A PDF is created with:

```
Pdf pdf = new Pdf();
```

A PDF is written with:

```
pdf.write(outputStream);
```

A page is created with:

```
Page page1 = pdf.createPage(pageWidth, pageHeight);
```

A text with font times-roman (default font) is added with:

```
page1.addText(positionX, positionY, fontSize, "The text to display");
```

A text width can be calculated with:

```
Font.TIMES_ROMAN.getWidth(FontsVariant.PLAIN, fontSize, "The text to search width for.");
```

A text with another font and font variant is added with:

```
page1.addText(20, 700, 12, Font.COURIER.getVariant(FontStyle.BOLD), "The text to display");
```

Below are texts with different fonts, colors, sizes and underlines:

Times-Roman

Times-Roman Bold

Times-Roman Italic

Times-Roman Bold Italic

Courier

Courier Bold

Courier Italic

Courier Bold Italic

Helvetica

Helvetica Bold

Helvetica Italic

Helvetica Bold Italic

Σψμβολ

❊❁❐❆✍✤❉■❇❂❁▼▲

An image is created with:

```
Image image = pdf.createImage(this.getClass().getResourceAsStream("/logo.png"));
```

An image is sized and positioned with:

```
image.setX(20);

image.setY(70);

image.setWidth(60);

image.setHeight(30);
```

An image is added with:

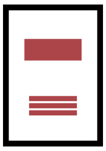
```
page1.addImage(image);
```

When an image is used multiple times, no need to re-create its data.

Just save space in PDF by cloning it’s references using:

```
Image image2 = image.cloneReference();
```

Change its size, position, etc and draw it again.

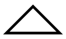


A straight line can be added with:

```
page2.addPath(new StraightPath(new Point(20, 640), new Point(150, 650)));
```

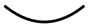
A line can be added with more points using:

```
StraightPath straightPath = new StraightPath(new Point(20, 510), new Point(20, 500), new Point(150, 500));

straightPath.setClosed(true);

page2.addPath(straightPath);
```



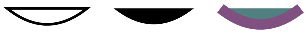
A Bezier line can be added with:

```
page2.addPath(new BezierPath(new Point(20, 440), new BezierPoint(50, 440, 30, 432, 40, 432))
```



Straight and Bezier lines can be filled/stroked/closed with:

```
BezierPath bezierPath2 = new BezierPath(new Point(20, 400), new BezierPoint(50, 440, 30, 432, 40, 432));

bezierPath2.setFillColor(new Color(80, 128, 128));

bezierPath2.setFilled(true);

bezierPath2.setStrokeColor(new Color(128, 80, 128));

bezierPath2.setLineWidth(4.0f);

bezierPath2.setClosed(true);

page2.addPath(bezierPath2);
```



click on this text to see internal link in action.

```
Text linkedText = new Text(20, 200, 12, "click on this text to see internal link in action.");

linkedText.setLink(image);
```

$^{\text{Super}}$ and $_{\text{sub}}$ -scriptedtext

```
text.setScript(TextScript.SUPER);

text.setScript(TextScript.SUB);
```

A character not supported (like this one: à•) does not compute width correctly and logs a warning.

Color is computed not when text is added (or this text should be black) but when pdf is written.

So be sure to control your object references.

Document meta-data can be added with setter on document (like document.setAuthor("author")).

Or specific meta-data can be added with document.addMetaData(name, value).

Key words are added with pdf.addKeyword(keyword)