**JQuery en JQuery Mobile**

Inhoudstafel

[Inhoudstafel 2](#_Toc358365626)

[Inleiding 4](#_Toc358365627)

[JQuery 5](#_Toc358365628)

[JQuery Mobile 6](#_Toc358365629)

[1. Pages & Dialogs 6](#_Toc358365630)

[1.1 Pages 6](#_Toc358365631)

[1.2 Dialogs 8](#_Toc358365632)

[2. Transitions 10](#_Toc358365633)

[3. Themes 13](#_Toc358365634)

[4. Buttons 15](#_Toc358365635)

[4.1 Thema en Stijl 15](#_Toc358365636)

[4.2 Icons 15](#_Toc358365637)

[4.3 Disabled 17](#_Toc358365638)

[4.4 Controlgroup 17](#_Toc358365639)

[4.5 Voorbeeld 18](#_Toc358365640)

[5. Navigation Bar 21](#_Toc358365641)

[6. Listview en Accodrion 25](#_Toc358365642)

[6.1 Listview 25](#_Toc358365643)

[6.2 Accordion 26](#_Toc358365644)

[7. Forms 28](#_Toc358365645)

[7.1 Inputs 28](#_Toc358365646)

[7.2 Selects 31](#_Toc358365647)

[7.3 Sliders 33](#_Toc358365648)

[7.4 Checkboxes & Radiobuttons 35](#_Toc358365649)

[7.5 Retrieve Information using JQuery 37](#_Toc358365650)

[8. Events 38](#_Toc358365651)

[8.1 Events Basic 38](#_Toc358365652)

[8.2 Orientation Events 39](#_Toc358365653)

[8.3 Scroll Events 40](#_Toc358365654)

[8.4 Touch Events 42](#_Toc358365655)

[8.5 JQuery Mouse Events 44](#_Toc358365656)

[8.6 Page Initialization Events 44](#_Toc358365657)

[8.7 Page Load and Remove Events 44](#_Toc358365658)

[8.8 Page Change Events 44](#_Toc358365659)

[8.9 Page Transitions and Animations Events 45](#_Toc358365660)

[8.10 Using Layout Events 45](#_Toc358365661)

Inleiding

Dit 2e deel gaat over JQuery en JQuery Mobile. Een basis van JQuery is wel vereist voor dit gedeelte te kunnen volgen. Daarom is er eerst een kleine inleiding van JQuery met een overzicht van een heel goede bron TheNewBoston. Daarna is er een overzicht van alle functies en widgets binnen het JQuery Mobile framework. We gaan over het gebruik van pagina’s en dialogen, gevolgd door transities. Daarna kijken we naar thema’s en allerhande widgets zoals buttons, forms en listviews. Tot slot gaan we over de events die het JQuery Mobile framework aanbiedt.

JQuery

JQuery Mobile gaat voort op het framework JQuery en het is aangeraden de basis van JQuery te kennen.

De basis van JQuery zou normaal gezien moeten gekend zijn als je van de keuzeopleiding software komt. Indien je een opfrissing nodig hebt van de gezien e leerstof, of je van een andere richting komt, kan je hieronder een overzicht van 200 filmpjes van TheNewBoston. Deze zijn een heel goede tutorial om te volgen, of om een aantal aspecten te leren kennen van JQuery.

Bron: http://thenewboston.org/list.php?cat=32

1. Intro

5. Uitleg rond inline & external

6-10: Basis rond pageloading

11-26: Selecteren van elementen uit HTML (niets speciaal)

27-30: Selectors gebruiken voor aantal voorbeelden

31 - 41: Events en Eventhandling

42 : Bind Events (!)

43 : Live code toeveogen in HTML (!)

44-51: Hovertext, eventdetectie

52- : HTML en toevoegen en verwijderen an attirbutes

66 -78: effects

97-105: AJAX Basis (!)

115-138 : plugins

JQuery UI: (Widgets)

141-143 : Draggable

144-146 : Dropable

147-148 : Sortable

149-151 : Resizable

152-153 : Accordion

154-155 : Datepicker

156-158 : Dialog

159-160 : Progressbar

161-162 : Slider

163-165 : Tabs

166-167 : Drag&Drop lists

168-170 : Link met php & mysql voor username check (!)

171-172 : Voorbeeld met werken met slidedown

173-174 ; Countdown & redirect voorbeeld

178-181 : Instant search. Voorbeeld met ajax info ophalen en results onmiddelijk weergeven

182-186 ; Online & Offline push voorbeelden

187-188 : Multiple file upload

189-191 : Div statisch blijft

192-194 : Textarea emoticons

195-198 : show-hide fields

199-200 : submit form on element change

JQuery Mobile

## Pages & Dialogs

### Pages

JQuery mobile gebruikt van een header, content en footer structuur door het attribuut data-role aan te spreken.. Het maakt altijd 4 divs voor de pagina te definieren. ‘page’ geeft het begin van elke pagina aan. Binnen een page moeten altijd de volgende 3 divs zitten: header, content en footer.

<div data-role=”page”>

<div data-role=”header”></div>

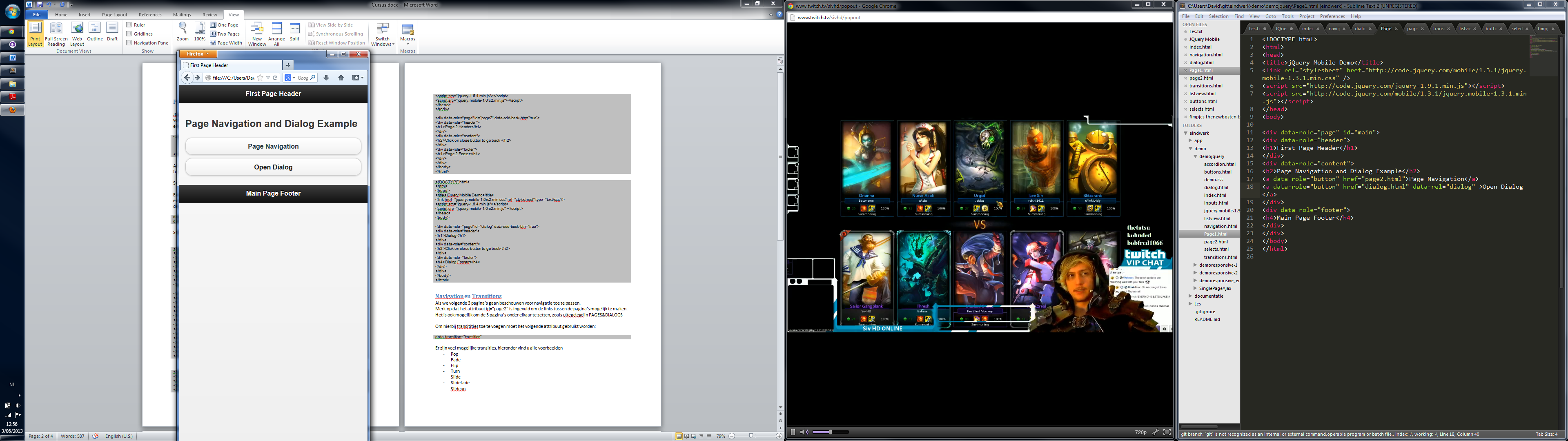
<div data-role=”content”></div>

<div data-role=”footer”></div>

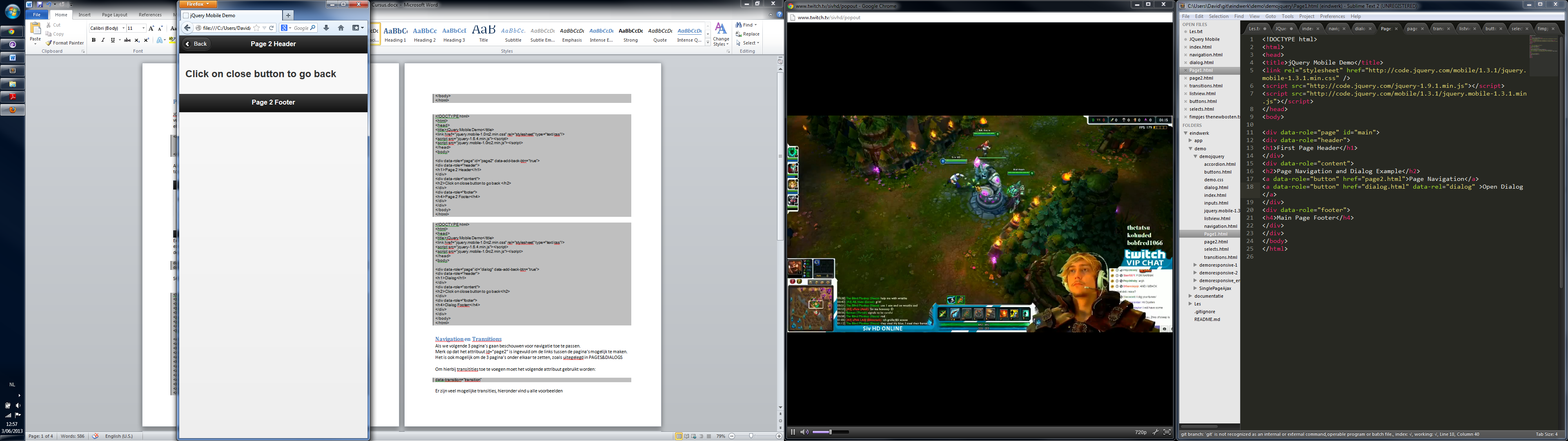
</div>

JQuery mobile gebruikt het attribuut data-role voor pagina’s te detecteren. Het is mogelijk om verschillende pagina’s in 1 html bestand te verwerken door steeds de volgende structuur dezelfde div’s elkaar te plakken en het id=”page1” in te vullen.

Als je deze structuur gebruikt ga je steeds de JQuery mobile layout hebben, welke automatisch zal toegepast worden via de jQuery Mobile css.



Screenshot 1 : First Page



Screenshot 2 : Second Page

<!DOCTYPE html>

<html>

<head>

<title>jQuery Mobile Demo</title>

<link href="jquery.mobile-1.0rc2.min.css" rel="stylesheet" type="text/css"/>

<script src="jquery-1.6.4.min.js"></script>

<script src="jquery.mobile-1.0rc2.min.js"></script>

</head>

<body>

<div data-role="page" id="main">

<div data-role="header">

<h1>First Page</h1>

</div>

<div data-role="content">

<h2>Page Navigation and Dialog Example</h2>

<a data-role="button" href="page2.html">Page Navigation</a>

<a data-role="button" href="dialog.html" data-rel="dialog" >Open Dialog </a>

</div>

<div data-role="footer">

<h4>Main Page Footer</h4>

</div>

</div>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<title>jQuery Mobile Demo</title>

<link href="jquery.mobile-1.0rc2.min.css" rel="stylesheet" type="text/css"/>

<script src="jquery-1.6.4.min.js"></script>

<script src="jquery.mobile-1.0rc2.min.js"></script>

</head>

<body>

<div data-role="page" id=”page2" data-add-back-btn=”true”>

<div data-role="header">

<h1>Page 2 Header</h1>

</div>

<div data-role="content">

<h2>Click on close button to go back </h2>

</div>

<div data-role="footer">

<h4>Page 2 Footer</h4>

</div>

</div>

</body>

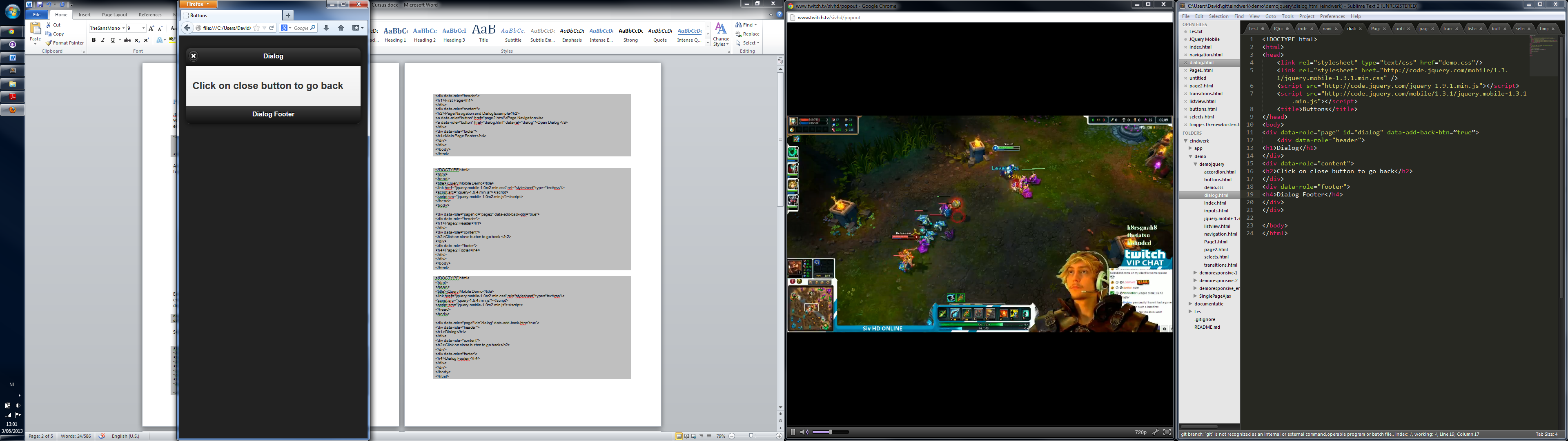
</html>

### Dialogs

Een dialog gaat vanzelf gegenereerd worden als je dezelfde structuur gebruikt. Daarbij heb je een extra attribuut die de mogelijkheid gaat bieden om een back-button toe te voegen in de header van de dialog. Dit doe je door het volgende attribuut toe te voegen:

data-add-back-btn=”true”

data-rel=”dialog”



Screenshot 3 : Dialog Page

<!DOCTYPE html>

<html>

<head>

<title>jQuery Mobile Demo</title>

<link href="jquery.mobile-1.0rc2.min.css" rel="stylesheet" type="text/css"/>

<script src="jquery-1.6.4.min.js"></script>

<script src="jquery.mobile-1.0rc2.min.js"></script>

</head>

<body>

<div data-role="page" id=”dialog" data-add-back-btn=”true”>

<div data-role="header">

<h1>Dialog</h1>

</div>

<div data-role="content">

<h2>Click on close button to go back</h2>

</div>

<div data-role="footer">

<h4>Dialog Footerr</h4>

</div>

</div>

</body>

</html>

## Transitions

Als we de 3 pagina’s van vorig hoofdstuk gaan beschouwen voor navigatie op toe te passen.

Merk op dat het attribuut id=”page2” is ingevuld om de links tussen de pagina’s mogelijk te maken.

In het eerste deel van deze cursus wordt uitgebreid uitgelegd hoe je navigatie best toepast aan de hand van AJAX calls. Hier komen we niet op terug en we gebruiken directe links tussen onze pagina’s.

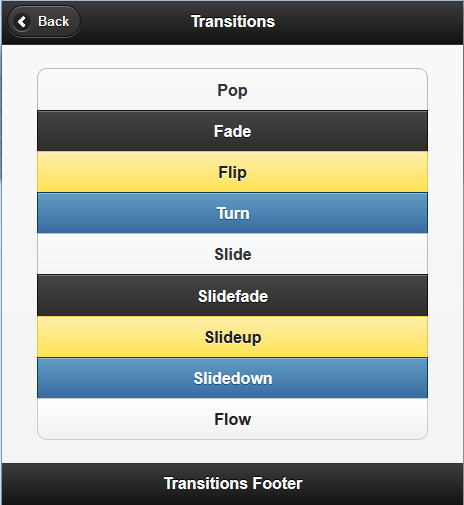
Om hierbij transitities toe te voegen moet het volgende attribuut gebruikt worden:

data-transition=”transition”

Er zijn veel mogelijke transities, hieronder vind u alle voorbeelden

* Pop
* Fade
* Flip
* Turn
* Slide
* Slidefade
* Slideup
* Slidedown
* Flow

Met onderstaand voorbeeld heb je een overzicht van alle verschillende effecten.



<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

</head>

<body>

<div data-role="page" data-url="transitions.html" id="transitions" data-add-back-btn="true">

<div data-role="header">

<h1>Transitions</h1>

</div>

<div data-role="content" id="content">

<div data-role="controlgroup">

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="pop" data-theme="d">Pop</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="fade" data-theme="a">Fade</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="flip" data-theme="e">Flip</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="turn" data-theme="b">Turn</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="slide" data-theme="d">Slide</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="slidefade" data-theme="a">Slidefade</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="slideup" data-theme="e">Slideup</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="slidedown" data-theme="b">Slidedown</a>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="flow">Flow</a>

</div>

</div>

<div data-role="footer">

<h4>Transitions Footer</h4>

</div>

</div>

<!DOCTYPE html>

<html>

<head>

<title>jQuery Mobile Demo</title>

<link href="jquery.mobile-1.0rc2.min.css" rel="stylesheet" type="text/css"/>

<script src="jquery-1.6.4.min.js"></script>

<script src="jquery.mobile-1.0rc2.min.js"></script>

</head>

<body>

<div data-role="page" id=”dialog" data-add-back-btn=”true”>

<div data-role="header">

<h1>Dialog</h1>

</div>

<div data-role="content">

<h2>Click on close button to go back</h2>

</div>

<div data-role="footer">

<h4>Dialog Footerr</h4>

</div>

</div>

</body>

</html>

## Themes

In vorig voorbeeld van transities kan u

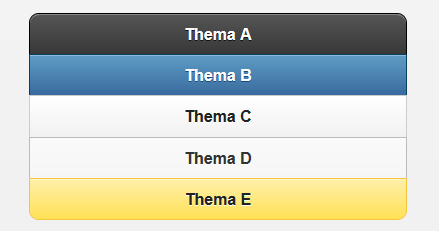
Op elk element van een JQuery mobile widget kan je volgende attributen toevoegen:

data-theme=”thema”

Standaard heb je volgende elementen beschikbaar met de basis JQuery css theme:

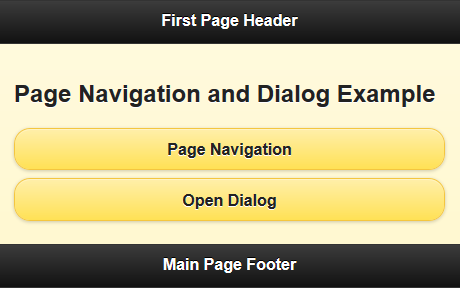
* A
* B
* C
* D
* E

In onderstaande screenshot kan u de verschillende thema’s toegepast zien op buttons.



Dit kan je ook toepassen op een hele page, waardoor de buttons en alle onderliggende elementen vanzelf het thema hebben toegepast.



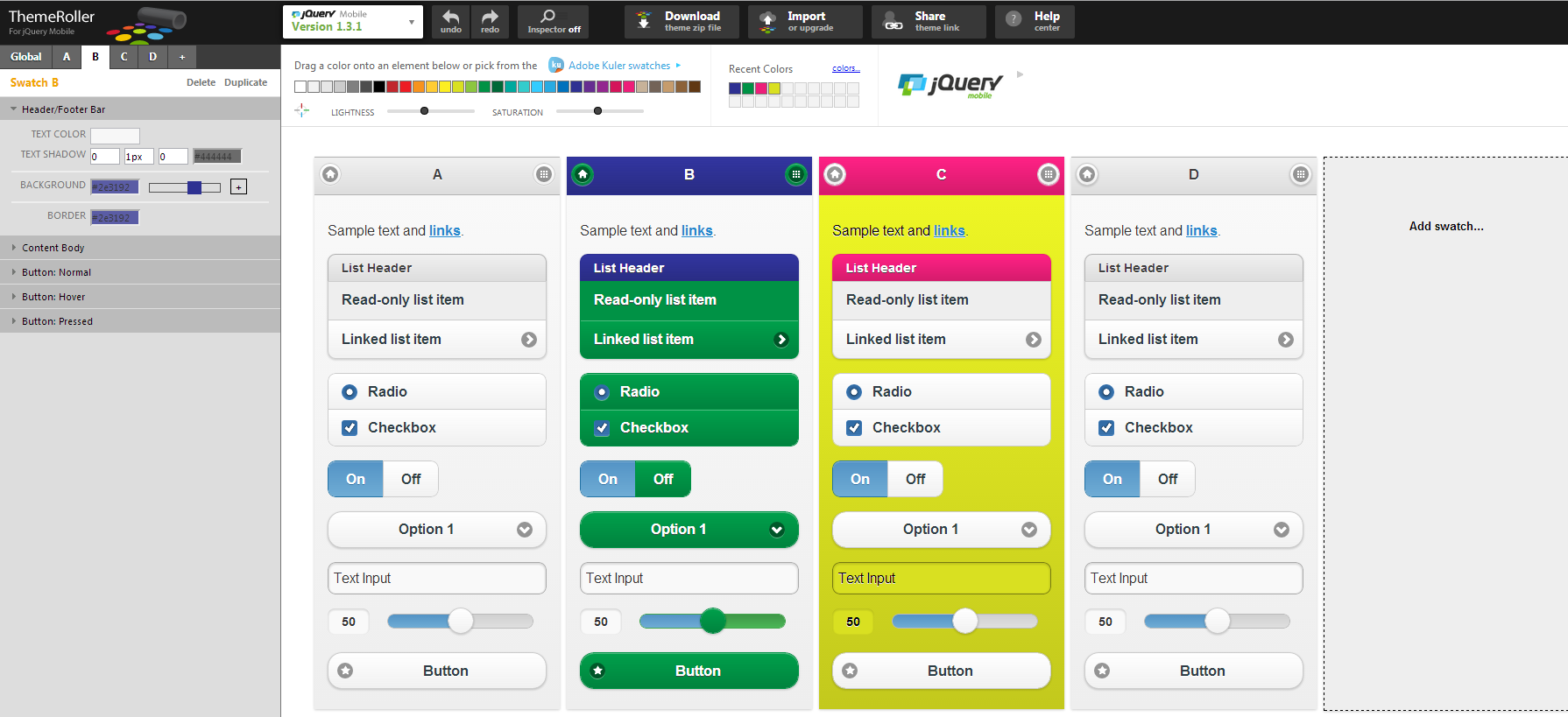


Voor heel makkelijk thema’s toe te voegen of aan te passen kan ik u verwijzen naar de volgende link.

<http://jquerymobile.com/themeroller/>

Deze biedt de mogelijkheid om via drag&drop makkelijk kleuren en thema’s aan te passen.

Je kan heir zelf het aantal ‘swatches’ kiezen dat je beschikbaar hebt, en elk daarvan kan je personaliseren.



Eens je je gepersonaliseerd thema gemaakt heb, kan je het heel eenvouding downloaden en integreren in je applicatie door de verwijzing van je css bestanden aan te passen.

## Buttons

Het grootste deel van de buttons verkrijg je door een <a> element te nemen en daarbij het data-role element toe te voegen:

data-role=”button”

Er zijn veel verschillende mogelijk attributen toe te voegen aan een button, en hieronder overlopen we deze.

### Thema en Stijl

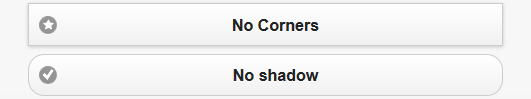
data-theme=”thema”

Uitleg rond het data-theme attribuut kan je vinden in het vorige hoofdstuk Thema’s.

data-corners=”false”

data-shadow=”false”

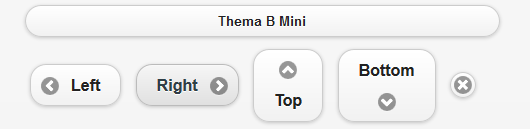
Via het element data-corners kan je buttons bekomen zonder ronde hoeken, en het attribuut data-shadows kan je gebruiken voor schaduwen te verwijderen of toe te voegen.



Je kan ook alle buttons als mini aanmaken of ze inline ingeven. Dit zorgt ervoor dat ofwel de buttons kleiner zijn maar nog steeds de volle breedte van de beschikbare ruimte gaan gebruiken. Het inline element gaat de knop verkleinen en ervoor zorgen dat het nog maar enkel de ruimte inneemt dat het nodig heeft.

data-inline=”true”

data-mini=”true”



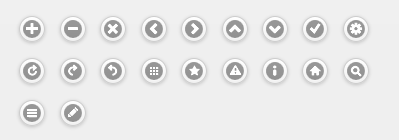
### Icons

data-icon=”plus”

Dit attribuut biedt de mogelijkheid om aan een button een icoon toe te voegen. Je kan een aantal voor gedefinieerde iconen gebruiken die beschikbaar zijn via de css bestanden van JQuery, maar je kan ook je eigen iconen hieraan toevoegen.

Hieronder vind je een overzicht van alle iconen:

* Plus
* Minus
* Delete
* Arrow-l
* Arrow-r
* Arrow-u
* Arrow-d
* Check
* Gear
* Refresh
* Forward
* Back
* Grid
* Star
* Alert
* Info
* Home
* Search
* Bars
* Edit

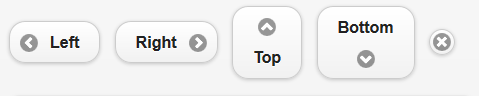


Men kan ook de positie van de icons kiezen door volgend attribuut aan te spreken:

data-iconpos="left"

Er zijn volgende posities mogelijk voor de iconen in de buttons:

* Left
* Right
* Top
* Bottom
* Notext



### Disabled

Je kan een button of een element uit een form op 2 verschillende manieren disablen.

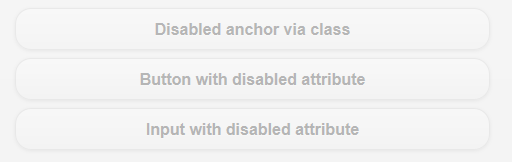
De eerste manier is via de class.

class="ui-disabled"

Met deze classe toe te voegen aan het element dat je wil disablen.

Je kan het ook doen via het volgende attribuut toe te voegen

disabled=””



### Controlgroup

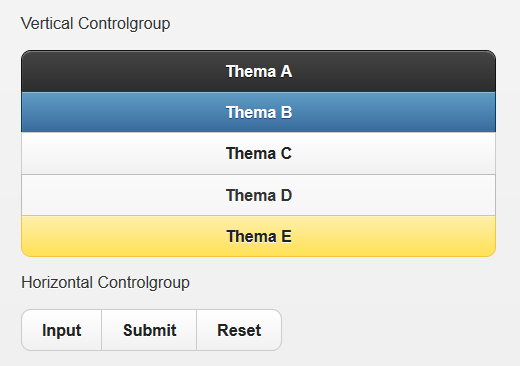
Een controlgroup kan gebruikt worden voor het groeperen van verschillende buttons.

Er zijn 2 soorten controlgroups, namelijk horizontale en verticale. Beiden hebben dezelfde functie, enkel de layout is verschillend.

<div data-role="controlgroup" data-type="horizontal"></div

<div data-role="controlgroup" data-type="vertical"></div

Binnen de controlgroup kan je buttons of inputs van een form groeperen.



### Voorbeeld

In onderstaand voorbeeld zijn alle knoppen, iconen, disables,.. samengevoegd in 1 voorbeeld.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

</head>

<body>

<div data-role="page" data-url="buttons.html">

<div data-role="header">

<h1>Buttons</h1>

</div>

<div data-role="content" id="content"><h2> Knoppen </h2>

<a href="dialog.html" data-role="button" data-rel="dialog" data-transition="pop">Button</a>

<a href="#" data-role="button" data-icon="star" data-corners="false">No Corners</a>

<a href="#" data-role="button" data-icon="check" data-shadow="false">No shadow</a>

<a href="#" data-role="button" data-mini="true">Data-Mini</a>

<a href="#" data-role="button" data-icon="arrow-l" data-iconpos="left" data-inline="true">Left</a>

<a href="#" data-role="button" data-icon="arrow-r" data-iconpos="right" data-inline="true">Right</a>

<a href="#" data-role="button" data-icon="arrow-u" data-iconpos="top" data-inline="true">Top</a>

<a href="#" data-role="button" data-icon="arrow-d" data-iconpos="bottom" data-inline="true">Bottom</a>

<a href="#" data-role="button" data-icon="delete" data-iconpos="notext" data-inline="true">Icon only</a>

<a href="#" data-role="button" class="ui-disabled">Disabled anchor via class</a>

<form>

<button disabled="">Button with disabled attribute</button>

<input type="button" value="Input with disabled attribute" disabled="">

</form>

<h2> Controlgroup </h2>

<p> Vertical Controlgroup </p>

<div data-role="controlgroup" data-type="vertical">

<a href="#" data-role="button" data-theme="a" >Thema A</a>

<a href="#" data-role="button" data-theme="b" >Thema B</a>

<a href="#" data-role="button" data-theme="c" >Thema C</a>

<a href="#" data-role="button" data-theme="d" >Thema D</a>

<a href="#" data-role="button" data-theme="e" >Thema E</a>

</div>

<p>Horizontal Controlgroup</p>

<form>

<div data-role="controlgroup" data-type="horizontal">

<input type="button" value="Input">

<input type="submit" value="Submit">

<input type="reset" value="Reset">

</div>

</form>

<h2> Icons </h2>

<a href="#" data-role="button" data-icon="plus" data-iconpos="notext" data-theme="c" data-inline="true">Plus</a>

<a href="#" data-role="button" data-icon="minus" data-iconpos="notext" data-theme="c" data-inline="true">Minus</a>

<a href="#" data-role="button" data-icon="delete" data-iconpos="notext" data-theme="c" data-inline="true">Delete</a>

<a href="#" data-role="button" data-icon="arrow-l" data-iconpos="notext" data-theme="c" data-inline="true">Arrow left</a>

<a href="#" data-role="button" data-icon="arrow-r" data-iconpos="notext" data-theme="c" data-inline="true">Arrow right</a>

<a href="#" data-role="button" data-icon="arrow-u" data-iconpos="notext" data-theme="c" data-inline="true">Arrow up</a>

<a href="#" data-role="button" data-icon="arrow-d" data-iconpos="notext" data-theme="c" data-inline="true">Arrow down</a>

<a href="#" data-role="button" data-icon="check" data-iconpos="notext" data-theme="c" data-inline="true">Check</a>

<a href="#" data-role="button" data-icon="gear" data-iconpos="notext" data-theme="c" data-inline="true">Gear</a>

<a href="#" data-role="button" data-icon="refresh" data-iconpos="notext" data-theme="c" data-inline="true">Refresh</a>

<a href="#" data-role="button" data-icon="forward" data-iconpos="notext" data-theme="c" data-inline="true">Forward</a>

<a href="#" data-role="button" data-icon="back" data-iconpos="notext" data-theme="c" data-inline="true">Back</a>

<a href="#" data-role="button" data-icon="grid" data-iconpos="notext" data-theme="c" data-inline="true">Grid</a>

<a href="#" data-role="button" data-icon="star" data-iconpos="notext" data-theme="c" data-inline="true">Star</a>

<a href="#" data-role="button" data-icon="alert" data-iconpos="notext" data-theme="c" data-inline="true">Alert</a>

<a href="#" data-role="button" data-icon="info" data-iconpos="notext" data-theme="c" data-inline="true">Info</a>

<a href="#" data-role="button" data-icon="home" data-iconpos="notext" data-theme="c" data-inline="true">Home</a>

<a href="#" data-role="button" data-icon="search" data-iconpos="notext" data-theme="c" data-inline="true">Search</a>

<a href="#" data-role="button" data-icon="bars" data-iconpos="notext" data-theme="c" data-inline="true">Bars</a>

<a href="#" data-role="button" data-icon="edit" data-iconpos="notext" data-theme="c" data-inline="true">Edit</a>

</div>

<div data-role="footer">

<p>footer</p>

</div>

</div>

</body>

</html>

## Navigation Bar

De navbar wordt gebruikt door volgend attribuut toe te voegen aan het header of footer element

<div data-role=”navbar”>

<ul>

<li>…</li>

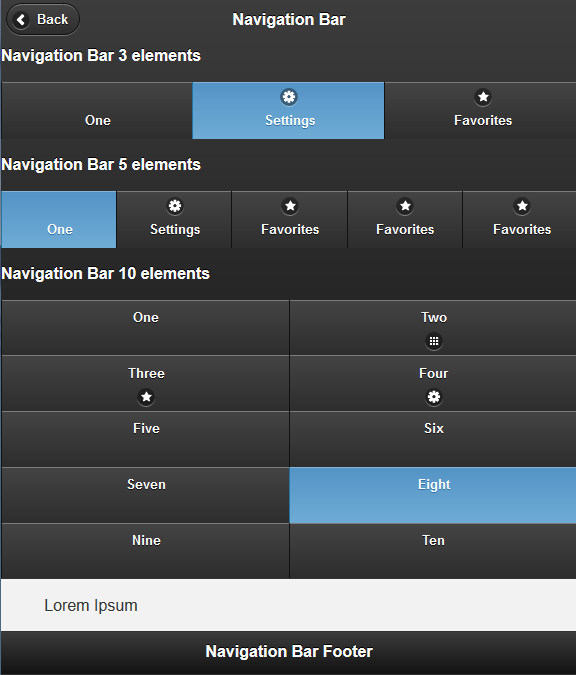
<li>…</li>

<li>…</li>

</ul>

</div>

Afhankelijk van het aantal elementen gaat de navigatiebalk zichzelf opsplitsen. Tot en met 5 elementen gaat JQuery Mobile de ruimte vanzelf evenredig opsplitisen tussen de lijstitems. Eens boven de 5 elementen gaat de UI dit opsplitsen in 2 rijen en krijg je volgend resultaat



De actieve knop aanduiden kan door de volgende class toe te voegen aan het element:

class="ui-btn-active"

Men kan in het voorbeeld zien dat de iconen van de buttons verschillen, alsook hun positie. Voor dit te doen verwijs ik jullie naar een vorig hoofdstuk (Buttons).

Een navigatie bar kan ook persistent worden opgesteld. Dit wil zeggen dat wanneer er gescrolled wordt, de navigatie bar steeds bovenaan het scherm blijft, en de rest van de pagina eronder schuift. Dit is mogelijk door het volgende attribuut toe te voegen aan de header of footer:

data-position="fixed"



<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Navigation</title>

</head>

<body>

<div data-role="page" id="navigation" data-add-back-btn="true">

<div data-role="header" data-id="foo1" data-position="fixed">

<h1>Navigation Bar</h1>

<p> Navigation Bar 3 elements</p>

<div data-role="navbar" data-theme="b">

<ul>

<li><a href="#" >One</a></li>

<li><a href="#" data-icon="gear" class="ui-btn-active">Settings</a></li>

<li><a href="#" data-icon="star">Favorites</a></li>

</ul>

</div><!-- /navbar -->

<p> Navigation Bar 5 elements</p>

<div data-role="navbar">

<ul>

<li><a href="#" class="ui-btn-active">One</a></li>

<li><a href="#" data-icon="gear">Settings</a></li>

<li><a href="#" data-icon="star">Favorites</a></li>

<li><a href="#" data-icon="star">Favorites</a></li>

<li><a href="#" data-icon="star">Favorites</a></li>

</ul>

</div><!-- /navbar -->

<p> Navigation Bar 10 elements</p>

<div data-role="navbar" data-iconpos="bottom">

<ul>

<li><a href="#" >One</a></li>

<li><a href="#" data-icon="grid">Two</a></li>

<li><a href="#" data-icon="star">Three</a></li>

<li><a href="#" data-icon="gear">Four</a></li>

<li><a href="#">Five</a></li>

<li><a href="#">Six</a></li>

<li><a href="#">Seven</a></li>

<li><a href="#" class="ui-btn-active">Eight</a></li>

<li><a href="#">Nine</a></li>

<li><a href="#">Ten</a></li>

</ul>

</div><!-- /navbar -->

</div>

<div id="content">

<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui.

Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc, quis gravida magna mi a libero. Fusce vulputate eleifend sapien. Vestibulum purus quam, scelerisque ut, mollis sed, nonummy id, metus. Nullam accumsan lorem in dui. Cras ultricies mi eu turpis hendrerit fringilla. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; In ac dui quis mi consectetuer lacinia.</p>

</div>

<div data-role="footer">

<h4>Navigation Bar Footer</h4>

</div>

</div>

</body>

</html>

## Listview en Accodrion

### Listview

Een listview kan je gebruiken voor een lijst van elementen weer te geven. Dit bereik je door aan het <ul> element een extra attribuut mee te geven:

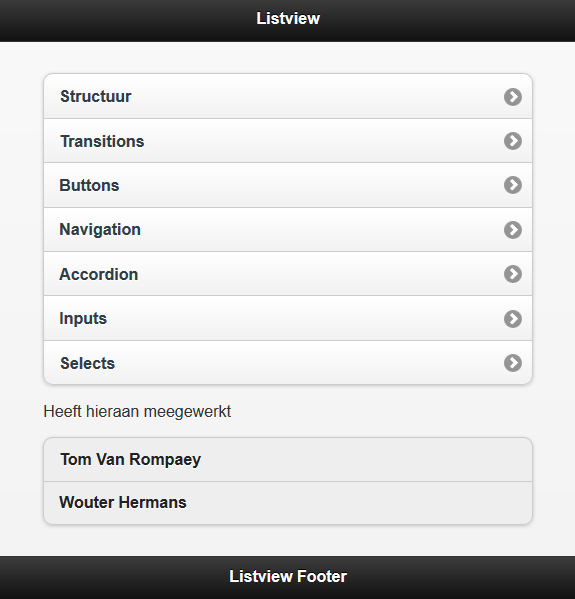
<ul data-role=”listview” data-inset=”true”>

<lI>…</li>

<li>…</li>

</ul>

Je kan ook een gelinkte listview bekomen door binnen het <li> element een link te zetten.



<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

</head>

<body>

<div data-role="page" id="listview">

<div data-role="header">

<h1>Listview</h1>

</div>

<div data-role="content" id="content">

<ul data-role="listview" data-inset="true">

<li><a href="page1.html">Structuur</a></li>

<li><a href="transitions.html">Transitions</a></li>

<li><a href="buttons.html">Buttons</a></li>

<li><a href="navigation.html">Navigation</a></li>

<li><a href="accordion.html">Accordion</a></li>

<li><a href="inputs.html">Inputs</a></li>

<li><a href="selects.html">Selects</a></li>

</ul>

<p> Heeft hieraan meegewerkt </p>

<ul data-role="listview" data-inset="true">

<li> Tom Van Rompaey </li>

<li> Wouter Hermans </li>

</div>

<div data-role="footer">

<h4>Listview Footer</h4>

</div>

</div>

</body>

</html>

### Accordion

De accordion is een gelijkaardige widget. Met volgende code bekom je een collapsible:

<div data-role="collapsible-set" >

<div data-role="collapsible">

<h3>…</h3>

<p>….</p>

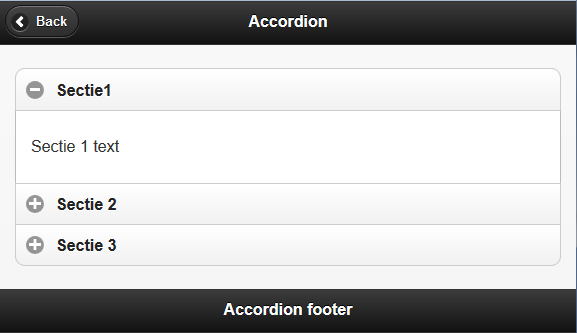
</div>

<div data-role="collapsible">

…..

</div>

Het h3 element gaat getoond worden als de collapsible niet open staat. Eens het opengeklikt wordt toont hij de <p> elementen erbinnen.



<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Accordion</title>

</head>

<body>

<div data-role="page" id="accordion" data-add-back-btn="true">

<div data-role="header">

<h1> Accordion </h1>

</div>

<div data-role="content">

<div data-role="collapsible-set" data-theme="c" data-content-theme="d">

<div data-role="collapsible">

<h3>Sectie1</h3>

<p>Sectie 1 text</p>

</div>

<div data-role="collapsible">

<h3>Sectie 2</h3>

<p>Sectie 2 text</p>

</div>

<div data-role="collapsible">

<h3>Sectie 3</h3>

<p>Sectie 3 text</p>

</div>

</div>

</div>

<div data-role="footer">

<h4>Accordion footer</h4>

</div>

</div>

</body>

</html>

## Forms

### Inputs

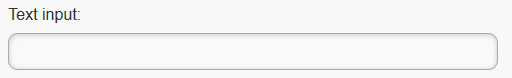
Er zijn veel verschillende inputvelden beschikbaar in JQuery Mobile.

De eerste dat we gaan bekijken is het basis tekst input veld.

<form>

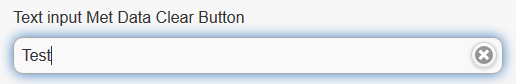
<input type=”text”>

</form>



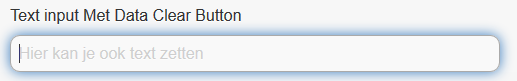
Je kan bij dit inputveld ook het data-clear-btn attribuut toevoegen welke een knop beschikbaar maakt om het veld leeg te maken eens er tekst in staat.

data-clear-btn=”true”



Een uitgegrijsde boodschap toevoegen dat verdwijnt eens je het veld selecteerd kan met het placeholder attribuut.

placeholder=”Zet hier je text”

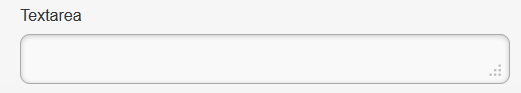


Voor tekst in te geven is e rook een groter veld beschikbaar welke de textarea is. De textarea gebruikt de attributen cols en rows voor de grootte van het veld te bepalen.

<form>

<textarea cols=”40” rows=”8” ></textarea>

</form>

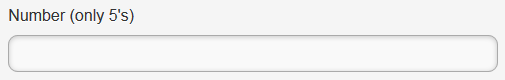


Voor getallen zijn er ook verschillende velden beschikbaar. Het eerste is een normaal veld van type number.

<form>

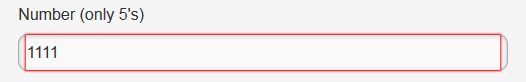
<input type=”text”>

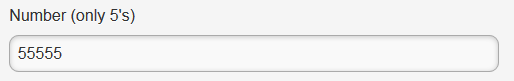
</form>



7Je kan ook een soort van autocorrectie creëren door patronen toe te voegen op de velden. Dit doe je door het pattern attribuut toe te voegen. Enkel en alleen een ingevoegde waarde dat voldoet aan de REGEX expressie binnen het pattern gedeelte wordt aanvaard.

pattern=”getal”



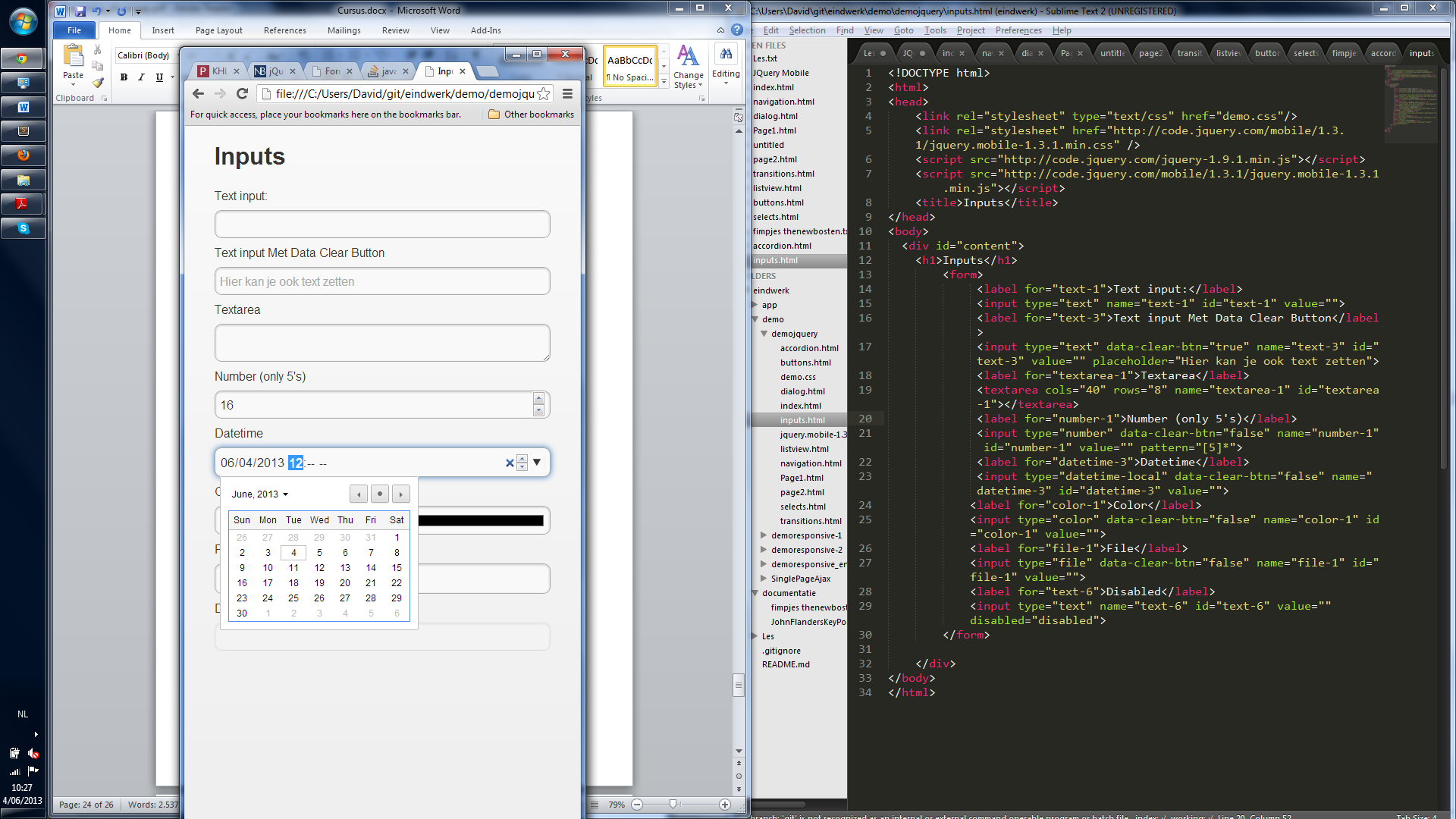


Vervolgens heb je datums en tijd ingavevelden. Hiervoor zijn er een aantal beschikbaar maar de meest complete is de volgende:

<form>

<input for=”datetime-local”>

</form>



Ingavevelden voor kleuren en files bekom je door volgende codes toe te passen

<form>

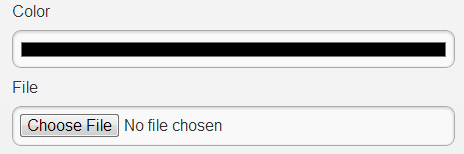
<input for=”color”>

</form>

<form>

<input for=”file”>

</form>



Het gehele voorbeeld vind u hieronder

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Inputs</title>

</head>

<body>

<div id="content">

<h1>Inputs</h1>

<form>

<label for="text-1">Text input:</label>

<input type="text" name="text-1" id="text-1" value="">

<label for="text-3">Text input Met Data Clear Button</label>

<input type="text" data-clear-btn="true" name="text-3" id="text-3" value="" placeholder="Hier kan je ook text zetten">

<label for="textarea-1">Textarea</label>

<textarea cols="40" rows="8" name="textarea-1" id="textarea-1"></textarea>

<label for="number-1">Number (only 5's)</label>

<input type="number" data-clear-btn="false" name="number-1" id="number-1" value="" pattern="[5]\*">

<label for="datetime-3">Datetime</label>

<input type="datetime-local" data-clear-btn="false" name="datetime-3" id="datetime-3" value="">

<label for="color-1">Color</label>

<input type="color" data-clear-btn="false" name="color-1" id="color-1" value="">

<label for="file-1">File</label>

<input type="file" data-clear-btn="false" name="file-1" id="file-1" value="">

<label for="text-6">Disabled</label>

<input type="text" name="text-6" id="text-6" value="" disabled="disabled">

</form>

</div>

</body>

</html>

### Selects

Je kan een aantal verschillende elementen toevoegen aan selects maar eerst zullen we een basis select maken.

<form>

<div data-role="fieldcontain">

<label for="select-native-1">Basic:</label>

<select name="select-native-1" id="select-native-1">

<option value="1">The 1st Option</option>

<option value="2">The 2nd Option</option>

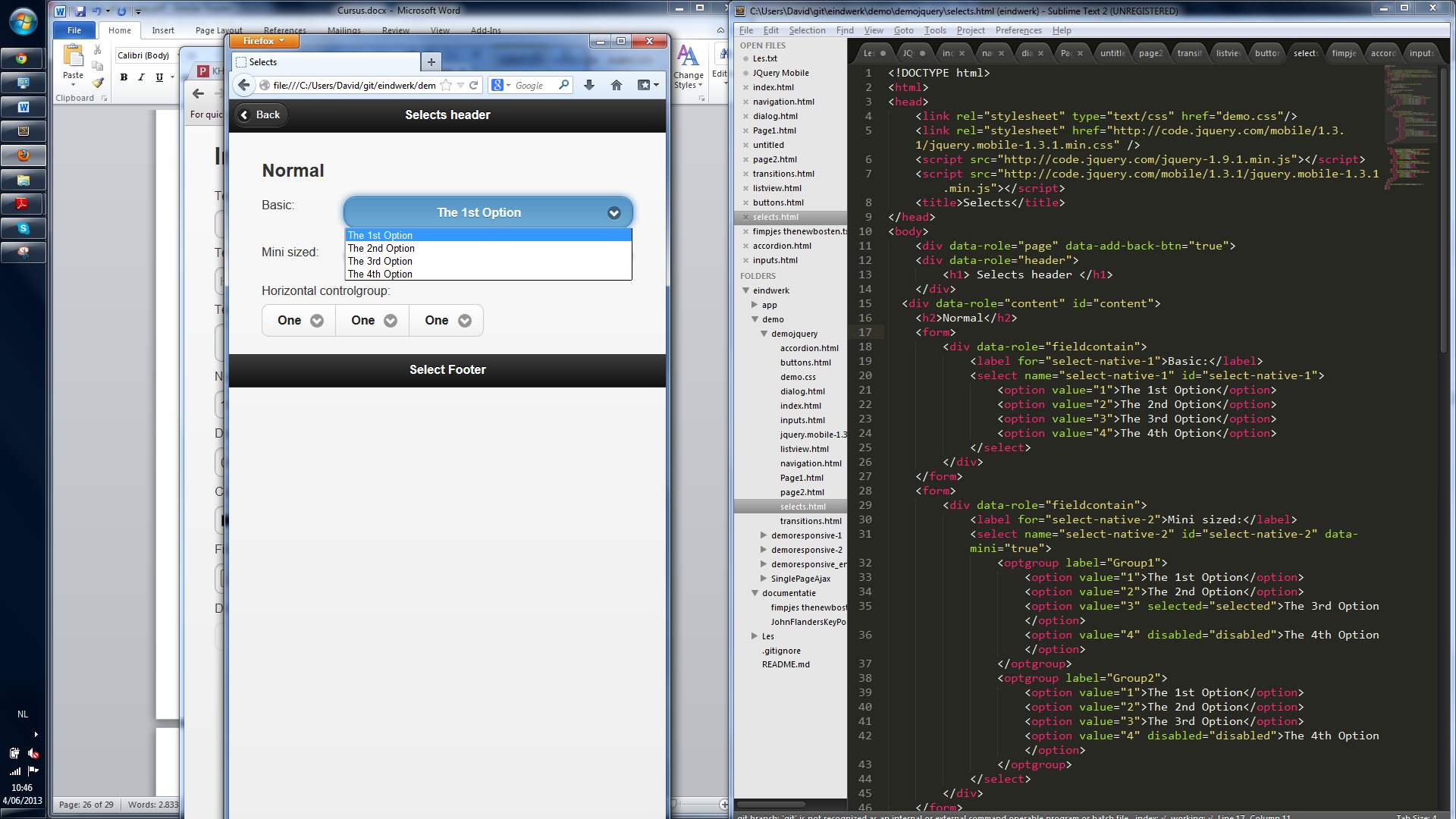
<option value="3">The 3rd Option</option>

<option value="4">The 4th Option</option>

</select>

</div>

</form>



Merk op dat we een div toevoegen binnen de form met fieldcontain voor het attribuut data-role. Vervolgens maken we een select aan met verschillende opties. We kunnen ook deze opties groeperen in optgroups. Deze optgroups gaan een logische en visuele structuur steken in de verschillende options beschikbaar in de select.

<form>

<div data-role="fieldcontain">

<label for="select-native-2">Mini sized:</label>

<select name="select-native-2" id="select-native-2">

<optgroup label="Group1">

<option value="1">The 1st Option</option>

<option value="2">The 2nd Option</option>

<option value="3" selected="selected">The 3rd Option</option>

<option value="4" disabled="disabled">The 4th Option</option>

</optgroup>

<optgroup label="Group2">

<option value="1">The 1st Option</option>

<option value="2">The 2nd Option</option>

<option value="3">The 3rd Option</option>

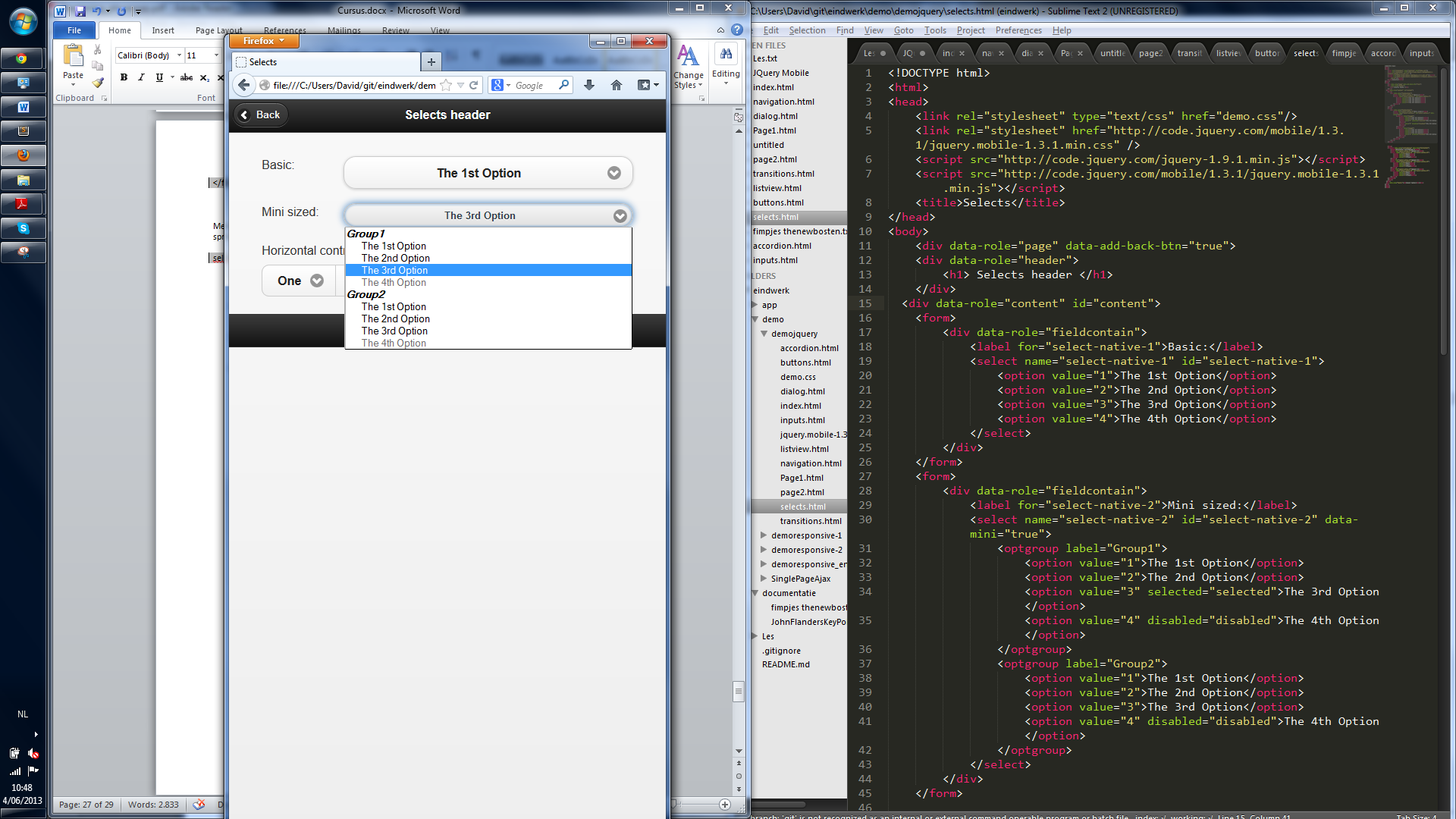
<option value="4" disabled="disabled">The 4th Option</option>

</optgroup>

</select>

</div>

</form>



Men kan ook een optie al op voorhand geselecteerd zetten door het selected attribuut aan te spreken.

selected=”selected”

Een global voorbeeld vind je hieronder.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Selects</title>

</head>

<body>

<div data-role="page" data-add-back-btn="true">

<div data-role="header">

<h1> Selects header </h1>

</div>

<div data-role="content" id="content">

<form>

<div data-role="fieldcontain">

<label for="select-native-1">Basic:</label>

<select name="select-native-1" id="select-native-1">

<option value="1">The 1st Option</option>

<option value="2">The 2nd Option</option>

<option value="3">The 3rd Option</option>

<option value="4">The 4th Option</option>

</select>

</div>

</form>

<form>

<div data-role="fieldcontain">

<label for="select-native-2">Mini sized:</label>

<select name="select-native-2" id="select-native-2" data-mini="true">

<optgroup label="Group1">

<option value="1">The 1st Option</option>

<option value="2">The 2nd Option</option>

<option value="3" selected="selected">The 3rd Option</option>

<option value="4" disabled="disabled">The 4th Option</option>

</optgroup>

<optgroup label="Group2">

<option value="1">The 1st Option</option>

<option value="2">The 2nd Option</option>

<option value="3">The 3rd Option</option>

<option value="4" disabled="disabled">The 4th Option</option>

</optgroup>

</select>

</div>

</form>

<form>

<fieldset data-role="controlgroup" data-type="horizontal">

<legend>Horizontal controlgroup:</legend>

<label for="select-native-11">Select A</label>

<select name="select-native-11" id="select-native-11">

<option value="#">One</option>

<option value="#">Two</option>

<option value="#">Three</option>

</select>

<label for="select-native-12">Select B</label>

<select name="select-native-12" id="select-native-12">

<option value="#">One</option>

<option value="#">Two</option>

<option value="#">Three</option>

</select>

<label for="select-native-13">Select C</label>

<select name="select-native-13" id="select-native-13">

<option value="#">One</option>

<option value="#">Two</option>

<option value="#">Three</option>

</select>

</fieldset>

</form>

</div>

<div data-role="footer"><h4>Select Footer</h4></div>

</div>

</body>

</html>

### Sliders

Een slider wordt gemaakt met volgende code:

<form>

<input type="range" name="slider" id="slider" value="40" min="0" max="100" data-highlighted=”true>

</form>

Je kan ook een dubbele slider maken, maar daarvoor moet je een div toevoegen met volgende attributen. Merk op dat er 2 inputs zijn van type range, elk met dezelfde min en max waarden enkel een andere value. Dit zorgt ervoor dat je een dubbele slider krijgt zoals hieronder.

<form>

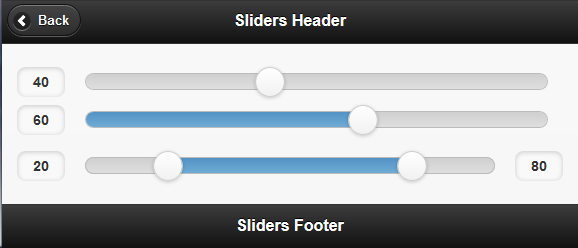
<div data-role="rangeslider">

<input type="range" min="0" max="100" value="20">

<input type="range" min="0" max="100" value="80">

</div>

</form>



Het gehele voorbeeld vindt u hieronder.

<!DOCTYPE html>

<html>

<head>

<title>jQuery Mobile Demo</title>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

</head>

<body>

<div data-role="page" id="main" data-add-back-btn="true">

<div data-role="header">

<h1>Sliders Header</h1>

</div>

<div data-role="content">

<form>

<input type="range" name="slider" id="slider" value="40" min="0" max="100">

<input type="range" name="slider" id="slider" value="60" min="0" max="100" data-highlight="true">

</form>

<form>

<div data-role="rangeslider">

<input type="range" min="0" max="100" value="20">

<input type="range" min="0" max="100" value="80">

</div>

</form>

</div>

<div data-role="footer">

<h4>Sliders Footer</h4>

</div>

</div>

</body>

</html>

### Checkboxes & Radiobuttons

Radiobuttons en checkboxes zijn vrijwel hetzelfde als in simpele html, daarvoor ga ik enkel de code en het voorbeeld tonen aangezien de code voor zichzelf spreekt.

<!DOCTYPE html>

<html>

<head>

<title>jQuery Mobile Demo</title>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

</head>

<body>

<div data-role="page" id="main" data-add-back-btn="true">

<div data-role="header">

<h1>Checkboxes and Radiobuttons header</h1>

</div>

<div data-role="content">

<fieldset data-role="controlgroup">

<legend>Checkboxes</legend>

<input type="checkbox" name="checkbox-1a" id="checkbox-1a" checked="">

<label for="checkbox-1a">Check 1</label>

<input type="checkbox" name="checkbox-2a" id="checkbox-2a">

<label for="checkbox-2a">Check 2</label>

<input type="checkbox" name="checkbox-3a" id="checkbox-3a">

<label for="checkbox-3a">Check 3</label>

<input type="checkbox" name="checkbox-4a" id="checkbox-4a">

<label for="checkbox-4a">Check 4</label>

</fieldset>

<fieldset data-role="controlgroup">

<legend>Radio buttons, vertical controlgroup:</legend>

<input type="radio" name="radio-choice-1" id="radio-choice-1" value="choice-1" checked="checked">

<label for="radio-choice-1">Radio 1</label>

<input type="radio" name="radio-choice-1" id="radio-choice-2" value="choice-2">

<label for="radio-choice-2">Radio 2</label>

<input type="radio" name="radio-choice-1" id="radio-choice-3" value="choice-3">

<label for="radio-choice-3">Radio 3</label>

<input type="radio" name="radio-choice-1" id="radio-choice-4" value="choice-4">

<label for="radio-choice-4">Radio 4</label>

</fieldset>

<p>Flipswitch</p>

<select name="slider2" id="slider2" data-role="slider">

<option value="off">Off</option>

<option value="on">On</option>

</select> </div>

<div data-role="footer">

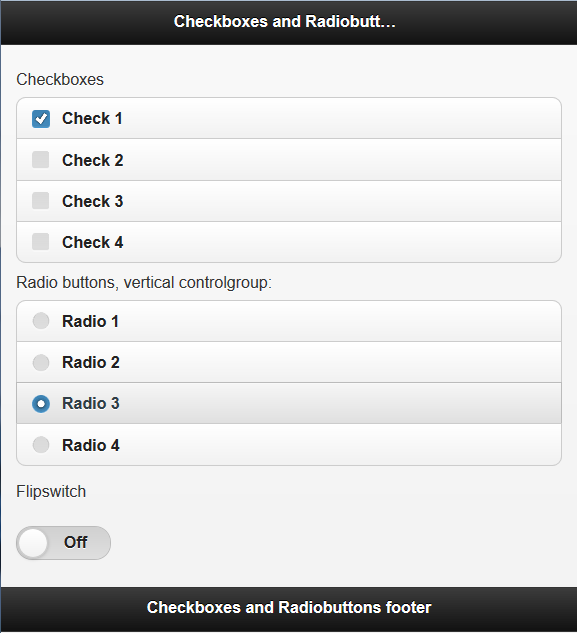
<h4>Checkboxes and Radiobuttons footer</h4>

</div>

</div>

</body>

</html>



### Retrieve Information using JQuery

JQuery vergemakkelijkt het aanspreken van elementen in de HTML. Je kan op verschillende manieren elementen gaan selecteren.

De eerste is via het element. Dit voorbeeld selecteert alle p elementen binnen de html pagina.

$(“p”)

Ook kan je selecteren via de ID’s van de elementen. Let op het # voor de id.

$(“#id”)

Men kan ook selecteren op basis van de CSS classes. Als men bijvoorbeeld een .blauw classe heeft die men op verschillende elementen van verschillende soorten toepast, kan men via de CSS-based selector toch aan al deze elementen geraken.

$(“.blauw”)

Een combinatie gebruiken van deze verschillende selectoren is mogelijk. Het onderstaande voorbeeld gaat alle elementen <p class=”.blauw”> selecteren.

$(“p.blauw”)

## Events

### Events Basic

Je om het even welke functie aan om het even welke event binden met de volgende functie.

$(window).bind(“element”, function());

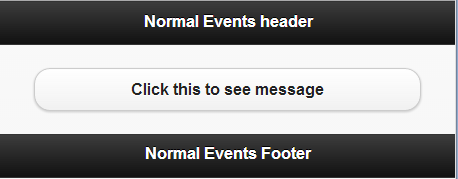
Dit biedt je de mogelijkheid om een pagina te creeeren dat interacteif omgaat met verschillende events. Best ga je bij het laden van de pagina al je functies gaan binden aan de overeenkomstige events. Dit kan je doen door volgende code te gebruiken

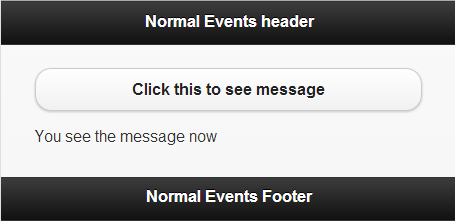
$(document).ready(function(){

//bindt functies aan events

});

In onderstaand voorbeeld kan je zien hoe eerst er geen extra bericht staat, en na het klikken de div met id message wel content heeft.





<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Normal Events</title>

<script>

$(document).ready(function(){

$("#mybutton").bind("click",function(event){

$("#message").html("<p>You see the message now</p>");

});

});

</script>

</head>

<body>

<div data-role="page" data-add-back-btn="true">

<div data-role="header">

<h1> Normal Events header </h1>

</div>

<div data-role="content" id="content">

<button id="mybutton">Click this to see message</button>

<div id="message"></div>

</div>

<div data-role="footer"><h4>Normal Events Footer</h4></div>

</div>

</body>

</html>

### Orientation Events

JQuery biedt ook de mogelijkheid om een de orientatie van het scherm te detecteren. Dit kan heel handig zijn als er verschillende css moet ingeladen worden, of bepaalde functies nodig zijn bij een breder scherrm.

Je kan je binden aan de verandering van de orientatie aan volgende code:

$(document).ready(function(){

$(window).bind("orientationchange", function(event){

//Doe iets

});

});

Onderstaand voorbeeld gaat een bericht aanpassen in de div message.

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Orientation Events</title>

<script>

$(document).ready(function(){

$(window).bind("orientationchange", function(event){

$("#message").html("Orientation changed to "+event.orientation);

});

});

</script>

</head>

<body>

<div data-role="page" data-add-back-btn="true">

<div data-role="header">

<h1> Orientation Events header </h1>

</div>

<div data-role="content" id="content">

<h2>Orientation Test</h2>

<div id="message"></div>

</div>

<div data-role="footer"><h4>Orientation Events Footer</h4></div>

</div>

<script>

$("#mybutton").bind("click",function(event){

$("#message").html("<p>You see the message now</p>");

});

</script>

</body>

</html>

### Scroll Events

Op dezelfde manier kan je je inschrijven op het scrolling event. Er zijn hier 2 events beschikbaar namelijk scrollstart en scrollend. Scrollstart wordt niet overal ondersteunt, dus ben je best met scrollend te gebruiken.

$(document).ready(function(){

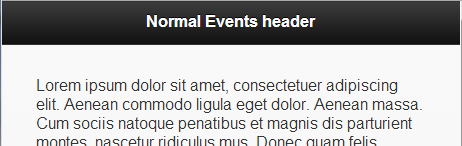
$(window).bind("scrollstart", function(event){

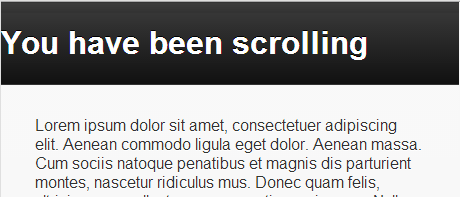
$("#header").html("<h1>You have been scrolling</h1>");

});

});

In onderstaan voorbeeld zie je de header veranderen nadat je gescrolled hebt.





<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Normal Events</title>

<script>

$(document).ready(function(){

$(window).bind("scrollstart", function(event){

$("#header").html("<h1>You have been scrolling</h1>");

});

});

</script>

</head>

<body>

<div data-role="page" data-add-back-btn="true">

<div data-role="header" id="header">

<h1> Normal Events header </h1>

</div>

<div data-role="content" id="content">

<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. </p>

</div>

<div data-role="footer"><h4>Normal Events Footer</h4></div>

</div>

<script>

$("#mybutton").bind("click",function(event){

$("#message").html("<p>You see the message now</p>");

});

</script>

</body>

</html>

### Touch Events

Touch events zijn waarschijnlijk de meest belangrijke events voor JQuery mobile. Niet alle touch events worden ondersteund, enkel een basis aantal. Voor alle touch events, of zekere een aantal complexere te hebben stel ik volgend framwork voor: <http://eightmedia.github.io/hammer.js/>.

Dit is hammer.js en zit vrij goed in elkaar.

Wij gaan over de basis touch events gaan. Je hebt volgende events:

* Tap: het aanraken van het scherm, vergelijkbaar met clicken.
* Tap and Hold: het aanraken van het scherm en blijven houden, vergelijkbaar met clicken en muis ingedrukt houden
* Swipe: met je vinger over het scherm glijden, vergelijkbaar met een slectie van tekst te maken
* Swipe Left: swipe maar van rechts naar links
* Swipe Right: swipe van links naar rechts.

Je inschrijven op deze events gaat als volgt:

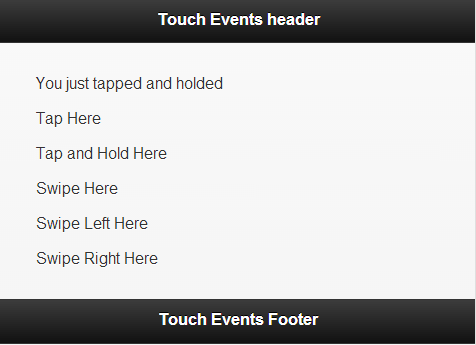
$(document).ready(function(){

$("#tap").bind("tap", function(){

$("#message").html("<p>You just tapped</p>");

});

Hier heb ik me bijvoorbeeld ingeschreven op het tap event.



Het volledige voorbeeld vindt je hieronder:

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="demo.css"/>

<link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.css" />

<script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>

<script src="http://code.jquery.com/mobile/1.3.1/jquery.mobile-1.3.1.min.js"></script>

<title>Touch Events</title>

<script>

$(document).ready(function(){

$("#tap").bind("tap", function(){

$("#message").html("<p>You just tapped</p>");

});

$("#taphold").bind("taphold", function(){

$("#message").html("<p>You just tapped and holded</p>");

});

$("#swipe").bind("swipe", function(){

$("#message").html("<p>You just swiped</p>");

});

$("#swipeleft").bind("swipeleft", function(){

$("#message").html("<p>You just swiped left</p>");

});

$("#swiperight").bind("swiperight", function(){

$("#message").html("<p>You just swiped right</p>");

});

});

</script>

</head>

<body>

<div data-role="page" data-add-back-btn="true">

<div data-role="header">

<h1> Touch Events header </h1>

</div>

<div data-role="content" id="content">

<div id="message"></div>

<p id="tap"> Tap Here </p>

<p id="taphold"> Tap and Hold Here </p>

<p id="swipe"> Swipe Here </p>

<p id="swipeleft"> Swipe Left Here </p>

<p id="swiperight"> Swipe Right Here </p>

</div>

<div data-role="footer"><h4>Touch Events Footer</h4></div>

</div>

</body>

</html>

### JQuery Mouse Events

JQuery mobile heeft ook specifieke virtuele mouse events, om deze te differentieren van de touch events.

Aan de basis is het inschrijven op deze events en het gebruikt ervan hetzelfde. Daarom lijst ik hier de verschillende virtuele muis events op:

* Vmousemove
* Vmouseover
* Vmousedown
* Vmouseup
* Vmouseclick
* Vmousecancel

### Page Initialization Events

Er zijn ook aantal events die de mogelijkheid bieden om de initialisatie van de pagina te besturen en zelfs ervoor in te springen. Dit kan met volgende events:

* Pagebeforecreate, dit event is gelanceerd voor de creatie van de pagina
* Pagecreate, dit events is gelanceerd na de create van de pagina, maar voor de initialisatie van de widgets
* Pageinit, dit event wordt pas gelanceerd na de hele create van de pagina

$(document).ready() funcite wordt steeds maar 1 keer gelanceerd. Daarom is het beter om met pageinit te werken, want deze gaat zelfs met AJAX calls gelanceerd worden.

### Page Load and Remove Events

Op dezelfde manier als page initialization heb je een aantal events die betrekking hebbne op het laden van de pagina:

* Pagebeforeload
* Pageload
* Pageloadfailed, deze wordt enkel gelanceerd als het laden van de pagina mislukt.
* Pageinit

### Page Change Events

In hetzelfde genre hebben we ook de page change events. Opnieuw hebben zij ongeveer dezelfde betekenis:

* Pagebeforechange
* Pagechange
* Pagechangefailed

Hieraan zie je dat het belangrijk is om de goede events te kiezen bij wat je juist zoekt.

### Page Transitions and Animations Events

JQuery Mobile biedt de mogelijkheid om te detecteren als de pagina zichtbaar is of niet. Volgende events worden hiervoor gebruikt:

* Pagebeforehide
* Pagehide
* Pagebeforeshow
* Pageshow

Deze events geven je de kans om als je pagina geminimized wordt, dat als het opengeklikt wordt, je afbeeldingen en css bijvoorbeeld te refreshen. Of kijken als er een bepaalde nieuwere versie is van data.

### Using Layout Events

Er zijn een aantal widgets die veranderen van grootte of herschalen door user interaction. Bijvoorbeeld list views, accordions, textareas, … Om dit te voorkomen, of er anders mee om te gaan kan je volgend event gebruiken:

* Updatelayout

Dit event geeft je de kans om tussen te komen voor het veranderen van de layout van de elementen en widgets.