แนะนำการใช้งาน PYTHON ทางด้านภูมิสารสนเทศ

อ.ดร.ศักดิ์ดา หอมหวล

ภาควิชาภูมิศาสตร์ มหาวิทยาลัยเชียงใหม่

หัวข้อบรรยาย

- ภาษา Python
- Python package สำหรับงานภูมิสารสนเทศ
- การเขียนโปรแกรม
- แนะนำ ArcPy
- ทดลองใช้งาน

ภาษา Python

Paradigm	Multi-paradigm: functional, imperative, object-oriented, structured, reflective
Designed by	Guido van Rossum
Developer	Python Software Foundation
First appeared	1990; 30 years ago[1]
Stable release	3.8.3 / 13 May 2020; 58 days ago[2]
Preview release	3.9.0b4 / 3 July 2020; 7 days ago[3]
OS	Linux, macOS, Windows Vista (and newer) and more
License	Python Software Foundation License
Filename extensions	.py, .pyi, .pyc, .pyd, .pyo (prior to 3.5),[5] .pyw, .pyz (since 3.5)[6]
Website	www.python.org

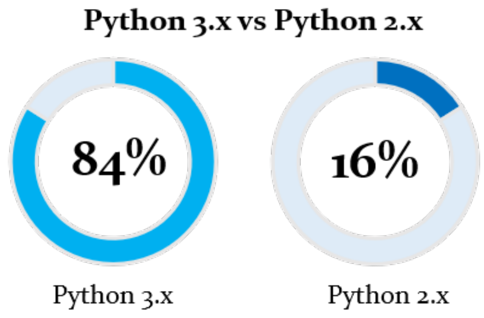
Guido van Rossum

python™

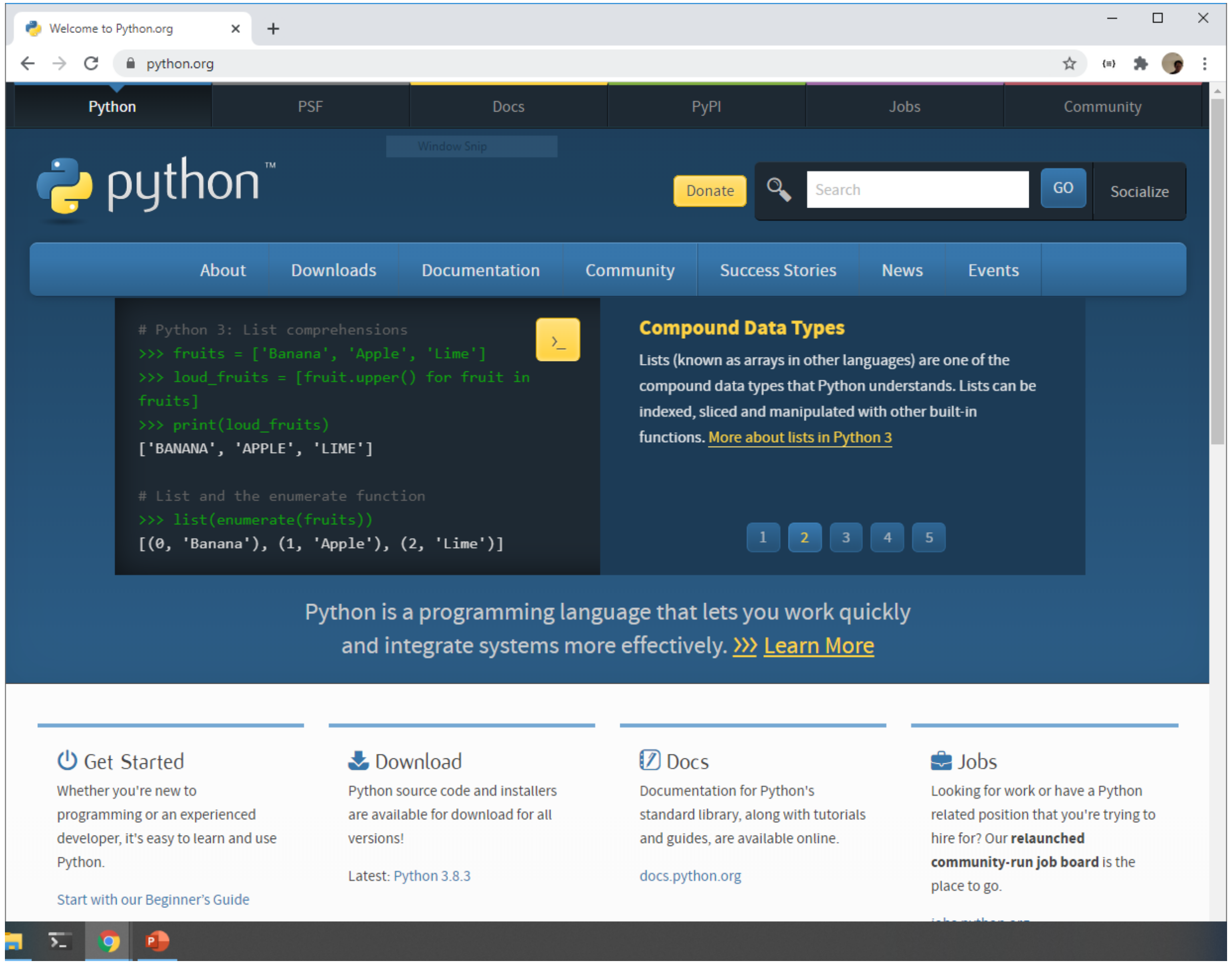
ข้อมูลจาก https://en.wikipedia.org/wiki/Python_(programming_language)

ภาษา Python

- Open-source programming language
- Easy to learn
- Cross-platform
- Embeddable (making ArcGIS scriptable)
- Stable and mature
- Python 2.0, released in 2000
- Python 3.0, released in 2008
- A large user community



https://www.9experttraining.com/articles/



```
# Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

# List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]
```

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. More about lists in Python 3

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Start with our Beginner's Guide

Download

Python source code and installers are available for download for all versions!

Latest: Python 3.8.3

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

docs.python.org

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

ศึกษารายละเอียดเพิ่มเติม python.org.

Python package สำหรับงานภูมิสารสนเทศ

https://gisgeography.com/python-libraries-gis-mapping/

- ArcPy: arcgisscripting module https://desktop.arcgis.com/en/arcmap/latest/analyze/arcpy/what-is-arcpy-.htm
- **Geopandas: open source spatial data analysis tools** https://geopandas.org/
- **GDAL/OGR: translator library for raster and vector geospatial data formats** https://gdal.org/
- **RSGISLib:** remote sensing tools for raster processing and analysis https://www.rsgislib.org/
- **PyProj:** cartographic projections and coordinate transformations library http://pyproj4.github.io/pyproj/stable/

การเขียนโปรแกรม Python เบื้องต้น

Case sensitivity

Space and tabs don’t mix

Objects (วิธีเขียนโปรแกรมเชิงวัตถุ)

Scope (การตั้งชื่อตัวแปร กำหนดตัวแปร)

ใช้ Colons (:) หลังคำสั่ง if, for, while, def

Variable and String

```
a = 1
b = 2
c = a + b
Name = “Sakda”
```

Hello world

```
print("Hello world!")
```

Hello world with a variable

```
msg = "Hello world!"
print(msg)
```

Concatenation

```
first_name = 'albert’
last_name = 'einstein’
full_name = first_name + ' ' + last_name
print(full_name)
```

การเขียนโปรแกรม Python เบื้องต้น

Types and Type Conversion

เราสามารถแปลงค่าไปมาระหว่างตัวแปรต่างชนิดกันได้

```
#String to Integer
x = int("17")
y = int(4.8)
print ("x =",x, ",y =",y, " and x - y =",x - y)

#String to Float
x = float(17)
y = float("4")
print ("x =",x, ",y =",y, " and x - y =",x - y)
x = 17.0 ,y = 4.0 and x - y = 13.0
```

การเขียนโปรแกรม Python เบื้องต้น

ตัวแปร List

List is a collection which is ordered and changeable. Allows duplicate members.

Make a list

```
bikes = ['trek', 'redline', 'giant’]
```

Get the first item in a list

```
first_bike = bikes[0]
```

Get the last item in a list

```
last_bike = bikes[-1]
```

Looping through a list

```
for bike in bikes:
      print(bike)
```

Adding items to a list

```
bikes = []
bikes.append('trek’)
bikes.append('redline’)
bikes.append('giant’)
```

Remove

```
bikes.pop(1)
```

การเขียนโปรแกรม Python เบื้องต้น

Numerical lists

```
squares = []
for x in range(1, 11):
      squares.append(x**2)
```

Slicing a list

```
finishers = ['sam', 'bob', 'ada', 'bea’]
first_two = finishers[:2]
```

Copying a list

```
copy_of_bikes = bikes[:]
```

การเขียนโปรแกรม Python เบื้องต้น

ตัวแปร Dictionary

Dictionary is a collection which is unordered, changeable and indexed. No duplicate members.

A simple dictionary

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964,
    "year": 1964
}
print(thisdict)
```

Accessing a value

```
print("alien['color’])
```

Adding a new key-value pair

```
alien['x_position'] = 0
```

Looping through all key-value pairs

```
fav_numbers = {'eric': 17, 'ever': 4}
for name, number in fav_numbers.items():
       print(name + ' loves ' + str(number))
```

การเขียนโปรแกรม Python เบื้องต้น

ตัวแปร Tuple

A tuple is a collection which is **ordered** and **unchangeable.**

```
fruit = ("apple", "banana", "cherry", "orange", "kiwi”)

print(fruit)

print(fruit[1])   # "banana"

print(fruit[2:3]) # "banana", "cherry"
```

การเขียนโปรแกรม Python เบื้องต้น

ตัวแปร Set

A set is a collection which is unordered and unindexed. In Python sets are written with curly brackets.

```
theset = {"apple", "banana", "cherry"}

for x in theset:
  print(x)
```

การเขียนโปรแกรม Python เบื้องต้น

Python Loops

For loops

```
fruits = ["apple", "banana", "cherry"]

for x in fruits:
  print(x)
```

While Loops

```
i = 1
while i < 6:
  print(i)
  i += 1
```

การเขียนโปรแกรม Python เบื้องต้น

Conditional

equals

```
x == 42
```

not equal

```
x != 42
```

greater than

```
x > 42
```

or equal to

```
x >= 42
```

less than

```
x < 42
```

or equal to

```
x <= 42
```

Conditional test with lists

```
‘apple' in fruit
‘banana' not in fruit
```

การเขียนโปรแกรม Python เบื้องต้น

If statements

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

แบบ elif

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

Indentation สำคัญที่สุด!!

แบบ else

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

การเขียนโปรแกรมด้วย Python

การเขียน Function

A simple function

```
def greet_user():
      print("Hello!")

greet_user()
```

Passing an argument

```
def greet_user(username):
      print("Hello, " + username + "!")

greet_user('jesse')
```

การเขียนโปรแกรมด้วย Python

การเขียน Function

Default values for parameters

```
def make_pizza(topping='bacon’):
      print("Have a " + topping + " pizza!")

make_pizza()
make_pizza('pepperoni')
```

Returning a value

```
def add_numbers(x, y):
      return x + y

sum = add_numbers(3, 5)
print(sum)
```

การเขียนโปรแกรมด้วย Python

การเรียกใช้ Python package

Import

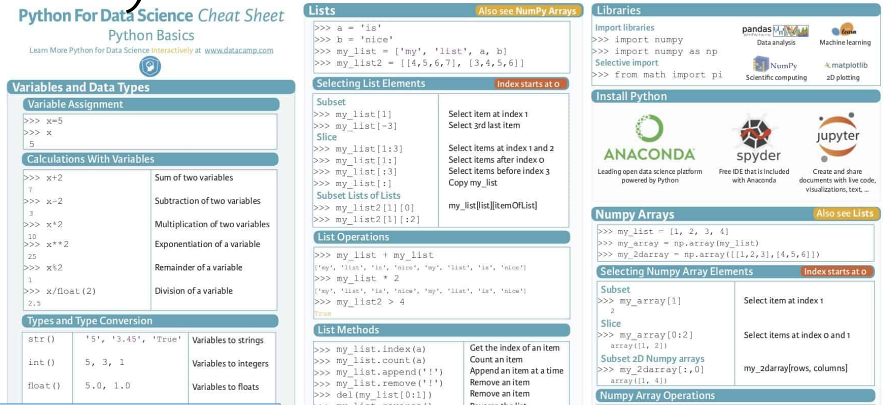
```
import libraries
import arcpy
import numpy as np
```

Selective import

```
from math import pi
```

ศึกษาด้วยตนเอง

Python Document



Python For Data Science Cheat Sheet

Python Basics

Learn More Python for Data Science Interactively at www.datacamp.com

Variables and Data Types

Variable Assignment

```
>>> x=5
>>> x
 5
```

Calculations With Variables

`>>> x+2`
`7`
	Sum of two variables

`>>> x-2`
`3`
	Subtraction of two variables

`>>> x*2`
`10`
	Multiplication of two variables

`>>> x**2`
`25`
	Exponentiation of a variable

`>>> x%2`
`1`
	Remainder of a variable

`>>> x/float(2)`
`2.5`
	Division of a variable

Types and Type Conversion

str()	'5', '3.45', 'True'	Variables to strings
int()	5, 3, 1	Variables to integers
float()	5.0, 1.0	Variables to floats

Lists Also see NumPy Arrays

```
>>> a = 'is'
>>> b = 'nice'
>>> my_list = ['my', 'list', a, b]
>>> my_list2 = [[4,5,6,7], [3,4,5,6]]
```

Selecting List Elements — Index starts at 0

Subset	
`>>> my_list[1]`	Select item at index 1
`>>> my_list[-3]`	Select 3rd last item
Slice	
`>>> my_list[1:3]`	Select items at index 1 and 2
`>>> my_list[1:]`	Select items after index 0
`>>> my_list[:3]`	Select items before index 3
`>>> my_list[:]`	Copy my_list
Subset Lists of Lists	

`>>> my_list2[1][0]`
`>>> my_list2[1][:2]`
	my_list[list][itemOfList]

List Operations

```
>>> my_list + my_list
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list * 2
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list2 > 4
True
```

List Methods

`>>> my_list.index(a)`	Get the index of an item
`>>> my_list.count(a)`	Count an item
`>>> my_list.append('!')`	Append an item at a time
`>>> my_list.remove('!')`	Remove an item
`>>> del(my_list[0:1])`	Remove an item
`>>> my_list.reverse()`	Reverse the list
`>>> my_list.extend('!')`	Append an item
`>>> my_list.pop(-1)`	Remove an item
`>>> my_list.insert(0,'!')`	Insert an item
`>>> my_list.sort()`	Sort the list

String Operations — Index starts at 0

```
>>> my_string[3]
>>> my_string[4:9]
```

String Methods

`>>> my_string.upper()`	String to uppercase
`>>> my_string.lower()`	String to lowercase
`>>> my_string.count('w')`	Count String elements
`>>> my_string.replace('e', 'i')`	Replace String elements
`>>> my_string.strip()`	Strip whitespaces

Libraries

Import libraries

```
>>> import numpy
>>> import numpy as np
```

Selective import

```
>>> from math import pi
```

Install Python

ANACONDA
Leading open data science platform powered by Python
spyder
Free IDE that is included with Anaconda
jupyter
Create and share documents with live code, visualizations, text, ...

Numpy Arrays Also see Lists

```
>>> my_list = [1, 2, 3, 4]
>>> my_array = np.array(my_list)
>>> my_2darray = np.array([[1,2,3],[4,5,6]])
```

Selecting Numpy Array Elements — Index starts at 0

Subset	

`>>> my_array[1]`
`2`
	Select item at index 1

Slice	

`>>> my_array[0:2]`
`array([1, 2])`
	Select items at index 0 and 1

Subset 2D Numpy arrays	

`>>> my_2darray[:,0]`
`array([1, 4])`
	my_2darray[rows, columns]

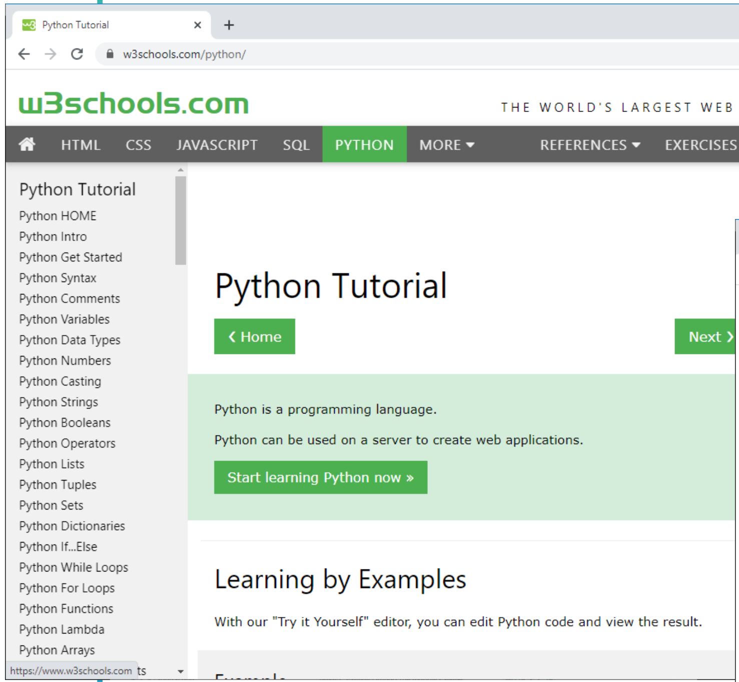
Numpy Array Operations

```
>>> my_array > 3
 array([False, False, False,  True], dtype=bool)
>>> my_array * 2
 array([2, 4, 6, 8])
>>> my_array + np.array([5, 6, 7, 8])
 array([6, 8, 10, 12])
```

Numpy Array Functions

`>>> my_array.shape`	Get the dimensions of the array
`>>> np.append(other_array)`	Append items to an array
`>>> np.insert(my_array, 1, 5)`	Insert items in an array
`>>> np.delete(my_array,[1])`	Delete items in an array
`>>> np.mean(my_array)`	Mean of the array
`>>> np.median(my_array)`	Median of the array
`>>> my_array.corrcoef()`	Correlation coefficient
`>>> np.std(my_array)`	Standard deviation

DataCamp
Learn Python for Data Science Interactively



Python Tutorial

- Python HOME
- Python Intro
- Python Get Started
- Python Syntax
- Python Comments
- Python Variables
- Python Data Types
- Python Numbers
- Python Casting
- Python Strings
- Python Booleans
- Python Operators
- Python Lists
- Python Tuples
- Python Sets
- Python Dictionaries
- Python If...Else
- Python While Loops
- Python For Loops
- Python Functions
- Python Lambda
- Python Arrays

Python Tutorial

❮ Home Next ❯

Python is a programming language.

Python can be used on a server to create web applications.

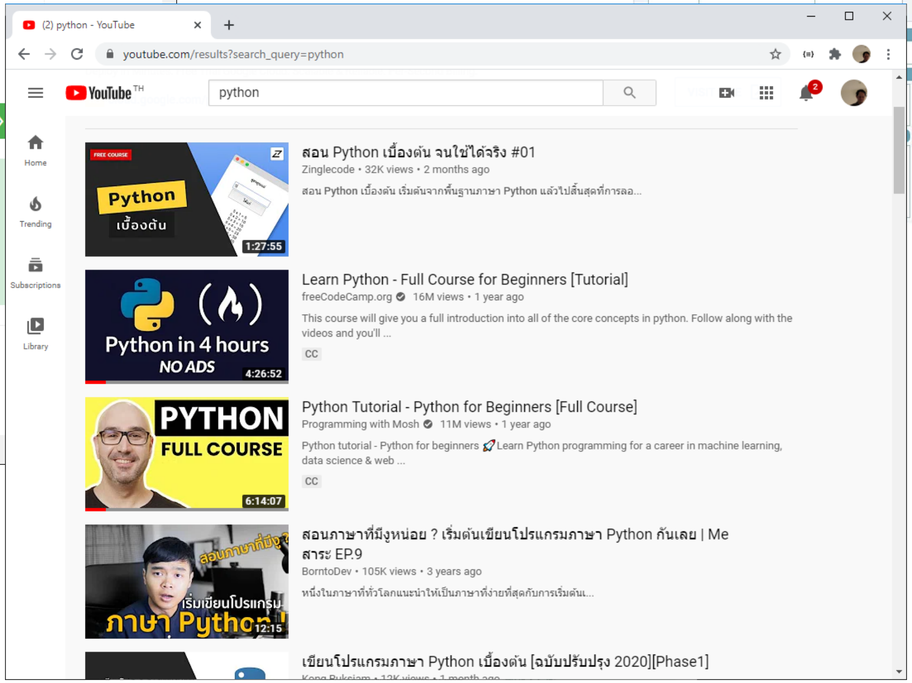
Start learning Python now »

Learning by Examples

With our "Try it Yourself" editor, you can edit Python code and view the result.

https://www.w3schools.com

เว็บไซต์



สอน Python เบื้องต้น จนใช้ได้จริง #01
Zinglecode • 32K views • 2 months ago
สอน Python เบื้องต้น เริ่มต้นจากพื้นฐานภาษา Python แล้วไปสิ้นสุดที่การลอ...

Learn Python - Full Course for Beginners [Tutorial]
freeCodeCamp.org • 16M views • 1 year ago
This course will give you a full introduction into all of the core concepts in python. Follow along with the videos and you'll ...
CC

Python Tutorial - Python for Beginners [Full Course]
Programming with Mosh • 11M views • 1 year ago
Python tutorial - Python for beginners 🚀Learn Python programming for a career in machine learning, data science & web ...
CC

สอนภาษาที่มีงูหน่อย ? เริ่มต้นเขียนโปรแกรมภาษา Python กันเลย | Me สาระ EP.9
BorntoDev • 105K views • 3 years ago
หนึ่งในภาษาที่ทั่วโลกแนะนำให้เป็นภาษาที่ง่ายที่สุดกับการเริ่มต้น...

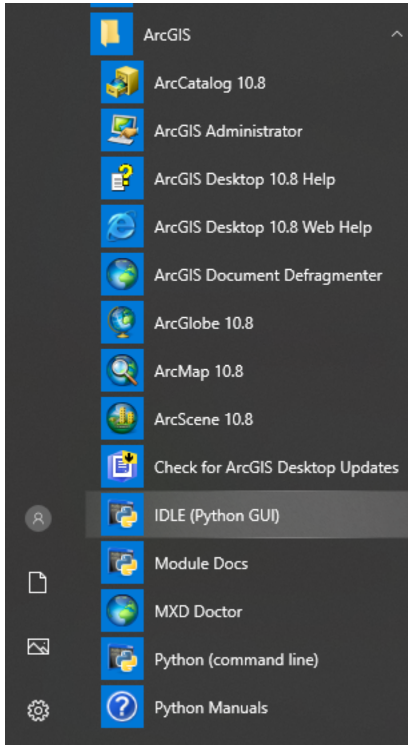
เขียนโปรแกรมภาษา Python เบื้องต้น [ฉบับปรับปรุง 2020][Phase1]
Kong Ruksiam • 12K views • 1 month ago

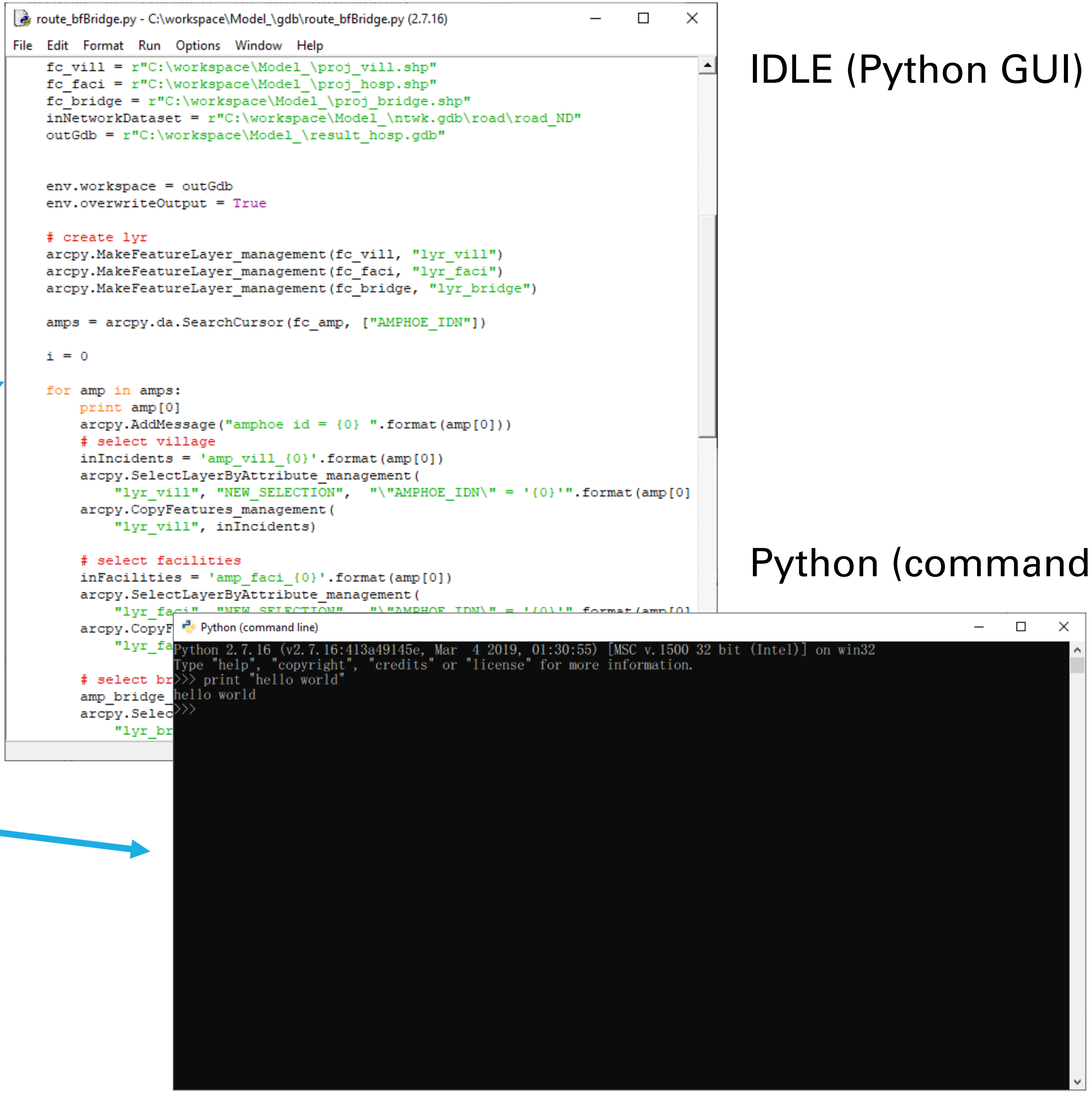
วิดีโอ

ArcPy

- Python module สามารถใช้งานกับข้อมูลภูมิสารสนเทศ
- ไม่ฟรี ต้องติดตั้ง ArcGIS
- มีเครื่องมือให้ใช้งานที่หลากหลาย
- มี document ของแต่ละ function, class, module
- มี Code completion รองรับหลาย Editor
- ArcGIS 10.x -> Python 2.x
- ArcGIS Pro -> Python 3.x

เครื่องมือที่ใช้พัฒนา



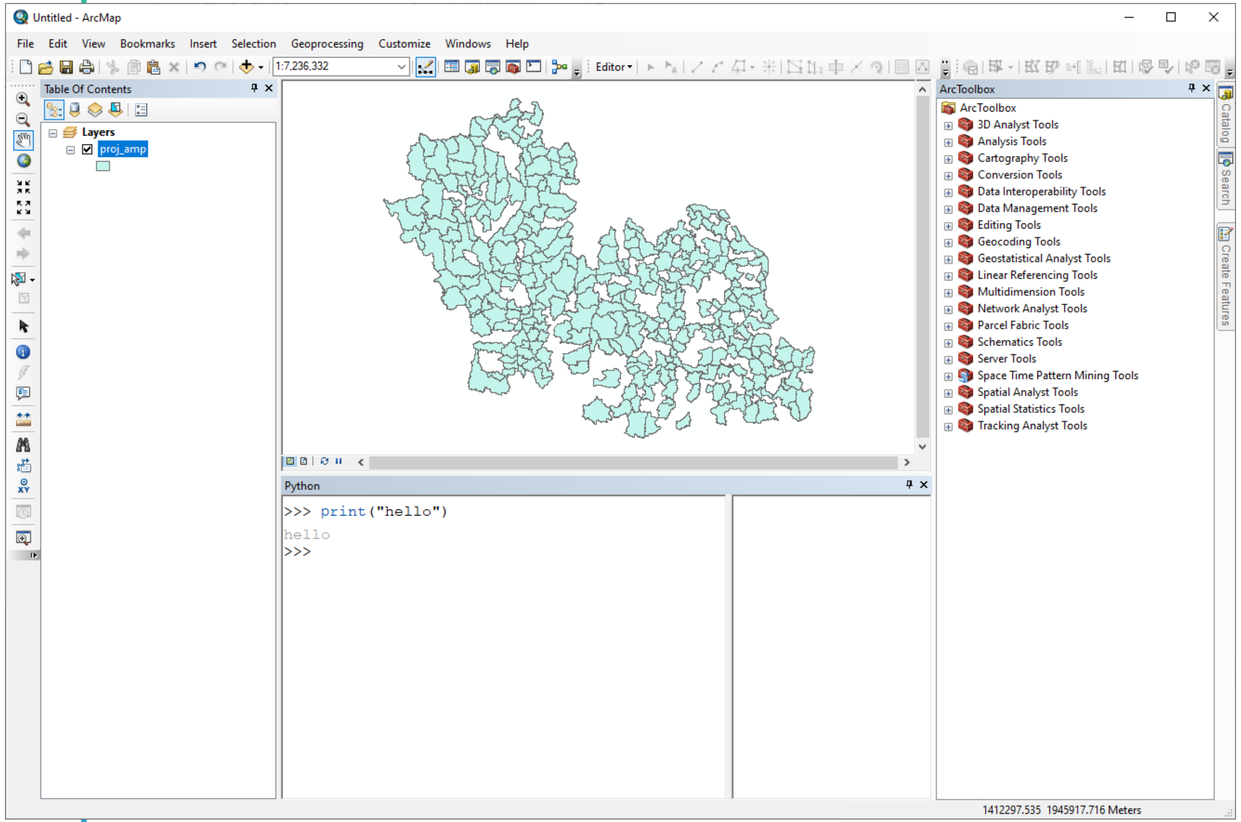

route_bfBridge.py - C:\workspace\Model_\gdb\route_bfBridge.py (2.7.16)
File Edit Format Run Options Window Help
Python (command line)

```
    fc_vill = r"C:\workspace\Model_\proj_vill.shp"
    fc_faci = r"C:\workspace\Model_\proj_hosp.shp"
    fc_bridge = r"C:\workspace\Model_\proj_bridge.shp"
    inNetworkDataset = r"C:\workspace\Model_\ntwk.gdb\road\road_ND"
    outGdb = r"C:\workspace\Model_\result_hosp.gdb"


    env.workspace = outGdb
    env.overwriteOutput = True

    # create lyr
    arcpy.MakeFeatureLayer_management(fc_vill, "lyr_vill")
    arcpy.MakeFeatureLayer_management(fc_faci, "lyr_faci")
    arcpy.MakeFeatureLayer_management(fc_bridge, "lyr_bridge")

    amps = arcpy.da.SearchCursor(fc_amp, ["AMPHOE_IDN"])

    i = 0

    for amp in amps:
        print amp[0]
        arcpy.AddMessage("amphoe id = {0} ".format(amp[0]))
        # select village
        inIncidents = 'amp_vill_{0}'.format(amp[0])
        arcpy.SelectLayerByAttribute_management(
            "lyr_vill", "NEW_SELECTION",  "\"AMPHOE_IDN\" = '{0}'".format(amp[0]
        arcpy.CopyFeatures_management(
            "lyr_vill", inIncidents)

        # select facilities
        inFacilities = 'amp_faci_{0}'.format(amp[0])
        arcpy.SelectLayerByAttribute_management(
            "lyr_faci", "NEW_SELECTION",  "\"AMPHOE_IDN\" = '{0}'".format(amp[0]
        arcpy.CopyF
            "lyr_fa

        # select br
        amp_bridge_
        arcpy.Selec
            "lyr_br
```

```
Python 2.7.16 (v2.7.16:413a49145e, Mar  4 2019, 01:30:55) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>>
```

IDLE (Python GUI)

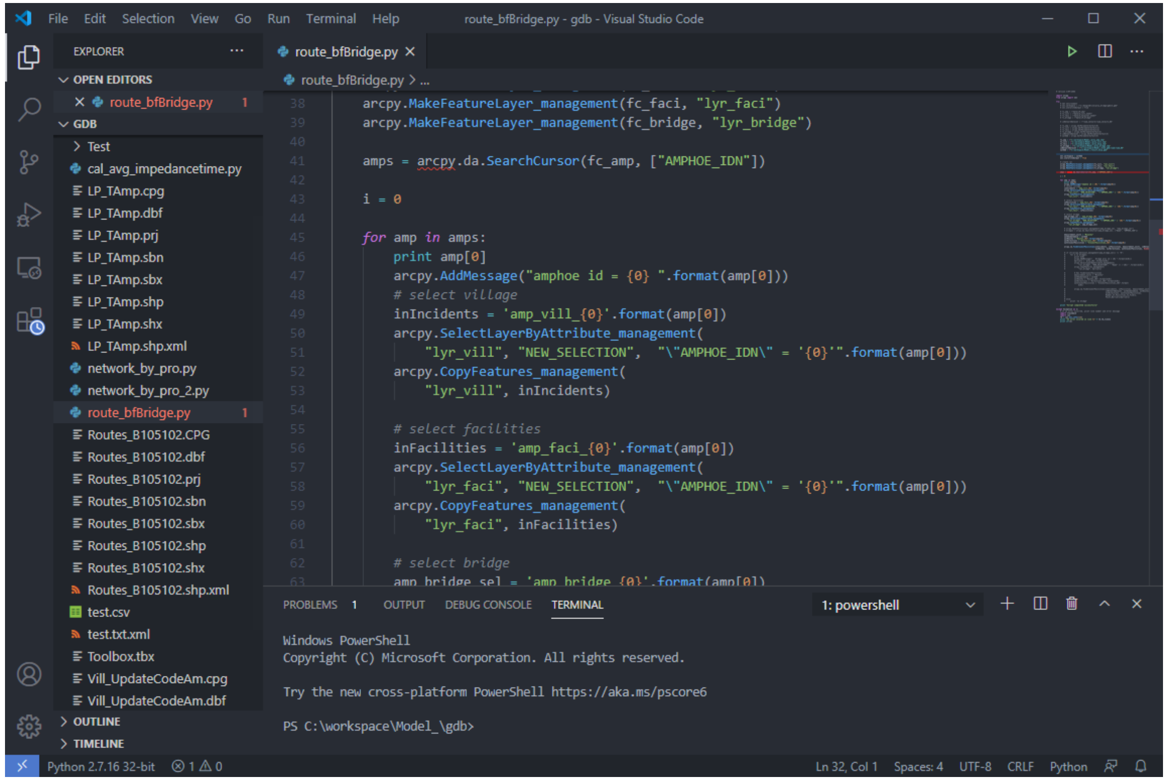
Python (command line)



```
>>> print("hello")
hello
>>>
```

Python console

Editor อื่นๆ



```
    arcpy.MakeFeatureLayer_management(fc_faci, "lyr_faci")
    arcpy.MakeFeatureLayer_management(fc_bridge, "lyr_bridge")

    amps = arcpy.da.SearchCursor(fc_amp, ["AMPHOE_IDN"])

    i = 0

    for amp in amps:
        print amp[0]
        arcpy.AddMessage("amphoe id = {0} ".format(amp[0]))
        # select village
        inIncidents = 'amp_vill_{0}'.format(amp[0])
        arcpy.SelectLayerByAttribute_management(
            "lyr_vill", "NEW_SELECTION",  "\"AMPHOE_IDN\" = '{0}'".format(amp[0]))
        arcpy.CopyFeatures_management(
            "lyr_vill", inIncidents)

        # select facilities
        inFacilities = 'amp_faci_{0}'.format(amp[0])
        arcpy.SelectLayerByAttribute_management(
            "lyr_faci", "NEW_SELECTION",  "\"AMPHOE_IDN\" = '{0}'".format(amp[0]))
        arcpy.CopyFeatures_management(
            "lyr_faci", inFacilities)

        # select bridge
        amp_bridge_sel = 'amp_bridge_{0}'.format(amp[0])
```

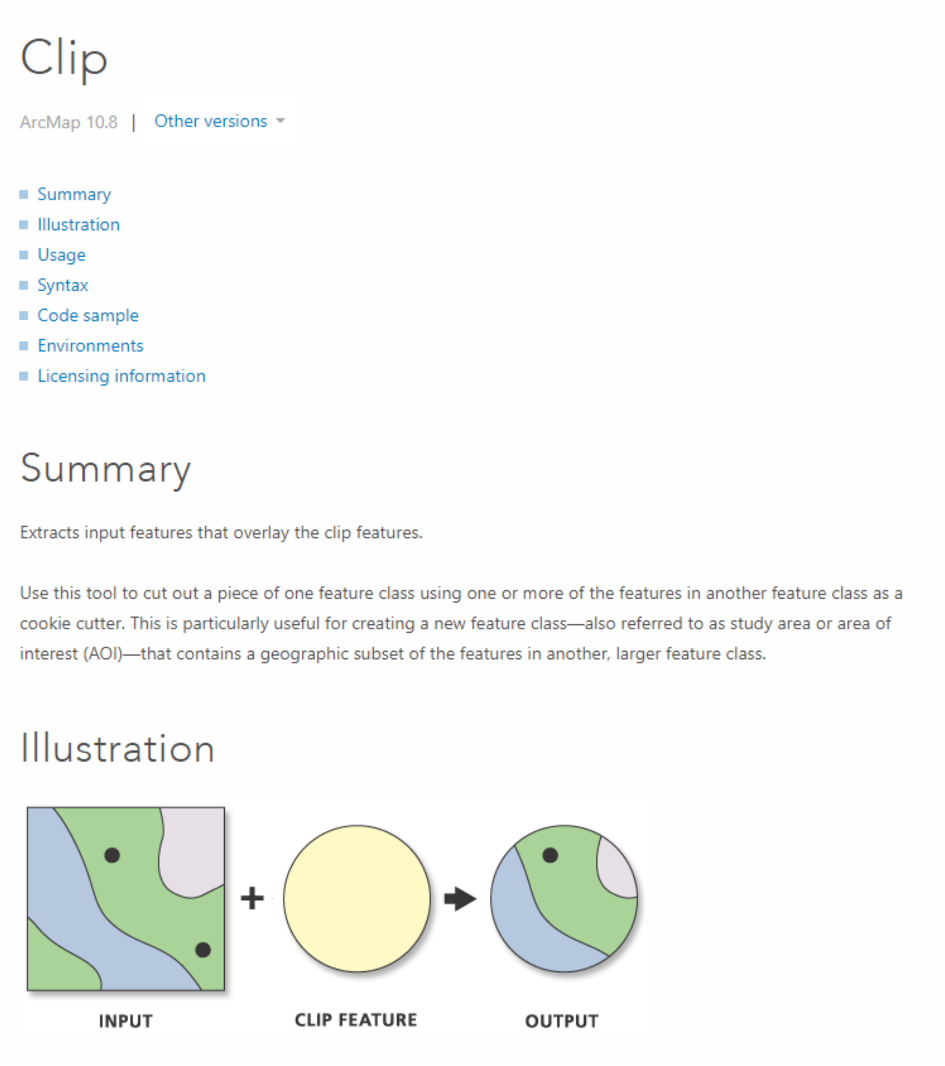
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\workspace\Model_\gdb>
```

ArcPy modules

- Data Access module (arcpy.da)*
- Mapping module (arcpy.mp)*
- Metadata module (arcpy.metadata)
- Sharing module (arcpy.sharing)
- Image Analystmodule (arcpy.ia)
- Spatial Analyst module (arcpy.sa)*
- Network Analyst modules (arcpy.nax and arcpy.na)*
- Workflow Manager module (arcpy.wmx)

https://pro.arcgis.com/en/pro-app/arcpy/get-started/a-quick-tour-of-arcpy.htm

ตัวอย่าง



Clip

ArcMap 10.8 | Other versions

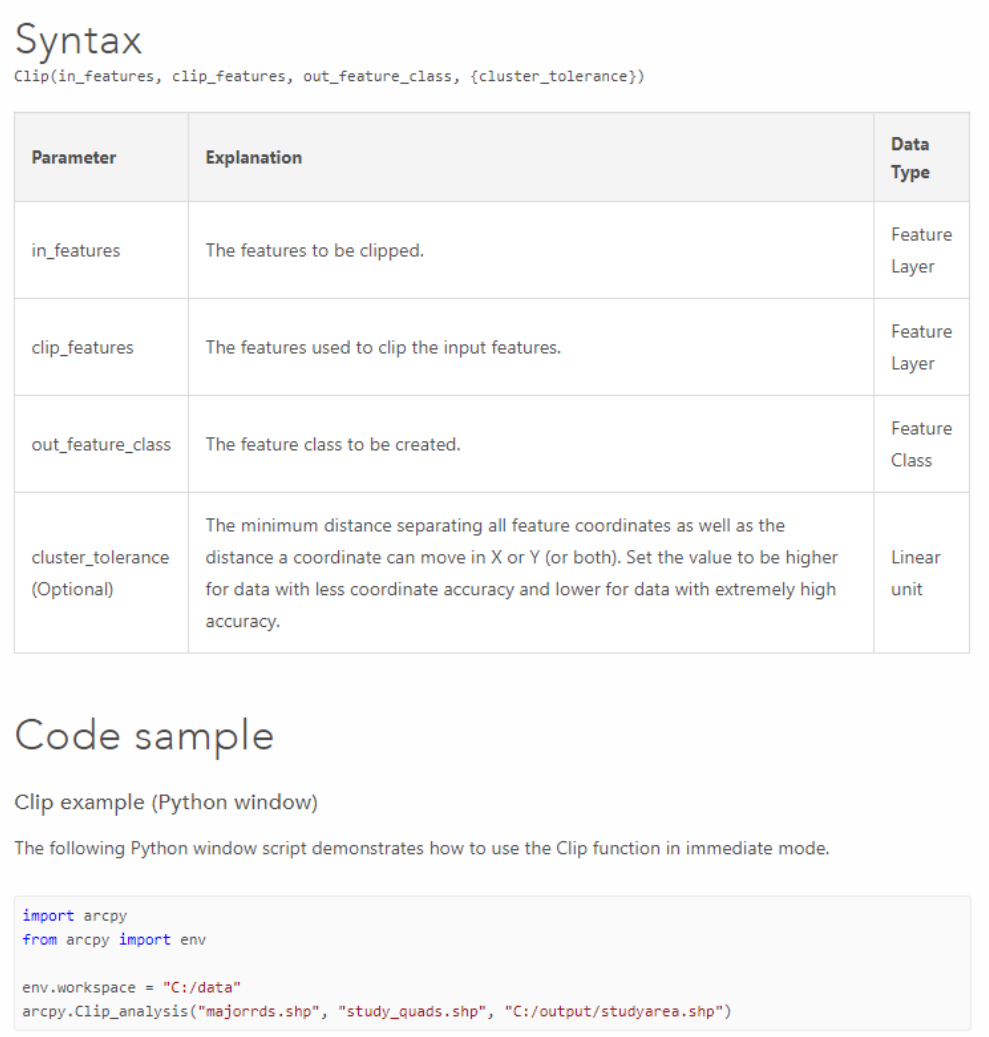
- Summary
- Illustration
- Usage
- Syntax
- Code sample
- Environments
- Licensing information

Summary

Extracts input features that overlay the clip features.

Use this tool to cut out a piece of one feature class using one or more of the features in another feature class as a cookie cutter. This is particularly useful for creating a new feature class—also referred to as study area or area of interest (AOI)—that contains a geographic subset of the features in another, larger feature class.

Illustration



Syntax

```
Clip(in_features, clip_features, out_feature_class, {cluster_tolerance})
```

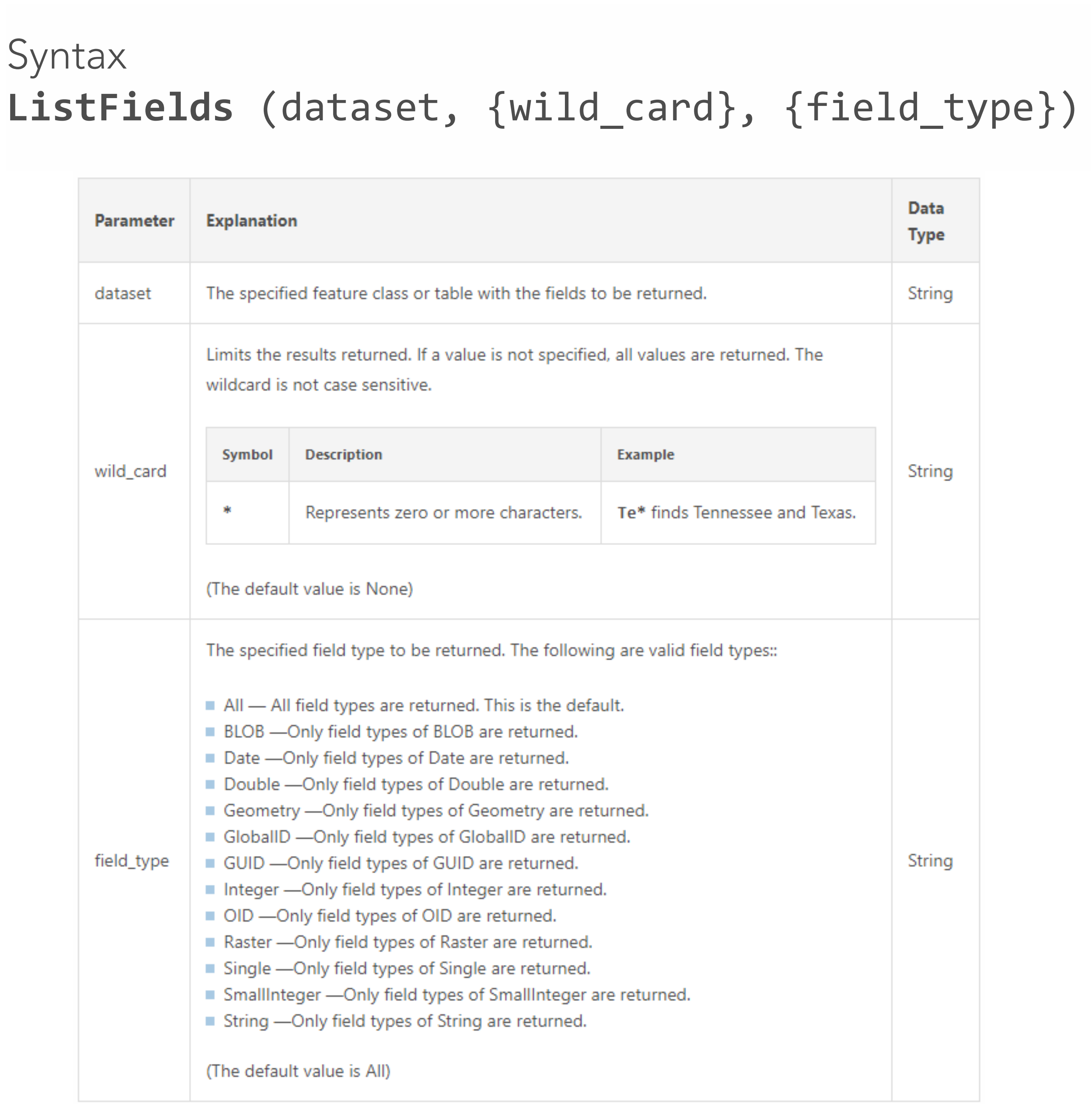
Parameter	Explanation	Data Type
in_features	The features to be clipped.	Feature Layer
clip_features	The features used to clip the input features.	Feature Layer
out_feature_class	The feature class to be created.	Feature Class
cluster_tolerance (Optional)	The minimum distance separating all feature coordinates as well as the distance a coordinate can move in X or Y (or both). Set the value to be higher for data with less coordinate accuracy and lower for data with extremely high accuracy.	Linear unit

Code sample

Clip example (Python window)

The following Python window script demonstrates how to use the Clip function in immediate mode.

```
import arcpy
from arcpy import env

env.workspace = "C:/data"
arcpy.Clip_analysis("majorrds.shp", "study_quads.shp", "C:/output/studyarea.shp")
```



Syntax

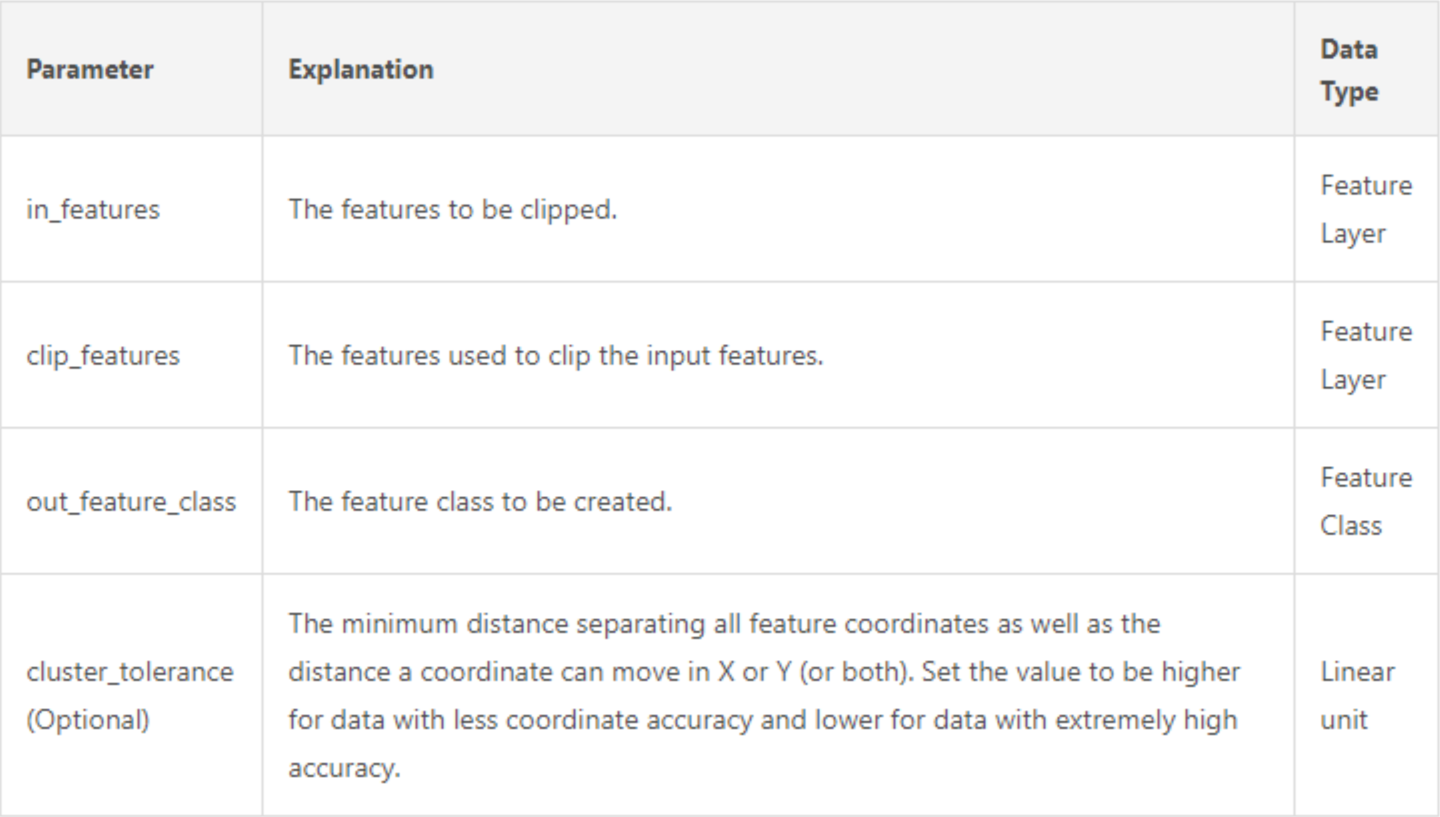
```
ListFields (dataset, {wild_card}, {field_type})
```

<table>
<tr><th>Parameter</th><th>Explanation</th><th>Data Type</th></tr>
<tr><td>dataset</td><td>The specified feature class or table with the fields to be returned.</td><td>String</td></tr>
<tr><td>wild_card</td><td>Limits the results returned. If a value is not specified, all values are returned. The wildcard is not case sensitive.
<table><tr><th>Symbol</th><th>Description</th><th>Example</th></tr><tr><td>*</td><td>Represents zero or more characters.</td><td><b>Te*</b> finds Tennessee and Texas.</td></tr></table>
(The default value is None)</td><td>String</td></tr>
<tr><td>field_type</td><td>The specified field type to be returned. The following are valid field types::
■ All — All field types are returned. This is the default.
■ BLOB —Only field types of BLOB are returned.
■ Date —Only field types of Date are returned.
■ Double —Only field types of Double are returned.
■ Geometry —Only field types of Geometry are returned.
■ GlobalID —Only field types of GlobalID are returned.
■ GUID —Only field types of GUID are returned.
■ Integer —Only field types of Integer are returned.
■ OID —Only field types of OID are returned.
■ Raster —Only field types of Raster are returned.
■ Single —Only field types of Single are returned.
■ SmallInteger —Only field types of SmallInteger are returned.
■ String —Only field types of String are returned.
(The default value is All)</td><td>String</td></tr>
</table>

```python
import arcpy
# For each field in the Hospitals feature class, print
# the field name, type, and length.
fields = arcpy.ListFields("c:/data/municipal.gdb/hospitals")
for field in fields:
    print("{0} is a type of {1} with a length of {2}"
          .format(field.name, field.type, field.length))
```

Syntax

```
Clip(in_features, clip_features, out_feature_class, {cluster_tolerance})
```

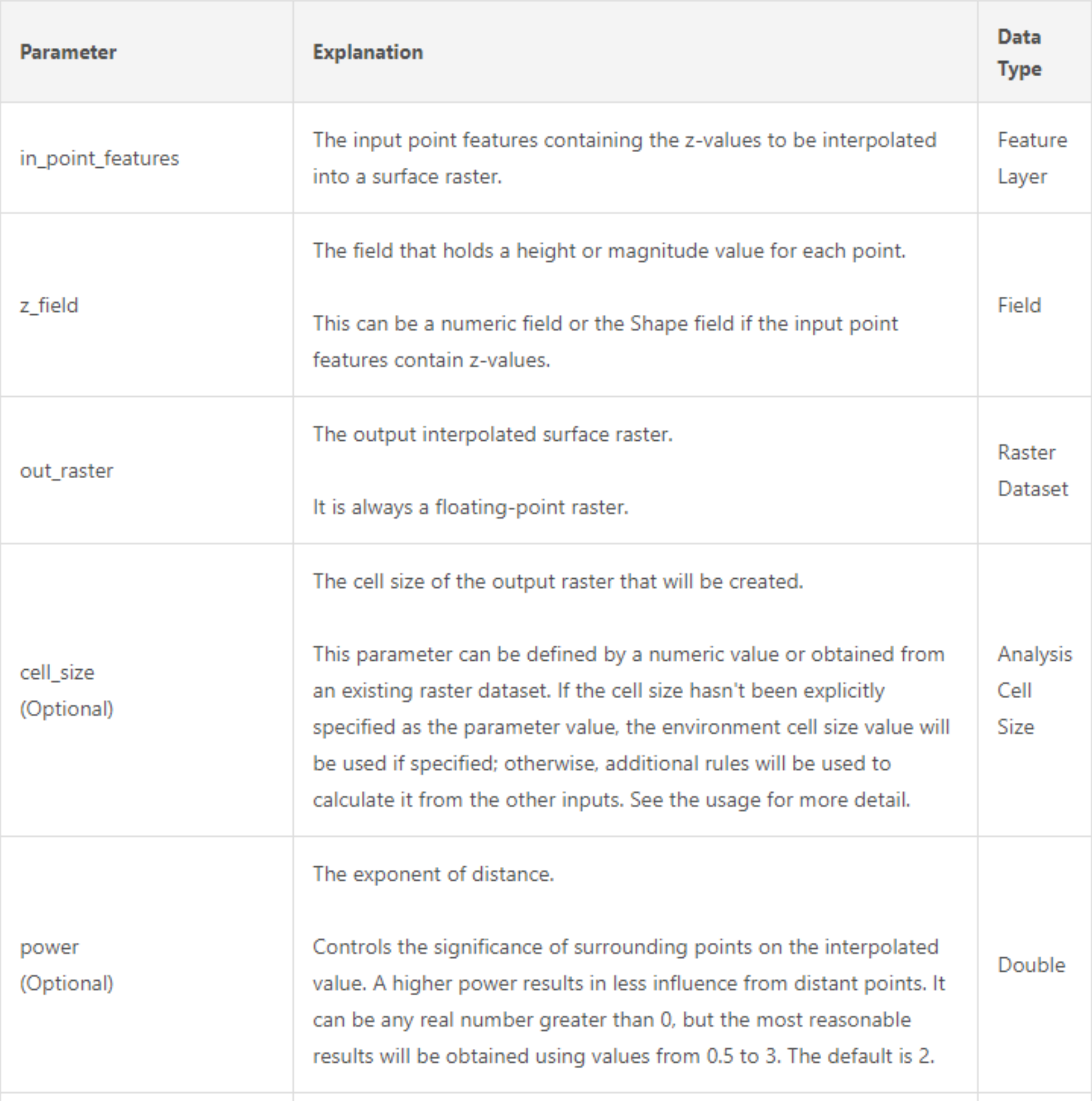


Parameter	Explanation	Data Type
in_features	The features to be clipped.	Feature Layer
clip_features	The features used to clip the input features.	Feature Layer
out_feature_class	The feature class to be created.	Feature Class
cluster_tolerance (Optional)	The minimum distance separating all feature coordinates as well as the distance a coordinate can move in X or Y (or both). Set the value to be higher for data with less coordinate accuracy and lower for data with extremely high accuracy.	Linear unit

```python
import arcpy
from arcpy import env
env.workspace = "C:/data"
arcpy.Clip_analysis("majorrds.shp", "study_quads.shp", "C:/output/studyarea.shp")
```

Syntax

Idw(in_point_features, z_field, out_raster, {cell_size}, {power}, {search_radius}, {in_barrier_polyline_features})



Parameter	Explanation	Data Type
in_point_features	The input point features containing the z-values to be interpolated into a surface raster.	Feature Layer
z_field	The field that holds a height or magnitude value for each point. This can be a numeric field or the Shape field if the input point features contain z-values.	Field
out_raster	The output interpolated surface raster. It is always a floating-point raster.	Raster Dataset
cell_size (Optional)	The cell size of the output raster that will be created. This parameter can be defined by a numeric value or obtained from an existing raster dataset. If the cell size hasn't been explicitly specified as the parameter value, the environment cell size value will be used if specified; otherwise, additional rules will be used to calculate it from the other inputs. See the usage for more detail.	Analysis Cell Size
power (Optional)	The exponent of distance. Controls the significance of surrounding points on the interpolated value. A higher power results in less influence from distant points. It can be any real number greater than 0, but the most reasonable results will be obtained using values from 0.5 to 3. The default is 2.	Double

```
import arcpy
from arcpy import env

# Set environment settings
env.workspace = "C:/data”

# Set local variables
inPointFeatures = "ca_ozone_pts.shp"
zField = "ozone"
outRaster = "C:/output/idwout01"
cellSize = 2000.0
power = 2
searchRadius = 150000

# Check out the ArcGIS 3D Analyst extension license
arcpy.CheckOutExtension("3D")

# Execute IDW
arcpy.Idw_3d(inPointFeatures, zField, outRaster, cellSize, power, searchRadius)
```

ลอง Dev ! กัน

- ลองเขียน ListFields
- ลองเขียน Buffer
- ลองเขียน Interpolation