

Java String.hashCode python implementation

Author : Anthony Lombard
Contact : anthony@_nospam_pimentech.fr
Revision : 1.9
Date : 2010-04-08
Copyright : 2010 PimenTech SARL
Tags : python hashCode java string

The problem

Recently we had to reproduce some java code in python and we faced this :

```
int i = someString.hashCode()
```

As we needed the real computed value and not the hash mecanism behind it (which python has as well), we had to reimplement this String.hashCode() in python.

Source

Get it here : <http://ftp.pimentech.net/src/libcommonPython/src/python/libcommon/javastringhashCode.py>

```
# -*- coding:utf-8 -*-

def convert_n_bytes(n, b):
    bits = b*8
    return (n + 2**(bits-1)) % 2**bits - 2**(bits-1)

def convert_4_bytes(n):
    return convert_n_bytes(n, 4)

def java_string_hashcode(s):
    h = 0
    n = len(s)

    for i, c in enumerate(s):
        h = h + ord(c)*31**(n-1-i)

    return convert_4_bytes(h)
```

Usage Example

```
>>> java_string_hashcode('Big Bisou')
477474450L
```

It is worth noticing that we had to reimplement 'dumb' int calculus in python, as this smart kid automatically switches from int to long. So we had to reimplement int overflow too. Thanks to this [guy](#) it was pretty simple :

```
>>> convert_4_bytes(2**31-1)
2147483647L
```

```
>>> convert_4_bytes(2**31)
-2147483648L
```