

Python - Klient SMTP

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

Ogólne

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>

Uruchamianie skryptu:

```
C:\> python naszSkrypt.py
```

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstęp. Kod należy przepisać ze zrozumieniem.

Kod skryptu - wersja podstawowa

```
import smtplib
from email.utils import formatdate

# Linie poniżej należy zmienić (wersja statyczna):
odbiorcaEmail = 'twójOdbiorca@email.pl'
tematEmail = 'Testujemy e-maila'
trescListu = "Pozdrowienia z programu Python!\n-- \nTwój Python" # Dowiedz się, czym jest
# delimiter i po co się go stosuje!
nadawcaEmail = 'twój@email.pl' # Czyli także: login.

# Serwer SMTP (wpisz swoje dane konfiguracyjne). Wersja z SSL:
adresSerweraSMTP = 'twójAdresSMTP' # np. smtp.poczta.onet.pl
portSSL = 465
password = 'twojeHasło'

# Poniżej niczego już nie zmieniamy!
# Nagłówki e-mail:
nadawcaNaglowek = f"From: {nadawcaEmail}"
odbiorcaNaglowek = f"To: {odbiorcaEmail}"
tematNaglowek = f"Subject: {tematEmail}"
dataWyslania = f>Date: {formatdate(localtime=True)}"

# Pierwszy nagłówek musi być w pierwszej linii:
calaWiadomosc = f"""{nadawcaNaglowek}
{odbiorcaNaglowek}
{tematNaglowek}
{dataWyslania}
{trescListu}
"""

# Wysyłka e-maila:
server = smtplib.SMTP_SSL(f"{adresSerweraSMTP}:{portSSL}")
```

Uwaga: Jeśli posiadasz konto w serwisie **WP.pl**, poniżej wpisz nazwę swojego **loginu** w apostrofach zamiast zmiennej `nadawcaEmail` (ale tylko w poniższej linii):

```
server.login(nadawcaEmail, password)
server.sendmail(nadawcaEmail, odbiorcaEmail, calaWiadomosc)
server.quit()
# Potwierdzenie wysyłki (opcjonalnie):
print("\\n\\nWiadomość została wysłana.\\n\\n")
input('\\n\\nNaciśnij ENTER, aby zakończyć...\\n')
```

Kod skryptu - wersja rozbudowana

```
#!/usr/bin/env python
# Powyższa linijka, tylko dla systemu Linux!
```

try:

```
from os import system # Wymagane do kolorowania składni w systemie Windows 10/11:
system("")
# Uwaga: Składnia nie będzie kolorowana, gdy uruchomimy kod w „IDLE Shell” (bo to nie
# jest prawdziwy terminal) oraz w Windows 7/8.
import smtplib, textwrap
from email.utils import formatdate

# Linie poniżej należy zmienić (wersja statyczne):
# odbiorcaEmail = 'jaskowalski@4bird.pl'
# tematEmail = 'Testujemy e-maila'
# Jeśli poniższy kod jest wewnątrz TRY, należy rozpisać go w jednej linii z użyciem \\n:
# trescListu = "Pozdrowienia z programu Python v. 2!\\nPolskie znaki: ą, ę, ś, ć, ń, ó, ź, ż.\\n-- \\nTwój Python"
odbiorcaEmail = input('\\nPodaj adres e-mail odbiorcy: ')
tematEmail = input('\\n\\nPodaj temat wiadomości: ')
przywitanie = input('\\n\\nNapisz treść przywitania (np. Dzień Dobry): ')
trescListu = input('\\n\\nWpisz treść wiadomości: ')
# Zawijamy tekst do 72 znaków (tego nie zmieniamy):
trescListu = textwrap.fill(trescListu, width=72)

nadawcaEmail = 'twój@email.pl' # Czyli także: login.
podpis = 'Jan Kowalski'
# Serwer SMTP (wpisz swoje dane konfiguracyjne). Wersja z SSL:
adresSerweraSMTP = 'twójAdresSMTP'
portSSL = 465
password = 'twojeHasło'

# Wersja z TLS (zamiast SSL):
# server = smtplib.SMTP(f"{adresSerweraSMTP}:587")
# server.starttls()

# Poniżej niczego już nie zmieniamy!
# Nagłówki e-mail:
```

```

nadawcaNaglowek = f"From: {nadawcaEmail}"
odbiorcaNaglowek = f"To: {odbiorcaEmail}"
tematNaglowek = f"Subject: {tematEmail}"
dataWyslania = f>Date: {formatdate(localtime=True)}"
klientEmail = 'X-Mailer: Python'
formatWiadomosci = 'Content-Type: text/plain; charset=UTF-8'
kodowanie = 'Content-Transfer-Encoding: 8bit'
delimiter = f"-- \n{podpis}" # Dowiedz się, czym jest delimiter!

# Pierwszy nagłówek musi być w pierwszej linii, kolejne nie mogą być rozdzielane
# tabulatorem ani spacją. Więc, jeśli calaWiadomosc jest wewnątrz sekcji TRY, musi być w jednej
# linijce!
calaWiadomosc = f"{nadawcaNaglowek}\n{odbiorcaNaglowek}\n{tematNaglowek}\n{dataWyslania}\n{klientEmail}\n{formatWiadomosci}\n{kodowanie}\n{przywitanie}\n{trescListu}\n\n{delimiter}"

print('\n\n033[1;36;40mPoczekaj... Wyślę... \n\n033[0m')

# Wysłanie e-maila:
server = smtplib.SMTP_SSL(f"{adresSerweraSMTP}:{portSSL}")
# Uwaga: Jeśli posiadasz konto w serwisie WP.pl, poniżej wpisz nazwę swojego loginu w
# apostrofach zamiast zmiennej nadawcaEmail (ale tylko w poniższej linii):
server.login(nadawcaEmail, password)
server.sendmail(nadawcaEmail, odbiorcaEmail, calaWiadomosc.encode("utf-8"))
# Info: Kodowanie wiadomości można również ustawić za pomocą polecenia
calaWiadomosc.set_charset("utf-8").
server.quit()
# Potwierdzenie wysyłki (opcjonalnie):
print("\n\n033[1;32;40mWiadomość została wysłana. \n\n033[0m")

# Kolejność występowania "except" ma znaczenie. Najpierw powinny znaleźć się szczegółowe
# błędy:
except (ModuleNotFoundError, ImportError):
    print('\n\n033[1;31;40mJest jakiś problem z importem modułu. Sprawdź dokładnie nazwy\ni
    połączenie z siecią. \n\n033[0m')
    raise SystemExit

except NameError as bladZmiennej:
    print('\n\n033[1;31;40mProblem z nazwami zmiennych. Szczegóły: {}. \n\n033[0m' .
    format(bladZmiennej))
    raise SystemExit

except AttributeError as bladAtrybutu:
    print('\n\n033[1;31;40mJest problem z nazwami / atrybutami poleceń. Szczegóły: {}. \n\n
    033[0m' . format(bladAtrybutu))
    raise SystemExit

except LookupError as bladLookup:

```

```
print('\n\n033[1;31;40mJest problem ze wpisanymi wartościami. Szczegóły: {}. \n\n033[0m' . format(bladLookup))
raise SystemExit
```

except smtplib.SMTPRecipientsRefused:

```
print('\n\n033[1;31;40mNiewłaściwy adres odbiorcy lub niewłaściwa kompozycja wiadomości. \n\n033[0m')
raise SystemExit
```

except (smtplib.SMTPAuthenticationError, smtplib.SMTPServerDisconnected):

```
print('\n\n033[1;31;40mNiewłaściwy login lub hasło do serwera SMTP. \n\n033[0m')
raise SystemExit
```

except (smtplib.socket.gaierror, OSError, ConnectionRefusedError):

```
print('\n\n033[1;31;40mProblem z przeprowadzeniem połączenia. Możliwe przyczyny: \n- niewłaściwy adres nadawcy/odbiorcy; \n- niewłaściwy adres serwera SMTP; \n- niewłaściwy numer portu; \n- lub niepoprawne hasło. \n\n033[0m')
```

```
raise SystemExit # Nie używać exit() ani quit() - one zamykają interpreter, os.exit() nieczyści buffora, zaś sys.exit() wymaga importu biblioteki.
```

finally:

```
input('\n\nNaciśnij ENTER, aby zakończyć \n\n')
```