



## Supported HTML tags

This section lists the built-in HTML tags supported by Flash Player. You can also create new styles and tags using Cascading Style Sheets; see [Formatting text with Cascading Style Sheets](#).

### Anchor tag (<a>)

The <a> tag creates a hyperlink and supports the following attributes:

- **href** Specifies the URL of the page to load in the browser. The URL can be absolute or relative to the location of the SWF file that is loading the page.
- **target** Specifies the name of the target window to load the page into.

For example, the following HTML snippet creates the link "Go home," which opens [www.macromedia.com](http://www.macromedia.com) in a new browser window.

```
<a href="../../home.htm" target="_blank">Go home</a>
```

You can also define `a:link`, `a:hover`, and `a:active` styles for anchor tags by using style sheets. See [Styling built-in HTML tags](#).

### Bold tag (<b>)

The <b> tag renders text as bold. A bold typeface must be available for the font used to display the text.

```
<b>This is bold text.</b>
```

### Break tag (<br>)

The <br> tag creates a line break in the text field, as shown in this example:

```
One line of text<br>Another line of text<br>
```

### Font tag (<font>)

The <font> tag specifies a font or list of fonts to display the text.

The font tag supports the following attributes:

- **color** Only hexadecimal color (`#FFFFFF`) values are supported. For example, the following HTML code creates red text.

```
<font color="#FF0000">This is red text</font>
```

- **face** Specifies the name of the font to use. You can also specify a list of comma-separated font names, in which case Flash Player chooses the first available font. If the specified font is not installed on the playback system, or isn't embedded in the SWF file, then Flash Player chooses a substitute font.

Example:

```
<font face="Times, Times New Roman">This is either Times or Times New Roman..</font>
```

For more information on embedding fonts in Flash applications, see [TextField.embedFonts](#) and [Setting dynamic and input text options](#).

- **size** Specifies the size of the font, in pixels. You can also use relative point sizes (+2 or -4).

```
<font size="24" color="#0000FF">This is green, 24-point text</font>
```

### Image tag (<img>)

The <img> tag lets you embed external JPEG files, SWF files, and movie clips inside text fields. Text automatically flows around images you embed in text fields. This tag is supported only in dynamic and input text fields that are multiline and wrap their text.

#### ► To create a multiline text field with word wrapping, do one of the following:

The <img> tag has one required attribute, `src`, which specifies the path to a JPEG file, a SWF file, or the linkage identifier of a movie clip symbol. All other attributes are optional.

The <img> tag supports the following attributes:

- **src** Specifies the URL to a JPEG or SWF file, or the linkage identifier for a movie clip symbol in the library. This attribute is required; all other attributes are optional. External files (JPEG and SWF files) are not displayed until they have downloaded completely.

**Note:** Flash Player does not support progressive JPEG files.

- **id** Specifies the name for the movie clip instance (created by Flash Player) that contains the embedded JPEG file, SWF file, or movie clip. This is useful if you want to control the embedded content with ActionScript.
- **width** The width of the image, SWF file, or movie clip, in pixels.
- **height** The height of the image, SWF file, or movie clip being inserted, in pixels.
- **align** Specifies the horizontal alignment of the embedded image within the text field. Valid values are `left` and `right`. The default value is `left`.
- **hspace** Specifies the amount of horizontal space that surrounds the image where no text will appear. The default value is 8.
- **vspace** Specifies the amount of vertical space that surrounds the image where no text will appear. The default value is 8.

For more information and examples of using the `<img>` tag, see [Embedding images, SWF files, and movie clips in text fields](#).

### Italic tag (`<i>`)

The `<i>` tag displays the tagged text in italics. An italic typeface must be available for the font used.

That is very `<i>interesting</i>`.

The above code would render as follows:

That is very *interesting*.

### List item tag (`<li>`)

The `<li>` tag places a bullet in front of the text that it encloses.

```
Grocery list:
<li>Apples</li>
<li>Oranges</li>
<li>Lemons</li>
```

The above code would render as follows:

Grocery list:

- Apples
- Oranges
- Lemons

### Paragraph tag (`<p>`)

The `<p>` tag creates a new paragraph. It supports the following attributes:

- **align** Specifies alignment of text within the paragraph; valid values are `left`, `right`, and `center`.
- **class** Specifies a CSS style class defined by an `TextField.StyleSheet` object. (For more information, see [Using style classes](#).)

The following example uses the `align` attribute to align text on the right side of a text field.

```
textField.htmlText = "<p align='right'>This text is aligned on the right side of the text fie
```

The following example uses the `class` attribute to assign a text style class to a `<p>` tag.

```
var myStyleSheet = new TextField.StyleSheet();
myStyleSheet.createTextField("test", 10, 0,0, 300,100);
createTextField("test", 10, 0,0, 300,100);
test.styleSheet = myStyleSheet;
test.htmlText = "<p class='body'>This is some body-styled text.</p>.";
```

### Span tag (`<span>`)

The `<span>` tag is available only for use with CSS text styles. (For more information, see [Formatting text with Cascading Style Sheets](#).) It supports the following attribute:

- **class** Specifies a CSS style class defined by an `TextField.StyleSheet` object. For more information on creating text style classes, see [Using style classes](#).

### Text format tag (`<textformat>`)

The `<textformat>` tag lets you use a subset of paragraph formatting properties of the `TextFormat` class within HTML text fields, including line leading, indentation, margins, and tab stops. You can combine `<textformat>`

tags with the built-in HTML tags.

The `<textformat>` tag has the following attributes:

- **blockindent** Specifies the block indentation in points; corresponds to `TextFormat.blockIndent`. (See [TextFormat.blockIndent](#).)
- **indent** Specifies the indentation from the left margin to the first character in the paragraph; corresponds to `TextFormat.indent`. (See [TextFormat.indent](#).)
- **leading** Specifies the amount of leading (vertical space) between lines; corresponds to `TextFormat.leading`. (See [TextFormat.leading](#).)
- **leftmargin** Specifies the left margin of the paragraph, in points; corresponds to `TextFormat.leftMargin`. (See [TextFormat.leftMargin](#).)
- **rightmargin** Specifies the right margin of the paragraph, in points; corresponds to `TextFormat.rightMargin`. (See [TextFormat.rightMargin](#).)
- **tabstops** Specifies custom tab stops as an array of non-negative integers; corresponds to `TextFormat.tabStops`. (See [TextFormat.tabStops](#).)

The following code example uses the `tabstops` attribute of the `<textformat>` tag to create a table of data with boldfaced row headers, as shown below:

Name	Age	Department
Tim	32	IMD
Edwin	46	Engineering

► [To create a formatted table of data using tab stops:](#)

### Underline tag (`<u>`)

The `<u>` tag underlines the tagged text.

This text is `<u>`underlined`</u>`.

The above code would render as follows:

This text is underlined.

