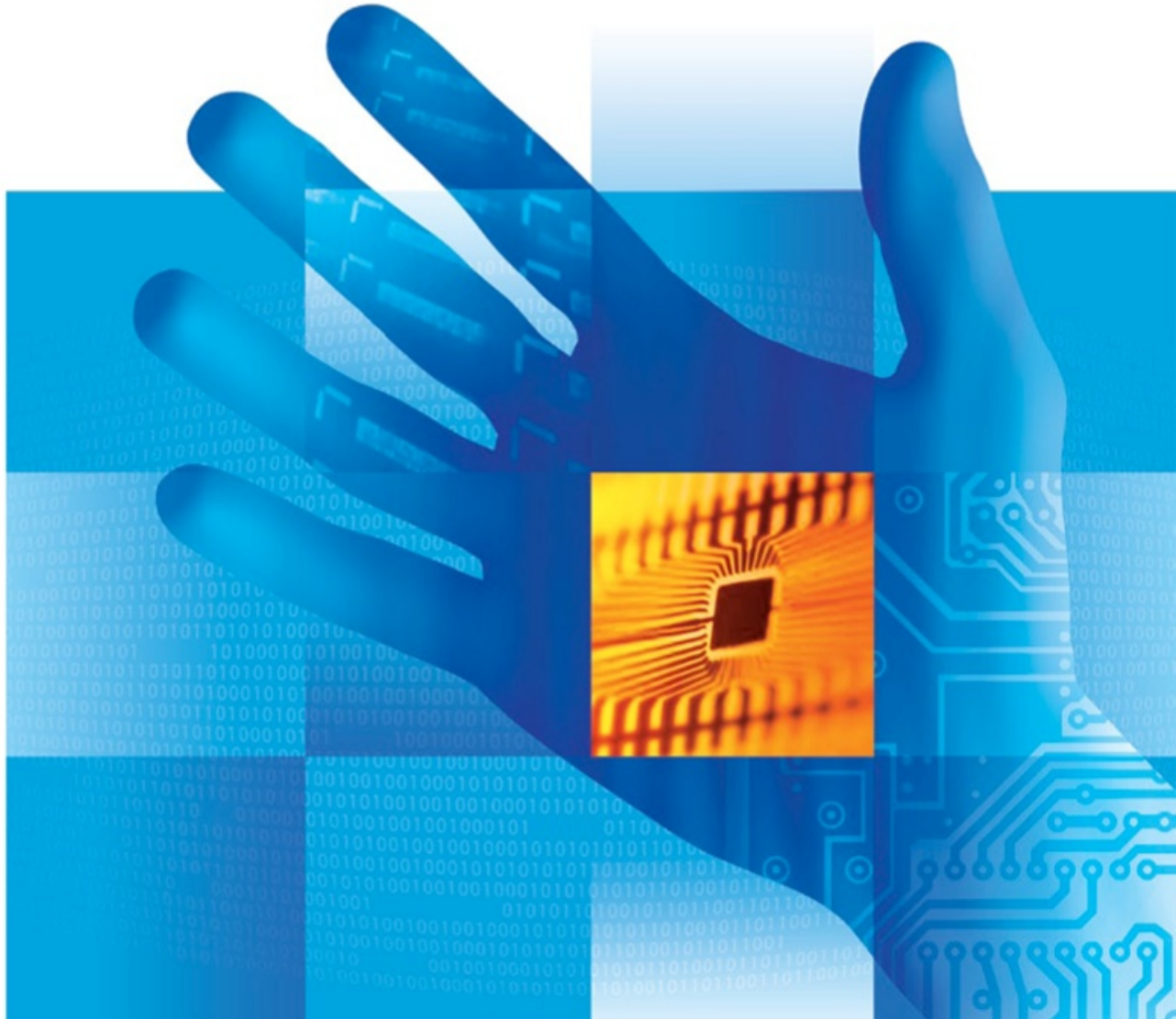
jQuery Mobile Extended High-level GUI elements



TAMZ 1

Lab 4

See: http://demos.jquerymobile.com/

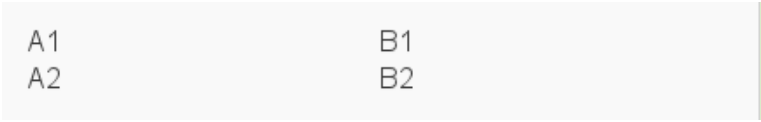
jQuery Mobile – extra GUI

- Besides the basic HTML form elements, already covered in the second lecture jQM offers several other features:
  - Page transitions
  - Grids
  - Lists (filterable)
  - Collapsibles and their groups (accordions)
  - Control groups
  - Popups
  - Flipswitches, range sliders
  - Tabs, panels
  - Tables with toggleable columns
- Normal sized widgets can be made smaller by using data-mini="true"
- For linking external page: rel="external"

Page transitions (grade A devices)

- Controlled by data-transition data attribute when changing page, opening dialog box, …
  - Reverse animation: data-direction="reverse"
- Transitions:
  - fade
  - pop
  - flip
  - turn
  - flow
  - slide
    - slidefade
  - slideup
  - slidedown
  - none

Grids



- Using class="ui-grid-X" (a-d), X+1 class="ui-block-Y" (a-e)

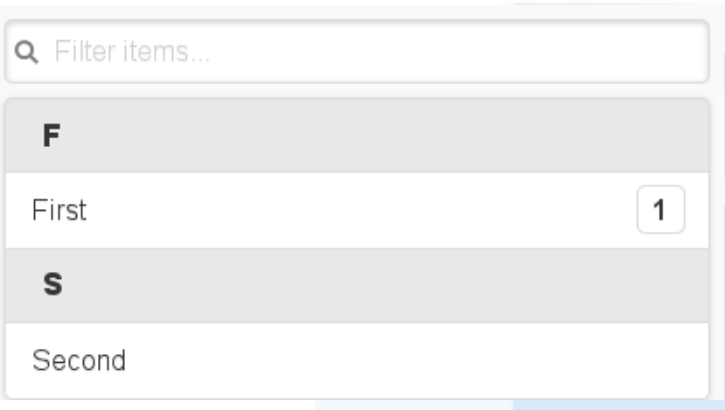
```
<div class="ui-grid-a">
    <div class="ui-block-a">A1</div>
    <div class="ui-block-b">B1</div>
    <div class="ui-block-a">A2</div>
    <div class="ui-block-b">B2</div>
</div>
```

- By default the blocks of grid have the same width
  - ui-grid-a: 50%, b: 33%, c: 25%, d: 20%
- Multiple lines of blocks possible when repeating the classes (green/blue example above)
- Responsive design for small screens can be enabled by using class="ui-grid-X ui-responsive" instead

Lists

- Use classic <ul> and <ol> tags with data-role="listview"
  - Read-only ones contain classic <li> items
  - Linked contain … inside of the
    - Multiple <a> tags inside list item provide split buttons
    - Border around the list view: data-inset="true"
    - Filterable listview: data-filter="true" (to offer a hint, you can use data-filter-placeholder="…"
      - To start with hidden list: data-filter-reveal="true"
  - List item with data-role="list-divider" will insert a separator
    - Or you can use auto division with data-autodividers="true"
  - A 123 inside of the list item can be used to provide a count of results

```
<ul data-role="listview" data-inset="true"
        data-filter="true" data-autodividers="true" >
       <li>First <span class="ui-li-count">1</span></li>
       <li>Second</li>
</ul>
```



Collapsibles

- Collapsible will allow you to collapse/expand the content
  - Collapsible content: …
  - The text in collapsed state is the content of the first <h1>-<h6> element inside of the collapsible.
    - For forms, use <legend> instead (and <fileldset> for content)
    - Custom header rendering is also possible
  - By default **+** for expanding content and **-** for collapsing
    - Can be changed, e.g. **data-collapsed-icon**="carat-d" and **data-expanded-icon**="carat-u"
  - Collapsed by default, opposite: data-collapsed="false"
  - Dynamic collapsibles are also possible
- Collapsible set (accordion) encloses several collapsibles
  - Only one of them can be expanded at a time
    - …

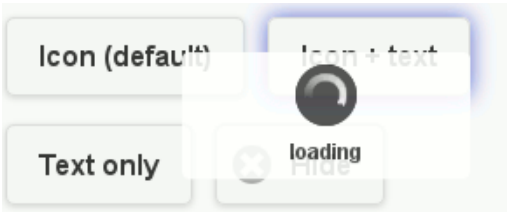
Control groups

- Allow to group a set of buttons into a single block
  - All buttons enclosed by
  - Default control group is vertical, can be changed by using data-type="horizontal".
  - Dynamic changes through JS also possible
  - If used on <fieldset> instead of , it can be used for radio buttons, checkboxes and selects
    - A horizontal layout is also possible:

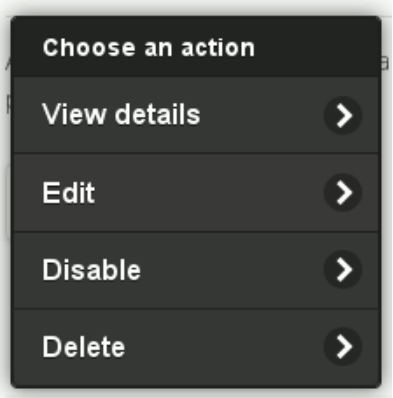

```
<form>
<fieldset data-role="controlgroup" data-type="horizontal">
    <legend>Horizontal, mini sized:</legend>
    <input name="ch-h" id="ch-h-a" type="checkbox">
    <label for="checkbox-h-6a">One</label>
                    …
    </fieldset>
</form>
```

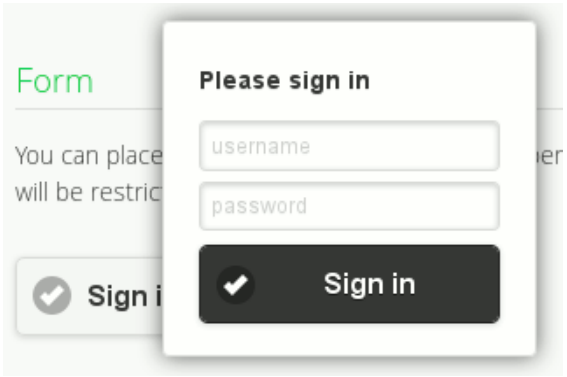
AJAX Loader

- A link can show/hide AJAX loader icon
  - class="hide-page-loading-msg"/"show-page-loading-msg"
  - use data-inline="true"
- The loader can contain text (together with/instead of) icon
  - data-textvisible="true" data-msgtext="My text loader"
  - Disable loader icon: data-textonly="true"
- We can even define loader as HTML code:
  - data-html="…"



Popups



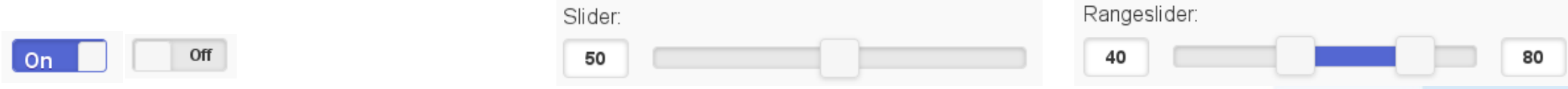


Will open a popup on some GUI element (typically by tapping it)

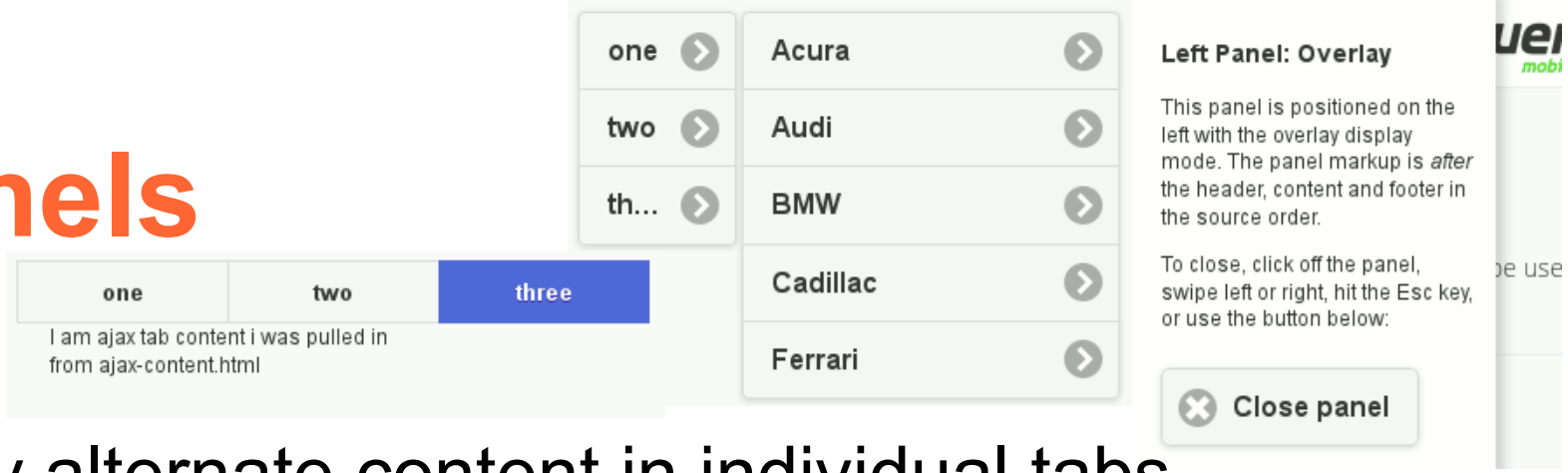
- From simple text info to complicated forms
- Contents of the popup inside of
- Referenced by a GUI element using link to popup ID with data-rel="popup"
  - e.g. Show
- Data transitions possible, popup is placed over the link
  - To change the position, we can use data-position-to
    - "window" – centered on window, "origin", any other sel.
- To create popup which can't be exited by ESC or clicking outside of it, we can use data-dismissible="false"
- To use a popup with arrow, we can specify data-arrow, by default "true", or specify arrow edges by -separated: l,t,r,b

Flip switches, sliders, range sliders

- Flipswitch – a two-state switch using classic checkbox or select GUI element, by default with On/Off text
  - To render as a flipswitch use data-role="flipswitch"
  - To set the labels, we can use
    - data-on-text="Enabled" data-off-text="Disabled"
- Sliders – based on HTML5 input type="range"
  - min="-99" – minimal possible value
  - max="99" – maximum possible value
  - step=".5" – slider increment
  - value="0" – default value
  - Read-only sliders: disabled="disabled"
  - 2 sliders enclosed in tag → Range slider (specifying min and max value)

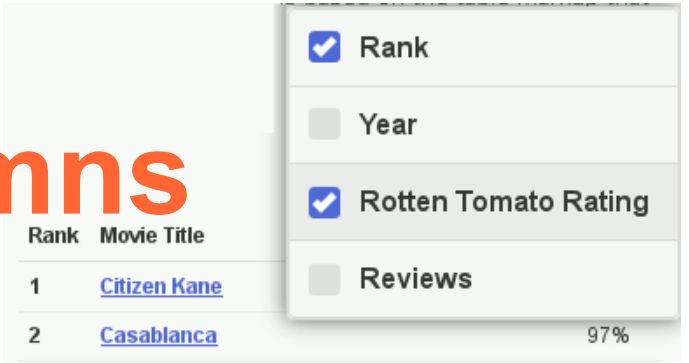


Tabs, Panels



- Tabs display alternate content in individual tabs
  - The whole content is enclosed in element
  - The tabs are defined in a list (<ul>) inside inside of a div with data-role="listview", containing links to content Ids
    - use data-ajax="false" on these links
  - Vertical tabs can be used e.g. with class="tablist-left"
- Panels can be shown/hidden, may contain menu, settings, etc., accessible as link targets
  - Defined as with data-role="panel"
    - data-display – how to show the panel (overlay/reveal/push)
    - data-position – where to place the panel (left/right)
    - data-position-fixed="true" – display independently on scroll
  - Limit closing: data-swipe-close="false", data-dismissible="false"

Customizable table columns



- Toggle the columns to make them (in)visible
  - Classic table with data-mode="columntoggle" and id
  - To change the button label, we can also include the data-column-btn-text="Which cols…" in the <table> tag
- Reflow a table – <table id="…" data-role="table" data-mode="reflow" class="ui-responsive">
- Columns defined in <th> tag with data-priority='X'
  - The priority defines, when to display these columns by default (data-priority="critical" – always), default settings:
    - data-priority="1" – displays column at 320px (20em)
    - …
    - data-priority="6" – displays the column at 1,120px (70em)
  - Use <abbr title="Very long title">VLT</abbr> inside of <th> where needed

Dynamic content updates

If we add/manipulate dynamic content in DOM with JavaScript, change values, etc., it may not be redrawn or created properly. To fix this, we must use one of following methods:

- Trigger updatelayout event – when showing/hiding content
- Trigger **create** event on changed element's container (parent) to ensure the content is regenerated
  - e.g. $("#mycset").parent().trigger("create");
- Call **refresh** method on GUI element which got changed (listview, collapsibleset, checkboxradio, …)
  - $(".selector").listview("refresh"); sel.button("refresh");
  - $(".selector").checkboxradio("refresh");
  - $(".selector").collapsibleset("refresh");
  - $(".selector").listview("refresh");
  - $(".selector").[range]slider("refresh");
  - $(".selector").selectmenu("refresh", force_rebuild);

Task (1 point)

Rich GUI written by hand using extended UI elements

- Form with text, date, choice group & selection
  - Entries from the form can be saved in a list and deleted/edited/shown
  - Collapsibles/Collapsible group can be used
- Example applications
  - Contacts (Name, phone number, E-mail, height (slider), gender, birthday)
  - Student list
  - …