
Vývoj geografických aplikací v GIS

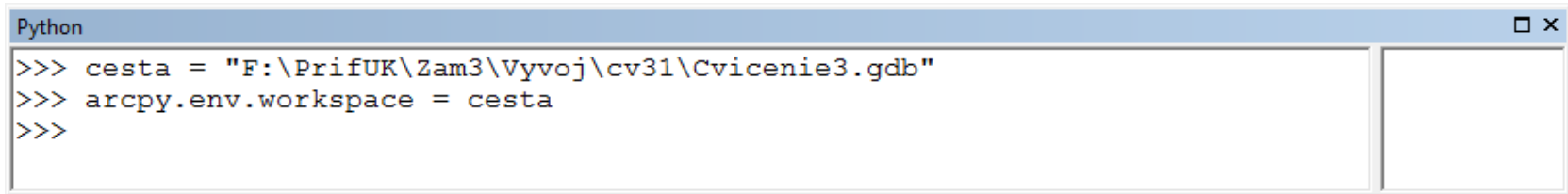
Hana Stanková, Vladimír Pelech

Lekcia 3

Dávkové spracovanie pomocou
ArcPy Listing

Príprava na cvičenie

- Vytvorte si premennú „cesta“ a ako reťazec do nej uložte cestu k databáze použitej v predchádzajúcom cvičení.
- Následne nastavte workspace podľa premennej „cesta“.



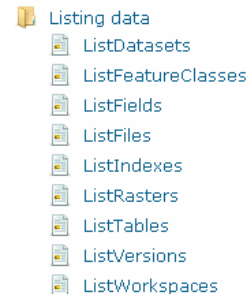
```
Python □ ×  
>>> cesta = "F:\PrifUK\Zam3\Vyvoj\cv31\Cvicenie3.gdb"  
>>> arcpy.env.workspace = cesta  
>>>
```

Funkcia ListFeatureClasses

- vytvorí zoznam tried objektov v pracovnom prostredí (v adresári alebo v geodatabáze)

```
Python
>>> zoznam_vrstiev = arcpy.ListFeatureClasses()
>>> zoznam_vrstiev
[u'cesty_4', u'clc_2012_4', u'vodne_toky_4', u'zeleznice_4',
 u'Zaujmová_oblasť']
>>> |
```

- počet „listovacích“ funkcií sa líši podľa verzie



Funkcia ListFeatureClasses

- má tri nepovinné parametre, pomocou ktorých je možné obmedziť vytvorený zoznam:
 - {wild_card} –hľadaný reťazec v názve, kde * nahrádza ľubovoľný počet znakov
 - {feature_type} – reťazec na obmedzenie typu objektu, ukázať manuál
 - {feature_dataset} -ak chcem vyhľadávať iba v datase, pri jeho vynechaní sa vyhľadáva iba v databáze
-

Funkcia ListFeatureClasses

- Použitie wild_card, získame iba vrstvy začínajúce na malé c. wild card je case sensitive.

```
Python
>>> zoznam_vrstiev1 = arcpy.ListFeatureClasses("c*")
>>> zoznam_vrstiev1
[u'cesty_4', u'clc_2012_4']
>>> |
```

- Použitie feature_type na obmedzenie polygónovej geometrie.

```
Python
>>> zoznam_vrstiev1 = arcpy.ListFeatureClasses("c*", "Polygon")
>>> zoznam_vrstiev1
[u'clc_2012_4']
>>>
```

Nástroj

CreateFeatureDataset_management

- Vytvorí nový Feature Dataset
- dva povinné a jeden nepovinný parameter:
 - out_dataset_path -cesta k databáze, kde má byť nový dataset vytvorený,
 - out_name -názov nového datasetu,
 - {spatial_reference}-súradnicový systém nového datasetu.



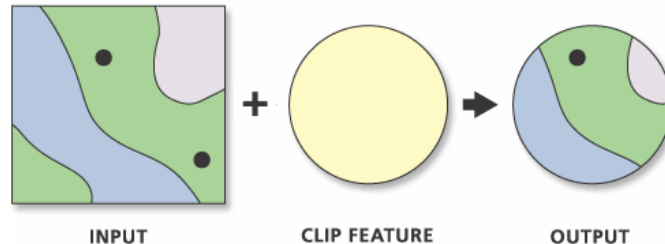
```
Python
>>> arcpy.CreateFeatureDataset_management(arcpy.env.workspace, "Orezane")
<Result 'F:\\PrifUK\\Zam3\\Vyvoj\\cv31\\Cvicenie3.gdb\\Orezane'>
>>> |
```

25
15:33:27
2021
(Elapsed
Time: 0,33
seconds)

Nástroj

Clip_analysis

- Vytvárá novú vrstvu na základe tzv. orezávacej vrstvy.
- Tri povinné a jeden nepovinný parameter:
 - in_features –vstupná vrstva na orezanie,
 - clip_features -tzv. orezávacia vrstva,
 - out_feature_class – názov novej, už orezanej vrstvy,
 - {cluster_tolerance} -minimálna vzdialenosť oddeľujúca objekty.



Nástroj

Clip_analysis

- Je ho možné použiť na dávkové orezanie viacerých vrstiev pomocou cyklu a vytvoreného zoznamu vrstiev.

```
Python □ ×  
>>> for vrstva in zoznam_vrstiev:  
...     if vrstva != "Zaujmová_oblasť":  
...         arcpy.Clip_analysis(vrstva, "Zaujmová_oblasť", "Orezane/%s_clip" %  
vrstva)  
...  
>>>
```

Formátovanie reťazcov v ArcPy

- V ArcPy je možné pre premennú použiť zástupný znak %s (ukážka aj na predchádzajúcom slide s clipom).
 - Tento reťazec v Python-e volá funkciu str(), ktorá (nasilu) konvertuje objekt na textový reťazec, ktorý sa dá „ľahko vytlačiť“.
 - Podobne pre konverziu na (*signed*) integer sa používa zástupný znak %i.
 - Formátovaných reťazcov môže byť aj viac.
-

Formátovanie reťazcov v ArcPy

- Ukážka použitia %s.

```
Python
>>> premenna = 'Som premenná.'
>>> print "Tento retazec ma premennú, ktorá nasleduje %s" %premenna
Tento retazec ma premennú, ktorá nasleduje Som premenná.
>>>
```

- Ukážka súčasného použitia %s aj %i v jednom príkaze. Pozor na zmenu odkazu cez zátvorku pri viacerých nahradeniach.

```
Python
>>> cislo = 25
>>> print "Tento retazec ma premennú, ktorá nasleduje %s a cislo %i" %
(premenna, cislo)
Tento retazec ma premennú, ktorá nasleduje Som premenná. a cislo 25
>>>
```

Lekcja 4

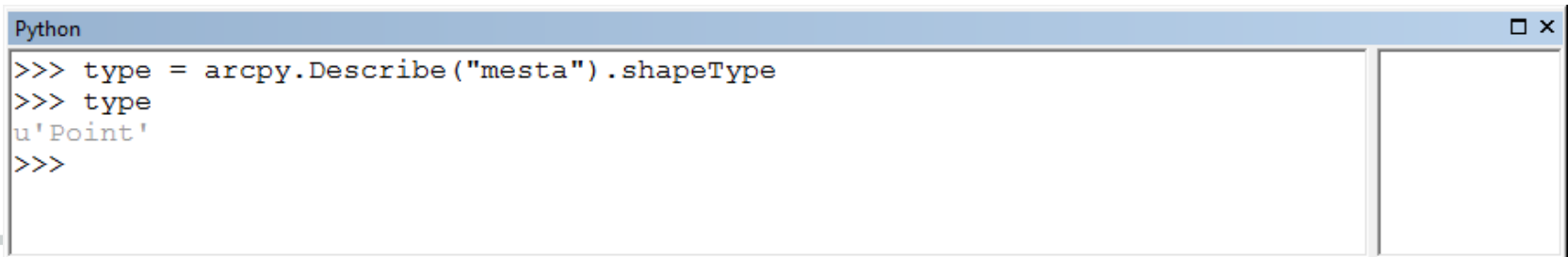
ArcPy kurzory

Príprava na cvičenie

- Stiahnuť si zazipovaný súbor k lekcii na disk a odzipovať ho.
 - Pridať si do okna ArcMapu vrstvy nachádzajúce sa v odzipovanom priečinku. Ide o dve vrstvy.
 - Pridať si do okna ArcMapu vrstvu *hranice* zo súboru cv1 pre lekciiu 1.
-

Funkcia Describe

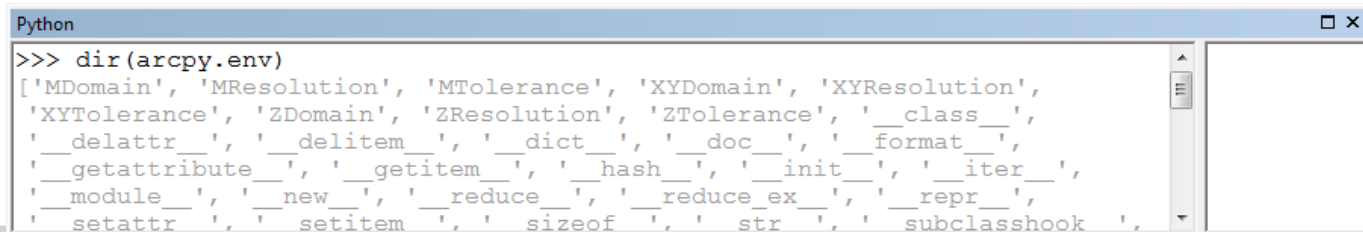
- Funkcia Describe() vracia vlastnosti určitých druhov dát ako sú tabuľky, triedy objektov, geodatabázy, coverage, layer súbory, relačné triedy, pracovné prostredia (workspace) a sady údajov (dataset), takisto ako geoprocených objektov ako sú FeatureLayers a TableViews.
- Vlastnosti: typ údajov, typ geometrie (bod, línia, polygón), názov poľa OID, názov poľa shape(ukážka), ... atď.



```
Python
>>> type = arcpy.Describe("mesta").shapeType
>>> type
u'Point'
>>>
```

Výpis názvov atribútov

- `arcpy.Describe(„název_vrstvy“).fields` – vracia pole objektov `fields` (stĺpce tabuľky)
- `arcpy.ListFields(„název_vrstvy“, {“wild_card“}, {“field_type“})`
– to isté
- získané objekty majú vlastnosti `name`, `type`, `length`, `precision`, `scale`, `aliasName`, `domain...atd.`
- vlastnosti sa líšia podľa objektu, na ich určenie sa dá použiť funkcia `dir`.



```
Python
>>> dir(arcpy.env)
['MDomain', 'MResolution', 'MTolerance', 'XYDomain', 'XYResolution',
 'XYTolerance', 'ZDomain', 'ZResolution', 'ZTolerance', '__class__',
 '__delattr__', '__delitem__', '__dict__', '__doc__', '__format__',
 '__getattr__', '__getitem__', '__hash__', '__init__', '__iter__',
 '__module__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
 'setattr', 'setitem', 'sizeof', 'str', 'subclasshook',
```

Výpis názvov atribútov s použitím Describe

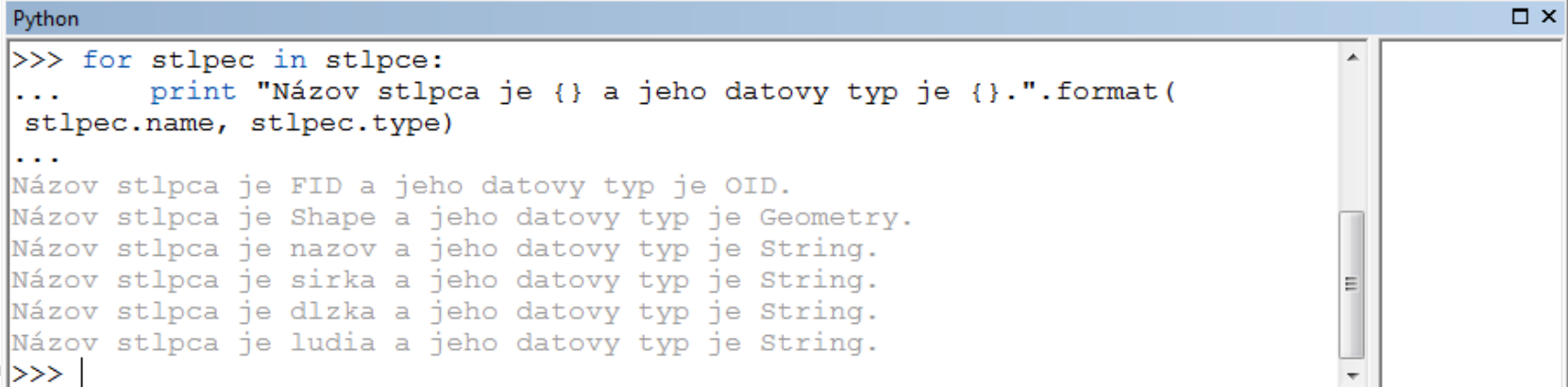
```
Python □ ×  
>>> stlpce = arcpy.Describe("mesta").fields  
>>> for stlpec in stlpce:  
...     print stlpec.name  
...  
FID  
Shape  
nazov  
sirka  
dlzka  
ludia  
>>> |
```


Výpis názvov atribútov s použitím listFields

```
Python □ ×  
>>> stlpce = arcpy.ListFields("mesta")  
>>> for stlpec in stlpce:  
...     print stlpec.name, stlpec.type  
...  
FID OID  
Shape Geometry  
nazov String  
sirka String  
dlzka String  
ludia String  
>>>
```

Výpis s použitím metódy format.

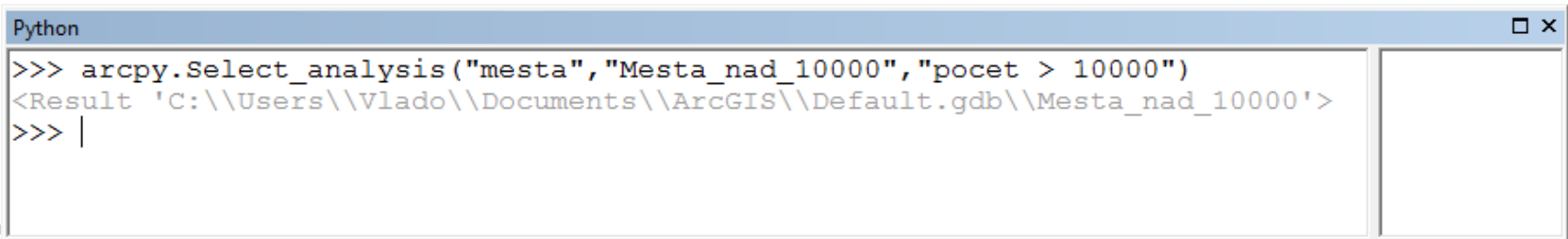
- Zástupné znaky {} v reťazci odkazujú na použitie premennej. V reťazci sa ich môže nachádzať aj viac a môžu byť aj v určitom formáte.
- Premenná je potom uvedená v metóde format() v poradí, v akom sa má v reťazci použiť.



```
Python
>>> for stlpec in stlpce:
...     print "Názov stlpca je {} a jeho datovy typ je {}".format(
...         stlpec.name, stlpec.type)
...
Názov stlpca je FID a jeho datovy typ je OID.
Názov stlpca je Shape a jeho datovy typ je Geometry.
Názov stlpca je nazov a jeho datovy typ je String.
Názov stlpca je sirka a jeho datovy typ je String.
Názov stlpca je dlzka a jeho datovy typ je String.
Názov stlpca je ludia a jeho datovy typ je String.
>>> |
```

Atribútové dopyty

- nástroje `Select_analysis`, `TableSelect_analysis`
- parametre:
 - `in_feature/ in_table` – vstupná vrstva/tabuľka
 - `out_feature_class /out_table` – vytvorená vrstva/tabuľka
 - `{where_clause}` – SQL výraz
- SQL výraz musí byť uzavretý v "...", pozor na miešanie s apostrofom.



```
Python
>>> arcpy.Select_analysis("mesta", "Mesta_nad_10000", "pocet > 10000")
<Result 'C:\\Users\\Vlado\\Documents\\ArcGIS\\Default.gdb\\Mesta_nad_10000'>
>>> |
```

ArcPy kurzory

- používajú sa na prístup k záznamom tabuľky:
 - prechádzajú cez jednotlivé záznamy
 - získavajú hodnoty atribútov z tabuliek, tried objektov, rastrov
 - získavajú geometriu objektov
 - Typy kurzorov:
 - SearchCursor – *read-only* prístup
 - UpdateCursor – aktualizuje alebo vymazáva hodnoty atribútov alebo geometriu (*read/write* prístup)
 - InsertCursor – pridáva nové záznamy do tabuľky , zapisuje hodnoty atribútov a geometriu
-

ArcPy kurzory

- Z historického hľadiska sú v arcpy dostupné kurzory, ktoré boli uvedené na predchádzajúcom slajde ako `arcpy.SearchCursor()`.
 - Avšak novšie a výkonnejšie kurzory sa nachádzajú v module Data Access (da) a sú dostupné ako `arcpy.da.SearchCursor()`.
-

Funkcie getValue a setValue

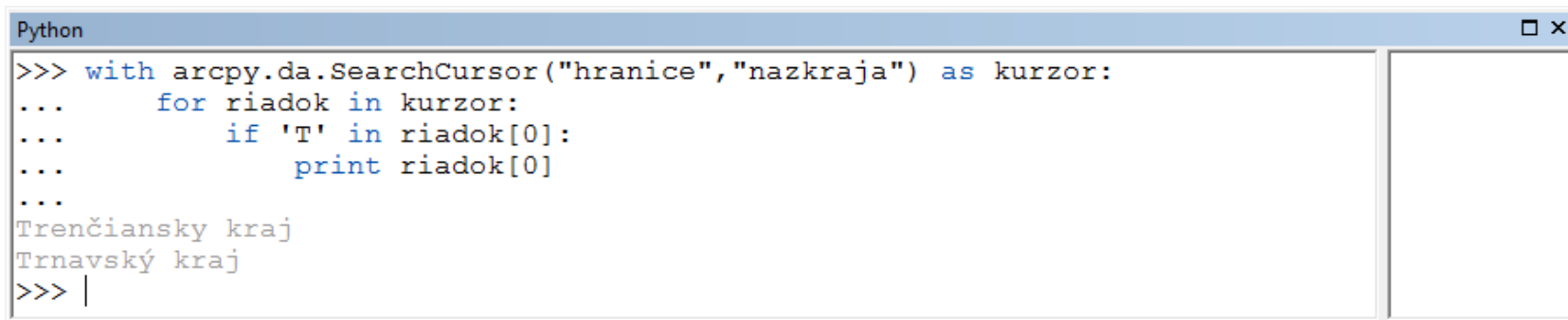
- Používajú sa na získanie alebo zapísanie hodnoty atribútov v rámci vytvorených starších kurzorov.
 - Parametre:
 - field_name – názov stĺpca ako reťazec
 - object - (iba pre setValue), čo sa má vložiť
 - Po novom už nie je potrebné používať getValue. Stačí uviesť príslušný index pre riadok v rámci kurzora a hodnota bude dostupná.
-

SearchCursor

- Parametre:
 - `in_table` – objekt, pre ktorý chcem vytvoriť kurzor,
 - `field_names`, [`field_names`,...] – stĺpec alebo pole stĺpcov, ktoré chcem využiť v kurzore, na všetky stĺpce použiť *. Je možné dopytovať aj vlastnosti geometrie (`SHAPE@AREA`).
 - `{where_clause}` – SQL výraz na obmedzenie výberu,
 - `{spatial_reference}` – súradnicový systém objektu,
 - a ďalšie ...
-

SearchCursor

- Kurzory je možné vytvárať pomocou syntaxe s použitím *with* vytvoreny_kurzor as názov_kurzora:

A screenshot of a Python terminal window titled "Python". The code entered is:

```
>>> with arcpy.da.SearchCursor("hranice","nazkraj") as kurzor:  
...     for riadok in kurzor:  
...         if 'T' in riadok[0]:  
...             print riadok[0]  
...  
Trenčiansky kraj  
Trnavský kraj  
>>> |
```

The output shows two lines: "Trenčiansky kraj" and "Trnavský kraj". The cursor is at the end of the last line.

- Uvedená syntax má tú výhodu, že na konci sa kurzor automaticky sám vymaže. Čiže na dáta neostane tzv. lock a môže sa s nimi ďalej pracovať.
-

SearchCursor

- Uloženia hodnoty do premennej s pomocou kurzora.

```
Python
>>> with arcpy.da.SearchCursor("mesta","*", "nazov like 'Zvolen'") as kurzor:
...     for row in kurzor:
...         ZV = row[6]
...
>>> ZV
44498
>>>
```

- Využitie v porovnaní s hodnotami v novom kurzore.

```
Python
>>> with arcpy.da.SearchCursor("mesta","*") as kurzor2:
...     for row in kurzor2:
...         if ZV < row[6]:
...             print "%s - %i" %(row[2],row[6])
...
Banská Bystrica - 85052
Martin - 60917
```

SearchCursor

- Starší typ kurzora umožnil využiť aj názov stĺpca. V novom by to už nemalo byť možné.

```
Python □ ×  
>>> stary_kurzor = arcpy.SearchCursor("mesta","nazov like 'Zvolen'")  
>>> for row in stary_kurzor:  
...     hodnota = row.pocet  
...  
>>> hodnota  
44498  
>>>
```

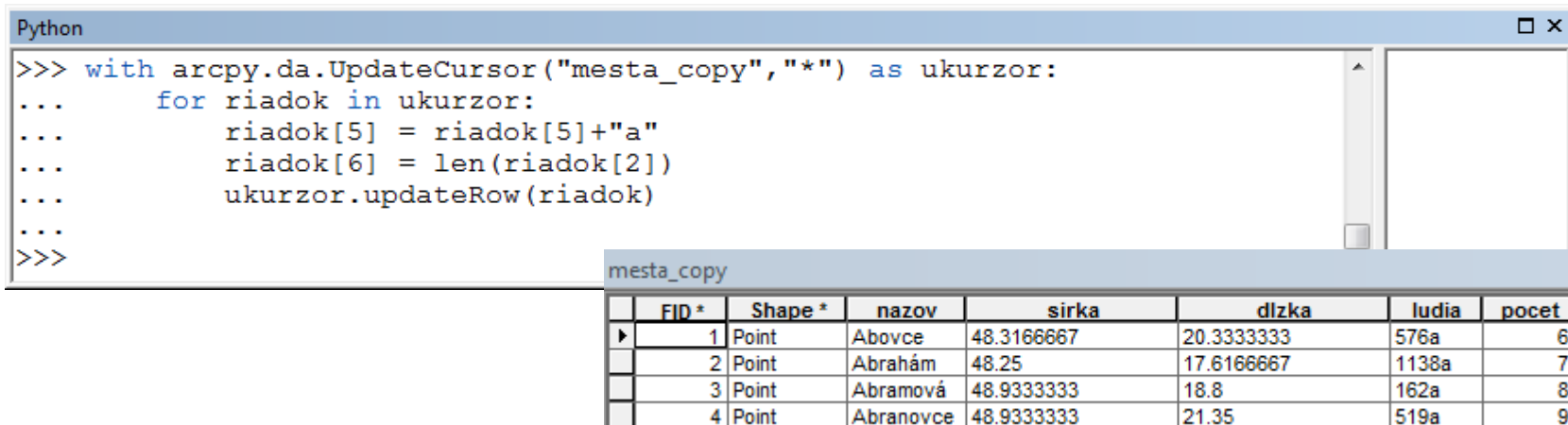
S použitím `arcpy.Select_management` vytvorte vrstvu s názvom `mesta_copy` tak, aby obsahovala iba záznamy začínajúce sa na písmeno A.

Metódy kurzorov

- všetky kurzory majú metódu next, ktorá automaticky získava nasledujúci záznam, taktiež reset()- vráti kurzor na začiatok
 - metódy kurzora Update:
 - updateRow – aktualizuje príslušný riadok
 - deleteRow – odstráni riadok z tabuľky
 - metódy kurzora Insert:
 - newRow – vytvorí prázdny riadok (prvý krok)
 - insertRow – vloží riadok do tabuľky (druhý krok)
-

UpdateCursor

- Používa sa v prípade aktualizácie hodnôt v tabuľke, aj geometrie, alebo na odstránenie záznamov z tabuľky. Po aktualizácii je nutné zmenu potvrdiť pomocou metódy *updateRow()*.



```
Python
>>> with arcpy.da.UpdateCursor("mesta_copy", "*") as ukurzor:
...     for riadok in ukurzor:
...         riadok[5] = riadok[5]+"a"
...         riadok[6] = len(riadok[2])
...         ukurzor.updateRow(riadok)
...
>>>
```

	FID *	Shape *	nazov	sirka	dlzka	ludia	pocet
▶	1	Point	Abovce	48.3166667	20.3333333	576a	6
	2	Point	Abrahám	48.25	17.6166667	1138a	7
	3	Point	Abramová	48.9333333	18.8	162a	8
	4	Point	Abranovce	48.9333333	21.35	519a	9

UpdateCursor

- Vymazanie záznamu alebo záznamov pomocou *deleteRow()*.

Python

```
>>> with arcpy.da.UpdateCursor("mesta_copy","*") as ukurzor:
...     for riadok in ukurzor:
...         if 'Ab' in riadok[2]:
...             ukurzor.deleteRow()
...
>>> |
```

Čítanie geometrie

- Geometrické pole (obyčajne Shape) vracia geometrický objekt, ktorý sa dá použiť pri geoprocessingu namiesto triedy objektov.

```
Python □ ×  
>>> with arcpy.da.SearchCursor("mesta_copy",["SHAPE@","nazov"]) as skurzor:  
...     for riadok in skurzor:  
...         arcpy.Buffer_analysis(riadok[0],"Buffer_%s" %riadok[1],500)  
...  
>>> |
```

Práca s geometrickými objektmi

- Geometrický objekt má vlastnosti: type, area, length, centroid, extent, first point, last point...atď.
 - Môžeme ich nájsť na:
<https://desktop.arcgis.com/en/arcmap/latest/analyze/arcpy-classes/geometry.htm> v časti *Properties*.
 - Geometrický objekt má metódu *getPart()*, ktorá vracia pole bodov objektu pre časť geometrie alebo pole obsahujúce polia, každé pre jednu časť objektu.
-

Práca s geometrickými objektmi

```
Python
>>> arcpy.Select_analysis("mesta","mesta_nad_50000", "pocet > 50000")
<Result 'C:\\Users\\Vlado\\Documents\\ArcGIS\\Default.gdb\\mesta_nad_50000'>
>>> pole=[]
>>> with arcpy.da.SearchCursor("mesta_nad_50000","Shape@") as skurzor:
...     for riadok in skurzor:
...         pole.append(riadok[0])
...
>>> for a in pole:
...     print a.getPart()
...
-417485,923700001 -1228783,8908 NaN NaN
-430523,172899999 -1190623,7206 NaN NaN
-501096,9965 -1326122,0112 NaN NaN
-331005,180100001 -1199269,9118 NaN NaN
-261922,871199999 -1208498,9525 NaN NaN
-458373,879799999 -1221904,8489 NaN NaN
-497702,5022 -1203632,857 NaN NaN
522251,057200000 1260210,2027 NaN NaN
```

Práce s geometrickými objekty

- Ukážka dostupných vlastností a metod geometrických objektov.



```
Python
```

```
>>> for a in pole:  
...     print a.
```

- JSON
- WKB
- WKT
- angleAndDistanceTo
- area
- boundary
- buffer
- centroid
- clip

```
a.  
A PointGeometry is a  
shape that  
has neither  
length nor  
area at a  
given scale.
```

InsertCursor

- Vkladá hodnoty do atribútových polí.
- Vkladá geometriu do stĺpca s geometriou.
- Nezabudnúť na vymazanie kurzora pomocou del, kvôli locku.

```
Python
>>> ikurzor = arcpy.da.InsertCursor("mesta_nad_50000", ["nazov", "Shape"])
>>> ikurzor.insertRow(("Nový bod", arcpy.Point(-473000,-1259000)))
10L
>>> del ikurzor
>>>
```

- Funkcia Point sa používa na tvorbu bodových objektov (voliteľné parametre X, Y, M, Z, ID).
-

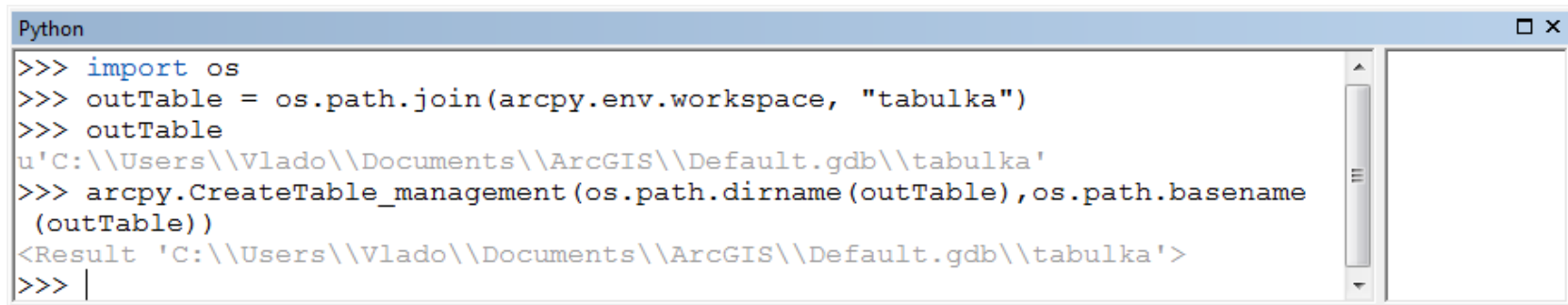
Zápis hodnôt do zoznamu

- vytvorenie prázdneho zoznamu:
zoznam = [] alebo zoznam = list()
- zápis hodnôt do zoznamu:
zoznam.append()

```
Python
>>> pole = []
>>> with arcpy.da.SearchCursor("mesta", ["pocet", "nazov"]) as kurzor:
...     for row in kurzor:
...         if row[0]>ZV:
...             pole.append([row[0],row[1]])
...
>>> pole
[[85052, u'Bansk\xel Bystrica'], [60917, u'Martin'], [87569, u'Nitra'],
 [55303, u'Poprad'], [93147, u'Pre\u0161lov'], [54395, u'Prievidza'], [59039,
 u'Tren\u010d\xeddn'], [70202, u'Trnava'], [86811, u'\u017dilina']]
>>>
```

Vytvorenie tabuľky

- Nástroj CreateTable_management
- Parametre: out_path, out_name a nepovinné parametre



```
Python
>>> import os
>>> outTable = os.path.join( arcpy.env.workspace, "tabulka")
>>> outTable
u'C:\\Users\\Vlado\\Documents\\ArcGIS\\Default.gdb\\tabulka'
>>> arcpy.CreateTable_management( os.path.dirname( outTable ), os.path.basename
(outTable) )
<Result 'C:\\Users\\Vlado\\Documents\\ArcGIS\\Default.gdb\\tabulka'>
>>> |
```

- Všimnite si použitie funkcií *os.path.dirname()* a *os.path.basename()* z knižnice *os*. Na ich použitie je potrebné importovať knižnicu *os*.
-

Vytvorenie tabuľky

- Pridanie stĺpcov do novovytvorenej tabuľky:

```
Python
>>> arcpy.AddField_management(outTable, "Nazov", "TEXT")
<Result 'C:\\Users\\Vlado\\Documents\\ArcGIS\\Default.gdb\\tabulka'>
>>> arcpy.AddField_management(outTable, "Pocet", "Long")
<Result 'C:\\Users\\Vlado\\Documents\\ArcGIS\\Default.gdb\\tabulka'>
>>>
```

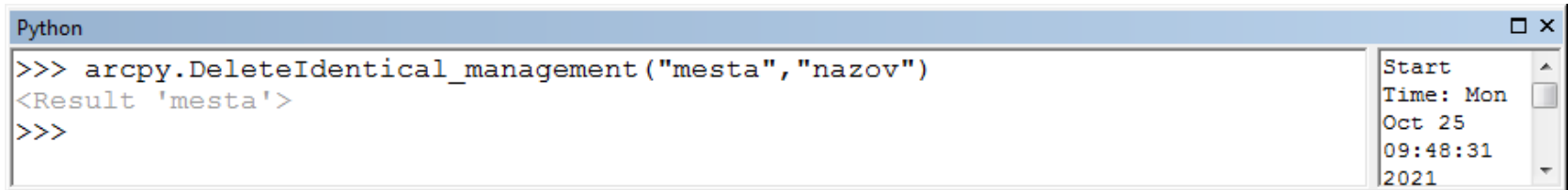
- A ich naplnenie hodnotami z poľa:

```
Python
>>> ikurzor = arcpy.da.InsertCursor("tabulka", ["Nazov", "Pocet"])
>>> for zaznam in pole:
...     ikurzor.insertRow([zaznam[1], zaznam[0]])
...
>>> del ikurzor
>>> |
```

- Hodnoty pre nový riadok sa zadávajú ako zoznam!
-

Vymazanie identických záznamov

- Nástroj DeleteIdentical_management
- Parametre:
 - in_dataset – vstupná tabuľka alebo feature class
 - fields – stĺpec alebo zoznam stĺpcov, v ktorých sa budú hľadať identické záznamy
 - {x,y tolerance}, {z tolerance}



```
Python
>>> arcpy.DeleteIdentical_management("mesta","nazov")
<Result 'mesta'>
>>>
```

Start
Time: Mon
Oct 25
09:48:31
2021

Vytvorenie dočasnej vrstvy

- Nástroj `MakeFeatureLayer_management()`
- Parametre:
 - `in_features` - vstupná vrstva
 - `out_layer` - názov dočasnej vrstvy
 - `{where_clause}` – kritériá definované pomocou SQL výrazu
- Dočasná vrstva sa dá použiť ako vstup do všetkých nástrojov, ktoré akceptujú feature layer ako vstup.

```
Python
>>> arcpy.MakeFeatureLayer_management("mesta","MestaLayer")
<Result 'MestaLayer'>
>>> arcpy.MakeFeatureLayer_management("hranice","HraniceLayer"," nazkraja
LIKE 'Trenčiansky kraj'")
<Result 'HraniceLayer'>
>>> |
```


Atribútové dopyty

- Nástroj `SelectLayerByAttribute_management()`
 - Parametre:
 - `in_layer_or_view` - vstup
 - `{selection_type}` – typ výberu
 - `{where clause}` – kritériá definované pomocou SQL výrazu
 - Na rozdiel od `Select_analysis` nevytvára novú vrstvu, iba robí výber (pridáva k výberu, maže z výberu, vyberá z výberu, robí obrátený výber...)
-

Atribútové dopyty

```
Python
>>> arcpy.SelectLayerByAttribute_management("MestaLayer", "NEW_SELECTION",
"pocet > 1000")
<Result 'MestaLayer'>
>>>
```

Priestorové dopyty

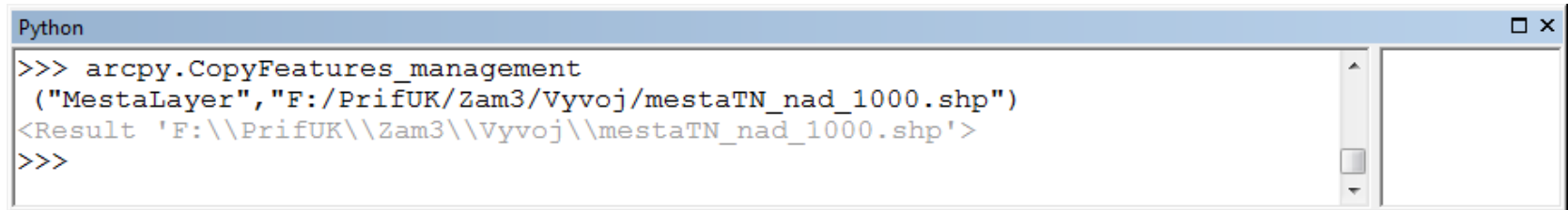
- Nástroj `SelectLayerByLocation_management()`
 - Parametre:
 - `in_layer`- vstupná vrstva
 - `{overlap type}` – typ priestorového vzťahu INTERSECT, CONTAINS, WITHIN, ...
 - `{select_features}`- objekty zo vstupnej vrstvy budú vybrané na základe priestorového vzťahu k objektom tejto vrstvy
 - `{search_distance}` – vzdialenosť, platná iba niektoré vzťahy
 - `{selection type}` - typ výberu
-

Priestorové dopyty

```
Python
>>> arcpy.SelectLayerByLocation_management
("MestaLayer", "WITHIN", "HraniceLayer", "", "SUBSET_SELECTION")
<Result 'MestaLayer'>
>>>
```

Uloženie výberu do novej vrstvy

- Nástroj CopyFeatures_management()
- Parametre:
 - in_features – vstupná vrstva
 - out_feature_class-cesta a názov novej vrstvy, do ktorej budú uložené vybrané objekty



```
Python
>>> arcpy.CopyFeatures_management
("MestaLayer", "F:/PrifUK/Zam3/Vyvoj/mestaTN_nad_1000.shp")
<Result 'F:\\PrifUK\\Zam3\\Vyvoj\\mestaTN_nad_1000.shp'>
>>>
```

Otázky?

Ďakujem za pozornosť.
