

Re: a print bug?

Re: a print bug?

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-08/msg01088.html>

- *From:* Roel Schroeven <rschroev_nospam_ml@xxxxxxxxxxx>
 - *Date:* Fri, 28 Jul 2006 08:19:52 GMT
-

Summercoolness@xxxxxxxxxx schreef:

Steve Holden wrote:

You obviously haven't yet passed your floating-point number proficiency test yet. Please restrict yourself to integers until you understand the difficulties that inaccuracies in floating-point can create ;-)

hm, actually, i understand the limitation of floating point.
but my real concern is, how come "print" favors one version over the other...
the first version will favor the correct rounding, while the second doesn't favor it. to me, this is biased computing.... i will feel happier if the behavior is consistent. (such as the first line printing out as 1.23449999999999) . if most people favor the behavior of line 1, that is, silently rounding off to about the 11th decimal places, then why doesn't printing with formatting also use that same behavior, which is rounding off to the 11th places first, and then round off to whatever the user wants.

```
print 1.2345
```

```
1.2345
```

```
print "%10.3f" % 1.2345
```

```
1.234
```

but then again, i just realize even if i do a

Re: a print bug?

```
round(1.23449999999999999999, 6)
```

```
1.2344999999999999
```

so even if you round it off, it is still represented the same way....
still, how "print" handles 1.2345 and treating it and printing it as
1.2345, i wonder why we don't make the "print with formatting" the same
behavior, treating it as 1.2345 first and round it off to 1.235

You do realize that your formatting uses less decimal places than the print statement, do you?

```
>>> print 1.2345
1.2345
>>> print "%0.3f" % 1.2345
1.234
>>> print "%0.4f" % 1.2345
1.2345
```

If you use 4 decimals after the decimal sign, you get the same result as with a plain print statement (at least in this case). That means that print is not treating it as 1.2345 first and then rounding it off to 1.235; it is just using the actual internal representation and rounding it off to 1.2345. Which IMO is the only sensible thing one can do.

—

If I have been able to see further, it was only because I stood
on the shoulders of giants. — Isaac Newton

Roel Schroeven

.