

macro-html Widgets

Max Rottenkolber

Wednesday, 2 December 2015

Table of Contents

1	macro-html.widgets	1
1.1	html-widget-document (Function)	1
1.2	html-widget-form (Function)	2
1.3	html-widget-head (Function)	2
1.4	html-widget-input (Function)	3
1.5	html-widget-list (Function)	4
1.6	html-widget-meta (Function)	4
1.7	html-widget-select (Function)	5
1.8	html-widget-table (Function)	5
1.9	html-widget-textarea (Function)	6

1 macro-html.widgets

HTML widget functions. All widgets print to `*standard-output*`.

1.1 html-widget-document (Function)

Syntax:

— Function: **html-widget-document** *title body &key stylesheets scripts feeds encoding viewport*

Arguments and Values:

title—a *string*.

body—a *function designator* for a *function* that prints the document body to `*standard-output*`.

stylesheets—a list of *stylesheet designators*. A *stylesheet designator* is a *property list* with the keys `:href` and `:media` reflecting the keys of a `stylesheet` link.

scripts—a list of *strings* designating URIs to Javascript programs.

feeds—a list of *feed designators*. A *feed designator* is a *property list* with the keys `:href`, `:type` and `:title` reflecting the keys of an alternate link.

encoding—a *keyword* denoting a character encoding. The default is `:utf-8`.

viewport—a *string* denoting the value of a `viewport` meta attribute. the default is `"width=device-width, initial-scale=1"`.

Description:

`html-widget-document` produces a complete HTML document including a head section including elements declaring *title*, *stylesheets*, *scripts*, *feeds*, *encoding* and *viewport*. The *body function* is called to produce the document body.

1.2 html-widget-form (Function)

Syntax:

— Function: **html-widget-form** *action fields &key method description*

Arguments and Values:

action—a *string* denoting a *form action*¹.

fields—a *function designator* for a *function* that prints the form's inputs to `*standard-output*`.

method—one of "GET" or "POST". The default is "GET".

description—a *string*. The default is "Submit".

Description:

`html-widget-form` produces a form element with *fields*. The resulting form will be bound to *action* and use *method*. It will also contain a submit button labeled with *description*.

See Also:

- 1. *Form action* (<http://www.w3.org/TR/html5/forms.html#attr-fs-action>)

1.3 `html-widget-head` (Function)

Syntax:

— Function: `html-widget-head` *title* &*key* *stylesheets* *scripts* *feeds* *encoding* *viewport*

Arguments and Values:

title—a string.

stylesheets—a list of stylesheet designators. A stylesheet designator is a property list with the keys `:href` and `:media` reflecting the keys of a stylesheet link.

scripts—a list of strings designating URIs to Javascript programs.

feeds—a list of feed designators. A feed designator is a property list with the keys `:href`, `:type` and `:title` reflecting the keys of an alternate link.

encoding—a keyword denoting a character encoding. The default is `:utf-8`.

viewport—a string denoting the value of a viewport meta attribute. the default is `"width=device-width, initial-scale=1"`.

Description:

`html-widget-head` produces a head section including elements declaring *title*, *stylesheets*, *scripts*, *feeds*, *encoding* and *viewport*.

1.4 `html-widget-input` (Function)

Syntax:

— Function: `html-widget-input` *name* *label* &*optional type*

Arguments and Values:

name, *label*—strings.

type—a string denoting an input type¹. The default is `"text"`.

Description:

`html-widget-input` produces an input element of *type* with *name* preceded by *label*.

See Also:

- *Input type* (<http://www.w3.org/TR/html5/forms.html#attr-input-type>)

1.5 `html-widget-list` (Function)

Syntax:

— Function: `html-widget-list` *list* &key *type to-string*

Arguments and Values:

list—a *list*.

type—one of `:ordered`, `:unordered` and `:definitions`. The default is `:unordered`.

to-string—a *function designator*. The default is `identity`.

Description:

`html-widget-list` produces an ordered, unordered or definition list containing the items in *list*. *To-string* is called on list items before they are included in the list.

If *type* is `:ordered` or `:unordered` an ordered or unordered list is produced respectively, containing the elements of *list*.

If *type* is `:definitions` a definition list is produced. *List* must be a *list* of two-element *lists*, the first and second elements being definition title and definition description respectively. *To-string* will be called with two arguments: `:title` or `:description` depending on the value *type* and the respective value of the two-element *list*.

1.6 `html-widget-meta` (Function)

Syntax:

— Function: `html-widget-meta` &key *content-type encoding*

Arguments and Values:

content-type—a *string* denoting a MIME type. The default is `"text/html"`.

encoding—a *keyword* denoting a character encoding. The default is `:utf-8`.

Description:

`html-widget-meta` produces a meta element declaring the *content-type* and *encoding* of a HTML document.

1.7 html-widget-select (Function)**Syntax:**

— Function: **html-widget-select** *name label options &key multiple*

options::= '({↓*option-group* | ↓*option*}*)

option-group::= (:group *label* {↓*option*}*)

option::= (*value label* [:selected])

Arguments and Values:

name, label, value—strings.

multiple—a generalized boolean. The default is *false*.

Description:

`html-widget-select` produces a select element *name* preceded by *label* containing option elements as declared in *options*. *Option-groups* can be used to produce optgroup elements accordingly. If an *option* declaration contains `:selected` as its third element, the resulting option element will be selected by default. If *multiple* is true, the select element will allow for multiple selections.

1.8 html-widget-table (Function)**Syntax:**

— Function: **html-widget-table** *head body*

Arguments and Values:

head—a list.

body—a list of lists.

Description:

`html-widget-table` produces a table element with *head* as its table head *body* as its rows.

1.9 `html-widget-textarea` (Function)

Syntax:

— Function: `html-widget-textarea` *name label initial-text*

Arguments and Values:

name, label, initial-text—strings.

Description:

`html-widget-text-area` produces a `textarea` element with *name* and *initial-text* preceded by *label*.