

# Tvorba nástrojů a pluginů s využitím QGIS Python API

Seminář GIS Ostrava 2021

Jan Čaha

17. 3. 2021

# Motivace

- snaha najít (zejména pro platformu Windows) co "nejpoužitelnější" nastavení pro vývoj v **Pythonu**, který hodně využívá API **GDAL/OGR**, **QGIS**, **PROJ**, **GRASS** a další **GIS knihovny**
- vytěžit pro práci maximum z vývojového prostředí (IDE) **PyCharm** (ať už v Community nebo Professional Edition)
- pro vývoj na platformě Linux/macOS je situace trochu lepší, ale některé tipy stále platí

# Python, GIS knihovny a Windows

- problém na Windows je nastavení systémových proměnných, cest ke knihovnám apod.
- instalace knihoven skrze **PyPI** je spíše problém než řešení
- lépe funguje instalace skrze prostředí **Conda** popř. **Anaconda**, ale je nutné věnovat trochu času nastavení
- některé instalace (např. **QGIS**) nesou veškeré nezbytnosti s sebou (Python, knihovny atd.)
- nejkompexnější řešení je instalace skrze instalátor **OSGEO4W** ([dostupný odsud](#))
  - umožňuje instalaci i vývojových verzí GIS knihoven
  - aktuálně trochu problém: starý instalátor nenabízí nové verze software a nový instalátor nefunguje úplně dobře, proto je pozastaven, ale dá se čekat, že se to v dohledné době vyřeší
  - nový instalátor je [dostupný zde](#)
- propojení instalací s IDE

# Zhodnocení instalací (podle náročnosti)

- [Standalone installer QGIS](#) - jednoduché, ale pouze stabilní verze software
- **OSGEO4W** - nepatrně komplexnější, ale možnost volby verzí software
- **Anaconda** - instalace složitější, stabilní verze software, ale poměrně stabilní
- běžná instalace **Pythonu** + knihoven - možné, ale velice složité

**Ukázky instalací a výsledných složek**

# Propojení instalací Pythonu s IDE

- nejčastější problém - IDE "nevidí" některé knihovny, nebo nelze Python programy spustit z IDE bez chyb
- nutnost spouštět napsané programy "jinde" - zdržení, riziko chyby
- většina problémů vzniká z chybějících nastavení systémových proměnných, cest ke knihovnám apod.
- řešení jsou buď environment v **Conda**, který toto řeší, a nebo **OSGEO4W**, kde jsou nadefinované `.bat` soubory, které nastaví prostředí před spuštěním programu

# Propojení instalátoru QGIS nebo OSGEO4W s PyCharm

- v principu shodné systémy
- dva kroky:
  - spuštění **PyCharm** s nastavením systémových proměnných a cest k souborům
  - nastavení cesty k Python knihovnám pro **QGIS**
- systémové proměnné a cesty můžeme získat z existujících `.bat` souborů - v našem případě `qgis-dev.bat`
- v `.bat` souboru nastavíme místo spuštění **QGIS** spuštění **PyCharm**
- v **PyCharm** nastavíme Python interpretor na správný soubor a přidáme cesty ke Python QGIS API

# Ukázka spuštění PyCharm s nastavením OSGEO4W



# Propojení Anaconda env s PyCharm

- pouze jeden krok
  - systémové proměnné jsou nastavené v environmentu
  - cesty k Python QGIS API je nutné dodat k Python interpretoru ručně

**Ukázka spuštění PyCharm s nastavením  
Anaconda env s QGIS, GDAL/OGR atd.**

# Ukázky práce s GDAL/OGR API a QGIS API

nápověda, kompletace kódu

# Programování s QGIS API

# Typy programů

- nástroje nebo pluginy pro QGIS
- samostatné nástroje využívající QGIS API
  - bez využití nástrojů QGIS
  - využívající nástroje QGIS

# Ukázka vytvoření nástroje pro QGIS

# Zadání

- vytvoření nástroje, který z polygonové vrstvy vytvoří bodovou vrstvu určením nejvzdálenějšího bodu od hranice (pole of inaccessibility)

# Ukázka vytvoření standalone aplikace bez nástrojů QGIS



# Zadání

- vytvoření skriptu, který z polygonové vrstvy vytvoří bodovou vrstvu určením nejvzdálenějšího bodu od hranice (pole of inaccessibility)

# Ukázka vytvoření standalone aplikace s nástroji QGIS

# Inicializace QGIS a prostředí

```
from qgis.core import QgsApplication
import processing
from processing.core.Processing import Processing
from qgis.analysis import QgsNativeAlgorithms

app = QgsApplication([], False)
Processing.initialize()
app.processingRegistry().addProvider(QgsNativeAlgorithms())
```

# Spouštění nástrojů QGIS

```
parameters = {'INPUT': path_input_data,  
              'POINTS_NUMBER': 100,  
              'MIN_DISTANCE': 0,  
              'MIN_DISTANCE_GLOBAL': 0,  
              'MAX_TRIES_PER_POINT': 10,  
              'SEED': None,  
              'INCLUDE_POLYGON_ATTRIBUTES': False,  
              'OUTPUT': path_output_points}  
  
processing.run("native:randompointsinpolygons",  
              parameters=parameters)
```

# Zadání

- vytvoření skriptu, který z bodové vrstvy vytvoří Voroniho polygony a následně spojí geometrie pro prvky se shodnou hodnotou atributu (dissolve)

**Děkuji za pozornost.**

**Dotazy???**