Here's about the simplest Python UDF you can write:

**from** pig\_util **import** outputSchema

**@outputSchema('word:chararray')**

**def** **hi\_world**():

**return** "hello world"

from \_\_future\_\_ import division

@outputSchema("num1:double")

def get\_gpa(credit, grade):

num,denom = 0,0 temp1, temp2 = [], [] for t in credit: temp1.append(t[0]) denom += t[0] for u in grade: temp2.append(u[0])

**Specifying the UDF output schema**

Now a UDF has input and output. This little section is all about the outputs. Here we'll go over the different ways you can specify the output format of a Python UDF through use of the outputSchema decorator. We have a few options, here they are:

*# our original udf*

*# it returns a single chararray (that's PigLatin for String)*

**@outputSchema('word:chararray')**

**def** **hi\_world**():

**return** "hello world"

*# this one returns a Python tuple. Pig recognises the first element*

*# of the tuple as a chararray like before, and the next one as a*

*# long (a kind of integer)*

**@outputSchema("word:chararray,number:long")**

**def** **hi\_everyone**():

**return** "hi there", 15

*#we can use outputSchema to define nested schemas too, here is a bag of tuples*

**@outputSchema('some\_bag:bag{t:(field\_1:chararray, field\_2:int)}')**

**def** **bag\_udf**():

**return** [

('hi',1000),

('there',2000),

('bill',0)

]

*#and here is a map*

**@outputSchema('something\_nice:map[]')**

**def** **my\_map\_maker**():

**return** {"a":"b", "c":"d", "e","f"}

REGISTER 'myudf.py' using jython as myudfs

users = LOAD 'user\_data' AS (firstname: chararray, lastname:chararray,some\_integer:int);

winning\_users = FOREACH users GENERATE myudfs.deal\_with\_a\_string(firstname);

full\_names = FOREACH users GENERATE myudfs.deal\_with\_two\_strings(firstname,lastname);

squared\_integers = FOREACH users GENERATE myudfs.square\_a\_number(some\_integer);

users\_by\_number = GROUP users by some\_integer;

indexed\_users\_by\_number = FOREACH users\_by\_number GENERATE group,myudfs.now\_for\_a\_bag(users);

<https://svn.apache.org/repos/asf/pig/trunk/src/python/streaming/pig_util.py>