# Editing mode with contenteditable, contentEditable and designMode

In October 1998 the World Wide Web Consortium (W3C) published the „Document Object Model (DOM) Level 1 Specification". This specification includes an API on how to alter DOM nodes and the document's tree[1]. This effectively provided a standardized way for changing a website's contents. With the implementations of Netscape's JavaScript and Microsoft's JScript this API has been made accessible to developers.

In July 2000, with the release of Internet Explorer 5.5, Microsoft introduced the contentEditable and designMode IDL attributes along with the contenteditable content attribute[2]. These attributes were not standardized and not part of the W3C DOM specifications.

| Attribute | Type | Can be set to | Possible values |
|---|---|---|---|
| designMode | IDL attribute | Document | „on", „off" |
| contentEditable | IDL attribute | Specific HTMLElements | boolean, „true", „false", „ inherit" |
| contenteditable | content attribute | Specific HTMLElements | empty string, „true", „false" |

By settings these settings to „true" (contenteditable / contentEditable) or „on" (designMode), Internet Explorer switches the affected elements and their children to an editing mode. In editing mode it is possible to

1. Let the user interactively click on and type inside text elements
2. Execute „commands" via JScript and JavaScript

Fetching user inputs (clicking on elements, accepting keyboard input and modifying text nodes) is handled entirely by the browser. No further scripting is necessary other than setting the mentioned attributes on elements. This behavior is inherited by child elements.

In editing mode, calling the method document.execCommand will format the currently selected text depending on the parameters passed. Calling document.execCommand('bold', false, null) will wrap the currently selected text in <b> tags. document.execCommand('createLink', false, 'http://google.com/') will wrap the selected text in a link to google.com. However, this command will be ignored, if the current selection is not contained by an element in editing mode.

While designMode can only be applied to the entire document, contentEditable and contenteditable attributes can be applied to a subset of HTML elements as described on Microsoft's Developer Network (MSDN) online documentation[2].

With the release of Internet Explorer 5.5 and the introduction of editing capabilities, Microsoft released a sparse documentation describing only the availability and the before-mentioned element restrictions of these attributes[2].

According to Mark Pilgrim, author of the „Dive into" book series and contributor to the the Web Hypertext Application Technology Working Group (WHATWG), Microsoft did not state a specific purpose for its editing API, but, its first use-case has been rich text editing[3].

In March 2003, the Mozilla Foundation introduced an implementation of Microsoft's designMode, named Midas, for their release of Mozilla 1.3. Mozilla names this „rich-text editing support" on the Mozilla Developer Network (MDN)[4]. In June 2008, Mozilla added support for contentEditable IDL and contenteditable content attributes with Firefox 3.

Mozilla's editing API resembles the API implemented for Internet Explorer, however, there are still differences (compare [5][6]). Most notably, Microsoft and Mozilla differ in the commands provided to pass to document.execCommand[6][7] and the markup generated by invoking commands[8]. In fact, Mozilla only provides commands dedicated to text editing while Microsoft offers a way to access lower-level browser components (like the browser's cache) using execCommand. This may show, that even though rich-text editing was its first use case and Mozilla implemented it naming it that, this editing API was not originally intended to be used as such.

[1] http://www.w3.org/TR/REC-DOM-Level-1/level-one-core.html

[2] https://msdn.microsoft.com/en-us/library/ms537837(VS.85).aspx

[3] https://blog.whatwg.org/the-road-to-html-5-contenteditable

[4] https://developer.mozilla.org/en/docs/Rich-Text_Editing_in_Mozilla

[5] https://msdn.microsoft.com/en-us/library/hh772123(v=vs.85).aspx

[6] https://developer.mozilla.org/en-US/docs/Midas

[7] https://msdn.microsoft.com/en-us/library/ms533049(v=vs.85).aspx

[8] https://developer.mozilla.org/en/docs/Rich-Text_Editing_in_Mozilla#Internet_Explorer_Differences