ИНЖЕНЕР ФАКУЛЬТЕТІ
«АҚПАРАТТЫҚ ЖҮЙЕЛЕР» ЖӘНЕ «КОМПЬЮТЕРЛІК ИНЖЕНЕРИЯ» КАФЕДРАЛАРЫ

“БАҒДАРАЛАМАЛЫҚ ҚАМТАМАСЫЗ ЕТУ ЖӘНЕ КОМПЬЮТЕРЛІК МОДЕЛЬДЕУ”

УДК: 004.420

COMPARISONS BETWEEN RUBY AND PYTHON

Asem Seysenbay
MSc Student, Suleyman Demirel University, Almaty
asemesboken@yahoo.com
Zhanat Kopbayeva, MSc Student, Suleyman Demirel University, Almaty
kpb_janat@mail.ru

Abstract

Nowadays web development programming is very popular. Using object-oriented programming (OOP) languages in it made it fun to design overall architectures, functionalities and ease of usability. As the scripting languages did not lose their popularity in this process, on their own field they also were making progress for making web development interesting for a programmer. Also scripting languages’ popularity is stable because of their compatibility and ease of use with the other languages. Considering these in this paper the comparison of basic items is made, that exist in Ruby and Python. Python itself was already popular, but coming of Ruby into the world of developers made them to begin many discussions. This work will give you basic view to make yourself a conclusion which one to choose and here it’s assumed that you already know one of the OOP languages or at least have some basic understanding of it.

Keywords: programming languages, script, Python, Ruby

В этой работе было приведено сравнение двух скриптовых языков Руби и Питон. Этот предварительный просмотр позволит для дальнейшего прогресса в ознакомлении с этими двумя языками.

Executing statements

Python statement example in Python as shown in table 1 and its output in figure 1.1[1]:

Table 1 Statement in Python

Code line
from time import time t1=time() print("Hello World!") print time()-t1

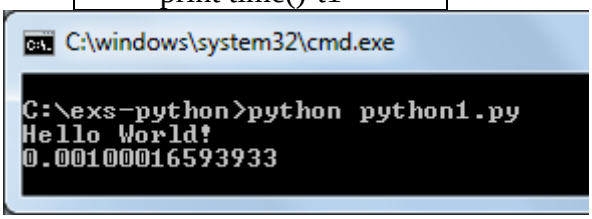

Figure 1.1 Output in Python

The same example in Ruby:

Table 2 Statement in Ruby

Code line

```
        t1=Time.now
        puts "Hello
 World!"
        puts Time.now-
 t1
```

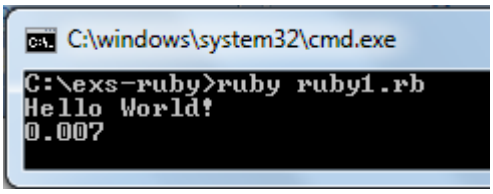

Figure 1.2. Output in Ruby

Comparison:

Code lines: Python: 4 lines, Ruby: 3 lines

Time for executing: Python time is less than Ruby time

Note: Actually in Python it was import statement taking one more line. And because of this kind of statement not required in Ruby rubiests claim it to be totally object oriented.

Method / Function defining

Method and function example in Python as shown in table 2 and its output in figure 2.1[2]:

Table 3 Method and function in Python

Code line
def sum(n1,n2): return n1+n2 from time import time t1=time() print(sum(3,4)) print time()-t1 t1=time() print(sum("Hello ","World!")) print time()-t1

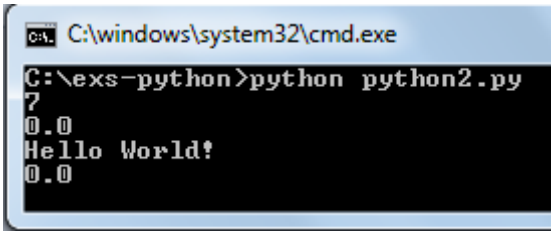

Figure 2.1. Output in Python

Ruby defining:

Table 4 Method and function in Ruby

Code line
def sum(n1,n2) n1+n2 end t1=Time.now puts sum(3,4) puts Time.now-t1 t1=Time.now puts sum("Hello ","World!")

```
puts Time.now-t1
```

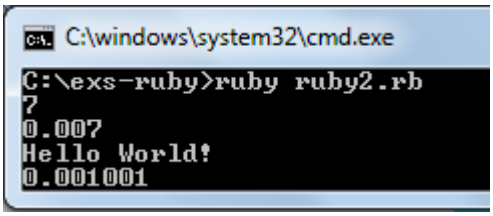

Figure 2.2. Output in Ruby

Comparison:

Code lines for defining method: Python: 2 lines, Ruby: 3 lines

Time: Python time is less than Ruby time

Note: Ruby needs additional end statement to finish defining and Python needs return keyword to be processed like a function not procedure. Without return or return without arguments will give us None result in Python. And in Ruby every called method returns a value (although no rule says you have to use that value).The value of a method is the value of the last statement executed during the method's execution. Also in Ruby because it looks at everything as objects nil argument calling will give undefined method error for NilClass as in Python calling with None argument.

IF Statement and Standard input, gets

Python IF statement:

Table 5 if statement in Python

Code line

from time import time
x=int(input("Please enter an integer:"))
t1=time()
if x<0:
  x=0
  print('Negative changed to zero')
elif x==0:
  print('Zero')
elif x==1:
  print('One')
else:
  print('More')
print time()-t1

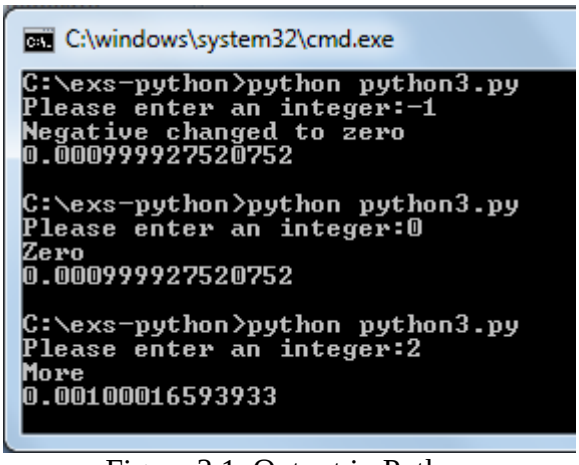

Figure 3.1. Output in Python

Ruby IF statement:

Table 6 If statement in Ruby

Code line
print "Please enter an integer:" x=Integer(gets) t1=Time.now if x<0 x=0 print(“Negative changed to zero\n”) elsif x==0 print “Zero\n” elsif x==1 puts 'One' else puts 'More' end puts Time.now-t1

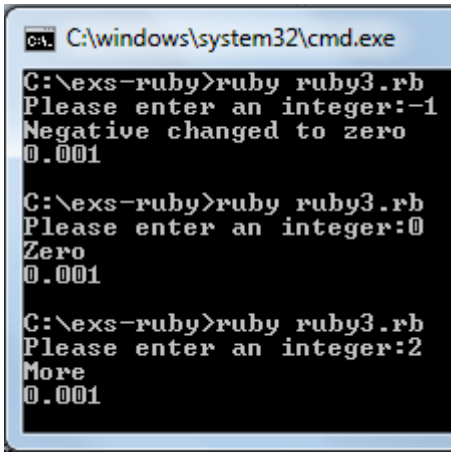

Figure 3.2. Output in Ruby

Comparison

Code lines for input, gets: Python: 1 line, Ruby: 2 lines

Code lines for IF statement: Python: 9 lines, Ruby: 10 lines

Time: Almost same

Note: In Python after the beginning of any statement we need to put colon. In Ruby there is no colon to be put and the brackets of any methods are optional if after method call there are no operations on return values of methods. In Ruby the brackets of print statement is must with one argument, with more arguments it must be omitted.

Conclusion

In this paper basic comparison between two scripting languages Ruby [1] and Python [2] was made. This preview will make it possible for your further progress in introduction with these two languages.

References

[1]. Python Tutorial, Release 3.2.3., Guido van Rossum, Fred L.Drake, Jr., September 2012.

[2]. Programming Ruby, The Pragmatic Programmers’ Guide, Dave Thomas with Chad Fowler and Andy Hunt, Second Edition.

УДК: 004.325