

# Editor types - Classic vs Inline

Textbox.io is able to edit content in 2 distinct ways - "*Classic*" mode, where the editor appears in a self-contained box, and "*Inline mode*" (also known as *in-context mode*) where the editor literally takes an element and makes it editable. In inline mode, you edit content exactly as it appears within the context of the page it is being hosted upon.

## Inline vs classic mode - a quick comparison

Inline	Classic
<ul style="list-style-type: none"><li>• Inherits stylesheets from the <i>host page</i> the editor is invoked upon</li><li>• Provides a 'true' WYSIWYG editing experience within the context of the host page</li><li>• Doesn't work on mobile (...yet...)</li><li>• Does not provide a fullscreen editing mode or source code view</li><li>• Container grows as the content inside grows</li><li>• Toolbar grows and can be dragged around the page</li></ul>	<ul style="list-style-type: none"><li>• Is self contained (within an iframe)</li><li>• Requires injection of outside CSS styles using the configuration API</li><li>• Works on mobile devices</li><li>• Editor toolbar is always visible (regardless of whether the editor is focused)</li><li>• Allows editing in fullscreen mode</li><li>• Editor container remains a fixed size</li></ul>



Inline editing does not currently work on Mobile devices. See: [Developing editor experiences optimized for mixed platforms](#)

## Demo - Classic Editing

## Demo - Inline Editing

## Which type of Editor should I choose?

It really depends on the kinds of problems you're looking to solve through the use of an editor.

**Classic Mode** is useful for generating content that requires isolation- such as form fields, or email compose fields.

**Inline Mode** is useful for editing content such as real-time WYSIWYG editing of complete web pages

## Inline-specific configuration options

### Toolbar visibility

The toolbar can be made invisible at initialization via the configuration object. Such functionality is useful when developing visually lightweight "minimal" applications.

### Toolbar draggability

By default, the user can drag the toolbar away from it's container object as if it were a floating window. This functionality may be disabled.

### Toolbar offset

An offset may be set on the toolbar to provide visual separation between the toolbar and the content being edited.

Note: the editor assumes a "pinned" behavior when it is attached to the content (i.e. when it has no offset). This gives it special properties (such as attaching it to the top of the viewport) when large blocks of content are scrolled. This functionality is no-longer executed when the toolbar has an offset from the container window.

## Features disabled for inline editing

The following features are disabled when inline editing is used (regardless of whether they are declared in configuration):

- Source code view
- Fullscreen editing

## See also

- [API Basics](#) - Learn the basics of using the editor API, such as Getting and setting content in the editor.
- [Toolbar configuration](#) object
- [Customizing the Editor](#) - Learn how to configure and customize Textbox.io [for your applications](#).
- [Spell Checking](#) - Find out more about how to enhance the editing experience with spell checking, image upload and other web services from Tiny.
- [API reference](#)