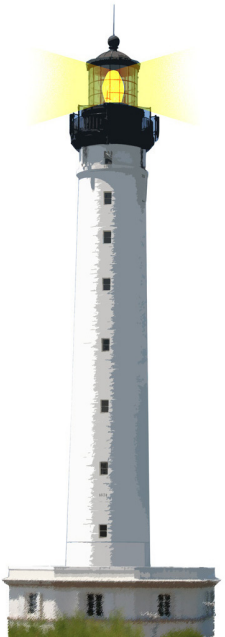
Seaside: Rendering Components

Damien Cassou, Stéphane Ducasse and Luc Fabresse

W4S09

Pharo

http://www.pharo.org



Remember

- renderContentOn: is dedicated to HTML generation
- parameter named html (WAHtmlCanvas) defines a DSL like API to generate valid HTML

```
WACounter >> renderContentOn: html
  html heading: count.
  html anchor
    callback: [ self increase ];
    with: '++'.
  html space.
  html anchor
    callback: [ self decrease ];
    with: '--'
```

Rendering components

Using a DSL

- Brushes to emit HTML
  - OO API to generate HTML
  - Generated HTML valid by construction
- CSS-based

Generating a Title

```
html div id: 'title'; with: 'Title'
```

```
<div id="title">Title</div>
```

Generating a List

```
html div id: 'list';
   with: [
      html span class: 'item'; with: 'Item 1'.
      html span class: 'item'; with: 'Item 2' ]
```

```
<div id="list">
   <span class="item">Item 1</span>
   <span class="item">Item 2</span>
</div>
```

Generating a List with a Loop

```
html unorderedList
   id: 'list';
   with: [
      1 to: 5 do: [ :i |
         html listItem
            class: 'item';
            with: 'Item ', i asString ]]
```

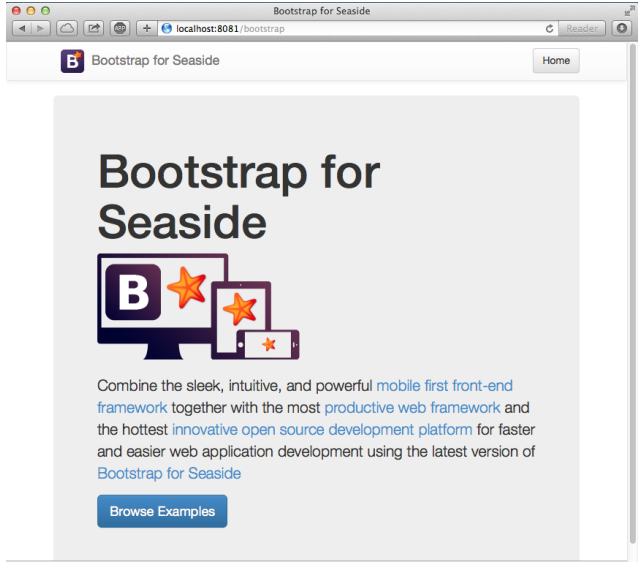
```
<ul id="list">
   <li class="item">Item 1</li>
   <li class="item">Item 2</li>
   <li class="item">Item 3</li>
   <li class="item">Item 4</li>
   <li class="item">Item 5</li>
</ul>
```

Generating a List with a Loop

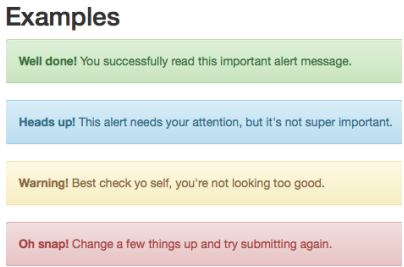
```
html unorderedList
   id: 'list';
   with: [
      1 to: 5 do: [ :i |
         html listItem
            class: 'itemodd' if: i odd;
            class: 'itemeven' if: i even;
            with: 'Item ', i asString ]]
```

```
<ul id="list">
   <li class="itemodd" >Item 1</li>
    <li class="itemeven">Item 2</li>
    <li class="itemodd" >Item 3</li>
    <li class="itemeven">Item 4</li>
    <li class="itemodd" >Item 5</li>
</ul>
```

Twitter Bootstrap



Rendering Twitter Alerts




```
renderExampleOn: html
    html heading level: 2; with: 'Examples'.
    html tbsAlert beSuccess;
        with: [ html strong: 'Well done!'. html text: ' You successfully
          read this important alert message.' ].
    html tbsAlert beInfo;
        with: [ html strong: 'Heads up!'. html text: ' This alert needs
          your attention, but it''s not super important.' ].
```

Rendering Twitter Buttons


```
renderExampleOn: html
   html tbsButtonGroup: [
      html tbsButton beDefault; with: 'Left'.
      html tbsButton beDefault; with: 'Middle'.
      html tbsButton beDefault; with: 'Right' ].
```

Implement a Counter using Twitter Bootstrap

- Subclass WACounter

```
WACounter subclass: #WATwitterCounter
  instanceVariableNames: ''
  classVariableNames: ''
  package: 'Seaside-Examples-Misc'
```

- Use JQuery and Bootstrap libraries

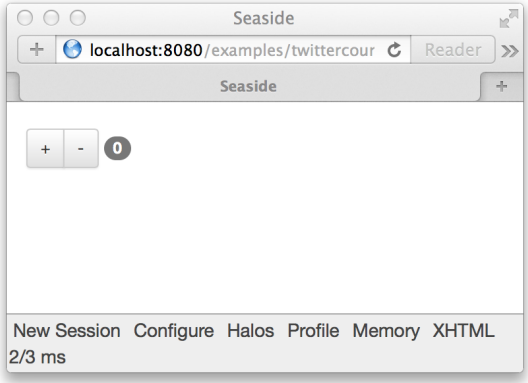
```
WATwitterCounter class >> initialize
  "self initialize"
  | app |
  app := WAAdmin register: self asApplicationAt: 'twittercounter
    '.
  app
    addLibrary: JQDeploymentLibrary;
    addLibrary: TBSDevelopmentLibrary
```

Implement a Counter using Twitter Bootstrap

- Redefine renderContentOn:

```
WATwitterCounter >> renderContentOn: html
  html tbsContainer: [
    html form: [
      html tbsButtonGroup beSmall; with: [
        html tbsButton beDefault;
          callback: [ self increase ]; with: '+'.
        html tbsButton beDefault;
          callback: [ self decrease ]; with: '–' ].
      html
        space;
        tbsBadge: self count ] ]
```

Implement a Counter using Twitter



Apply Your Own CSS Rules

```
WATwitterCounter >> renderContentOn: html
  html tbsContainer: [
    html form: [
      "..."
      html space.
      html tbsBadge
        class: 'odd' if: self count odd;
        with: self count ] ]

WATwitterCounter >> style
  ^ '.odd { color: red; }'
```

```
<span class="badge odd">7</span>
```

Conclusion

- A Web application = a root component
- A Component renders itself in HTML (renderContentOn:)
- An exensible DSL helps to easily generate HTML
  - using brushes
  - according to CSS frameworks such as Bootstrap, JQuery UI, ...
  - and scripting (loops, ...)

A course by

and

in collaboration with