

## Romani ite domum

2006-04-25 20:09:27

[Dies ist der Satz](#), den Brian in dem Kultfilm "Das Leben des Brain" von [Monty Python](#) an die Wand der römischen Stadt schreiben muss. Doch wer hätte gedacht, dass es tatsächlich, auf  $\frac{1}{2}$ er vielleicht [Albert Uderzo](#), dem Vater von Asterix und dem Papst, Lateinophile gibt, die daraus sogar eine Programmiersprache machen?

[Perligata](#) ist ein Versuch eine auf Perl basierende Sprache mit lateinischer Syntax und Semantik zu entwickeln. Da heißt es etwa:

In Latin, the same instruction would be (loosely) rendered:

Bonos tum malos tum Foedum pugnatoribus da.

Here the direct objects are bonos ("the good (people), accusative plural), malos ("the bad (people), accusative plural) and Foedum ("Mr Ugly, accusative singular). The indirect object is pugnatoribus ("fighters, dative plural) and the verb is da ("give, present imperative). The conjunction tum means "and then, and conveys the significance of the order of the direct objects.

Unlike the English "-s ending, the various Latin suffixes (-os, -um, -ibus) specify both the number and the role (or "case) of the nouns they inflect. This means that the positions of the various objects, and indeed of the verb itself, do not matter. The same sentence could equally well be written:

Pugnatoribus da bonos tum malos tum Foedum.

or

Da bonos tum malos tum Foedum pugnatoribus.

Semantically, all of these variants (and any other permutations of the verb and its objects) are equivalent to the same target/data model:

```
_assign( -target=>@gunslingers, -data=>[@good, @bad, $Ugly] );
```

and hence are equivalent to the standard Perl:

```
@gunslingers = ( @good, @bad, $Ugly );
```

Thus it is possible to write Perl programs in Latin.

Das ist zwar  $1337^3$ , aber da beleibe ich lieber bei der guten alten Perl Syntax, auch wenn diese manchmal sehr kryptisch ist.