

Using Python In ArcGIS: A Gentle Introduction

Lyzi Diamond

Oregon Department of Geology and Mineral Industries

July 25, 2013

Think about this:

"The notion that code is this hyperspecialized thing, scary punctuation soup on a dark screen, something that someone else does, is wrong, and it's toxic.

"There are people all over the world who don't consider how code might help them do their job, because they think it's a big leap. It's not. It's thousands of tiny steps, and everyone takes them in a different direction. A little bit of code goes a long way."

– "*Code, the newsroom, and self-doubt*", Noah Veltman, July 22, 2013.

<http://veltman.tumblr.com/post/56132893301/code-the-newsroom-and-self-doubt>

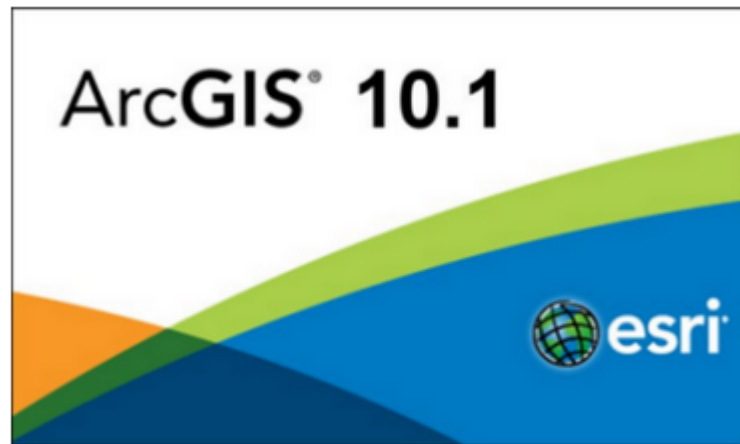
What is Python?

- High-level programming language
- Robust standard library
- Free and open
- Runs on almost every system
- Large module ecosystem
- Compiles strictly



Why would you use it?

- Desktop scripting
- Server-side languages and frameworks for powering web applications
- Full-scale program development
- **Scripting inside of other software programs**



Python and ArcGIS

- Anywhere you can use VB, you can use Python
- Field Calculations
- Label Expressions
- (Scripting with ArcPy)
- (Deployment of scripts on server)

What makes Python work?

- There are a few basic ideas to understand:
 - Variables
 - Operators
 - Conditionals
 - Loops
 - Functions
 - (Objects)
 - (Classes)

But first, some things of note.

- `# This is a comment.`
- `=` is an assignment operator. `==` is a way to test equivalence.
 - `x = 5` `# The variable x now holds the value 5.`
 - `x == 5` `# This statement will evaluate to True, because x is equal to 5.`
 - `x == 6` `# This statement will evaluate to False, because x is not equal to 6.`
 - `x = 6` `# The variable x now holds the value 6.`
 - `x == 6` `# This statement will now evaluate to True, because x is now equal to 6.`

Variables

- A variable is a placeholder name that can be changed and updated with a value; can use almost anything you want as the name (with some restrictions and reserved words).
- `x = 5` # `x` is the variable

Data Types

- Every data value in Python has a type.
 - Boolean: `x = True`
 - Integer: `x = 57`
 - Floating Number: `x = 3.14159`
 - String: `x = "Lyzi Diamond"`
 - List: `x = [3, 4, 7, 2, 5]`
- There are a few other data types, but we won't worry about them for now.

Boolean Operators

- Evaluating true/false statements. ArcGIS users, this should be familiar stuff.
 - `==` # checks if values of operands are equal
 - `!=` # checks if values of operands are not equal
 - `<>` # same as `!=`
 - `>` # checks if first operand is greater than second
 - `<` # checks if first operand is less than second
 - `>=` # checks if first operand is greater than or equal to second operand
 - `<=` #checks if first operand is less than or equal to second operand.

Conditionals

- Basic idea: if this is true, do this thing. Otherwise, do this other thing.
- Building block of many computing operations.
- Syntax:

```
x = 5  
y = 4
```

```
if (x > y):  
    print "x is greater than y!"  
elif (x < y):  
    print "x is less than y!"  
else:  
    print "x and y are the  
same!"
```

What will this print?

Loops

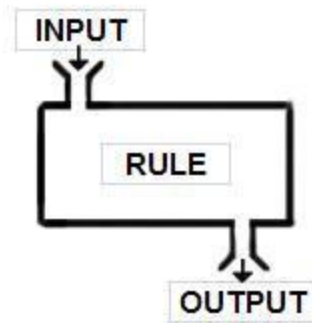
- Loops through a series of values and executes some operation on each iteration.
- `for` and `while` loops
- `for` loop:
 - ```
x = [3, 4, 7, 2, 5]
for number in x:
 print number + " sheep"
```
  - 3 sheep
  - 4 sheep
  - 7 sheep
  - 2 sheep
  - 5 sheep

# So far...

- Comments
- Assignment vs. equivalent operators
- Variables
- Data types
- Boolean operators
- Conditionals
- Loops
  
- Up next: Functions!

# Functions

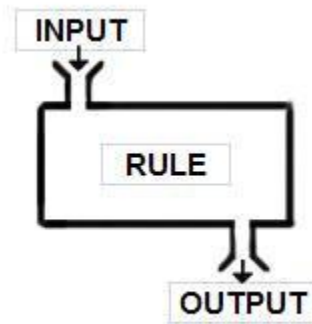
- Does anyone remember function machines from grade school math?



# Functions (cont'd)

- Functions in Python work the same way.
- Example time. Formula for converting meters to centimeters:  $m = cm * 100$
- ```
def metersToCentimeters(m):  
    cm = m * 100  
    return cm
```

Functions (cont'd)



- ```
def metersToCentimeters(m):
 cm = m * 100
 return cm
```
- Input (parameter/argument): m
- Rule (function): multiply by 100
- Output (return value): cm



# Functions (cont'd)

- ```
def metersToCentimeters(m):  
    cm = m * 100  
    return cm
```
- `metersToCentimeters(4)` will return 400
- `metersToCentimeters(5.8)` will return 580
- `metersToCentimeters(-1)` will return -100
- Notes:
 - The placeholder value in parentheses is called a “parameter.” The value you actually pass is called an “argument.”
 - When you use the function and pass an argument, you are “calling” the function.
 - Functions can largely be reused.
 - The name of the function is written in “camel-case.”
 - In Python, white space is important; all values inside of our function definition are two spaces indented.

Let's talk about ArcGIS, baby.

- Where can we use functions in ArcGIS?
 - Field Calculations
 - Label Expressions
 - (Batch Geoprocessing Scripts)
 - (Server-side Scripting)

Field Calculations

- What is really going on with field calcs?
Let's take a look.
- DEMO: Field Calculations with Python

Field Calculations (cont'd)

- So what actually happened there?
 - We defined a function `setRatio` that took two parameters: white and black.
 - The function contained a conditional (if/else) statement, saying that if black was equal to zero, set Ratio to be the same value as white. Otherwise, set Ratio to be the value of white divided by black.
 - We called the function with the fields WHITE and BLACK as arguments.
 - The function returned a value.

Field Calculations (cont'd)

- What was ArcGIS doing?
 - ArcGIS took our function and put it inside of a `for` loop.
 - That `for` loop iterated over each row/feature in the feature class, finding that row's value for WHITE, that row's value for BLACK, doing the calculation, and coming up with a return value.
 - That return value was then set as the value for Ratio for that particular row/feature.

Label Expressions

- Label Expressions are very similar, but have their own nuances. Let's take a look!
- DEMO: Label Expressions with Python

Label Expressions (cont'd)

- So what actually happened here?
 - We defined a function FindLabel with three parameters: [NAME], [Acres], and [SqYards].
 - When passed into this function, these three parameters became Strings.
 - We declared a variable x, and set it to the numerical value of [Acres].
 - We used a conditional statement to determine if [Acres] was less than 1, and had the function either return the area in yards or acres, depending.

Label Expressions (cont'd)

- What was ArcGIS doing?
 - ArcGIS took our function and put it inside of a `for` loop.
 - That `for` loop iterated over each row/feature in the feature class, finding that row's value for Acres, determining whether or not it was less than 1, and sending the appropriate return value to the Label Engine (Maplex).
 - Maplex then displayed the label, after styling and following the defined placement rules.

What did we learn today?

- Comments
- Assignment vs. equivalent operators
- Data types
- Boolean operators
- Variables
- Conditionals
- Loops
- Functions
- Python for Field Calcs
- Python for Label Expressions

In conclusion

- By writing one Python function, you can save yourself *tons* of time.
- What we went over today are the building blocks for writing full-scale geoprocessing scripts, desktop scripts, or solving a variety of other problems.
- I had to rush through this, but I'm happy to break it out into smaller, more in-depth sections. Talk to me!

Thank you!

"Coders change the world. They build new, amazing things faster than ever before. Anyone with imagination can learn to write code."

– *Jeff Wilke, Senior Vice President for Consumer Business, Amazon.com*

Interested in learning more?

- LearnPython.org – interactive tutorial
- LearnPythonTheHardWay.org/book
- PythonForBeginners.com – written tutorials
- Using Python with ArcGIS – <http://video.esri.com/watch/1158/using-python-with-arccgis> (ArcPy/scripting-specific)
- Hitchhiker's Guide to Python and ArcGIS – http://proceedings.esri.com/library/userconf/devsummit11/papers/tech/hitchhikers_guide_to_python_and_arccgis.pdf