

```

#!/usr/bin/python
from PyQt4.QtGui import QTextDocument, QPrinter, QApplication
import os
import sys
import logging
import argparse
import subprocess

class Py2pdf:

    """
        converts python file to pdf
    """

    def __init__(self, ifile=None, ofile=None, size="A4"):

        logging.getLogger().name = "py2pdf> "
        logging.getLogger().setLevel(logging.DEBUG)
        self.temp_location = "/tmp/temp.py"
        self.style = "Py2pdf/css/py2html.css"
        self.size = size
        logging.debug(os.path.exists(self.style))
        if not ifile:
            raise Exception("input file is required")
        else:
            self.py_file = ifile
            if ofile:
                self.pdf_file = ofile
            else:
                self.pdf_file = ifile.split('.')[0]+".pdf"
            self.create_clone()

    def create_clone(self):
        """
            creates a temporary file to copy python file
        """
        ftemp = open(self.temp_location, 'wb')
        fpy = open(self.py_file, 'rb')
        ftemp.write(fpy.read())
        ftemp.close()
        fpy.close()

    def init_print(self):
        app = QApplication(sys.argv)
        doc = QTextDocument()
        r = "pyhtmlizer -s %s %s" % (self.style, self.temp_location)
        s = subprocess.Popen(r, shell=True)
        s.wait()
        self.temp_location += ".html"
        html = open(self.temp_location).read()
        doc.setHtml(html)
        printer = QPrinter()
        printer.setOutputFileName(self.pdf_file)
        printer.setOutputFormat(QPrinter.PdfFormat)
        if self.size.lower() == "a2":
            printer.setPageSize(QPrinter.A2)
        elif self.size.lower() == "a3":
            printer.setPageSize(QPrinter.A3)
        elif self.size.lower() == "a4":
            printer.setPageSize(QPrinter.A4)

        printer.setPageMargins(15, 15, 15, 15, QPrinter.Millimeter)
        doc.print_(printer)
        logging.info("PDF created at %s" % (self.pdf_file))

    def parse_arg():
        parser = argparse.ArgumentParser(description=" \
            Convert Python code into pdf with syntax highlighting", epilog="\
            Author:tushar.rishav@gmail.com")

        parser.add_argument(
            "-i", "--ifile", help="Absolute path of the python file", type=str)
        parser.add_argument(
            "-o", "--ofile", help="Absolute path of the output pdf file", type=str)
        parser.add_argument(
            "-s", "--size", help="PDF size. A2,A3,A4,A5 etc", type=str, default="A3")
        args = parser.parse_args()
        if not args.ifile:
            raise Exception("input file is required")
        else:

```

```
    ifile = args.ifile # read python file which is to be converted
    if args.ofile:
        pdf_file = args.ofile
    else:
        pdf_file = args.ifile.split('.')[0]+".pdf"
    return (ifile, pdf_file, args.size)

def main():

    ifile, ofile, size = parse_arg()
    pdf = Py2pdf(ifile, ofile, size)
    pdf.init_print()

if __name__ == "__main__":
    main()
```