Powering Up Your FME workspaces with Python

Data Types

Integer	-256, 15
Float	-253.23, 1.253e-10
String	"Hello", 'Goodbye', """Multiline"""
Boolean	True, False
List	[value, ...]
Tuple	(value, ...)[1]
Dictionary	{ key: value, ... }
Set	{ value, value, ... }[2]

[1] Parentheses usually optional
[2] Create an empty set with set()

Statements

If Statement
if *expression*:
    statements
elif *expression*:
    statements
else:
    statements

While Loop
while *expression*:
    statements

For Loop
for *var* in *collection*:
    statements

Counting For Loop
for i in range(*start*, *end* [, *step*]):
    statements
(*start* is included; *end* is not)

Arithmetic Operators

x + y	add	x - y	subtract
x * y	multiply	x / y	divide
x % y	modulus	x ** y	x^y

Assignment shortcuts: x *op*= y
Example: x += 1 increments x

Comparison Operators

x<y	Less	x<=y	Less or eq
x>y	Greater	x>=y	Greater or eq
x==y	Equal	x!=y	Not equal

Boolean Operators

not x	x and y	x or y

Exception Handling

try:
    statements
except [*exception type* [as *var*]]:
    statements
finally:
    statements

Conversion Functions

int(*expr*)	Converts *expr* to integer
float(*expr*)	Converts *expr* to float
str(*expr*)	Converts *expr* to string
chr(*num*)	ASCII char *num*

String / List / Tuple Operations

len(*s*)	length of *s*
s[*i*]	*i*th item in *s* (0-based)
s[*start*:*end*]	slice of *s* from *start* (included) to *end* (excluded)
x in *s*	**True** if *x* is contained in *s*
x not in *s*	**True** if *x* is not contained in *s*
s + *t*	the concatenation of *s* with *t*
s * *n*	*n* copies of *s* concatenated
sorted(*s*)	a sorted copy of *s*
s.index(*item*)	position in *s* of *item*

More String Operations

s.lower()	lowercase copy of *s*
s.replace(*old*, *new*)	copy of *s* with *old* replaced with *new*
s.split(*delim*)	list of substrings delimited by *delim*
s.strip()	copy of *s* with whitespace trimmed
s.upper()	uppercase copy of *s*

See also http://docs.python.org/library/stdtypes.html#string-methods

Mutating List Operations

del *lst*[*i*]	Deletes *i*th item from *lst*
lst.append(*e*)	Appends *e* to *lst*
lst.insert(*i*, *e*)	Inserts *e* before *i*th item in *lst*
lst.sort()	Sorts *lst*

See also http://docs.python.org/library/stdtypes.html#typesseq-mutable

Dictionary Operations

len(*d*)	Number of items in *d*
del *d*[*key*]	Removes *key* from *d*
key in *d*	True if *d* contains *key*
d.keys()	Returns a list of keys in *d*

See also http://docs.python.org/library/stdtypes.html#mapping-types-dict

Function Definitions

def *name*(*arg1*, *arg2*, ...):
    statements
    return *expr*

String Formatting

```
name = "Abe"
surname = "Jones"
print(f"Hello, {name} {surname}")
```
Hello, Abe Jones

```
money = 253422.3
print(f"You owe me ${money:,.2f}")
```
You owe me $253,422.30

```
from time import strftime
ts = strftime("%Y-%m-%d %H:%M:%S")
print(ts)
```
2022-07-20 09:56:47

See also http://docs.python.org/library/string.html#format-specification-mini-language

Useful Functions

exit(*code*)	Terminate program with exit *code*
input("*prompt*")	Print *prompt* and readline() from stdin

Authors

Tino Miegel (t.miegel@conterra.de)

Dennis Wilhelm (d.wilhelm@conterra.de)

https://conterra.de

FME UC 2022

THE PEAK OF DATA INTEGRATION 2022 UC

https://fmeuc.com

Other References

https://www.cheatography.com/davechild/cheat-sheets/python/
More Python Cheatsheet Goodness

https://community.safe.com/
FME Community

https://community.safe.com/s/article/python-and-fme-basics
Tutorial: Python and FME Basics

https://docs.safe.com/fme/html/fmepython/index.html
Official FME Python API Documentation

Code Snippets

Loop Over Sequence
```
for index, value in enumerate(seq):
    print("{} : {}".format(index, value))
```
Loop Over Dictionary
```
for key in sorted(dict):
    print(dict[key])
```
Read a File
```
with open("filename", "r") as f:
    for line in f:
        # Strip newline
        line = line.rstrip("\n")
        print(line)
```

FME Execution Order

1. Python Scripted Parameter (in order)
2. FME Python Startup script
3. Workspace with PythonCaller and PythonCreator
4. FME Python Shutdown script

FMELogfile()

logMessageString(message, severity)	Write a Message to the log, severity is optional
FME_INFORM	The default log level
FME_WARN	Warning messages
FME_ERROR	Error messages
FME_FATAL	Fatal error messages

FMEFeature()

getAttribute(attributename)	Query attribute value
setAttribute(attributeName, attributeValue)	Set an attribute value
getGeometry()	Query geometry
removeAttribute()	Remove attribute
getAllAttributes()	Get a list with all attributes

FME Python Startup Script

fme.logFileName	The name of the log file used for this translation.
fme.macroValues	A Python dictionary, indexed by macro name, which holds the value of each macro known within the workspace or mapping file at the end of parsing.

FME Python Shutdown Script

Note: To import the 'fme' Python module, execute the statement import fme.

fme.failureMessage	The failure message if the translation failed, blank if the translation succeeded.
fme.featuresRead	A Python dictionary, indexed by feature type, which holds the number of features read for that feature type.
fme.featuresWritten	A Python dictionary, indexed by feature type, which holds the number of features written for that feature type.
fme.logFileName	The name of the log file used for this translation.
fme.status	False if the translation failed and True if it was successful
fme.totalFeaturesRead	The total number of features read
fme.totalFeaturesWritten	The total number of features written

FME Web Services API

Note: Use with 'import fmewebservices'

FMENamedConnectionManager()	Get the instance of the manager object.
FMENamedConnectionManager.getNamedConnection('name')	Obtain a named connection which does not block access on the DB to that given named connection. Will return None if the connection name doesn't exist.
FMEBasicConnection.getUsername()	Retrieves the user name associated with this connection.
FMEBasicConnection.getPassword()	Retrieves the user password associated with this connection.
FMEBasicConnection.getType()	Retrieves the type of the named connection.

Authors

Tino Miegel (t.miegel@conterra.de)

Dennis Wilhelm (d.wilhelm@conterra.de)

https://conterra.de

FME UC 2022