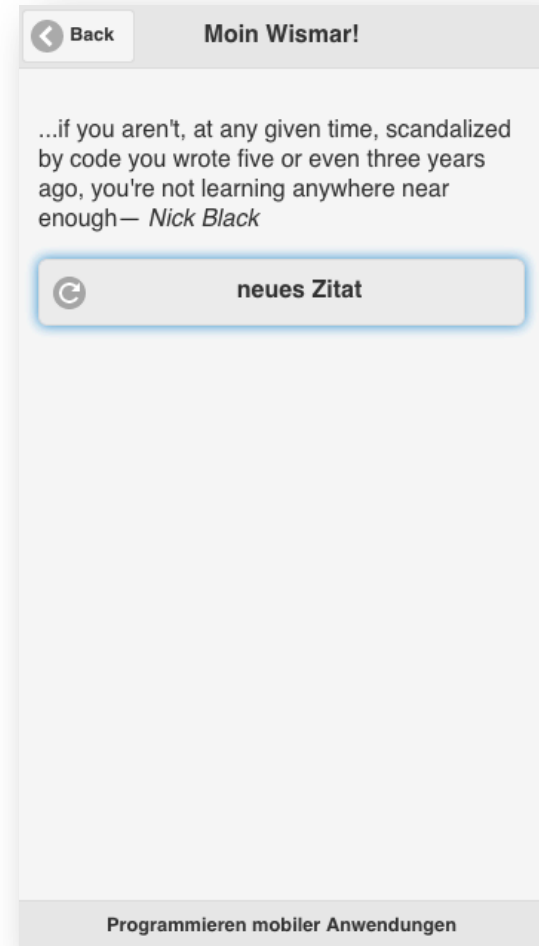
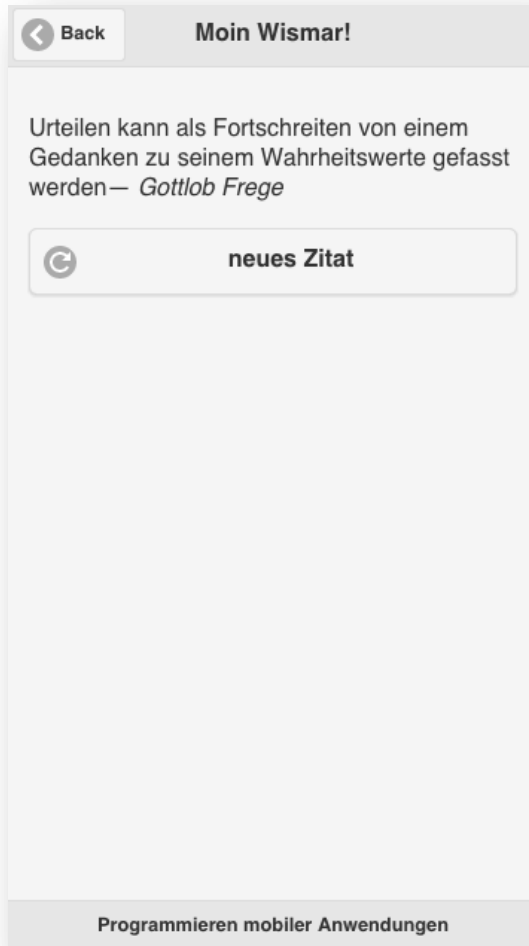
▣ Anzeigen eines Zitats

```
function displayQuote(quote){
    $("#quote").html(quote.quote + "&mdash; <i>" + quote.author + "</i>");
}
```

▣ Button-Click-Event

```
$(function() {
    $( "#newQuote" ).click(function( event ) {
        displayQuote(getNextQuote());
    });
});
```

Zwischenstand



Events

- Touchgesten: Vor / zurück
 - Registrieren des entsprechenden Events unter Angabe des Eventhandlers
 - *Swiperight* und *swipeleft*

```
$(document).swiperight(function(){  
    getNextQuoteAndDisplay();  
});  
  
$(document).swipeleft(function(){  
    getPreviousQuoteAndDisplay();  
});
```

REST

- Daten momentan statisch aus Array
- Ziel: Laden der Zitate über Webservice (REST-basiert)
 - z.B. über Jax-RS / Jersey (Java)

```
@GET
@Path("quotes/{i}")
@Produces(MediaType.APPLICATION_JSON)
public Response getQuote(@PathParam("i") int i){
    if (i>=0 && i<=quotes.size()) return Response.ok(quotes.get(i-1)).build();
    else return Response.seeOther(URI.create("quotes")).build();
}
```

- Wir wollen den Client implementieren
 - Per REST und AJAX

REST URIs

- Service steht zur Verfügung unter

- <http://.../QuoteService/quotes>

- Liefert Array aller Zitate

- <http://.../QuoteService/quotes/1>

- Liefert erstes Zitat

```
{"author":"Gottlob Frege","quote":"Urteilen kann als Fortschreiten von einem Gedanken zu seinem Wahrheitswerte gefasst werden"}
```

- <http://.../QuoteService/quotes/3>

- Liefert drittes Zitat

- <http://.../QuoteService/quotes/count>

- Liefert die Anzahl der Zitate, d.h. Größe des Arrays

JQuery & REST

- ▣ Abrufen von Informationen per GET-Request auf entsprechende Ressource (identifiziert durch URI)
- ▣ Auslösen eines GET-Requests per AJAX
 - ▣ Per JQuery über \$.get()

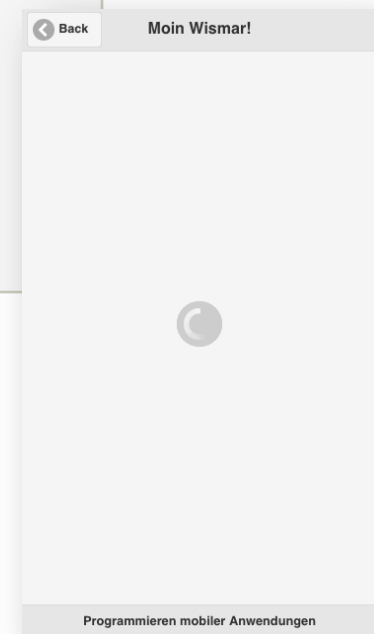
```
$.get("http://localhost:8080/QuoteService/quotes",  
function callback(data){  
    //Daten verarbeiten  
});
```

- ▣ Definition der Callback-Methode
- ▣ Alternativen
 - ▣ Beim Start komplettes Array laden und bei jedem Klick auf „nächstes Zitat“ lokal arbeiten
 - ▣ Bei jedem Klick genau ein Zitat abrufen

Ladebalken

- Anzeigen eines Ladekreises bei REST-Aktivität
 - Über Events *ajaxStart* und *ajaxStop*
 - Eventhandler zeigen eine integrierte Animation an bzw. verstecken sie (keine zusätzlichen Grafiken notwendig)

```
$(document).ajaxStart(function() {  
    $.mobile.loading('show');  
});  
  
$(document).ajaxStop(function() {  
    $.mobile.loading('hide');  
});
```



Geokoordinaten (HTML5)

- ▣ Geokoordinaten auslesen
 - ▣ `navigator.geolocation.getCurrentPosition`
- ▣ Reverse Geocoding (Koordinaten zu Adresse) über Google

```
navigator.geolocation.getCurrentPosition(  
  function(data){  
    $.get("https://maps.googleapis.com/maps/api/geocode/json?  
          latlng="+data.coords.latitude+", "+data.coords.longitude,  
          function callback(data){  
            var city=data.results[0].address_components[2].long_name; //get city name  
            $("#city").text(city);  
          });  
    },  
  function(err){console.log(err);}  
);
```

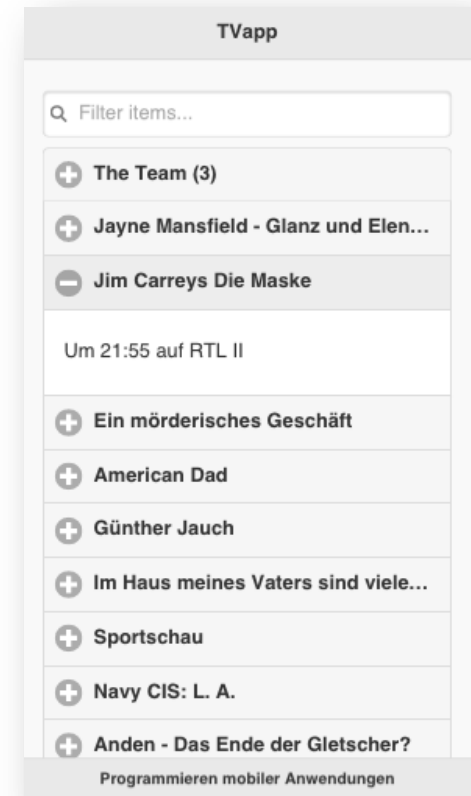
Praktikum

- Ziel: Fernsehzeitung als Webapp realisieren
- Gegeben:
 - REST-Service, der das aktuelle Programm ermittelt
 - Sender
 - Startzeit
 - Name der Sendung
 - <http://mmborg.net:8080/TVProgramm/now>

```
[{"sender":"ZDF","titel":"The Team (3)","zeit":"22:00"}, {"sender":"ARTE","titel":"Jayne Mansfield - Glanz und Elend einer Blondine","zeit":"22:00"}, {"sender":"RTL II","titel":"Jim Carreys Die Maske","zeit":"21:55"}, {"sender":"ZDFneo","titel":"Ein mörderisches Geschäft","zeit":"21:50"}, {"sender":"COMEDY CENTRAL","titel":"American Dad","zeit":"21:50"}]
```

Praktikum

- Aufgabe:
 - Implementieren Sie eine Webapp, die dem Nutzer das aktuelle Fernsehprogramm anzeigt!
 - Nutzen Sie JQuery Mobile
 - Wählen Sie eine Ansicht, die bei Bedarf zu einem Film Details anzeigen kann
 - Bieten Sie eine einfache Möglichkeit, die Sendungen zu filtern (Suche)
 - Abgabefrist: 14. April 2015
 - persönlich demonstrieren oder Abgabe per E-Mail (Link auf Webseite oder Archiv)



Nächste Woche

