

Simple Use Case: YAHOO.widget.Editor

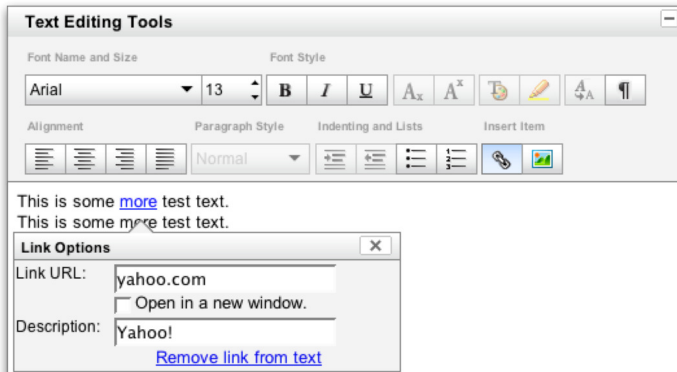
Markup:

```
<body class="yui-skin-sam">
<textarea id="msgpost">Preloaded HTML goes here.
</textarea>
</body>
```

Script:

```
var oEditor = new YAHOO.widget.Editor('msgpost',
{
    height: '300px',
    width: '500px'
});
oEditor.render();
```

Creates an Editor instance with default configurations.



Constructor: YAHOO.widget.Editor

`YAHOO.widget.Editor(str|el container[, obj cfg])`

Arguments:

- (1) **Container element:** <textarea> element or element id for the <textarea> that will be transformed into a Rich Text Editor.
- (2) **Configuration object:** When instantiating an Editor, you can pass all configurations in as an object argument or configure the instance after instantiation. See Configuration Options section for common configuration object members.

Dependencies

Editor: Yahoo, Dom, Event, Element, ContainerCore; Animation, Menu and Button are optional. **SimpleEditor:** YAHOO, Dom, Event, and Element; Animation and ContainerCore are optional.

Interesting Moments in Rich Text Editor & Toolbar

See online docs for complete list of Rich Text Editor and Toolbar events.

Event	Fires...
editorContentLoaded	Fires after the editor iframe's document fully loads.
editorMouseUp, editorMouseDown, editorDoubleClick, editorKeyUp, editorKeyDown	Fires in response to the corresponding Dom event.
beforeExecCommand, afterExecCommand	Fires at the beginning/end of the execCommand process. Reference YAHOO.util.Element.html#addListener for more details.
beforeOpenWindow, afterOpenWindow	Fires before/after an editor window is opened.
closeWindow	Fires after an editor window is closed.
toolbarExpanded, toolbarCollapsed	Fires when toolbar is expanded/collapsed via the collapse button.
buttonClick	Fires when a toolbar button receives a click event.

All Editor events are Custom Events (see Element docs); subscribe to these events using their subscribe method: `oEditor.on('afterNodeChange', fnMyHandler);`

Key Rich Text Editor Configuration Options

See online docs for complete list of Rich Text Editor configuration options.

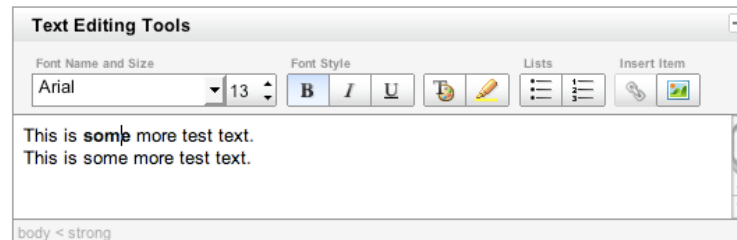
Option (type)	Default	Description
height, width	best guessed size of textarea	The height/width of the editor iframe container not including the toolbar.
animate	false	Indicates whether or not the editor should animate movements.
disabled	false	Toggle for the editor's disabled state. When disabled, design mode is off and a mask is placed over the iframe so no interaction can take place.
dompath	false	Toggles the display of the current Dom path below the editor.
toolbar	See editor.js.html	The default toolbar config.
handleSubmit	false	When true, the editor will attempt to attach a submit listener to the parent form that would trigger the editor's save handler and place the new content back into the textarea before the form is submitted.

Editor options can be set in the constructor's second argument (eg, `{height: '300px'}`) or at runtime via `set` (eg, `oEditor.set("height", "300px");`).

Constructor: YAHOO.widget.SimpleEditor

`YAHOO.widget.SimpleEditor(str | el ref container[, obj config])`

Creates a SimpleEditor instance with default configurations. SimpleEditor is a lighter version of the Editor Control.



YAHOO.widget.Editor:

Methods

render() Causes the toolbar and the editor to render and replace the textarea.

setEditorHTML(string html) Loads HTML into the editor's body.

getEditorHTML() Returns the unprocessed HTML from the editor.

saveHTML() Cleans the HTML with the cleanHTML method and places the string into the textarea.

cleanHTML(string html) Processes the HTML with a few regexes to clean it up and stabilize the output.

clearEditorDoc() Clears the editor doc.

destroy() Destroys the editor along with all of its elements and objects.

toString() Returns a string representing the Editor.

nodeChange() Handles toolbar setup, getting the Dom path, and fixing nodes.

execCommand(str command[, str arg]) Levels the differences in the support by various browsers of execCommand actions.

YAHOO.widget.Toolbar:

Methods

addButton(obj config) Add a new button to the toolbar.

addButtonGroup(obj config) Adds a new button group to the toolbar.

addButtonToGroup(obj config) Adds a new button group to a toolbar group.

addSeparator() Adds a new button separator to the toolbar.

getButtonByValue(str | obj command) Gets a button instance or a menuItem instance from the toolbar by its value.

disableButton(str | number | obj button) Disables a button in the toolbar.

enableButton(str | number | obj button) Enables a button in the toolbar.

selectButton(str | number | obj button) Selects a button in the toolbar.

deselectButton(str | number | obj button) Deselects a button in the toolbar.