

KSS Techniques

Plone Symposium East 2008

Author: Joel Burton <joel@joelburton.com>
Copyright: Copyright 2007 Joel Burton. All rights reserved.
Covering: Plone 2.5 or newer
Notice: Distribution outside of presentation prohibited.
Company: PloneBootcamps: www.plonebootcamps.com

Contents

1	KSS Techniques	2
1.1	Overview	2
1.1.1	Who Am I?	2
1.1.2	Overview	2
1.1.3	Challenges of JavaScript	2
1.1.4	What is KSS?	3
1.2	Client-Side	3
1.2.1	Simple Message	3
1.2.2	KSS	3
1.2.3	KSS	3
1.2.4	KSS	3
1.2.5	KSS	4
1.2.6	KSS	4
1.2.7	Parameter Provider	4
1.2.8	Parameter Provider	4
1.2.9	Registering KSS	4
1.3	Server-Side	5
1.3.1	Server-Side	5
1.3.2	Book Order Form	5
1.3.3	KSS	5
1.3.4	Script	5
1.3.5	Script	6
1.3.6	Script	6
1.3.7	No Submit KSS	6
1.3.8	No Submit KSS	7
1.3.9	Report Data	7
1.3.10	Book Report	7
1.3.11	Updating in KSS	7
1.3.12	Updating Script	8
1.3.13	Updating Automatically	8
1.3.14	Updating Script	8
1.4	KSS Cheat Sheet	8
1.4.1	Events: Basic	8

1.4.2	Events: Mouse	9
1.4.3	Events: Form	9
1.4.4	Events: Automatic	9
1.4.5	Actions: Changing HTML	9
1.4.6	Actions: Attributes	9
1.4.7	Actions: CSS Classes	10
1.4.8	Actions: Form Elements	10
1.4.9	Actions: Debugging	10
1.4.10	Parameter Providers: Forms	10
1.4.11	Parameter Providers: Content	10
1.4.12	Command Sets: Core	11
1.4.13	Commands Sets: Zope + Plone	11
1.4.14	Debugging KSS	11
1.4.15	Future of KSS	11
1.4.16	Thanks!	11

1 KSS Techniques

1.1 Overview

1.1.1 Who Am I?

- Plone Trainer / Developer
 - Offers training at <http://plonebootcamps.com>
 - Definitely not a JavaScript expert!

1.1.2 Overview

- Why KSS?
- Client-side convenience
- Server-side power
- Quick reference guide

1.1.3 Challenges of JavaScript

- Another language!
- Browser incompatibilities persist
- Even when done correctly ...
 - Do it again in Python

1.1.4 What is KSS?

- “Kinetic Style Sheets”
 - “Power of JS, syntax of CSS”
 - Allows you to declare behavior
 - Includes AJAX library
- Very powerful, very cool!

1.2 Client-Side

1.2.1 Simple Message

```

<form>                                     Show button
  <input type="submit" id="logButton" name="log"
    value="Show Message" />
</form>

<fieldset>                                  Show message
  <legend>Message:</legend>
  <p id="message"
    tal:condition="exists: request/log">Clicked (non-KSS)</p>
</fieldset>

```

1.2.2 KSS

```

#logButton:click {
  action-client: alert;
  alert-message: "Clicked!";
}

```

1.2.3 KSS

```

#logButton:click {                          Identifier & event
  action-client: alert;                       Client action: alert
  alert-message: "Clicked!";                 Parameters for alert
}

```

1.2.4 KSS

```

#logButton:click {
  action-client: replaceInnerHTML;           Action
  replaceInnerHTML-kssSelector: "#message"; Where
  replaceInnerHTML-html: "Clicked (via KSS)";
}

```

1.2.5 KSS

```
#logButton:click {
  action-client: replaceInnerHTML;           Action
  replaceInnerHTML-kssSelector: "#message"; Where
  replaceInnerHTML-html: "Clicked (via KSS)";
}
<p id="message">Clicked (non-KSS)</p>
```

1.2.6 KSS

```
#logButton:click {
  evt-click-preventdefault: True;   Don't do normal thing

  action-client: replaceInnerHTML;
  replaceInnerHTML-kssSelector: "#message";
  replaceInnerHTML-html: "Clicked (via KSS)";
}
```

1.2.7 Parameter Provider

```
#logButton:click {
  evt-click-preventdefault: True;   Don't do normal thing

  action-client: replaceInnerHTML;
  replaceInnerHTML-kssSelector: currentFormVar('note');
  replaceInnerHTML-html: "Clicked (via KSS)";
}
```

1.2.8 Parameter Provider

```
#logButton:click {
  evt-click-preventdefault: True;   Don't do normal thing

  action-client: replaceInnerHTML;
  replaceInnerHTML-kssSelector: currentFormVar('note');
  replaceInnerHTML-html: "Clicked (via KSS)";
}
<input type="text" name="note" />
```

1.2.9 Registering KSS

- Add your KSS file name to *portal_kss*
 - Just like *portal_css*
 - Put into debug mode to avoid cache problems

1.3 Server-Side

1.3.1 Server-Side

- Call server-side scripts via AJAX
 - Get results without page refresh
 - * Can show results
 - * Log them
 - * Do side-actions, etc.

1.3.2 Book Order Form

```
<form action="order_form">

  <input type="text" id="numcopies" name="numcopies"
    tal:attributes="value numCopies" />

  copies of <b>Mega Plone Development</b> at $19.99 =

  <span id="total" tal:content="cost">[cost]</span>

  <input type="submit" id="updateButton" value="Update" />

</form>
```

1.3.3 KSS

```
#updateButton:click {
  evt-click-preventdefault: True;           Skip regular

  action-server: kssOrderCalculate;         Call script
  kssOrderCalculate-qty:
    currentFormVar('numcopies');           Pass #
}
```

1.3.4 Script

- *orderCalculate.py* (PythonScript):


```
qty = int(qty)
return qty * 19.99
```

1.3.5 Script

- *kssOrderCalculate.py* (PythonScript):

```

from kss.core.ttwapi import (
    startKSSCommands,
    getKSSCommandSet,
    renderKSSCommands )

startKSSCommands(context, context.REQUEST)
core = getKSSCommandSet('core')

answer = str(context.orderCalculate(qty))

core.replaceInnerHTML('#total', answer)

return renderKSSCommands()

```

1.3.6 Script

- *kssOrderCalculate.py* (PythonScript):

```

from kss.core.ttwapi import (
    startKSSCommands,
    getKSSCommandSet,
    renderKSSCommands )
boilerplate

startKSSCommands(context, context.REQUEST) boilerplate
core = getKSSCommandSet('core')

answer = str(context.orderCalculate(qty)) get answer

core.replaceInnerHTML('#total', answer) replace

return renderKSSCommands() boilerplate

```

1.3.7 No Submit KSS

```

#updateButton:load {
  action-client: deleteNode;
}

#numcopies:change {
  action-server: kssOrderCalculate;
  kssOrderCalculate-qty: currentFormVar('numcopies');
}

```

1.3.8 No Submit KSS

```
#updateButton:load {           Get rid of button
  action-client: deleteNode;
}

#numcopies:change {           Update on change
  action-server: kssOrderCalculate;
  kssOrderCalculate-qty: currentFormVar('numcopies');
}
```

1.3.9 Report Data

```
def getSalesData(self):

  qty = random.randint(1,20)
  total = qty * 19.99

  return [{
    'id':'mpd',
    'title':'Mega Plone Development',
    'qty': qty,
    'total': total
  }]
```

1.3.10 Book Report

```
<tr tal:repeat="book books">
  <td class="title"
    tal:attributes="id string:${book/id}-title"
    tal:content="book/title"></td>
  <td class="qty"
    tal:attributes="id string:${book/id}-qty"
    tal:content="book/qty"></td>
  <td class="cost"
    tal:attributes="id string:${book/id}-cost"
    tal:content="book/total"></td>
</tr>
```

1.3.11 Updating in KSS

```
#salesUpdate:click {
  evt-click-preventdefault: True;   Skip regular

  action-server: kssUpdateBook;     Call script
  kssUpdateBook-bookid: "mpd";     Pass book to update
}
```

1.3.12 Updating Script

```

class KSSSalesReport(PloneKSSView):
    @kssaction
    def __call__(self, bookid):
        qty = random.randint(21,40)
        total = qty * 19.99

        core = self.getCommandSet('core')
        core.replaceInnerHTML('#%s-qty' % bookid, str(qty))
        core.replaceInnerHTML('#%s-cost' % bookid, str(total))

```

Simplifies our work
By doing boilerplate

1.3.13 Updating Automatically

```

#salesUpdate:load {
    action-client: deleteNode;
}

#sales:timeout {
    evt-timeout-delay: 2000;
    action-server: kssUpdateBook;
    kssUpdateBook-bookid: "mpd";
}

```

Get rid of button
Refresh every 2 secs

1.3.14 Updating Script

```

class KSSSalesReport(PloneKSSView):
    @kssaction
    def __call__(self, bookid):
        qty = random.randint(21,40)
        total = qty * 19.99

        core = self.getCommandSet('core')
        core.replaceInnerHTML('#%s-qty' % bookid, str(qty))
        core.replaceInnerHTML('#%s-cost' % bookid, str(total))

```

1.4 KSS Cheat Sheet

1.4.1 Events: Basic

- *click*
- *dblclick*
- *load*: Done on page load

- Useful for hiding stuff when KSS active

1.4.2 Events: Mouse

- *mousedown*, *mouseup*
- *mouseover*, *mouseout*

1.4.3 Events: Form

- *blur*: leaving a field
- *focus*: entering a field
- *change*: field value changes
- *select*: trigger when drop-down changes
- *submit*: on form submission

1.4.4 Events: Automatic

- *timeout*: done automatically after expiration
 - *evt-timeout-delay*: period in milliseconds
 - *evt-timeout-repeat*: (default true)

1.4.5 Actions: Changing HTML

- *replaceInnerHTML*: Replace all children of the given node with the given content.
 - *html*: the html to insert
- *insertHTMLAfter*: Add HTML after given node.
 - *html*: the html to insert
- *deleteNode*: Delete the node.

1.4.6 Actions: Attributes

- *setAttribute*: Sets a given HTML attribute of the node.
 - *name*: the attribute name.
 - *value*: the attribute value to set
- *setStyle*: Sets a given style element on the node.
 - *name*: the name of the style element.
 - *value*: the style element value to set

1.4.7 Actions: CSS Classes

- *addClass*: Add a class to the classes of the node.
 - value: the name of the class
- *removeClass*: Remove a class from the classes of the node.
 - value: the name of the class
- *toggleClass*: Toggle class on node.
 - value: the name of the class

1.4.8 Actions: Form Elements

- *focus*: Focus the given node that is a form input.

1.4.9 Actions: Debugging

- *error*: Throws an exception, when executed.
- *log*: Logs an informational message.
 - message
- *alert*: Javascript alert box.
 - message

1.4.10 Parameter Providers: Forms

- *formVar(formname, varname)*
 - Produces the value of a given variable within a given form.
- *currentFormVar(varname)*
 - Produces the value of a given variable within the current form.

1.4.11 Parameter Providers: Content

- *nodeAttr(attrname)*
 - Produces the value of a given html attribute of the selected node.
- *nodeContent()*
 - Produces the textual content of the node. Newlines are converted to spaces.

1.4.12 Command Sets: Core

- Commands are things used by scripts
- core command set are the same as the “actions”
 - *setAttribute*, *addClass*, etc.

1.4.13 Commands Sets: Zope + Plone

- Additional command sets for Zope and Plone specific stuff
 - Refreshing a viewlet
 - Refreshing a portlet
 - ... and more!

1.4.14 Debugging KSS

- Use Firebug!
 - Turn on *portal_javascript* debugging
 - Messages come out to Firebug console
- Try your scripts from URL directly

1.4.15 Future of KSS

- Other JavaScript libraries
 - For people who want to tinker deeply
- Possibly non-JS backends?
- In any event ...
 - ... lots more interactivity!

1.4.16 Thanks!

- Questions?
- Handouts at
<http://plonebootcamps.com/resources>