

# Creating a Paragraph

## Table of Contents

- [Paragraph Formatting](#)

To insert a paragraph in a section of your document, use one of the following methods. All of the methods insert a paragraph at a specified point in an Element. An Element object represents an editable region in a document, such as a section or a table cell. The document itself is an Element (the Document class extends Element). Later we will add a paragraph to a Document object.

- `Paragraph Element.InsertParagraphAfter(NamedStyle style)`

Inserts an empty paragraph at the end of the specified Element. This method applies the specified [named style](#) to the paragraph.

- `Paragraph Element.InsertParagraphAfter(NamedStyle style, ParagraphFormatting props)`

Inserts an empty paragraph at the end of the specified Element. This method applies a specified [named style](#) and a set of formatting properties to the paragraph. The formatting properties will override the style's properties

- `Paragraph Element.InsertParagraphBefore(NamedStyle style)`

Inserts an empty paragraph at the beginning of the specified Element. This method applies the specified [named style](#) to the paragraph.

- `Paragraph Element.InsertParagraphBefore(NamedStyle style, ParagraphFormatting props)`

Inserts an empty paragraph at the beginning of the specified Element. This method applies a specified [named style](#) and a set of formatting properties to the paragraph. The formatting properties will override the style's properties

- `Paragraph Position.InsertParagraph(NamedStyle style)`

Inserts an empty paragraph at a specified position within an Element. A Position object represents a cursor. This method applies the specified [named style](#) to the paragraph.

Before adding a paragraph to your file, you must create a content region in which to insert the paragraph. Each editable region in a Word file is represented by an Element object, or an object that extends Element, such as a Document object.

```
WordApplication wordApp = new WordApplication();
Document doc = wordApp.Create();
```

To insert an empty paragraph, use one of the methods listed at the beginning of this section. All of the methods take a style argument, which specifies a [named style](#) to apply to the inserted paragraph. The style argument may be null. If it is null, the Normal style will be applied to the paragraph. The Normal style is the default style used by Microsoft Word. The following inserts an empty paragraph at the end of the document and applies the BodyText style to the paragraph:

```
Paragraph para = doc.InsertParagraphAfter(doc.Styles[NamedStyle.BuiltIn.BodyText]);
```

To add text to the paragraph, call one of the Element class's InsertTextAfter or InsertTextBefore methods. You can [apply a font](#) to the text by passing a Font object to InsertTextAfter or InsertTextBefore. The following lines create a Font object that we will use later when adding text to the paragraph:

```
Font paragraphFont = doc.CreateFont();
paragraphFont.FontName = "Times New Roman";
paragraphFont.FontSize = 10;
```

The method `InsertTextAfter` takes a string and inserts it at the end of the paragraph. The method's second parameter specifies a `Font` object (created above) to apply to the text.

```
para.InsertTextAfter("This document was created with OfficeWriter for Word. It
contains a single, left-justified paragraph.",
paragraphFont);
```

The `Paragraph` class provides numerous formatting properties. The following left-justifies the paragraph:

```
//--- If para is a Paragraph object
para.Formatting.TextJustification = ParagraphFormatting.Justification.Left;

//--- If myFormat is a ParagraphFormatting object
myFormat.TextJustification = ParagraphFormatting.Justification.Left;
```

You can access existing paragraphs through the `Element.Elements` property, for example:

```
//--- Open an existing Word file and get the first
//--- paragraph.
WordApplication wordApp = new WordApplication();
Document doc = wordApp.Open(@"C:\sample.doc");
Paragraph firstParagraph = (Paragraph)doc.GetElements(Element.Type.Paragraph)[0];
```

## Paragraph Formatting

To apply a font to text in a paragraph, create a `Font` object and pass it to `Paragraph.InsertTextAfter` or `Paragraph.InsertTextBefore`, as demonstrated above.

Other paragraph formatting can be applied through the `Paragraph` object or the `ParagraphFormatting` object. To create a `ParagraphFormatting` object, call `Document.CreateParagraphFormatting`:

```
WordApplication wordApp = new WordApplication();
Document doc = wordApp.Create();
ParagraphFormatting bodyTextFormatting = doc.CreateParagraphFormatting();
```

Next, set `ParagraphFormatting` properties.

```
//--- Set the amount of whitespace that should be
//--- placed above a paragraph in twips (1/1440 in).
bodyTextFormatting.SpaceBefore = 240;

//--- Display line numbers.
bodyTextFormatting.LineNumberingAllowed = true;
```

To assign the `ParagraphFormatting` object that you created to a paragraph, pass it to `Element.InsertParagraphAfter` or `Element.InsertParagraphBefore` when you create a new paragraph.

```
Paragraph para = doc.InsertParagraphAfter(doc.Styles[NamedStyle.BuiltIn.BodyText],  
    bodyTextFormatting);
```