Project Examples

Python library 'PDFReport'

Python library for dynamic PDF reports using the FPDF library

Filename	Number of lines	Last modification	File-Size (Byte)
sample_15.py	53	03.11.2023	1568
sample_21.py	50	03.11.2023	1677
sample_31.py	68	03.11.2023	1957
sample_05.py	35	03.11.2023	1868
sample_25.py	59	03.11.2023	1951
sample_11.py	50	03.11.2023	1899
sample_01.py	30	03.11.2023	614
sample_24.py	71	03.11.2023	2215
sample_10.py	30	03.11.2023	762
sample_14.py	49	03.11.2023	1464
sample_20.py	37	03.11.2023	946
sample_30.py	363	03.11.2023	11739
sample_04.py	41	03.11.2023	1600
sample_27.py	55	03.11.2023	1680
sample_13.py	50	03.11.2023	1482
sample_03.py	26	03.11.2023	977
sample_17.py	67	03.11.2023	1887
sample_23.py	42	03.11.2023	1379
sample_07.py	32	03.11.2023	1307
sample_16.py	53	03.11.2023	1563
sample_22.py	84	03.11.2023	2709
sample_06.py	46	03.11.2023	2343
sample_26.py	93	03.11.2023	3340
sample_12.py	59	03.11.2023	1633
sample_02.py	26	03.11.2023	688
sample_19.py	43	03.11.2023	1184
sample_09.py	56	03.11.2023	2207
sample_29.py	126	03.11.2023	3510
sample_28.py	90	03.11.2023	2834
sample_18.py	59	03.11.2023	1764
sample_08.py	53	03.11.2023	1510

Content of file 'sample_15.py'

```
import tempfile
from PDFReport import *


def sample_15():
    """
    A table using more space than available - line break
    """
    TextStyle.reset_styles()

    report = Report()

    table = TableFrame(report.body)
    table.inter_row_space = 1.5

    col_ft = TableColumn(table, "Frame type", 50.0)
    col_co = TableColumn(table, "Container type", 30.0)
    col_de = TableColumn(table, "Description", 80.0)
    col_du = TableColumn(table, "", 50.0)
    col_nu = TableColumn(table, "Number", 30.0)

    row = TableRow(table)
    TableCell(row, col_ft, "col 1 width 50mm")
    TableCell(row, col_co, "col 2 width 30mm")
    TableCell(row, col_de, "col 3 width 80mm")
    TableCell(row, col_du, "col 4 width 50mm")
    TableCell(row, col_nu, "col 5 width 30mm")

    row = TableRow(table)
    TableCell(row, col_ft, "LineFrame")
    TableCell(row, col_co, "No")
    TableCell(row, col_de, "This frame type represents a line on the report.")
    TableCell(row, col_nu, "1")

    row = TableRow(table)
    TableCell(row, col_ft, "SerialFrame")
    TableCell(row, col_co, "Yes")
    TableCell(row, col_de, "This is a frame container for a series of frames which will be printed one
after the other.")
    TableCell(row, col_nu, "2")

    row = TableRow(table)
    TableCell(row, col_ft, "TextFrame")
    TableCell(row, col_co, "No")
    TableCell(row, col_de, "A simple frame type to print text.")
    TableCell(row, col_nu, "3")

    filename = tempfile.gettempdir() + "/output_15.pdf"
    report.output(filename, True)


if __name__ == '__main__':
    sample_15()
```

Content of file 'sample_21.py'

```
import tempfile
from PDFReport import *


shortText = "Gute Susanne sah im einer Augen erst der im gewesen."
text = "Gute Susanne sah im einer Augen erst der im gewesen. Staatliche einer als für diesmal der. Ihr wie
des bewegen Vorgang wieder, sagte wenn legitimen Ziel Vorsorge. Jemand man so zueinander für Schlimmste.
Es wichtiger die das eine auf nicht einer eine Ziel freien. Man Netz dreinblickte verbrachte derartige
neuen. Es ihm zum ihr Interesse den besass er sie ihr seine, die die in mit Spass, das Tage eine
beobachtete nicht und, machte umher zu Technologien zweifelhaft."


def sample_21():
    """
    Use BoxFrames to place and format texts
    """
    TextStyle.reset_styles()

    report = Report()
    body = report.body

    TextFrame(body, "", TextStyle(TS_BOLD))

    SerialFrame(body, Direction.VERTICAL, margin_bottom=5.0)

    box = BoxFrame(body, 50.0)
    TextFrame(box, text, TextStyle(TS_NORMAL), True)

    SerialFrame(body, Direction.VERTICAL, margin_bottom=10.0)

    box = BoxFrame(body, 120.0)
    TextFrame(box, text, TextStyle(TS_BOLD), True)

    SerialFrame(body, Direction.VERTICAL, margin_bottom=10.0)

    box = BoxFrame(body, 40.0)
    TextFrame(box, text, TextStyle(TS_BOLD), True, TextAlign.CENTER)

    SerialFrame(body, Direction.VERTICAL, margin_bottom=10.0)

    bf = BoxFrame(body, 140.0, border_extent=0.1)
    bf.margin_left = 20.0
    bf.set_padding(1.0)
    TextFrame(bf, text, TextStyle(TS_NORMAL), True, TextAlign.JUST)

    SerialFrame(body, Direction.VERTICAL, margin_bottom=10.0)

    filename = tempfile.gettempdir() + "/output_21.pdf"
    report.output(filename, True)


if __name__ == '__main__':
    sample_21()
```

Content of file 'sample_31.py'

```
import tempfile
import sqlite3

from PDFReport import *


def add_sqlite_table(body: SerialFrame, db_file: str, fields: [str], widths: [float], tab: str, where:
str):
    tf = TableFrame(body)

    tc = []

    con = sqlite3.connect(db_file)
    sql = "SELECT "
    for idx, f in enumerate(fields):
        tc.append(TableColumn(tf, f, widths[idx]))
        sql = sql + f
        sql = sql + ", "

    sql = sql.rstrip(", ")
    sql = sql + " FROM " + tab + " WHERE " + where
    cur = con.cursor()
    for row in cur.execute(sql):
        r = TableRow(tf)
        for idx, f in enumerate(fields):
            TableCell(r, tc[idx], str(row[idx]))
    con.close()

    SerialFrame(body, Direction.VERTICAL, margin_bottom=10.0)


def create_db(db_file: str):
    con = sqlite3.connect(db_file)
    cur = con.cursor()

    cur.execute("DROP TABLE IF EXISTS frames")
    cur.execute("CREATE TABLE frames(number, frame_type, container, description)")

    data = [
        (1, "LineFrame", 0, "This frame type represents a line on the report."),
        (2, "SerialFrame", 1, "This is a container for a series of frames which will be printed one after
the other."),
        (3, "TextFrame", 0, "A simple frame type to print text."),
    ]
    cur.executemany("INSERT INTO frames VALUES(?, ?, ?, ?)", data)
    con.commit()
    con.close()


def sample_31():
    """
    Report based on a sqlite database
    """
    db_file = str(tempfile.gettempdir()) + "/data31.db"

    create_db(db_file)

    TextStyle.reset_styles()
    report = Report()

    add_sqlite_table(report.body, db_file, ["number", "frame_type", "container", "description"], [20.0,
30.0, 20.0, 90.0], "frames", "number > 1")
    add_sqlite_table(report.body, db_file, ["number", "frame_type", "description"], [20.0, 30.0, 110.0],
"frames", "number = 1")

    filename = str(tempfile.gettempdir()) + "/output_31.pdf"
    report.output(filename, True)
```

```
if __name__ == '__main__':
    sample_31()
```