

iText를 이용한 PDF파일 만들기

iText 는 [Java](#) 및 .NET 에서 [PDF](#) 파일 을 만들고 조작하기 위한 [라이브러리](#) 입니다 .

iText는 Bruno Lowagie가 작성했습니다. 소스 코드는 처음에 [Mozilla Public License](#) 또는 GNU Library General Public License 오픈 소스 라이선스에 따라 오픈 소스로 배포되었습니다 . 그러나 버전 5.0.0(2009년 12월 7일 릴리스) 현재 [Afero General Public License](#) 버전 3에 따라 배포됩니다. iText의 LGPL/MPL 라이선스 버전의 포크는 현재 GitHub 에서 [OpenPDF](#) 라이브러리 로 활발하게 유지 관리되고 있습니다. [3] iText는 iText Software NV에서 배포한 [독점 라이선스](#) 를 통해서도 사용할 수 있습니다 .

iText는 최첨단 PDF에 대한 지원 등의 기능을 제공 [PKI](#)는 서명, 40 비트 및 128 비트 암호화 기반 [컬러 보정](#) , 태그의 PDF, PDF 형태 (AcroForms), PDF / X, [컬러 관리](#) 를 통해 [ICC](#)의 정보 및 [바코드](#) 되고 [Eclipse BIRT](#) , [Jasper Reports](#) , [JBoss Seam](#) , [Windward Reports](#) 및 [pdftk](#) 를 포함한 여러 제품 및 서비스에서 사용됩니다 . (위키백과 참조)

iText는 AGPL 라이선스로 사용하시되 소스를 모두 공개하시면 무료로 사용할 수 있고 그렇지 않은 경우 라이선스 구매 후 진행하셔야 합니다.

5버전과 7버전의 가장 큰 차이점은 5버전은 하나의 라이브러리 형태였다면 7버전은 모듈 형태로 세분화 되었다는 점 입니다. 좀 더 가볍게 필요한 기능만 의존관계를 추가하여 사용할 수 있습니다. 그리고 렌더링 방식이 개선되었습니다.

자세한 내용은 아래 링크에서 확인하시면 됩니다.

<https://itextpdf.com/en/blog/technical-notes/itext-7-and-itext-5-roadmaps-differences-updates>

1. 라이브러리 준비하기

<https://github.com/itext/itextpdf/releases/tag/5.5.13.2> 에서 아래의 파일들을 다운로드 받아 빌드 패스에 추가합니다.

```
itextpdf-5.5.13.2.jar
itext-pdfa-5.5.13.2.jar
itext-xtra-5.5.13.2.jar
commons-imaging-1.0-SNAPSHOT.jar
commons-io-2.4.jar
xmlworker-5.5.13.2.jar
itext-asian-5.2.0.jar
```

maven을 사용할 경우 pom.xml에 다음을 추가합니다.

```
<!-- properties 태그안에 iText버전을 등록합니다.-->
<properties>
    <itext.version>5.5.13.2</itext.version>
</properties>
<!-- dependencies 태그안에 라이브러리를 등록합니다. -->
<dependencies>
    <!-- itext5 -->
    <dependency>
```

```

        <groupId>com.itextpdf</groupId>
        <artifactId>itextpdf</artifactId>
        <version>${itext.version}</version>
    </dependency>
    <dependency>
        <groupId>com.itextpdf</groupId>
        <artifactId>itext-pdfa</artifactId>
        <version>${itext.version}</version>
    </dependency>
    <dependency>
        <groupId>com.itextpdf</groupId>
        <artifactId>itext-xtra</artifactId>
        <version>${itext.version}</version>
    </dependency>
    <dependency>
        <groupId>com.itextpdf.tool</groupId>
        <artifactId>xmlworker</artifactId>
        <version>${itext.version}</version>
    </dependency>
    <dependency>
        <groupId>com.itextpdf</groupId>
        <artifactId>itext-asian</artifactId>
        <version>5.2.0</version>
    </dependency>
</dependencies>

```

2. 비어있는 PDF파일을 만들어 보자.

다음은 지정 파일을 크롬을 이용하여 볼 수 있도록하는 유틸리티 파일이다.

```

import java.io.File;
import java.io.IOException;

public class Chromeview {
    public static void view(String destFileName) {
        String chrome = "C:\\\\Program Files
(x86)\\\\Google\\\\Chrome\\\\Application\\\\chrome.exe";
        try {
            File file = new File(destFileName); // 파일 객체 생성
            new ProcessBuilder(chrome, file.getAbsolutePath()).start(); // 프로세
스 시작
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

PDF 내의 측정 단위는 전통적인 그래픽 산업 측정 단위인 포인트 입니다. Adobe는 다음 정의를 사용합니다.

1 pt = 1/72 inch

1인치는 정확히 25.4mm로 정의되므로(정말로!) 공식을 사용하여 포인트에서 mm로 변환할 수 있습니다.

$$\text{mm} = \text{pt} * 25.4 / 72$$
$$\text{pt} = \text{mm} / 25.4 * 72$$

다음은 단위를 변경해주는 유틸리티 파일이다.

```
public class PtMmConverter {
    public static void main(String[] args) {
        // 595.27563, 841.8898
        System.out.println(mm2pt(210));
        System.out.println(mm2pt(297));

        System.out.println(pt2mm(595.27563) );
        System.out.println(pt2mm(841.8898) );

        System.out.println(pt2mm(mm2pt(210)));
        System.out.println(pt2mm(mm2pt(297)));
    }
    // point를 mm로 변환
    public static double pt2mm(double pt) {
        return pt * 25.4 / 72;
    }
    // mm를 point로 변환
    public static double mm2pt(double mm) {
        return mm / 25.4 * 72;
    }
}
```

이제 PDF파일을 만들어 보자

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex01_BlankPage {
    public static void main(String[] args) {
        // 지정 폴더가 존재하지 않으면 폴더를 자동 생성한다.
        File file = new File("pdf");
        if(!file.exists()) {
            file.mkdir();
        }
        String destFileName = "pdf/Ex01_BlankPage.pdf";
        Document document = null;
        try {
            // 1. 문서 작성
            document = new Document();
```

```

        // 2. 출력 지정
        PdfWriter.getInstance(document, new FileOutputStream(destFileName));
        // 3. 열기
        document.open();
        // 문서의 크기 출력
        getPageInfo(document);
        // 4. 빈 내용 작성
        document.add(new Chunk(""));
        // 5. 닫기
        document.close();
        // 작성된 PDF문서 확인하기
        ChromeView.view(destFileName);
    } catch (FileNotFoundException | DocumentException e) {
        e.printStackTrace();
    } finally {
        if(document!=null) document.close();
    }
}

private static void getPageInfo(Document document) {
    Rectangle rectangle = document.getPageSize(); // 만들어진 문서의 페이지 사이
    즈 구하기
    System.out.println(rectangle);
    // 폭과 높이를 mm로 변경해 보았다.
    double width = PtMmConverter.pt2mm(rectangle.getWidth());
    double height = PtMmConverter.pt2mm(rectangle.getHeight());
    // 크기 출력
    System.out.printf("크기 : (%.0f, %.0f)\n", width, height);
}
}

```

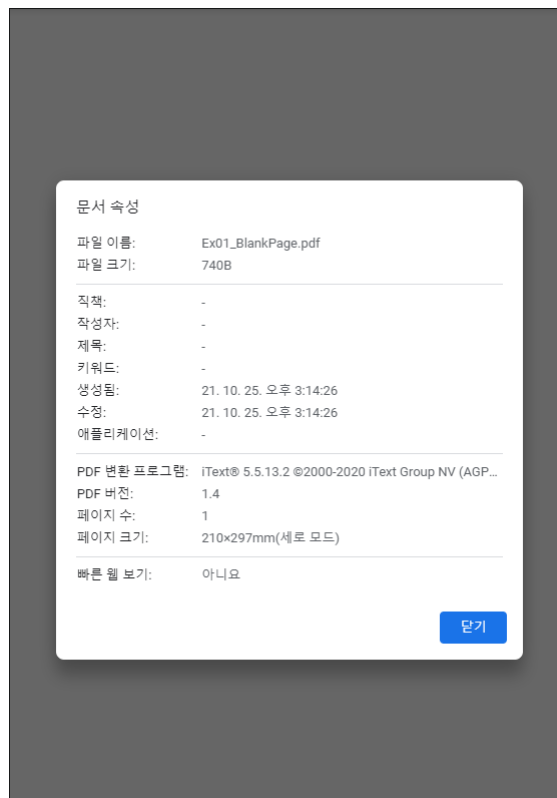
결과

```

RectangleReadOnly: 595.0x842.0 (rot: 0 degrees)
크기 : (210, 297)

```

크롬에서 문서 정보를 확인하면 다음과 같이 만들어진 문서의 정보를 확인 가능하다. 우측 상단의



3. PDF파일에 Hello World!!!를 출력해 보자.

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.net.MalformedURLException;

import com.itextpdf.text.BadElementException;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Image;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex02_Helloworld {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex02_Helloworld.pdf";
        Document document = null;
        try {
            // 1. 문서 작성
            document = new Document();
            // 2. 출력 지정
            PdfWriter writer = PdfWriter.getInstance(document, new
FileOutputStream(destFileName));
            // 3. 열기
            document.open();
            // 4. 문단 작성
            document.add(new Paragraph("Hello world!"));
            // 이미지 넣기
            String imageFile = "src/main/resources/logo.jpg";
            addImage(writer, imageFile, 10, 750);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        // 5. 닫기
        document.close();
        // 작성된 PDF문서 확인하기
        Chromeview.view(destFileName);
    } catch (FileNotFoundException | DocumentException e) {
        e.printStackTrace();
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if(document!=null) document.close();
    }
}
// 지정 위치에 이미지 넣기
private static void addImage(PdfWriter writer, String imageFile, float x,
float y)
    throws BadElementException, MalformedURLException, IOException,
DocumentException {
    PdfContentByte canvas = writer.getDirectContentUnder();
    Image image = Image.getInstance(imageFile);
    image.setAbsolutePosition(x, y);
    canvas.addImage(image);
}
}

```

이미지 파일



결과



4. PDF파일에 한글을 출력해 보자.

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex03_Helloworldkorean {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex03_Helloworldkorean.pdf";
        try {
            // 1. 문서 작성
            Document document = new Document();
            // 2. 출력 지정
            PdfWriter.getInstance(document, new FileOutputStream(destFileName));
            // 3. 열기
            document.open();
            // 4. 문단 작성
            // 4) 한글 입력을 위해 폰트를 선택해줍니다. iTextAsian.jar에서는 다음 3개의
            // 폰트를 사용할 수 있습니다.
            // HYGoThic-Medium, HYSMyeongJo-Medium, HYSMyeongJoStd-Medium
            String fontFace = "HYGoThic-Medium";
            String fontFace2 = "HYSMyeongJo-Medium";

            // 5) 글자 방향을 결정하는 CMap은 두가지가 있습니다.
            // UniKS-UCS2-H : 가로, UniKS-UCS2-V : 세로
            String fontNameH = "UniKS-UCS2-H";
            String fontNameV = "UniKS-UCS2-V";

            // 6) 준비한 설정값들을 활용해 Font 객체를 생성해줍니다. 생성자에 들어가는 인
            // 자는 BaseFont 와 사이즈 입니다.
            BaseFont bf = BaseFont.createFont(fontFace, fontNameH,
            BaseFont.NOT_EMBEDDED);
            BaseFont bf2 = BaseFont.createFont(fontFace2, fontNameV,
            BaseFont.NOT_EMBEDDED);
            Font font = new Font(bf, 10);
            Font font2 = new Font(bf2, 12);

            String content =
            Files.readString(Paths.get("src/main/resources/NationalAnthem2.txt"));
            document.add(new Paragraph(content, font));
            document.add(new Paragraph(" ", font));
            document.add(new Paragraph("안녕하세요 iText로 만든 PDF파일 입니다.",
            font));
            document.add(new Paragraph("안녕하세요 iText로 만든 PDF파일 입니다.",
            font2));

            // 5. 닫기
            document.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

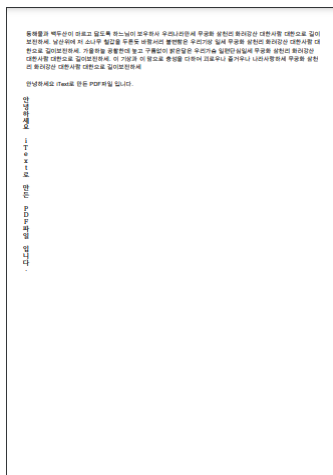
```
// 저장된 문서 보기
Chromeview.view(destFileName);
} catch (DocumentException | IOException e) {
    e.printStackTrace();
}
}
}
```

데이터 파일

NationalAnthem2.txt

동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리 화려강산 대한사
 람 대한으로 길이 보전하세. 남산위에 저 소나무 철갑을 두른듯 바람서리 불변함은 우리기상 일세
 무궁화 삼천리 화려강산 대한사람 대한으로 길이보전하세. 가을하늘 공활한데 높고 구름없이 밝은달
 은 우리가슴 일편단심일세 무궁화 삼천리 화려강산 대한사람 대한으로 길이보전하세. 이 기상과 이
 맘으로 충성을 다하여 괴로우나 즐거우나 나라사랑하세 무궁화 삼천리 화려강산 대한사람 대한으로
 길이보전하세.

결과



폰트 파일을 직접 사용하여 한글 문서를 만들어 보자.

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex04_Helloworldkorean2 {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex04_Helloworldkorean2.pdf";
        try {
            // 1. 문서 작성
            Document document = new Document();
```



```

// 2. 출력 지정
PdfWriter.getInstance(document, new FileOutputStream(destFileName));
// 3. 열기
document.open();
// 4. 문단 작성
// 폰트 파일을 이용한 폰트 작성
BaseFont bf =
BaseFont.createFont("font/NanumGothicCoding.ttf",BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
BaseFont bf2 =
BaseFont.createFont("font/stkaiti.ttf",BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
Font font = new Font(bf, 10);
Font font2 = new Font(bf2, 12);

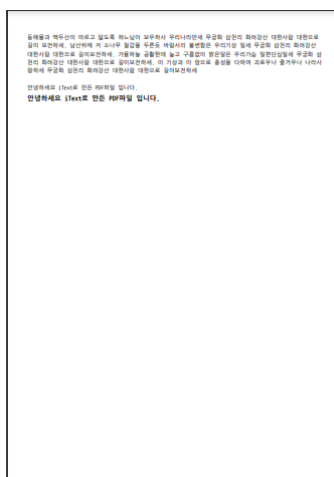
String content =
Files.readString(Paths.get("src/main/resources/NationalAnthem2.txt"));
document.add(new Paragraph(content, font));
document.add(new Paragraph(" ", font));
document.add(new Paragraph("안녕하세요 iText로 만든 PDF파일 입니다.",
font));
document.add(new Paragraph("안녕하세요 iText로 만든 PDF파일 입니다.",
font2));

// 5. 닫기
document.close();

// 저장된 문서 보기
ChromeView.view(destFileName);
} catch (DocumentException | IOException e) {
e.printStackTrace();
}
}
}

```

결과



5. 여러 페이지 문서를 만들어 보고 페이지 번호도 붙여 보자.

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.ColumnText;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfReader;
import com.itextpdf.text.pdf.PdfStamper;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex05_HelloWorldMultiplePages {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex05_HelloWorldMultiplePages.pdf";
        try {
            // 1. 문서 작성
            Document document = new Document();
            // 2. 출력 지정
            PdfWriter.getInstance(document, new FileOutputStream(destFileName));
            // 3. 열기
            document.open();
            // 4. 문단 작성
            BaseFont bf =
                BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
                    BaseFont.EMBEDDED);
            Font font = new Font(bf, 15);

            String content =
                Files.readString(Paths.get("src/main/resources/NationalAnthem2.txt"));
            for(int i=1; i<=4; i++) {
                document.add(new Paragraph(i + "번째 페이지 입니다.", font));
                document.add(new Paragraph(" ", font));
                document.add(new Paragraph(content, font));
                document.newPage();
            }
            // 5. 닫기
            document.close();

            // 페이지 번호 달기
            PdfReader reader = new PdfReader(destFileName);
            int n = reader.getNumberOfPages();
            String destFileName2 = "pdf/Ex05_HelloWorldMultiplePages2.pdf";
            PdfStamper stamper = new PdfStamper(reader, new
                FileOutputStream(destFileName2));
            PdfContentByte pagecontent;
            for (int i = 0; i < n; ) {
                pagecontent = stamper.getOverContent(++i);
                ColumnText.showTextAligned(pagecontent,
```

```

        Element.ALIGN_RIGHT, new Phrase(String.format("%s/%s
Page", i, n)), 559, 806, 0);
    }
    stamper.close();
    reader.close();
    // 저장된 문서 보기
    // ChromeView.view(destFileName);
    Chromeview.view(destFileName2);
} catch (DocumentException | IOException e) {
    e.printStackTrace();
}
}
}
}

```

결과

<p>1번째 페이지입니다.</p> <p>1/4 Page</p> <p>동해물과 백두산이 마르고 닳도록 한느니라 부루허사 우리나라에는 무궁화 삼천의 화려강산 대천사할 대천으로 길러 보관하세, 남산위에 저 소나무 칠엽을 푸른듯 여왕사의 불연꽃은 우리기삼 들에 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 가평하늘 공룡하늘 높고 구름같이 푸른 달은 우리기삼 일편단심될세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 이 기쁘고 이 일기로 충성을 다하여 고조우사 불가무사 나 라시말하세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세</p>	<p>2번째 페이지입니다.</p> <p>2/4 Page</p> <p>동해물과 백두산이 마르고 닳도록 한느니라 부루허사 우리나라에는 무궁화 삼천의 화려강산 대천사할 대천으로 길러 보관하세, 남산위에 저 소나무 칠엽을 푸른듯 여왕사의 불연꽃은 우리기삼 들에 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 가평하늘 공룡하늘 높고 구름같이 푸른 달은 우리기삼 일편단심될세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 이 기쁘고 이 일기로 충성을 다하여 고조우사 불가무사 나 라시말하세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세</p>
<p>3번째 페이지입니다.</p> <p>3/4 Page</p> <p>동해물과 백두산이 마르고 닳도록 한느니라 부루허사 우리나라에는 무궁화 삼천의 화려강산 대천사할 대천으로 길러 보관하세, 남산위에 저 소나무 칠엽을 푸른듯 여왕사의 불연꽃은 우리기삼 들에 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 가평하늘 공룡하늘 높고 구름같이 푸른 달은 우리기삼 일편단심될세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 이 기쁘고 이 일기로 충성을 다하여 고조우사 불가무사 나 라시말하세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세</p>	<p>4번째 페이지입니다.</p> <p>4/4 Page</p> <p>동해물과 백두산이 마르고 닳도록 한느니라 부루허사 우리나라에는 무궁화 삼천의 화려강산 대천사할 대천으로 길러 보관하세, 남산위에 저 소나무 칠엽을 푸른듯 여왕사의 불연꽃은 우리기삼 들에 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 가평하늘 공룡하늘 높고 구름같이 푸른 달은 우리기삼 일편단심될세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세, 이 기쁘고 이 일기로 충성을 다하여 고조우사 불가무사 나 라시말하세 무궁화 삼천의 화려강산 대천사할 대천으로 길러보관하세</p>

6. PDF문서에 텍스트를 추가해 보자.

텍스트의 입력에 있어서 3개의 클래스를 가지고 있다.

Chunk < Phrase < Paragraph 를 가지고 있으며 Chunk 가 가장 작은 단위의 표현이다.

```

import java.io.FileOutputStream;

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Font;
import com.itextpdf.text.FontFactory;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

```

```

// 텍스트의 입력에 있어서 3개의 클래스를 가지고 있다.
// Chunk - Phrase - Paragraph 를 가지고 있으며 chunk 가 가장 작은 단위의 표현이다.
public class Ex06_ChunkPhraseParagraph {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex06_ChunkPhraseParagraph.pdf";
        try {
            Document document = new Document(PageSize.A4, 20, 20, 30, 30); // 페이지
            // 크기 지정
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            pdfWriter.setInitialLeading(26); // 행간지정

            document.open();
            // Chunk - Phrase - Paragraph 차이점
            document.add(new Chunk("Hello",
            FontFactory.getFont(FontFactory.COURIER, 12, Font.BOLD)));
            document.add(new Phrase("PDF",
            FontFactory.getFont(FontFactory.COURIER, 12, Font.ITALIC))); // 옆으로
            document.add(new Paragraph("WORLD",
            FontFactory.getFont(FontFactory.COURIER, 12, Font.NORMAL))); // 줄분리
            // 줄바꿈
            document.add(Chunk.NEWLINE);

            // 한글 폰트로 변경
            BaseFont baseFont = BaseFont.createFont("HYGoThic-Medium", "UniKS-
            UCS2-H", BaseFont.NOT_EMBEDDED);
            Font font = new Font(baseFont, 10);
            BaseFont baseFont2 =
            BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
            BaseFont.EMBEDDED);
            Font font2 = new Font(baseFont2, 25);
            font2.setColor(BaseColor.RED);

            // 조각 만들고
            Chunk chunk1 = new Chunk("하나", font);
            chunk1.setBackground(BaseColor.PINK);

            Chunk chunk2 = new Chunk("둘", font2);
            chunk2.setBackground(BaseColor.PINK);

            Chunk chunk3 = new Chunk("셋", font2);
            chunk3.setBackground(BaseColor.CYAN);

            // 조각을 이용하여 구절만들고
            Phrase phrase1 = new Phrase();
            phrase1.add(chunk1);
            phrase1.add(chunk2);

            Phrase phrase2 = new Phrase();
            phrase2.add(chunk3);

            // 구절을 이용하여 문단을 만들고
            Paragraph paragraph = new Paragraph();
            paragraph.add(phrase1);
            paragraph.add(phrase2);

            // 문서에 추가 한다.

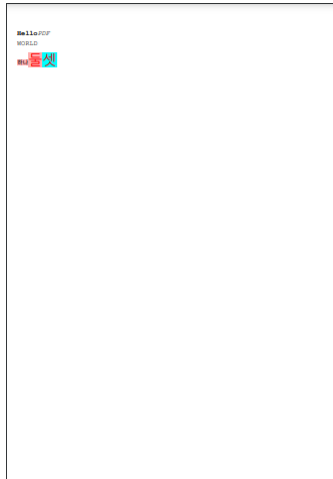
```

```

        document.add(paragraph);
        // 닫기
        document.close();
        // 작성된 문서 보기
        ChromeView.view(destFileName);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

결과



7. Chunk 배경색/글자색 적용

```

import java.io.FileOutputStream;

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Font;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex07_ChunkBackground {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex07_ChunkBackground.pdf";
        float fontSize = 25;
        float leading = 28;
        try {
            Document document = new Document(PageSize.A4, 20, 20, 30, 30); // 페이지
            // 사이즈와 좌우상하 여백
            PdfWriter pdfwriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            pdfwriter.setInitialLeading(leading); // 행간지정

            document.open();
            // 한글 폰트로 변경
            BaseFont baseFont1 = BaseFont.createFont("HYGoThic-Medium", "Uniks-
            UCS2-H", BaseFont.NOT_EMBEDDED);

```

```

        Font font1 = new Font(baseFont1, fontSize);
        font1.setColor(BaseColor.GREEN);
        BaseFont baseFont2 =
BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
        Font font2 = new Font(baseFont2, fontSize);
        font2.setColor(BaseColor.MAGENTA);

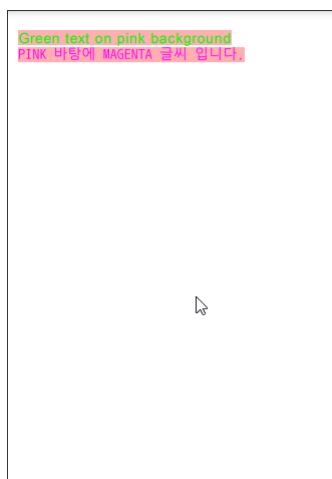
        // 지정한 font를 이용하여 chunk객체를 만듭니다.
        Chunk chunk1 = new Chunk("Green text on pink background", font1);
        // 배경색을 지정합니다.
        chunk1.setBackground(BaseColor.PINK);
        // 문서에 붙입니다.
        document.add(chunk1);
        // 빈줄을 삽입합니다.
        document.add(Chunk.NEWLINE);

        Chunk chunk2 = new Chunk("PINK 바탕에 MAGENTA 글씨 입니다.", font2);
        chunk2.setBackground(BaseColor.PINK);
        document.add(chunk2);
        document.add(Chunk.NEWLINE);

        // 닫기
        document.close();
        // 작성된 문서 보기
        ChromeView.view(destFileName);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

결과



8. 윗첨자/아랫첨자/특수문자 출력하기

```

import java.io.FileOutputStream;

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Document;
import com.itextpdf.text.Font;

```

```

import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex08_ChunkSubSuperscript {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex08_ChunkSubSuperscript.pdf";
        float fontSize = 25;
        float leading = 28;
        try {
            Document document = new Document(PageSize.A4, 20, 20, 30, 30); // 페이지
            // 크기, 좌우상하 여백
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            pdfWriter.setInitialLeading(leading);

            document.open();

            // 윗첨자, 아래첨자
            // https://www.htmlsymbols.xyz 이곳에서 코드값 확인이 가능하다.
            BaseFont baseFont1 = BaseFont.createFont("font/Cardo-Regular.ttf",
            BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
            Font font1 = new Font(baseFont1, fontSize);
            Paragraph paragraph1 = new Paragraph("H\u2082SO\u2074", font1);
            document.add(paragraph1);

            // Emoji Icon 출력
            BaseFont baseFont2 = BaseFont.createFont("font/NotoEmoji-
            Regular.ttf", BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
            Font font2 = new Font(baseFont2, fontSize);
            font2.setColor(BaseColor.RED);
            Paragraph paragraph2 = new
            Paragraph("\uD83D\uDC96\uD83D\uDE3B\uD83C\uDFAF", font2);
            document.add(paragraph2);

            document.close();
            // 저장된 문서 보기
            Chromeview.view(destFileName);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

결과

H₂SO₄
❤️🐱🐶

9. 기준선을 기준으로 텍스트 변위를 설정하기

```
import java.io.FileOutputStream;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Font;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex09_ChunkOrdinalNumbers {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex09_ChunkOrdinalNumbers.pdf";
        float fontSize = 10;
        float leading = 28;
        try {
            Document document = new Document(PageSize.A4, 20, 20, 30, 30); // 페이지
            PdfWriter pdfwriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            pdfwriter.setInitialLeading(leading);

            document.open();
            // Chunk 클래스는 기준선을 기준으로 텍스트 변위를 설정하는 setTextRise 메서
            드를 제공합니다.
            // 양수 값 상승 텍스트, 음수 값은 텍스트를 낮춥니다.
            Font small = new Font(FontFamily.HELVETICA, fontSize);
            Chunk st = new Chunk("st", small);
            st.setTextRise(7);
            Chunk nd = new Chunk("nd", small);
            nd.setTextRise(8);
            Chunk rd = new Chunk("rd", small);
            rd.setTextRise(9);
            Chunk th = new Chunk("th", small);
            th.setTextRise(10);

            Paragraph first = new Paragraph();
            first.add("The 1");
            first.add(st);
```



```

        first.add(" of May");
        document.add(first);

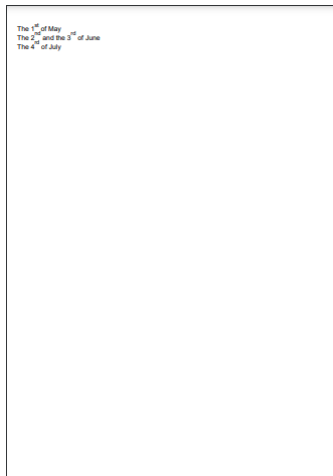
        Paragraph second = new Paragraph();
        second.add("The 2");
        second.add(nd);
        second.add(" and the 3");
        second.add(rd);
        second.add(" of June");
        document.add(second);

        Paragraph fourth = new Paragraph();
        fourth.add("The 4");
        fourth.add(rd);
        fourth.add(" of July");
        document.add(fourth);

        document.close();
        // 저장된 문서 보기
        Chromeview.view(destFileName);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

결과



10. FontFamily.SYMBOL 사용하기

```

import java.io.FileOutputStream;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Font;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex10_ChunkStandardDeviation {

```

```

public static void main(String[] args) {
    String destFileName = "pdf/Ex10_ChunkStandardDeviation.pdf";
    float fontSize = 18;
    float leading = 28;
    try {
        Document document = new Document(PageSize.A4, 20, 20, 30, 30); // 페이지
        PdfWriter pdfwriter = PdfWriter.getInstance(document, new
        FileOutputStream(destFileName));
        pdfwriter.setInitialLeading(leading);

        document.open();

        BaseFont baseFont =
        BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
        BaseFont.EMBEDDED);
        Font font = new Font(baseFont, fontSize);
        Chunk chunk = new Chunk("표준편차 기호는 Helvetica에 존재하지 않습니다.",
        font);
        document.add(chunk);

        document.add(Chunk.NEWLINE);

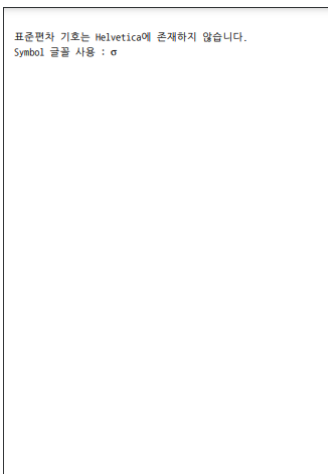
        Font symbol = new Font(FontFamily.SYMBOL);
        Phrase phrase = new Phrase("Symbol 글꼴 사용 : ", font);
        phrase.add(new Chunk("σ", symbol));

        document.add(phrase);

        document.close();
        // 저장된 문서 보기
        Chromeview.view(destFileName);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

결과



11. Chunk Bullets

```
import java.io.FileOutputStream;

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Font;
import com.itextpdf.text.Font.FontFamily;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex11_ChunkBullets {
    public static final String[] ITEMS = {
        "Insurance system", "Agent", "Agency", "Agent Enrollment",
        "Agent Settings",
        "Appointment", "Continuing Education", "Hierarchy",
        "Recruiting", "Contract",
        "Message", "Correspondence", "Licensing", "Party"
    };

    public static void main(String[] args) {
        String destFileName = "pdf/Ex11_ChunkBullets.pdf";
        float fontSize = 10;
        float leading = 28;
        try {
            Document document = new Document(PageSize.A4, 20, 20, 30, 30); // 페이지
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            pdfWriter.setInitialLeading(leading);

            document.open();

            Font zapfdingbats = new Font(FontFamily.ZAPFDINGBATS, 8);
            BaseFont baseFont =
            BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
            BaseFont.EMBEDDED);
            Font font = new Font(baseFont, fontSize);

            Chunk bullet = new Chunk(String.valueOf((char) 108), zapfdingbats);

            Paragraph paragraph = new Paragraph("아이템 ", font);
            document.add(paragraph);
            font.setSize(10);
            for (String item: ITEMS) {
                Paragraph paragraph1 = new Paragraph();
                paragraph1.add(bullet);
                paragraph1.add(new Phrase(" " + item , font));
                document.add(paragraph1);
            }
            document.add(Chunk.NEWLINE);

            font.setSize(15);
```

```

paragraph = new Paragraph("아이템 ", font);
document.add(paragraph);

font.setSize(10);
font.setColor(BaseColor.BLACK);

BaseFont baseFont2 = BaseFont.createFont("font/NotoEmoji-
Regular.ttf", BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
Font font2 = new Font(baseFont2, 10);
font2.setColor(BaseColor.RED);
for (String item: ITEMS) {
    Paragraph paragraph1 = new Paragraph("\ud83c\udfaf", font2);
    paragraph1.add(new Phrase(" " + item, font));
    document.add(paragraph1);
}
document.add Chunk.NEWLINE;
document.close();
// 저장된 문서 보기
Chromeview.view(destFileName);
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

결과



12. 정렬을 해보자.

```

import java.io.FileOutputStream;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;
import com.itextpdf.text.pdf.draw.VerticalPositionMark;

```

```

public class Ex12_Alignment {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex12_Alignment.pdf";
        float fontSize = 18;
        float leading = 28;
        try {
            Document document = new Document(PageSize.A4, 30, 30, 30, 30); // 페이지
            // 이지 사이즈와 좌우상하 여백
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            pdfWriter.setInitialLeading(leading);

            document.open();

            // 한글 폰트로 변경
            BaseFont baseFont1 = BaseFont.createFont("HYGoThic-Medium", "UniKS-
            UCS2-H", BaseFont.NOT_EMBEDDED);
            Font font1 = new Font(baseFont1, fontSize);
            BaseFont baseFont2 =
            BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
            BaseFont.EMBEDDED);
            Font font2 = new Font(baseFont2, fontSize);

            Chunk title= new Chunk("월간 보고서", font1);
            Phrase phrase1 = new Phrase();
            phrase1.add(title);
            Paragraph paragraph1 = new Paragraph();
            paragraph1.add(phrase1);
            paragraph1.setAlignment(Element.ALIGN_CENTER);
            document.add(paragraph1);

            font2.setSize(fontSize-8);
            Chunk subTitle = new Chunk("작성자 : 한사람", font2);
            Phrase phrase2 = new Phrase();
            phrase2.add(subTitle);
            Paragraph paragraph2 = new Paragraph();
            paragraph2.add(phrase2);
            paragraph2.setAlignment(Element.ALIGN_RIGHT);
            document.add(paragraph2);

            document.add(Chunk.NEWLINE);
            document.add(Chunk.NEWLINE);

            Chunk glue = new Chunk(new VerticalPositionMark());
            Paragraph paragraph3 = new Paragraph();
            paragraph3.add(new Chunk("왼쪽", font2));
            paragraph3.add(glue);
            paragraph3.add(new Chunk("가운데", font2));
            paragraph3.add(glue);
            paragraph3.add(new Chunk("오른쪽", font2));
            document.add(paragraph3);

            document.add(Chunk.NEWLINE);
            String nationalAnthem =
            Files.readString(Paths.get("src/main/resources/NationalAnthem2.txt"));
            Paragraph paragraph4 = new Paragraph(nationalAnthem, font2);
            paragraph4.setAlignment(Element.ALIGN_LEFT);

```

```

        document.add(paragraph4);
        document.add(Chunk.NEWLINE);
        paragraph4.setAlignment(Element.ALIGN_CENTER);
        document.add(paragraph4);
        document.add(Chunk.NEWLINE);
        paragraph4.setAlignment(Element.ALIGN_RIGHT);
        document.add(paragraph4);
        document.add(Chunk.NEWLINE);
        paragraph4.setAlignment(Element.ALIGN_JUSTIFIED);
        document.add(paragraph4);

        document.close();

        // 저장된 문서 보기
        Chromeview.view(destFileName);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

결과



```

import java.io.FileOutputStream;
import java.nio.file.Files;
import java.nio.file.Paths;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Paragraph;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex13_Alignment2 {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex13_Alignment2.pdf";
        try {
            Document document = new Document(PageSize.A4, 30, 30, 30, 30); // 페
            // 이지 사이즈와 좌우상하 여백

```

```

Pdfwriter.getInstance(document, new FileOutputStream(destFileName));

document.open();

// 한글 폰트로 변경
BaseFont baseFont =
BaseFont.createFont("font/NanumGothicCoding.ttf", BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
Font fontTitle = new Font(baseFont, 20, Font.BOLD|Font.UNDERLINE);
Font fontNormal = new Font(baseFont, 10);
Font fontBold = new Font(baseFont, 15, Font.BOLD);
// 애국가 읽기
String nationalAnthem =
Files.readString(Paths.get("src/main/resources/NationalAnthem2.txt"));

document.add(getParagraph("애국가", fontTitle, Element.ALIGN_CENTER));
document.add(Chunk.NEWLINE);

document.add(getParagraph("애국가(왼쪽정렬)", fontBold,
Element.ALIGN_CENTER));
document.add(getParagraph(nationalAnthem, fontNormal, 10,
Element.ALIGN_LEFT, 10));
document.add(Chunk.NEWLINE);

document.add(getParagraph("애국가(오른쪽정렬)", fontBold,
Element.ALIGN_CENTER));
document.add(getParagraph(nationalAnthem, fontNormal, 10,
Element.ALIGN_RIGHT, 20));
document.add(Chunk.NEWLINE);

document.add(getParagraph("애국가(가운데정렬)", fontBold,
Element.ALIGN_CENTER));
document.add(getParagraph(nationalAnthem, fontNormal, 10,
Element.ALIGN_CENTER, 30));
document.add(Chunk.NEWLINE);

document.add(getParagraph("애국가(양쪽정렬)", fontBold,
Element.ALIGN_CENTER));
document.add(getParagraph(nationalAnthem, fontNormal, 10,
Element.ALIGN_JUSTIFIED, 40));
document.add(Chunk.NEWLINE);

document.close();

// 저장된 문서 보기
ChromeView.view(destFileName);
} catch (Exception e) {
e.printStackTrace();
}
}
private static Paragraph getParagraph(String content, Font font,int align){
Paragraph paragraph = new Paragraph(content, font);
paragraph.setAlignment(align);
paragraph.setLeading(30);
return paragraph;
}
private static Paragraph getParagraph(String content, Font font,float
fontSize, int align, int leading){

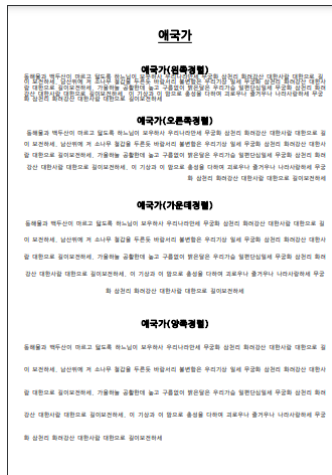
```

```

font.setSize(fontSize);
Paragraph paragraph = new Paragraph(content, font);
paragraph.setAlignment(align);
paragraph.setLeading(leading); // 행간
return paragraph;
}
}

```

결과



13. 절대좌표를 이용한 Text출력

```

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.BaseColor;
import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.Element;
import com.itextpdf.text.Font;
import com.itextpdf.text.Phrase;
import com.itextpdf.text.pdf.BaseFont;
import com.itextpdf.text.pdf.ColumnText;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex14_TextAtAbsolutePositions {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex14_TextAtAbsolutePositions.pdf";
        try {
            // 1. 문서 작성
            Document document = new Document();
            // 2. 출력 지정
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
FileOutputStream(destFileName));
            pdfWriter.setInitialLeading(12);

            // 3. 열기
            document.open();

```



```

        BaseFont baseFont1 =
BaseFont.createFont("font/NanumGothicCoding.ttf",BaseFont.IDENTITY_H,
BaseFont.EMBEDDED);
        BaseFont baseFont2 = BaseFont.createFont("font/NanumGothicCoding-
Bold.ttf",BaseFont.IDENTITY_H, BaseFont.EMBEDDED);
        Font font1 = new Font(baseFont1, 10);
        Font font2 = new Font(baseFont2, 12);

String text = "나는 어디에 나타날까요?";
PdfContentByte contentByte = pdfWriter.getDirectContent();
contentByte.saveState(); // 현재 상태 저장
contentByte.setLineWidth(0.05f); // 선의 폭
// 수직선 2개
contentByte.moveTo(400, 806);
contentByte.lineTo(400, 626);
contentByte.moveTo(508, 806);
contentByte.lineTo(508, 626);
// 수평선 5개
contentByte.moveTo(280, 788);
contentByte.lineTo(520, 788);
contentByte.moveTo(280, 752);
contentByte.lineTo(520, 752);
contentByte.moveTo(280, 716);
contentByte.lineTo(520, 716);
contentByte.moveTo(280, 680);
contentByte.lineTo(520, 680);
contentByte.moveTo(280, 644);
contentByte.lineTo(520, 644);
contentByte.stroke(); // 그리기
contentByte.restoreState(); // 상태복원
// 텍스트 출력 : PdfContentByte를 이용한 출력
contentByte.beginText();
contentByte.setFontAndSize(baseFont1, 12);
contentByte.showTextAligned(Element.ALIGN_LEFT, text+1, 400, 788,
0);
contentByte.showTextAligned(Element.ALIGN_RIGHT, text+2, 400, 752,
0);
contentByte.showTextAligned(Element.ALIGN_CENTER, text+3, 400, 716,
0);
contentByte.showTextAligned(Element.ALIGN_CENTER, text+4, 400, 680,
30); // 마지막 인수 회전 각도
contentByte.showTextAlignedKerned(Element.ALIGN_LEFT, text+5, 400,
644, 0);
contentByte.endText();

// 선그리기
contentByte.saveState();
contentByte.setLineWidth(0.05f);
// 수직선 2개
contentByte.moveTo(200, 590);
contentByte.lineTo(200, 410);
contentByte.moveTo(400, 590);
contentByte.lineTo(400, 410);
// 수평선 5개
contentByte.moveTo(80, 572);
contentByte.lineTo(520, 572);
contentByte.moveTo(80, 536);
contentByte.lineTo(520, 536);

```

```

        contentByte.moveTo(80, 500);
        contentByte.lineTo(520, 500);
        contentByte.moveTo(80, 464);
        contentByte.lineTo(520, 464);
        contentByte.moveTo(80, 428);
        contentByte.lineTo(520, 428);
        contentByte.stroke();
        contentByte.restoreState();
        // ColumnText.showTextAligned()을 이용한 텍스트 출력
        font2.setColor(BaseColor.BLUE);
        Phrase phrase = new Phrase(text+6, font2);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_LEFT, phrase,
200, 572, 0);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_RIGHT, phrase,
200, 536, 0);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_CENTER,
phrase, 200, 500, 0);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_CENTER,
phrase, 200, 464, 30);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_CENTER,
phrase, 200, 428, -30);

        // Chunk 속성을 변경한 출력
        Chunk chunk = new Chunk(text+7, font1);
        chunk.setHorizontalScaling(0.5f); // 크기 변경
        phrase = new Phrase(chunk);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_LEFT, phrase,
400, 572, 0);
        chunk = new Chunk(text+8, font1);
        chunk.setSkew(15, 15); // 비스듬한 정도
        phrase = new Phrase(chunk);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_LEFT, phrase,
400, 536, 0);
        chunk = new Chunk(text+9, font1);
        chunk.setSkew(0, 15);
        phrase = new Phrase(chunk);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_LEFT, phrase,
400, 500, 0);
        chunk = new Chunk(text+10, font1);
        chunk.setTextRenderMode(PdfContentByte.TEXT_RENDER_MODE_STROKE,
0.1f, BaseColor.RED); // 모드, 두께, 색상
        phrase = new Phrase(chunk);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_RIGHT, phrase,
400, 464, 0);
        chunk = new Chunk(text+11, font1);
        chunk.setTextRenderMode(PdfContentByte.TEXT_RENDER_MODE_FILL_STROKE,
2, BaseColor.BLUE);
        phrase = new Phrase(chunk);
        ColumnText.showTextAligned(contentByte, Element.ALIGN_CENTER,
phrase, 400, 428, -0);
        // 5. 닫기
        document.close();
        // 작성된 PDF문서 확인하기
        Chromeview.view(destFileName);
    } catch (FileNotFoundException | DocumentException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

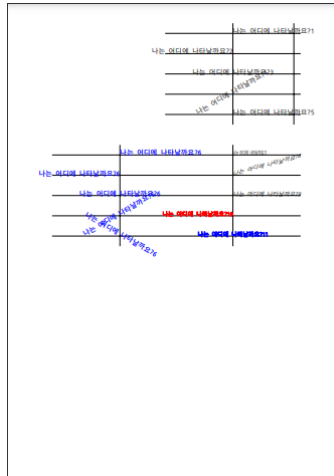
```

```

    }
}
}

```

결과



14. 선을 그려보자

```

import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex15_DrawLine {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex15_DrawLine.pdf";
        try {
            // 여백
            float marginLeft = 20, marginRight = 20, marginTop = 30,
            marginBottom = 30;
            Document document = new Document(PageSize.A4, marginLeft,
            marginRight, marginTop, marginBottom);
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            document.open();
            // 페이지 정보 출력
            getPageInfo(document);
            Rectangle rectangle = document.getPageSize();
            float width = rectangle.getWidth();
            float height = rectangle.getHeight();
            System.out.println(width + ", " + height);

            PdfContentByte contentByte = pdfWriter.getDirectContent();
            // 여백을 제외한 부분에 사각형그리기
            contentByte.saveState();
            contentByte.setRGBColorStroke(0xFF, 0x45, 0x00); // 선색상

```

```

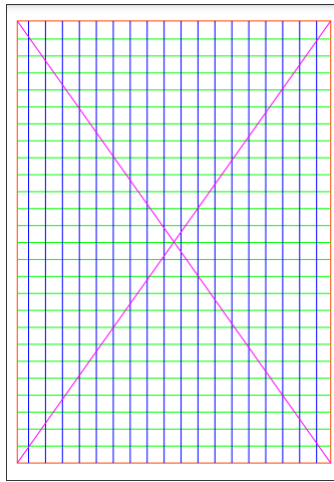
        contentByte.rectangle(marginLeft, marginBottom, width-
marginRight*2,height-marginTop*2); // 영역
        contentByte.stroke(); // 그리기
        contentByte.restoreState();
        // 수평선그리기
        contentByte.saveState();
        contentByte.setRGBColorStroke(0x00, 0xFF, 0x00); // 선색상
        for(float i = marginBottom*2;i<=height-marginTop*2; i+= 30) {
            contentByte.moveTo(marginLeft, i);
            contentByte.lineTo(width-marginRight, i);
        }
        contentByte.stroke(); // 그리기
        contentByte.restoreState();
        // 수직선그리기
        contentByte.saveState();
        contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선색상
        for(float i = marginLeft*2;i<=width-marginLeft; i+= 30) {
            contentByte.moveTo(i, marginTop);
            contentByte.lineTo(i, height-marginTop);
        }
        contentByte.stroke(); // 그리기
        contentByte.restoreState();
        // 대각선그리기
        contentByte.saveState();
        contentByte.setRGBColorStroke(0xFF, 0x00, 0xFF); // 선색상
        contentByte.moveTo(marginLeft, marginTop);
        contentByte.lineTo(width-marginRight, height-marginTop);
        contentByte.moveTo(marginLeft, height-marginTop);
        contentByte.lineTo(width-marginRight, marginTop);
        contentByte.stroke(); // 그리기
        contentByte.restoreState();

        document.close();
        // 작성된 PDF문서 확인하기
        Chromeview.view(destFileName);
    } catch (DocumentException | IOException e) {
        e.printStackTrace();
    }
}

private static void getPageInfo(Document document) {
    Rectangle rectangle = document.getPageSize();
    System.out.println(rectangle);
    double width = PtMmConverter.pt2mm(rectangle.getWidth());
    double height = PtMmConverter.pt2mm(rectangle.getHeight());
    System.out.printf("크기 : (%.0f, %.0f)\n", width, height);
}
}

```

결과



15. 사각형을 그려보자

```
import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex16_DrawRectangle {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex16_DrawRectangle.pdf";
        try {
            // 여백
            float marginLeft = 20, marginRight = 20, marginTop = 30,
            marginBottom = 30;
            Document document = new Document(PageSize.A4, marginLeft,
            marginRight, marginTop, marginBottom);
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            document.open();
            // 페이지 정보 출력
            getPageInfo(document);
            Rectangle rectangle = document.getPageSize();
            float width = rectangle.getWidth();
            float height = rectangle.getHeight();
            System.out.println(width + ", " + height);

            PdfContentByte contentByte = pdfWriter.getDirectContent();
            // 여백을 제외한 부분에 사각형그리기
            contentByte.saveState();
            contentByte.setRGBColorStroke(0xFF, 0x45, 0x00); // 선택상
            contentByte.rectangle(marginLeft, marginBottom, width-
            marginRight*2,height-marginTop*2); // 영역
            contentByte.stroke(); // 그리기
            contentByte.restoreState();

            contentByte.saveState();
```

```

        contentByte.setRGBColorStroke(0xFF, 0x00, 0x00); // 선색상
        contentByte.rectangle(200,height-200,100, 100); // 영역
        contentByte.stroke(); // 외곽선 그리기
        contentByte.restoreState();

        contentByte.saveState();
        contentByte.setRGBColorStroke(0xFF, 0x00, 0x00); // 선색상
        contentByte.setRGBColorFill(0xFF, 0x00, 0xFF); // 채우기 색상
        contentByte.rectangle(220,height-220, 100, 100); // 영역
        contentByte.fill(); // 배경그리기
        contentByte.restoreState();

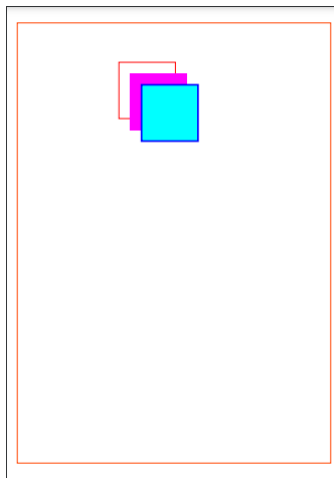
        contentByte.saveState();
        contentByte.setLineWidth(3); // 테두리 두께
        contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선색상
        contentByte.setRGBColorFill(0x00, 0xFF, 0xFF); // 채우기 색상
        contentByte.rectangle(240,height-240, 100, 100); // 영역
        contentByte.fillStroke(); // 외각선과 배경 모두 그리기
        contentByte.restoreState();

        document.close();
        // 작성된 PDF문서 확인하기
        Chromeview.view(destFileName);
    } catch (DocumentException | IOException e) {
        e.printStackTrace();
    }
}

private static void getPageInfo(Document document) {
    Rectangle rectangle = document.getPageSize();
    System.out.println(rectangle);
    double width = PtMmConverter.pt2mm(rectangle.getWidth());
    double height = PtMmConverter.pt2mm(rectangle.getHeight());
    System.out.printf("크기 : (%.0f, %.0f)\n", width, height);
}
}

```

결과



```

import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.Document;

```

```

import com.itextpdf.text.DocumentException;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex17_DrawRectangle2 {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex17_DrawRectangle2.pdf";
        try {
            // 여백
            float marginLeft = 20, marginRight = 20, marginTop = 30,
marginBottom = 30;
            Document document = new Document(PageSize.A4, marginLeft,
marginRight, marginTop, marginBottom); // 크기, 여백 좌, 우, 상, 하
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
FileOutputStream(destFileName));
            document.open();
            // 페이지 정보 출력
            getPageInfo(document);
            Rectangle rectangle = document.getPageSize();
            float width = rectangle.getWidth();
            float height = rectangle.getHeight();
            System.out.println(width + ", " + height);

            PdfContentByte contentByte = pdfWriter.getDirectContent();
            // 여백을 제외한 부분에 사각형그리기
            contentByte.saveState();
            contentByte.setRGBColorStroke(0xFF, 0x45, 0x00); // 선색상
            contentByte.rectangle(marginLeft, marginBottom, width-
marginRight*2,height-marginTop*2); // 영역
            contentByte.stroke(); // 그리기
            contentByte.restoreState();

            contentByte.saveState();
            contentByte.setLineWidth(2); // 테두리 두께
            contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선색상
            contentByte.setRGBColorFill(0xFF, 0xFF, 0x00); // 채우기 색상
            contentByte.rectangle(100,height-100, 50, 50); // 영역
            contentByte.fillStroke(); // 외각선과 배경 모두 그리기
            contentByte.rectangle(110,height-110, 50, 50); // 영역
            contentByte.stroke(); // 외각선 그리기
            contentByte.rectangle(120,height-120, 50, 50); // 영역
            contentByte.fill(); // 배경 그리기

            // concatCTM : 행렬을 현재 변환 행렬에 연결합니다.
            // 이동 (Translation) [1 0 0 1 tx ty]
            contentByte.setRGBColorFill(0xFF, 0x00, 0xFF); // 채우기 색상
            contentByte.concatCTM(1, 0, 0, 1, 50, -50); // 우측으로 50 아래로 50 이
동

            contentByte.rectangle(120,height-120, 50, 50); // 영역
            contentByte.fillStroke(); // 외각선과 배경 모두 그리기
            contentByte.restoreState();

            contentByte.saveState();
            // 크기 변환 (Scaling) : [sx 0 0 sy 0 0]
            contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선색상
            contentByte.setRGBColorFill(0x00, 0xFF, 0xFF); // 채우기 색상

```

```

contentByte.concatCTM(0.5f, 0, 0, 0.5f, 0, 0);
contentByte.rectangle(200,200, 50, 50); // 영역
contentByte.fill(); // 배경 그리기
contentByte.restoreState();

contentByte.saveState();
contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선색상
contentByte.setRGBColorFill(0xFF, 0xFF, 0x00); // 채우기 색상
contentByte.concatCTM(0.5f, 0, 0, 1f, 0, 0);
contentByte.rectangle(250, 100, 50, 50); // 영역
contentByte.fillStroke(); // 외각선과 배경 모두 그리기
contentByte.restoreState();

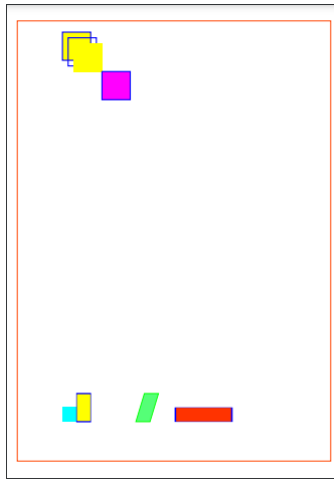
contentByte.saveState();
contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선색상
contentByte.setRGBColorFill(0xFF, 0x33, 0x00); // 채우기 색상
contentByte.concatCTM(2f, 0, 0, 0.5f, 0, 0);
contentByte.rectangle(150, 200, 50, 50); // 영역
contentByte.fillStroke(); // 외각선과 배경 모두 그리기
contentByte.restoreState();

/* Rotation: [cos(q) sin(q) -sin(q) cos(q) 0 0]
 * 90 degrees CCW: [0 1 -1 0 0 0]
 * 180 degrees: [-1 0 0 -1 0 0]
 * 270 degrees: [0 -1 1 0 0 0]
 * Skew: [1 tan(a) tan(b) 1 0 0]
 */
contentByte.saveState();
contentByte.setRGBColorStroke(0x00, 0xFF, 0x00); // 선색상
contentByte.setRGBColorFill(0x55, 0xFF, 0x77); // 채우기 색상
contentByte.concatCTM(1, 0, 0.3f, 1, 0, 0);
contentByte.rectangle(200, 100, 25, 50); // 영역
contentByte.fillStroke(); // 외각선과 배경 모두 그리기
contentByte.restoreState();

document.close();
// 작성된 PDF문서 확인하기
ChromeView.view(destFileName);
} catch (DocumentException | IOException e) {
    e.printStackTrace();
}
}
private static void getPageInfo(Document document) {
    Rectangle rectangle = document.getPageSize();
    System.out.println(rectangle);
    double width = PtMmConverter.pt2mm(rectangle.getWidth());
    double height = PtMmConverter.pt2mm(rectangle.getHeight());
    System.out.printf("크기 : (%.0f, %.0f)\n", width, height);
}
}

```

결과



16. 원을 그려보자

```
import java.io.FileOutputStream;
import java.io.IOException;

import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.PageSize;
import com.itextpdf.text.Rectangle;
import com.itextpdf.text.pdf.PdfContentByte;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex18_DrawCircle {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex18_DrawCircle.pdf";
        try {
            // 여백
            float marginLeft = 20, marginRight = 20, marginTop = 30,
            marginBottom = 30;
            Document document = new Document(PageSize.A4, marginLeft,
            marginRight, marginTop, marginBottom);
            PdfWriter pdfWriter = PdfWriter.getInstance(document, new
            FileOutputStream(destFileName));
            document.open();
            // 페이지 정보 출력
            getPageInfo(document);
            Rectangle rectangle = document.getPageSize();
            float width = rectangle.getWidth();
            float height = rectangle.getHeight();
            System.out.println(width + ", " + height);

            PdfContentByte contentByte = pdfWriter.getDirectContent();
            // 여백을 제외한 부분에 사각형그리기
            contentByte.saveState();
            contentByte.setRGBColorStroke(0xFF, 0x45, 0x00); // 선택색상
            contentByte.rectangle(marginLeft, marginBottom, width-
            marginRight*2,height-marginTop*2); // 영역
            contentByte.stroke(); // 그리기
            contentByte.restoreState();

            contentByte.saveState();
            contentByte.setLineWidth(2); // 테두리 두께
            contentByte.setRGBColorStroke(0x00, 0x00, 0xFF); // 선택색상
```

```

        contentByte.setRGBColorFill(0xFF, 0xFF, 0x00); // 채우기 색상
        contentByte.rectangle(200, 200, 5,5);
        contentByte.circle(200, 200, 50);
        contentByte.stroke();

        contentByte.rectangle(250, 250, 5,5);
        contentByte.circle(250, 250, 50);
        contentByte.fillStroke();

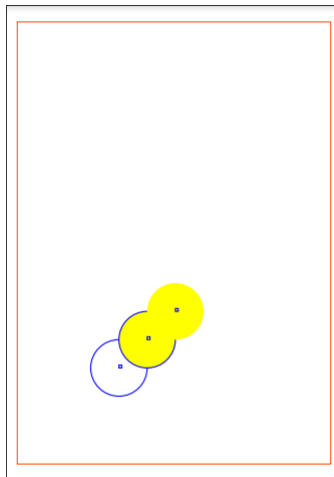
        contentByte.circle(300, 300, 50);
        contentByte.fill();
        contentByte.rectangle(300, 300, 5,5);
        contentByte.stroke();
        contentByte.restoreState();

        document.close();
        // 작성된 PDF문서 확인하기
        ChromeView.view(destFileName);
    } catch (DocumentException | IOException e) {
        e.printStackTrace();
    }
}

private static void getPageInfo(Document document) {
    Rectangle rectangle = document.getPageSize();
    System.out.println(rectangle);
    double width = PtMmConverter.pt2mm(rectangle.getWidth());
    double height = PtMmConverter.pt2mm(rectangle.getHeight());
    System.out.printf("크기 : (%.0f, %.0f)\n", width, height);
}
}

```

결과



25. 문서에 속성 및 버전 변경

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;

import com.itextpdf.text.Chunk;
import com.itextpdf.text.Document;
import com.itextpdf.text.DocumentException;
import com.itextpdf.text.pdf.PdfWriter;

public class Ex25_AddMetaData {
    public static void main(String[] args) {
        String destFileName = "pdf/Ex25_AddMetaData.pdf";
        try {
            Document document = new Document();
            PdfWriter writer = PdfWriter.getInstance(document, new
FileOutputStream(destFileName));
            writer.setPdfVersion(PdfWriter.PDF_VERSION_1_7); // PDF버전 변경(기본
1.4)

            document.open();
            // 문서의 속성을 추가한다.
            document.addTitle("My Title");
            document.addSubject("My Subject");
            document.addKeywords("My Keyword");
            document.addAuthor("My Author");
            document.addCreator("My Creator");
            document.addProducer();
            document.addCreationDate();

            document.add(new Chunk(""));
            document.close();
            Chromeview.view(destFileName);
        } catch (FileNotFoundException | DocumentException e) {
            e.printStackTrace();
        }
    }
}
```

결과



