Eonums

Convert between integer numbers and Esperanto strings.

Author: Dinu Gherman <gherman@darwin.in-berlin.de>

Homepage: http://www.dinu-gherman.net/

Version: Version 0.9.0

Date: 2008-09-15

Copyright: GNU Public Licence v3 (GPLv3)

About

Eonums is a simple module providing conversion between normal integer numbers and the corresponding textual expression in the Esperano language. It was mainly developped in order to explore the regularity of Esperanto expressions for big integer numbers.

Names for 10**k (k = 6, 9, 12, ...) like "miliono" (10**6) or "miliardo" (10**9) are chosen from the so-called "Longa Skalo" as described on this page about big numbers (in Esperanto).

The integer numbers *eonums* can convert to or from such Esperanto expressions can be arbitrarily large, but are limited in practice by the largest number for which there is a name in Esperanto (on the "Longa Skalo")", which is, on the previous page, 10**63 (dekiliardo). Hence, the largest integer you can handle with this module is 10**66 - 1. (This module makes no attempt to extend the Esperanto naming rules by introducing names like "undekiliono", "undekiliardo", "dudekiliono" etc.)

This module can be fully translated automatically to Python 3.0 using its migration tool named `2to3`.

Features

- convert Python integers to Esperanto integer strings (Unicode)
- convert Esperanto integer strings (Unicode) to Python integers
- validate Esperanto integer strings (Unicode)
- handle integers from 0 to 10**66 - 1
- provide conversion functions and command-line scripts
- provide a Unittest test suite
- can be automatically migrated to Python 3.0 using `2to3`

Examples

You can use *eonums* as a Python module e.g. like in the following interactive Python session:

```
>>> from eonums import int2eo, eo2int, validate_eo
>>>
>>> int2eo(22334455)
u'dudek du milionoj tricent tridek kvar mil kvarcent kvindek kvin'
>>>
>>> eo2int(u"cent dudek tri")
123
>>> validate_eo(u"dudek cent tri")
False
```

In addition there are two (very simple) conversion scripts, `int2eo` and `eo2int`, which can be used from the system command-line like this:

```
$ int2eo 22334455
dudek du milionoj tricent tridek kvar mil kvarcent kvindek kvin
$
$ eo2int "cent dudek tri"
123
```

Installation

There are two ways to install *eonums*, depending on whether you have the *easy_install* command available on your system or not.

1. Using *easy_install*

With the *easy_install* command on your system and a working internet connection you can install *eonums* with only one command in a terminal:

```
$ easy_install eonums
```

If the *easy_install* command is not available to you and you want to install it before installing *eonums*, you might want to go to the Easy Install homepage and follow the instructions there.

2. Manual installation

Alternatively, you can install the *eonums* tarball after downloading the file `eonums-0.9.0.tar.gz` and decompressing it with the following command:

```
$ tar xfz eonums-0.9.0.tar.gz
```

Then change into the newly created directory `eonums` and install *eonums* by running the following command:

```
$ python setup.py install
```

This will install a Python module named *eonums* in the `site-packages` subfolder of your Python interpreter and two scripts tool named `int2eo` and `eo2int` in your `bin` directory, usually in `/usr/local/bin`.

Testing

The *eonums* module contains a Unittest test suite which can be run by simply executing the module itself like the following on the system command-line:

```
$ python eonums.py
..........
----------------------------------------------------------------------
Ran 11 tests in 17.477s

OK
```

It takes a short while because it contains a roundtrip test in which the first 100,000 integers are converted to Esperanto strings and back to normal integers.

Bug reports

Please report bugs and patches to Dinu Gherman <gherman@darwin.in-berlin.de>. Don't forget to include information about the operating system and Python versions being used.