

printing plurals (Python recipe) by Oliver Steele

ActiveState Code (<http://code.activestate.com/recipes/81058/>)



Sometimes it's useful to print singular or plural versions of a word, depending on the value of the number of items that the word is describing: for example, "0 files", "1 file", "2 files" (note that "file" is used in the case of 1; "files" otherwise). This idiom does this inline, as a single expression.

```
1 >>> n = 0
2 >>> print "%d item%s" % (n, "s"[n==1:])
3 0 items
4 >>> n = 1
5 >>> print "%d item%s" % (n, "s"[n==1:])
6 1 item
7 >>> n = 2
8 >>> print "%d item%s" % (n, "s"[n==1:])
9 2 items
10
11 # If you might want to print negative items, add abs to the test:
12 >>> n = -1
13 >>> print "%d item%s" % (n, "s"[abs(n)==1:])
14 2 items
15
16 # If a word has irregular plural morphology, use a list:
17 >>> n=1
18 >>> print "%d %s" % (n, ['abacus', 'abaci'][n!=1])
19 1 abacus
20 >>> n=2
21 >>> print "%d %s" % (n, ['abacus', 'abaci'][n!=1])
22 2 abaci
```

Python, 22 lines

The idiom is "%d item%s" % (n, "s"[n==1:]). "s"[n==1:] evaluates to the empty string when n==1, and to "s" otherwise.

The code in the example could be wrapped into a utility function (e.g. 'printItems') or turned into a function to pluralize a word, but usually you want to do this once or twice in a program that doesn't otherwise do any text processing, so it's nice to avoid the conceptual overhead of adding a function, especially in problem domain that's unrelated to the rest of your program. Of course, it's not very readable unless you've seen it before. That's why it's an idiom.

(Common Lisp provides this functionality with the ~P directive: (format nil "~D item~:P" n) => "3 items")

Tags: [text](#)

1 comment



Robin Parmar 14 years, 10 months ago

function version. Cute! Here is a useful function version.

```
def Plural(num=0, text=''):
    return "%d %s%s" % (num, text, "s"[num==1:])
```

