

# **SimplyPDF-layout documentation**

## **Starting guide**

Creating a document	3
Adding texts, paragraphs and pages	3
Adding images and graphics	3
Text styles	3
Paragraph styles	3
Page styles	5
Adding headers and footers	6
Adding tables	6
Adding table of content	6
Adding document internal links	6
Adding footnotes	6
Using stops	6

## Creating a document

A PDF is created with:

```
Document document = Document.create();
```

and rendered with;

```
document.write(out);
```

## Adding texts, paragraphs and pages

A document created with no page style uses the default one: A4 portrait.

No need to add new pages, when an element does not fit into the page, a new page with same style than the previous one is created.

The same is applicable to lines. When a line does not fit into the page width, the text is automatically splitted on the last space that fits the page and the text that left is wrapped to a new line. This same principle is applicable indefinitely to all new lines created.

Of course, new lines and new pages can be created at any time.

Text can be <sup>super-scripted</sup> or <sub>sub-scripted</sub>.

A paragraph is created with:

```
Paragraph paragraph = document.createParagraph("Text of paragraph");
```



A new page is created with;

```
document.addPage();
```

Paragraph style (margins, first line margin, text color, size, font, etc.) can be defined using the constructor for most of parameters. For others, getters and setters can be used.

The same applies to most of the classes (Page, Text, etc.).

## Adding images and graphics

A paragraph can contain text but also images like this one:  and this one: .

## Text styles

A color can be defined for an entire paragraph but also for *a subset* of the same paragraph.

*This paragraph demonstrate that text can simply be put in **bold**, *italic* and underlined. Text size can also be changed within a paragraph.*

*Text styles needs to be used inside a "Text" object. A paragraph allow a simple String or Text object as parameter.*

## Paragraph styles

This paragraph has a margin of 20 on each side plus a first line margin of 20.

It's also composed of few lines that are part of this same paragraph, so this paragraph configuration is applied to all lines of it.

So, the first line is shifted by 40 pt from the left while other lines of the

paragraph are shifted by 20.

This paragraph also ends from 20 pt before page margins and have an top and bottom margin of 20 too.

Text of this paragraph has a size of 14 instead of 12 in a default one.

This paragraph is aligned to right. This shows that the first line margin is taken into account. So, the text will not be displayed until start of paragraph but with a space of the size of the first line margin. The others line will start with the default paragraph margin. This even works with new lines inside a paragraph. New lines will not be sensitive to first line of paragraph margin.

This paragraph is aligned to center with a first line margin. It demonstrate the same principles than the previous paragraph.

This paragraph shows that a vertical line spacing ratio can be applied between each line of a paragraph. The standard line spacing is just the size of the font. The line spacing ratio is the multication of itself by the font size. So if line spacing is set to 2, an empty space of the size of the font is left blank between two lines. Both the lines wrapped automatically and

The ones created specifically

will be affected by vertical line spacing ratio.



## Page styles

This page show a page with a different orientation and size.  
You can use default ones or create the ones you need.

## Adding headers and footers

Headers and footers can be set once for all for every page or when adding a new page.

```
PageHeader pageHeader = new PageHeader();
pageHeader.addElement(document.createParagraph("SimplyPDF-layout - web4enterprise"));
document.addPage(pageHeader, pageFooter);
```

## Adding tables

How to ...	Code
Create a table	<code>Table table = new Table();</code>
Add a row	<code>table.addRow(TableCell...)</code>
Add a cell to a row	<code>table.createTableCell("text of cell")</code>
Control column width	<code>table.setColumnWidth(columnIndex, width);</code>
Add table to page	<code>document.addElement(table)</code>
Merge rows	<code>cell.setMergedRows(nbRows)</code>
Merge columns	<code>cell.setMergedColumns(nbColumns)</code>
Repeat header on a new page	<code>setRepeatHeaderOnNewPage(true)</code>
Create a cell style	<code>TableCellStyle cellStyle = table.createTableCellStyle()</code>
Affect style to cell	<code>table.createTableCell(cellStyle, paragraph)</code>
Add footer	<i>Just create a new row with a different style.</i>

More information on tables in [table.pdf](#)

## Adding table of content

This still have to be coded.

## Adding document internal links

Document internal links can simply be added to any text.

An internal link target is nothing special, it's just a simple element like this one but is generally a title or a figure.

## Adding footnotes

A footnote can be added simply by adding a footnote object to a paragraph element.<sup>1 2</sup>

## Using stops

<sup>1</sup> Any paragraph with texts and images can be added to a footnote.

Footnotes indices are added to text automatically as super-scripts.

<sup>2</sup> Many foot-notes can be placed on the same text.

A stop can be added anywhere in a paragraph to:

- place text right to it.
- place text left to it.
- place text centered on it.