目次へ

リストをprintするとバケてやだ、というときのワンポイント

ひとつ、説明しそびれていたワザがあったので、これだけのためにひとつ記事にしてしまおう。

リストをそのままprintすると、おかしな表示になってイヤだなあという話。

```
>>> a = ['金色', '銀色', '青息吐息']
>>> print a
['¥x8b¥xe0¥x90F', '¥x8b¥xe2¥x90F', '¥x90¥xc2¥x91¥xa7¥x93f¥x91¥xa7']
```

前にもいったように、要素を直接指定してprintすればいいんだけど...

```
>>> print a[0], a[1], a[2]
金色 銀色 青息吐息
```

ちょっとした途中確認したいだけなのにこんなの打ち込むのも面倒だな、と。

こういうときに、リストの中身を全部一つの文字列につなげてしまう方法があって、こいつを使えば確認用のprintがちょっと手軽になるのです。

ちょっと変な書き方になるので、タイミングを逸し気味でした。今こそ、joinメソッドをご紹介しましょう。書き方は下のように！

```
>>> print "".join(a)
金色銀色青息吐息
```

ほう...

何？ この変な書き方は。

もうひとつ、例。

```
>>> print "--".join(a)
金色--銀色--青息吐息
```

えーと、最初に書いた文字列を「つなぎ」にして、次に指定するリストの中身をぜんぶ繋げてひとつの文字列にしてくれるのが、joinメソッドの機能です。splitの逆って感じですね。

空文字列（""）を「つなぎ」にすると、当然、隙間なくつめられた文字列が得られます。

リストの中身は必ずすべて文字列であることが、joinが使える条件です。ひとつでも数が混じってたらエラーを起こしますので注意。

で、なんでこんな書き方なの？ という件。直観的にわかりやすいのは、たとえば

```
>>> print a.join("--")
```

とかそんな感じだろうに。うん、そうですよねえ。でもこれは動作しません。

そのうち、「joinは文字列固有のメソッドなのでこんな書き方なんですわ」という説明を受けて、ああそんなものかもね、と理解できるときも来るかもしれません。でも今のところは、リストをひとつの文字列につなげるときは、こんな感じのちょっとヘンな書き方をするんだと覚えちゃってください。

ワンポイントアドバイスってことで、こんな記事を挟んでみましたよ、っと。