

## 「稀にある質問」の抄訳

The Python IAQ: Infrequently Answered Questions の抄訳です。

出展は [お勉強スレ Part2 \(#80-84\)](#) です。

Q: 多態ってすばらしいね。どんな型のリストもソートできるんでしょ？

それはちがいます。たとえばこれを見てください：

```
>>> x = [1, 1j]
>>> x.sort()
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in ?
    x.sort()
TypeError: cannot compare complex numbers using <, <=, >, >=
```

(1j という数値は -1 の平方根) ここでの問題は、sort メソッドが (現在の実装では) `__lt__` メソッドを使って比較をおこなうため、複素数との比較ができなくなってしまうところにあります。おもしろいことに `complex.__lt__` は数値と文字列やリスト、およびその他の複素数以外の型との比較は苦もなく行なうんですけどね。だから答えとしては、`__lt__` メソッド (実装が変わったときは別のメソッドになるかもしれまんが) をサポートするオブジェクトのソートならできる、ということになりますね。

質問の最初の「多態ってすばらしい」という部分についてはわたしも同意します。でも Python ではときどきそれが難しい場合もあるんですよ。なぜなら多くの Python の型 (シーケンスや数値など) はきちんと形式的に定義されていないから。

Q: Python で `++x` とか `x++` ってできる？

定義通りにいえば、イエスであり、ノーです。でも実用的な用途としては、できません。どういふことでしょうか？

- ・ できる: `++x` はちゃんと Python してます。でもそれはあなたが C++ や Java で望んでいたようなものではありません。ここでの `+` は単項演算子であり、よって `++x` は `+(+(x))` と解釈されます。この答えは (少くとも数値に関しては) ただの `x` になります。
- ・ できない: `x++` 自身は合法的な式ではありません。ただし文脈によっては正しいときもあります。たとえば `x++ - y` は `x++ + -(y)` と解釈されるので、これは数値に関しては `x - y` と等価です。なので、もちろんクラスによっては `++x` が意味を (限られた範囲でですが) 持つようなものを作ることができます。たとえば数値を保持していて、単項演算子 `+` がそれを 0.5 だけ増やすようなものです (あるいはこのアルゴリズムをランダムにしたけりゃ、0.5 の確率で 1 だけ増やすことにしてもいいです)、が...
- ・ できない: アホらしいから。 `x += 1` にしときなさい。Python 2.0 からこれはできるようになりました。

より本質的な質問としては、なぜ Python では `x++` はできないのでしょうか？わたしはこれは、Python が式の中での代入を許していない理由と同じもののように思えます。つまり Python は文と式を明確に分離させたいのです。もしあなたがこれらは区別されるべきであると考えたら、`++` を禁止するのはたぶんベストな選択でしょう。いっぽう関数型プログラミングの唱道者たちは、文は式であるべきだと主張しています。わたしは同僚の Dane, Bjarne Stroustrup とともにこち

らの立場です。彼は著書 The Design and Evolution of C++ の中でこう述べています。「もし自分がゼロから新しい言語を作るなら、わたしは Algol68 がたどった道をならって、すべての文と宣言が値をもった式になるようにするだろう」