

# Anchor Callback

- ★ Ask the rendering canvas for an anchor and configure it with a callback-block:

```
html anchor
```

```
  callback: [ self someAction ];
```

```
  with: 'Some Action'
```

- ★ The callback-block won't be evaluated right away, it is cached and will only be evaluated if the user clicks the link.

# Anchor Example

**WACounter>>renderContentOn: html**

html heading

level: 1;

with: count.

html anchor

callback: [ count := count + 1 ];

with: '++'.

html space.

html anchor

callback: [ count := count - 1 ];

with: '--'

# Form Callback

- ★ Render a form around all your controls:  
html form: [ ... ]
- ★ When using file-upload control, don't forget to specify the multipart-encoding:  
html form  
    multipart;  
    with: [ ... ]

# Text Input

★ Render a text-input field:

```
html textInput
```

```
  value: self text;
```

```
  callback: [ :value | self text: value ]
```

★ Render a password-input field:

```
html passwordInput
```

```
  callback: [ :value | self text: value ]
```

# Text Area

- ★ Render a text-area field:  
html textArea  
    callback: [ :value | self text: value ];  
    with: self text

# Submit Button

★ Render a submit button:  
html submitButton  
    callback: [ self save ];  
    text: 'Save'

# Check-Box



Render a check-box:

```
html checkbox
```

```
value: self selected;
```

```
callback: [ :bool | self selected: bool ]
```

# Radio-Group



Render a group of radio-buttons:

```
html radioGroup: [ :group |  
  html option  
    group: group;  
    selected: self selected = #a;  
    callback: [ self selected: #a ].  
  html option  
    group: group;  
    selected: self selected = #a;  
    callback: [ self selected: #a ] ]
```

# Select-List

★ Render a (drop-down) select-list:

```
html select
```

```
list: self collection;
```

```
selected: self selected;
```

```
callback: [ :value | self selected: value ];
```

```
labels: [ :value | value printString ]
```

# File-Upload



Render a file-upload field:

```
html fileUpload callback: [ :file |  
  self contentType: file contentType.  
  self fileName: file fileName.  
  self contents: file contents ]
```