佐藤 聖

4-1 Pythonの式

リスト1 Pythonで使える演算子

```
+   -   *   **   /   //   %   @   <<   >>   &   |   ^   ~   <   >   <=   >=   ==   !=
```

リスト2 Pythonで区切り文字として使われる記号

```
(   )   [   ]   {   }   ,   :   .   ;   @   =   ->   +=   -=
*=   /=   //=   %=   @=   &=   |=   ^=   >>=   <<=   **=
```

Python言語の式について知らなければならないことは，次のように多くあります．

算術変換，アトム，原子的要素，プライマリ，Await式，べき乗演算，単項算術演算とビット単位演算，二項算術演算，シフト演算，ビット単位演算の二項演算，比較，ブール演算，条件式，ラムダ，式のリスト，評価順序，演算子の優先順位

などです．しかしプログラミング入門者が，これらをすぐに利用することは少ないと思います．Pythonに慣れてくるとライブラリやパッケージを利用したソースコードを記述することが多くなると思いますので，最初からこれら全てを覚える必要はありません．

本記事では，よく利用される式だけを取り上げます．また，使い方を紹介する式には利用方法以外にもソースコードの記述方法があります．将来的に必要になったところで，そうした利用法を学ぶとよいと思います．

Pythonで使われる記号

他のコンピュータ言語と同じようにPythonでも1つまたは複数の記号を組み合わせて作られる，プログラム上の意味を持つトークンと呼ばれるキーワードのようなものを使います．トークンにはいろいろな記号が使われます．

● 演算子

公式ドキュメント(1)で紹介されている演算子を**リスト1**に示します．利用方法は後ほど紹介します．

● 区切り文字としての機能を持つデリミタ

リスト2に，文法上のデリミタ(区切り文字)として働くトークンを示します．

ピリオド(.)は浮動小数や虚数リテラル中にも置けます．

リスト中の累算代入演算子は字句的には，デリミタとしての振る舞いだけでなく，演算子としても働きます．詳しくは後ほど紹介します．

● 特殊な意味を持つ記号

以下の記号は印字可能なASCII文字で，他の記号と組み合わせることで特殊な意味を持つトークンとなったり，字句解析器にとって重要な意味を持ったりします．

`'  "  #  \`

▶シングル・クォートとダブル・クォート

シングル・クォート(')とダブル・クォート(")は文字列を作るときに使用します

▶コメント文

シャープ(#)は，コメント文をソースコードに記述するときに利用します．

▶エスケープ・シーケンス

バックスラッシュ(\)はエスケープ・シーケンスと

表1 演算子の優先順位

演算子	説　明
lambda	ラムダ式
if -- else	条件式
or	ブール演算OR
and	ブール演算AND
not x	ブール演算NOT
in, not in, is, is not, <, <=, >, >=, !=, ==	所属や同一性のテストを含む比較
|	ビット単位OR
^	ビット単位XOR
&	ビット単位AND
<<, >>	シフト演算
+, -	加算および減算
*, @, /, //, %	乗算，行列乗算，除算，切り捨て除算，剰余
+x, -x, ~x	正数，負数，ビット単位NOT
**	べき乗
await x	Await式
x[index], x[index:index], x(arguments...), x.attribute	添字指定，スライス操作，呼び出し，属性参照
(expressions...), [expressions...], {key: value...}, {expressions...}	結合式または括弧式，リスト表示，辞書表示，集合表示