



Seaside

Web 2.0

Lukas Renggli
renggli@iam.unibe.ch

Why should I upgrade?

Collaboration

Design

Liveliness

Sharing

Usability

How can I upgrade?

Can I do it with Seaside?

XHTML

Semantically valid markup



XHTML

Semantically valid markup

CSS

Cascading Style-Sheets



CSS

Cascading Style-Sheets

RSS

Really Simple Syndication



RSS

Really Simple Syndication



AJAX

Asynchronous JavaScript and XML

COMET

HTTP Streaming, Server Push

Seaside Web 2.0

Challenge



**Concentrate onto
the Web application**



Browser Bugs



Browser Differences



Browser Incompatibilities

prototype

script.aculo.us



Tight Integration

Feature Complete

Up-To-Date

Say it in Smalltalk

Avi Bryant
www.smallthought.com

Lukas Renggli
www.lukas-renggli.ch

Say it in Smalltalk

No JavaScript Programming

**Still, you need to be aware that
your code will result in JavaScript**

Think of it as Magic

```
html effect  
  id: 'hint';  
  shake
```



```
new Effect.Shake('hint')
```

Adding JavaScript Events

```
html div  
  onClick: (html effect toggle);  
  with: [ ... ]
```

AJAX – Step by Step

```
html (updater|periodical|request|evaluator)  
  [ OPTIONS; ]*  
  [ HANDLER; ]*  
  [ TRIGGER; ]*  
  [ CALLBACK; ]*
```

AJAX – Step by Step

Choose a strategy



```
html (updater|periodical|request|evaluator)  
  [ OPTIONS; ]*  
  [ HANDLER; ]*  
  [ TRIGGER; ]*  
  [ CALLBACK; ]*
```

AJAX – Step by Step

Choose a strategy

html (updater|periodical|request|evaluator)

[OPTIONS;]*

[HANDLER;]*

[TRIGGER;]*

[CALLBACK;]*

Define options

AJAX – Step by Step

Choose a strategy

html (updater|periodical|request|evaluator)

[OPTIONS;]*

[HANDLER;]*

[TRIGGER;]*

[CALLBACK;]*

Define options

Assign http
event handlers

AJAX – Step by Step

Choose a strategy

html (updater|periodical|request|evaluator)

[OPTIONS;]*

[HANDLER;]*

[TRIGGER;]*

[CALLBACK;]*

Define options

Assign http
event handlers

Set triggers to
serialize client state

AJAX – Step by Step

Choose a strategy

html (updater | periodical | request | evaluator)

[OPTIONS;]*

[HANDLER;]*

[TRIGGER;]*

[CALLBACK;]*

Define options

Assign http
event handlers

Define primary (and
secondary) callbacks

Set triggers to
serialize client state

AJAX Strategies

- Request
- Updater
 - Insertion
 - Periodical
- Evaluator



AJAX Request

- Perform a request and evaluate the callback `ajaxCallback` on the server.
- Note: A bug in your code doesn't show up in the Web browser, but directly as a debugger within your development image.

`html request`

`callback: [self ajaxCallback]`

AJAX Updater

- Replace the contents of the DOM element named `domId` with XHTML rendered by `renderAjaxOn:`.

```
html updater
```

```
  id: domId;
```

```
  callback: [ :r | self renderAjaxOn: r ]
```

Triggers

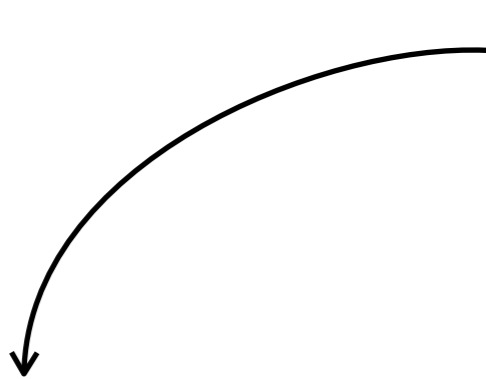
- `triggerForm: aDomId`
Trigger the form `aDomId` and all containing form-fields and associated callbacks.
- `triggerFormElement: aDomId`
Trigger the callback of the form element named `aDomId`. Does not work for checkbox and multi-select list.

Sortable List

- Zork
 - Foo
- Bar

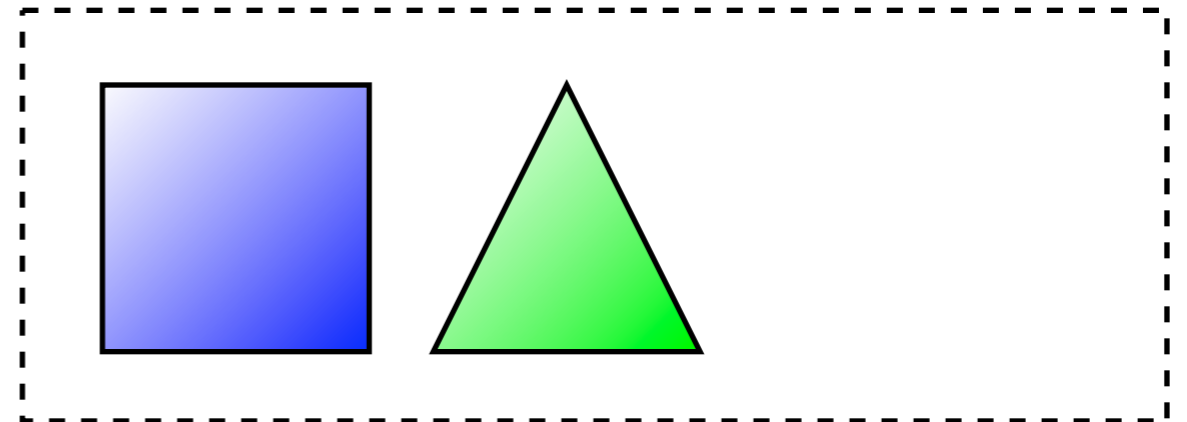
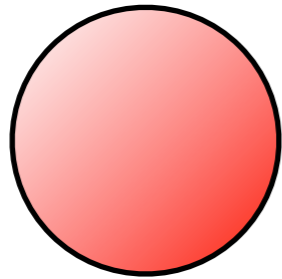
Sortable List

- Zork
- Foo
- Bar

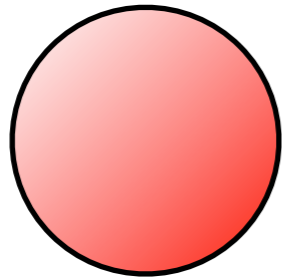


```
html unorderedList
  id: 'list';
  script: (html sortable
    onUpdate: (html request
      triggerSortable: 'list'
      callback: [ :v | list := v ]));
  with: [
    list do: [ :each |
      html listItem
        passenger: each;
        with: each label ] ]
```

Drag & Drop



Drag & Drop

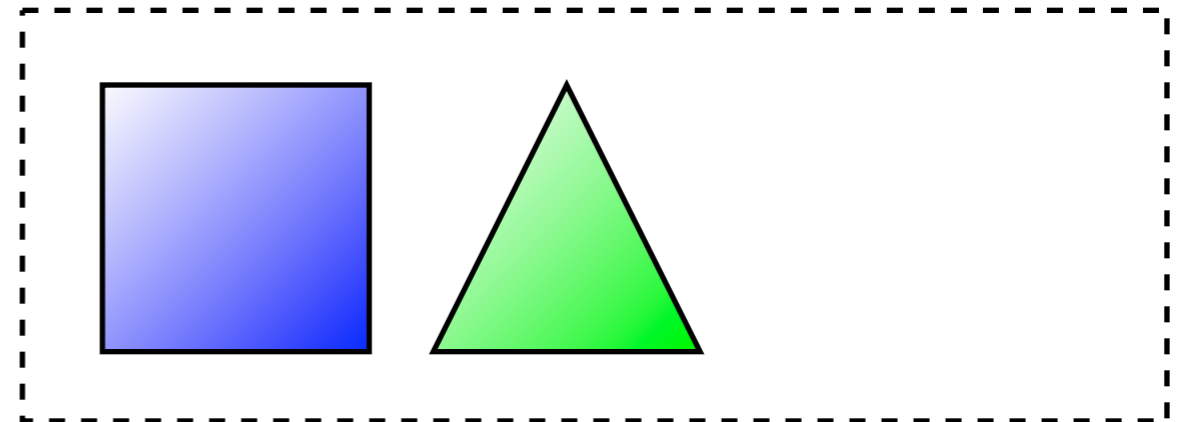


html div

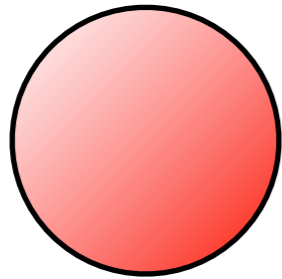
class: 'shape';

passenger: aCircle;

script: html draggable



Drag & Drop

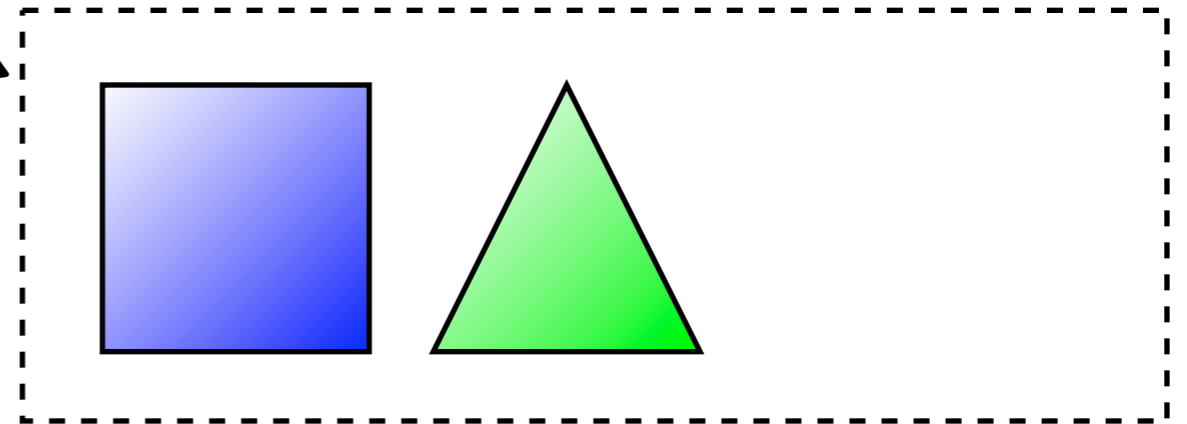


html div

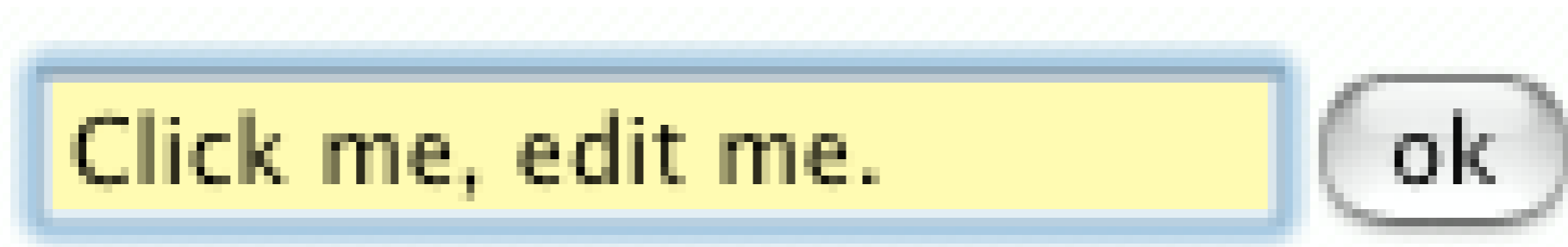
```
class: 'shape';  
passenger: aCircle;  
script: html draggable
```

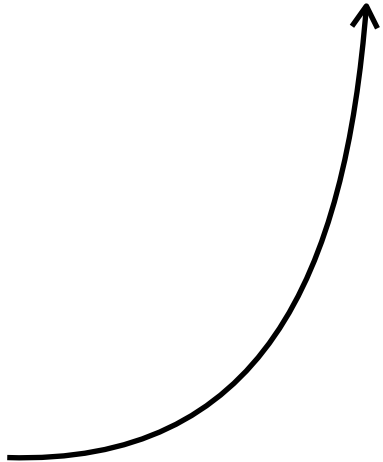
html div

```
id: 'dropbox';  
script: (html droppable  
  accept: 'shape';  
  onDrop: (html updater  
    id: 'dropbox';  
    triggerPassenger: [ :v | box add: v ];  
    callback: [ :r | self renderBox: r ]));  
with: [ html renderBox: r ]
```



In Place Editing



html paragraph 

```
script: (html inplaceEditor
  triggerInPlaceEditor: [ :v | text := v ];
  callback: [ :r | r render: text ]);
with: text
```

What about debugging?







FireBug

web debugging evolved

scriptaculous.seasidehosting.st

**“It’s more like
Web 3.0”**