Utilising the data attribute

adding client side behaviour in Oracle APEX

smart4 solutions®

Agenda

- Introducing the data attribute
- Introducing jQuery
- Changing Page-items into HTML items
- Record sorting
- Deleting records from a report
- Putting it all together

Introducing the data attribute

- Problem: how to unambiguously identify elements in a page
- Your HTML will have lots of hooks you can try to catch:
  - ID's
    - `$('#yourid').whateverfunction();`
  - classes
    - `$('.yourclass').anotherfunction();`
  - DOM traversal
    - `$('#aclass').parents('tr').children('td a[href~="a-record-id"]').whatever();`

Introducing the data attribute

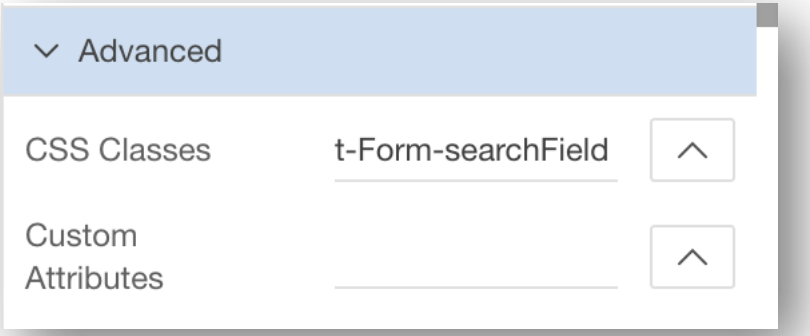
- The **data-*** global attributes form a class of attributes called **custom data attributes**, that allow proprietary information to be exchanged between the HTML and its DOM representation by scripts. (https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/data-*)

- HTML Attributes can be selected by:

  `$('[data-attribute="1234"]')`

  or

  `$('[data-attribute]')`



- And they can be simply added by using the "Custom Attributes" attribute

smart4 solutions

Introducing jQuery

- “jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.” (http://jquery.com/)
- Key functionality:
  - element selection (incl. DOM traversal)
  - event handling
  - AJAX
- Included in APEX by default !!

Introducing jQuery

- Element selection through the use of CSS selectors:
  - `.class`
  - `#static-id`
  - `[attribute]`
  - `[attribute='value']`

  - `[attribute~='value']` word contains
  - `[attribute^='value']` begins with
  - `[attribute$='value']` ends in
  - `[attribute*='value']` substring contains
  - `[attribute*='value' i]` substring contains (case insensitive)

Changing page-items to HTML5

- Oracle only supports 4 HTML5 types:
  - Text, Email, Phone Number and URL
- HTML5 offers a lot more:
  - color, date, datetime-local, month, number, range, search, time, week

- Run DA on pageload:
  - ```
    $('input[data-html5-type]').each( function() {
       $(this).attr( 'type'
                   , $(this).attr('data-html5-type')
                   );
    });
    ```
- Change your item into regular tekst item
- Add attribute: `data-html5-type` and set it to the type you need
- For date fields don't forget to set the format to "YYYY-MM-DD"

smart4 solutions
    ```

Changing page-items to HTML5

- of course we need a bit of javascript:

```
$('input[data-html5-type]').each( function() {
   $(this).attr( 'type'
                , $(this).attr('data-html5-type'));
});
```

- We can include this in an “onload” DA

Record Sorting

- using jQuery “sortable”
- Sortable is not enabled by default
- More info on:
  https://goo.gl/kKvrnW

smart4 solutions

Record Sorting

- Create classic report
  - the SQL should include an “order by” clause
  - All “important” regions should have a static ID
- Use the static ID to select your region:

```
// initiate sortability
$('#p10-emp').on('apexafterrefresh', function() {
     setSortable('#p10-emp’
          , 'table.t-Report-report tbody'
          , 'data-empno'
          , 'EMPNO'
          , 'AJAX_SORT_EMP');
});
setSortable(....);
```

Record Sorting

- And we need a function to handle everything..
  - a Handler function

```
function setSortable(pRegion, pSelector, pAttrName, pHeader, pServerProcess) {    // add some CSS to make user aware
of sortability    $(pRegion + ' ' + pSelector).css('cursor', 'pointer');    // add data-empno attribute    $(pRegion +
' ' + pSelector + ' td[headers="' + pHeader + '"]').each( function() {      $(this).parent('tr').attr(pAttrName,
$(this).text() );    } );    $(pRegion + ' ' + pSelector).sortable({        helper: function(e, ui) {
ui.children().each(function() {                $(this).width($(this).width());            });            return ui;
},        stop: function( event, ui ) {            // var employees will hold empno's colon deparated            var
employees = [];            // walk through all items that have data-empno attribute            $('[' + pAttrName +
']').each( function(){                employees.push ( $(this).attr(pAttrName) );            } );
apex.server.process ( pServerProcess, {                f01: employees            }, {dataType:"text",
success: function( pData ) {                $(pRegion).trigger('apexrefresh');                }              });
//$(ui.item).find('[data-empno]').attr('data-empno')        }    });    $(pRegion).on('apexafterrefresh', function() {
setSortable(pRegion, pSelector, pAttrName, pHeader, pServerProcess);    });
```

  - Let’s watch this in APEX

Deleting records from a report

- Again: regular report IR, CR or IG
- Add column with delete icon:
  - `<i data-delete-id="#EMPNO#" class="fa fa-trash-o"></i>`
- Add Event handler:
  - are you sure?
  - save the value in a hidden item
  - execute some pl/sql code (hidden item goes in and out)
  - alert that the records has been deleted
  - refresh the report

Requirements

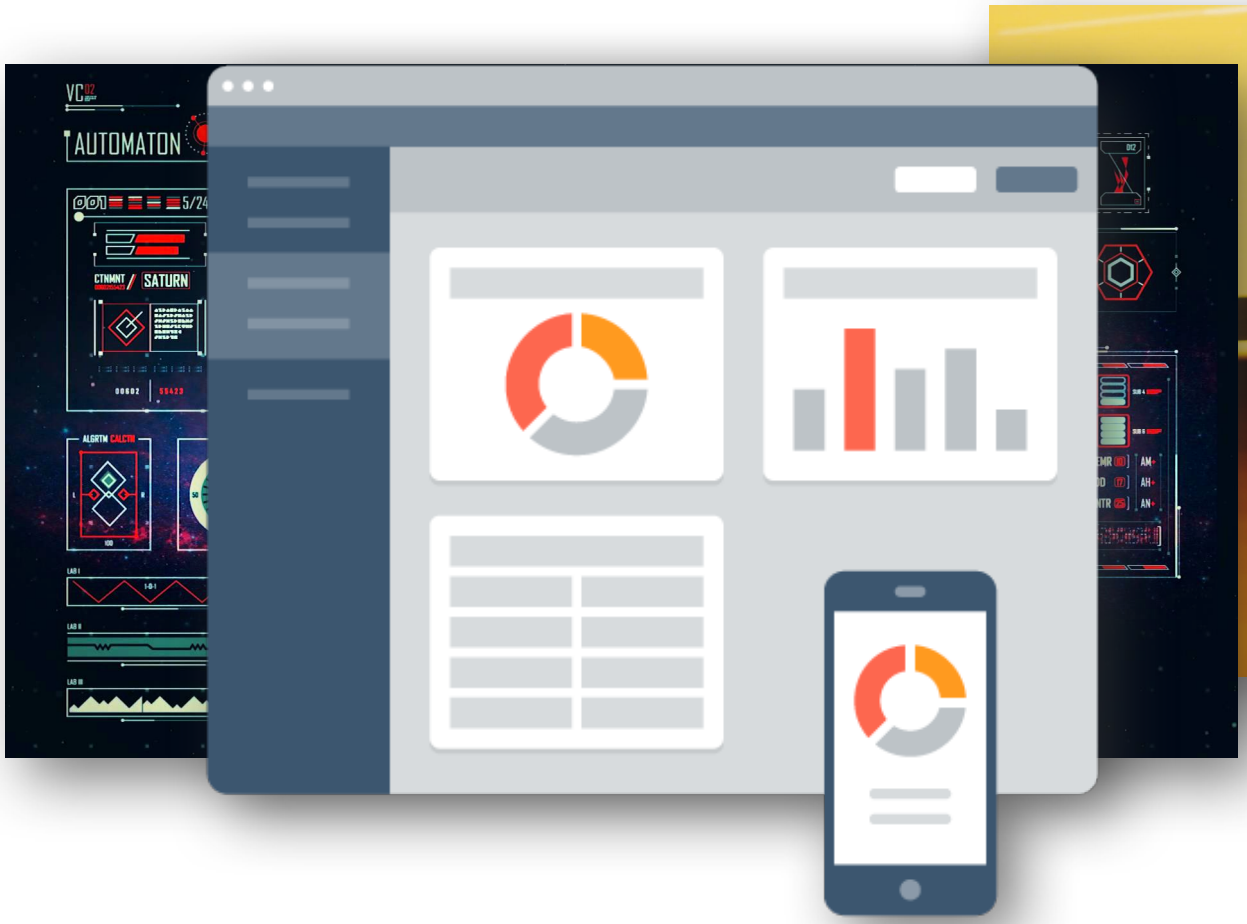
- We must send a letter to an emp
- We must be able to choose from predefined alineas
- We must be able to change the order of which those alineas appear on the letter
- We must be able to change the alinea’s texts without modifying the original predefined text

- We only need to send one letter per emp

APEX out of the box

- We could use an interactive grid for this!
- Let’s see how that looks

Let’s hack the user interface



smart4 solutions

Questions ?

smart4 solutions®