

# “Pythonic” kód

Aby tā programátori ~~nechceli zabíť~~ mali radi

# Veľmi stručne o mne



- Večný študent FIIT STU na odbore Internetové technológie
- Najmenší a najmladší back-end programátor v BACKBONE s.r.o
- Pestovateľ chilli
- Pythonista



# Ako byť lepší programátor?

A hlavne: prečo byť lepší programátor?

```

3506
3507         # Ak tlacim, na fisdkalny aj na nefiskalny doklad ako NNOUNCEMENTNF
3508         lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3509     elif tnLineType == 47.3: # Platba: (pre ine ako kompenzaciu zlavou)
3510         # Ak tlacim, na fisdkalny aj na nefiskalny doklad ako NNOUNCEMENTNF
3511         lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3512     elif tnLineType == 47.4: # Texty k jednotlivym platbam
3513         # Ak tlacim, na fisdkalny aj na nefiskalny doklad ako NNOUNCEMENTNF
3514         ### L.L. (25.03.2013) Zmena kvoli UPOSUPL
3515         #*lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3516         lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(self.parse_line("TXT", tcLine), 40)+chr(0x00)+chr(0x04)
3517     elif tnLineType == 48: # Medzera pred typom platby, BOWA netlac
3518         if not self.isFiscalParagon(): # Tlacim, len na nefiskalny doklad, tak pouzijem ANNOUNCEMENTNF
3519             lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3520         else: # Inak pouzijem ANNOUNCEMENT
3521             lcTrLine = None
3522
3523     elif tnLineType == 49: # Cislo stola a cislo uctu (pre FISKAL BOWA)
3524         ### L.L. (26.03.2013) Prerobene kvoli UPOSUPL
3525         #*lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3526         #LOCAL lcStulTxt, lcUcisTxt, lcStul, lcUcis
3527         lcUcisTxt = self.parse_line("TXTUCIS", tcLine)
3528         lcUcis = self.parse_line("UCIS", tcLine)
3529         lcStulTxt = self.parse_line("TXTSTUL", tcLine)
3530         lcStul = self.parse_line("STUL", tcLine)
3531         lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+\
3532             left(lcUcisTxt + lcUcis + iif(empty(lcStulTxt) and empty(lcStul), "", " /") + lcStulTxt + lcStul, 40) +chr(0x00)+chr(0x04)
3533     # Pre BOWU to ignorujem, pretoze FISK.UZ nebude sucastou tlacovej ulohy
3534     elif tnLineType == 50: # Pokracovanie, to pri BOWE netlacim
3535         lcTrLine = None
3536     elif tnLineType == 51: # TIP+TOTAL, PRE FISKAL ALTO, BOWA NETLACI
3537         ### L.L. (14.04.2011) Oprava kvoli producetu, tlacim aj na nefiskalny doklad a producet
3538         if not self.isFiscalParagon(): # Ak ide o nefiskalny doklad, tak tento TIP netlacim
3539             lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+tcLine+chr(0x00)+chr(0x04)
3540         else:
3541             lcTrLine = None
3542
3543     elif tnLineType == 52: # Potvrdenie pohybu na karte
3544         lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3545     elif tnLineType == 53: # Objednavky
3546         lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3547     elif tnLineType == 60: # Uzavierka "X" alebo "Z"
3548         lcTrLine = None
3549     elif tnLineType == 101: # Nadpis "VKLAD/VYBER => tlacim len pre nefiskalny vklad a vyber
3550         if not self.isFiscalParagon(): # Tlacim, len na nefiskalny doklad a pouzijem ANNOUNCEMENT
3551             lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3552         else:
3553             lcTrLine = None
3554
3555     elif tnLineType == 102: # VKLAD/VYBER - Cislo dokladu vo foodu - pre vyhľadavanie tlacim vzdy, pre fiskal - ANNOUNCEMENTNF, inak ANNOUNCEMENT
3556         if not self.isFiscalParagon(): # Tlacim, len na nefiskalny doklad a pouzijem ANNOUNCEMENT
3557             lcTrLine = chr(0x1C)+chr(0x20)+chr(0x31)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,40)+chr(0x00)+chr(0x04)
3558         else:
3559             lcTrLine = chr(0x1C)+chr(0x20)+chr(0x32)+_ alltrim(STR(tnLineNo%10,1,0))+chr(0x00)+left(tcLine,39)+chr(0x00)+chr(0x04)

```

# Programuj po anglicky

- Väčšina knižníc a frameworkov je napísaná v angličtine
- Inštrukcie jazyka už v angličtine sú
- Ľahšie sa opisujú tvoje problémy a postupy
- Zadefinované pojmy
- Slovensko-anglický kód vyzerá smiešne





# Pomenovávanie

```
def token_create(ip_id_kas, ip_id, ip_typ, ip_exp):
    lvs_token = ''
    lvs_kluc = settings.SECRET_KEY
    lvs_payload = '{' + '"exp":"' + ip_exp + '", "typ": ' + str(ip_typ) + ','
    lvs_payload += '"id":"' + str(ip_id) + '", "id_kas":"' + str(ip_id_kas) + '"}'
    lvs_token_hmac = hashlib.pbkdf2_hmac('sha256', bytes(lvs_payload, 'utf-8'), bytes(lvs_kluc, 'utf-8'), 100000)
    lvs_token_ba64 = base64.b64encode(bytearray(lvs_payload, 'utf-8'))
    lvs_token = lvs_token_ba64.decode() + '.' + binascii.hexlify(lvs_token_hmac).decode()

    return lvs_token

def token_create(kas_id, user_id, token_type, expiration):
    ITERATIONS = 100000

    payload = {
        "exp": expiration,
        "type": str(token_type),
        "id": str(user_id),
        "kas_id": kas_id
    }

    hmac = hashlib.pbkdf2_hmac('sha256', json.dumps(payload).encode(), settings.SECRET_KEY.encode(), ITERATIONS)
    payload_hash = base64.b64encode(json.dumps(payload).encode())

    return payload_hash.decode() + '.' + binascii.hexlify(hmac).decode()
```

# Ako správne pomenovať?

Súbor: snake\_case.py

Objekt: PascalCase

Metóda: snake\_case(email, password)

Premenná: first\_name

Konštanta: CONSTANT

# Zalamovanie riadkov

```
REST_FRAMEWORK = {
    'DEFAULT_PARSER_CLASSES': (
        'rest_framework.parsers.JSONParser',
    ),
    'DEFAULT_RENDERER_CLASSES': (
        'rest_framework.renderers.JSONRenderer',
        'rest_framework.renderers.BrowsableAPIRenderer'
    ),
    'DEFAULT_FILTER_BACKENDS': (
        'django_filters.rest_framework.DjangoFilterBackend',
        'rest_framework.filters.OrderingFilter'
    ),
    'DEFAULT_AUTHENTICATION_CLASSES': (), # no authentication classes
    'DEFAULT_PERMISSION_CLASSES': (
        'rest_framework.permissions.AllowAny', # will allow unrestricted access
    ),
    'DATETIME_FORMAT': None,
    'TEST_REQUEST_DEFAULT_FORMAT': 'json',
}
```

# Chcem vedieť viac

- [PEP8 – Style Guide for Python Code](#)
- [PEP20 – The Zen of Python](#)
- Nehryzim



# Ďakujem za pozornosť

Jakub Dubec – FIIT STU, BACKBONE s.r.o.

<https://jakubdubec.me>

jakub.dubec@gmail.com