

=== File printout ===

File name: txt4print.py
File size: 5495 bytes
File sha256: 06f212c252a6aded950236b6f71deb01fe641cc6ec9c7a3bc16ad6870dd201cb
Printed: 2021-07-22T11:31:14 Pages: 3 Random ID: cdc39056

Bytes are printed as ASCII chars if possible. Spaces (0x20) are printed as ".", newlines (0x0a) are printed as "¶". All other bytes are printed as "¿xx", with "xx" being the value of the byte as hex number, e.g., "¿09" for a tab, "¿0d" for a carriage return.

Text printed in bold is not part of the file. On the left is the line number, on right a per line checksum (CRC-16-IBM). Lines longer than 80 chars are wrapped.

Created with txt4print version 0.1 - for additional information see <https://github.com/lobaro/txt4print>

===== BEGIN OF FILE =====

0001	#!/usr/bin/env.py python3¶	61f5
0002	import base64¶	6d0f
0003	import datetime¶	56e8
0004	import hashlib¶	77ac
0005	import os.path¶	d566
0006	import random¶	1119
0007	import string¶	1a8f
0008	import sys¶	89d0
0009	import tempfile¶	a247
0010	¶	473f
0011	from fpdf import FPDF¶	5a1b
0012	from font import FONT_REGULAR, FONT_BOLD¶	28d5
0013	¶	473f
0014	printable = string.ascii_letters + string.digits + string.punctuation¶	7133
0015	printable_byte = printable.encode()¶	a262
0016	¶	473f
0017	__version__ = "0.1"¶	a007
0018	¶	473f
0019	def calc_crc(data):¶	4330
0020crc = 0xFFFF¶	a946
0021for pos in data:¶	afc8
0022crc ^= pos¶	5702
0023for i in range(8):¶	7959
0024if (crc & 1) != 0:¶	c346
0025crc >>= 1¶	a0eb
0026crc ^= 0xA001¶	13c3
0027else:¶	6a94
0028crc >>= 1¶	a0eb
0029return crc¶	b050
0030	¶	473f
0031	¶	473f
0032	class PDF(FPDF):¶	05bb
0033def __init__(self):¶	7201
0034super().__init__()¶	1dbe
0035self.regular = None¶	9fb1
0036self.bold = None¶	2639
0037self.alias_nb_pages("¿c3¿a4")¶	e159
0038self.load_font()¶	a3b0
0039	¶	473f
0040self.font_regular()¶	df20
0041self.ln_width = self.get_string_width("0000.")¶	f729
0042self.content_width = self.get_string_width("x"*80 + ".")¶	3735
0043self.crc_width = self.get_string_width("0000")¶	5d67
0044self.total_width = self.get_string_width("a"*(5+80+5))¶	cbed
0045m = (self.w - self.total_width) / 2¶	9d02
0046self.set_margins(m, 10, m)¶	606b
0047	¶	473f
0048def load_font(self):¶	d0ee
0049self.regular = tempfile.NamedTemporaryFile()¶	af15

0050self.regular.write(base64.b64decode(FONT_REGULAR))¶	b374
0051self.add_font('Font',·'',·self.regular.name,·uni=True)¶	0454
0052self.bold.=·tempfile.NamedTemporaryFile()¶	6099
0053self.bold.write(base64.b64decode(FONT_BOLD))¶	cc8e
0054self.add_font('Font',·'B',·self.bold.name,·uni=True)¶	b305
0055	¶	473f
0056def·font_regular(self,·size=10):¶	2092
0057self.set_font("Font",·"·",·size)¶	bd35
0058	¶	473f
0059def·font_bold(self,·size=10):¶	54db
0060self.set_font("Font",·"·B",·size)¶	5489
0061	¶	473f
0062def·footer(self):¶	e38f
0063self.font_regular()¶	df20
0064self.set_y(-15)¶	d849
0065self.set_x(self.l_margin)¶	4991
0066self.cell(0,·4,·self.fname)¶	03d5
0067self.set_y(-15)¶	d849
0068self.set_x(self.l_margin)¶	4991
0069self.cell(self.total_width,·4,·self.ident+·"·-·"·+·str(self.page_no())· +·"/¿c3¿a4",·0,·0,·"R")¶	9495
0070	¶	473f
0071def·text_line(self,·s,·bold=False):¶	c76b
0072if·bold:¶	1e52
0073self.font_bold()¶	0e7b
0074else:¶	4f65
0075self.font_regular()¶	8ba3
0076self.cell(0,·4,·"·....."·+·s,·0,·1)¶	90d8
0077	¶	473f
0078def·to_line(self,·raw):¶	5738
0079cs.=·[]¶	bfc7
0080for·b·in·raw:¶	b9e9
0081if·b==ord("·"):¶	304a
0082cs.append('¿c2¿b7')¶	e821
0083elif·b==ord("\n"):¶	fdc0
0084cs.append('¿c2¿b6')¶	1420
0085elif·b·in·printable_byte:¶	d6cf
0086cs.append(chr(b))¶	3a2b
0087else:¶	6a94
0088cs.append(r'¿c2¿bf{0:02x}'.format(b))¶	756d
0089return·"".join(cs)¶	7b1e
0090	¶	473f
0091def·do(self,·fpath):¶	4260
0092self.ident.=·"".join(["%02x"·%·random.randint(0,·0xff)·for·_·in·range(4)])¶	8c67
0093self.date.=·datetime.datetime.now().replace(microsecond=0).isoformat()¶	0c50
0094fname.=·os.path.basename(fpath)¶	8a4f
0095self.fname.=·fname¶	6684
0096with·open(fpath,·"rb")·as·f:¶	2bae
0097raw.=·f.read()¶	789d
0098l.=·len(raw)¶	97b3
0099sha.=·hashlib.sha256(raw).hexdigest()¶	1895
0100	¶	473f
0101self.add_page()¶	0404
0102self.font_bold()¶	8b6d
0103self.text_line("===·File·printout·===",·True)¶	1f29
0104self.text_line("File·name:····%s"·%·fname)¶	b809
0105self.text_line("File·size:····%d·bytes"·%·l)¶	9f03
0106self.text_line("File·sha256:····%s"·%·sha)¶	ccfd
0107self.text_line("Printed:····%s········Pages:·¿c3¿a4········Random·ID :·%s"·%·(self.date,·self.ident))¶	3387
0108self.ln()¶	e6de
0109self.text_line('Bytes·are·printed·as·ASCII·chars·if·possible··Spaces·(0x 20)·are·printed·as·"¿c2¿b7",·')¶	57c9
0110self.text_line('newlines·(0x0a)·are·printed·as·"¿c2¿b6"··All·other·bytes ·are·printed·as·"¿c2¿bfx",·with·')¶	d957

0111self.text_line("xx".being.the.value.of.the.byte.as.hex.number,.e.g.,";c2;bf09".for.a.tab,";c2;bf0d".for')¶	d8e1
0112self.text_line('a.carriage.return.')¶	82bc
0113self.text_line("Text.printed.in.bold.is.not.part.of.the.file.On.the.left.is.the.line.number,.on")¶	fb89
0114self.text_line("right.a.per.line.checksum.(CRC-16-IBM).Lines.longer.than.80.chars.are.wrapped.")¶	7626
0115self.ln()¶	e6de
0116self.text_line("Created.with.txt4print.version."+"__version__"+"--for.additional.information.see")¶	c6fb
0117self.text_line("https://github.com/lobaro/txt4print")¶	1482
0118self.ln()¶	e6de
0119self.text_line("===== =====BEGIN.OF.FILE===== ===== =====",.True)¶	3416
0120raw_lines=.raw.splitlines(True)¶	a3d7
0121if.raw_lines[-1][-1]==ord(b'\n'):¶	d130
0122raw_lines.append(b"")¶	ff4b
0123for.n,.raw_line.in.enumerate(raw_lines):¶	3fbc
0124crc=.calc_crc(raw_line)¶	14b3
0125line=.self.to_line(raw_line)¶	681e
0126self.font_bold()¶	0e7b
0127self.cell(self.ln_width,.4,"%04d"%.(n+1),.0,.0)¶	e8db
0128self.font_regular()¶	8ba3
0129while.len(line)>.80:¶	ee20
0130self.cell(self.content_width,.4,.line[:80],.0,.1)¶	33a8
0131self.cell(self.ln_width,.4,".....",.0,.0)¶	40b3
0132line=.line[80:]¶	d1a6
0133self.cell(self.content_width,.4,.line,.0,.0)¶	150a
0134self.font_bold()¶	0e7b
0135self.cell(self.crc_width,.4,"%04x"%crc,.0,.1)¶	168c
0136self.text_line("===== =====END.OF.FILE===== ===== =====",.True)¶	ba82
0137	¶	473f
0138self.output(fname+"".pdf")¶	b968
0139	¶	473f
0140	¶	473f
0141	if.__name__=="__main__":¶	39d0
0142	...if.len(sys.argv)==2:¶	045e
0143pdf=.PDF()¶	1298
0144pdf.do(sys.argv[1])¶	b97d
0145	...else:¶	b6ca
0146print("txt4print.v"+__version__+"--Basic.text.file.printing.tool")¶	da0a
0147print("Create.a.pdf.for.printing.out.files.so.that.they.can.be.recovered. .")¶	1049
0148print("See.https://github.com/lobaro/txt4print.for.more.information.")¶	34f9
0149print("This.will.only.work.well.with.small.text.files.with.short.lines.")¶	8fef
0150print("Usage:%s.<filename>"%sys.argv[0])¶	51c3
0151	===== END OF FILE =====	ffff