Generisk programmering og biblioteksudvikling - Week assignment 2 (Python scripting)

Lasse Jon Fuglsang Pedersen Mads Ruben Burgdorff Kristensen
Anders Sabinsky Tøgern

Department of Computing, University of Copenhagen
Universitetsparken 1, DK-2100 Copenhagen East, Denmark

Abstract. The Python scripting language is well known for its extendability and embedded application. Numerous larger open-source software projects use Python in one form or another to enhance productivity or to automate tedious tasks. It is also used sometimes as the main component, drawing on an extensive library of usermade extensions.

1. Introduction

Python is a cross platform scripting language invented by Guido van Rossum in 1991 with the philosophy that "'programmer effort should be over computer effort"' and the code should rather be readable than fast.

2. The Python language

The language is strongly typed and there is no need to specify the type of variables. The type of a variable will be assigned by the first use of that variable. The language supports classes. Classes' member functions are all static functions whose first parameter is always assumed to be the class instance, unlike `C` where the class instance is passed to member functions by the hidden `this` pointer.

Much like `C` the Python language employs use of scopes for structuring code in loops and blocks, but unlike C where scopes are declared explicitly with curly-brackets, scopes in the Python language are delimited by levels of indention and are as such implicit in the code. A nice effect of scoping by indentation is that the Python language enforces programmers to indent in similar fashion.

While `C` only support arrays of static length Python support lists which is an equivalent to `C`'s arrays except that the lists in Python are dynamic through manipulating methods like `extend` and `append`. Thus, the programmer does not have to know beforehand how many elements must fit into a list; he can simply declare it, and then append more data as more data arrives. This is probably rather inefficient in terms of memory usage and running time, but on the other hand it is nice to have for prototyping.

Python also support a special type of lists called a `dictionary`, whose structure is similar to a hash table except there are no predefined equivalent to a hash

Written 1 May 2007.

Python	**C**
if x == 42:	if(x == 42) {
print "x = 42"	printf "x = 42";
else:	} else {
printf "x != 42"	printf "x != 42";
	}

Figure 1. Example of the difference between how scopes are written in `C` and Python. Notice that Python have no characters delimiting the scopes of the `if` and `else`. Also notice how both scopes begin with a colon (:).

function. It is possible to think of the dictionary structure as an array or a list where indexing is done by key instead of by index.

Python has a comprehensive collection of standard libraries that support everything from file-access to HTTP servers. It is also possible to integrate Python scripts with libraries written in other languages, like `C`, through its `COM` integration which makes Python very agile. Along with this agility it is also possible to embed Python in `C`/`C++` applications to make it possible for the user to customize the applications by running custom scripts.

If the standard library lack some needed functionality Python has a very large community in which it might be possible to find an implementation for that functionality.

As a special feature Python supports arbitrarily large integers which sets no limitations for computations. Although this only applies to integers and not fractional numbers, it potentially enhances the usability of Python for scientific applications: One might implement very large or precise fractional numbers within an application in Python, given the existing support for arbitrarily large integers.

The Python language is a scripting language and programs written in Python can be run using a Python interpreter. Several interpreters, written in different languages like `C` (CPython), `Java` (Jython), `C#` (IronPython) and Python itself (PyPy), exist. The interpreters feature advanced memory management which makes the dynamic types, like lists, easy to manipulate through native methods. The memory manager also support garbage collection. This is a requirement of the language and so you cannot have a correct interpreter of the language without supporting these features.

For further information about Python, see [1], [2], [3].

3. Common uses

Python can be used for many things, given its extendability, and the fact that it can be embedded as an interpreter in an existing C/C++ application. Python is mostly used to do routine work like unit testing, maintenance and batch jobs, not to mention prototyping and doing quick-and-dirty implementations.

Examples of uses of Python scripts is in the benchmarking tool Benz where the Python scripts integrates with GNUPlot as an external library. Python is also

extensively used in the MiG (Minimum Intrusion Grid) project at DIKU where it, among other things, is running a web server.

An example of how to use a Python program to create a nightly build of an application could be a script that performed the following actions:

1. Check that all dependencies are exists and are valid
2. Create the nightly build
3. Check that the nightly build is valid
4. Publish the nightly build

4. Conclusion

Because of the prioritisation that with Python "'programmer effort should be over computer effort"' the language is not the best choice for real-time critical applications. On the other hand it is potentially much faster to prototype in Python than in C/C++ especially because of Python's language features, dynamic list allocation and memory management. Large integers is also a huge bonus.

References

[1] G. van Rossum, *Extending and Embedding the Python Interpreter*, `http://docs.python.org/ext/ext.html` (2006).

[2] G. van Rossum, *Python Library Reference*, `http://docs.python.org/lib/lib.html` (2006).

[3] G. van Rossum, *Python Reference Manual*, `http://docs.python.org/ref/ref.html` (2006).