

Python文法I



2024/03/19

- 文法I
 - 演算
 - 変数
 - データ型
- 文法II
 - コレクション
 - List
 - Tuple
 - Dict
- 文法III
 - 条件分岐、繰り返し
 - 条件分岐(if文)
 - 繰り返し(for文)
- 文法IV
 - 関数
 - モジュール

プログラミングでは、計算処理を指示する**演算子**によって新たな値に**評価**されます
特に、足し算や引き算などの**算術演算子**は、数学のように数値間に置いて計算します



(例) 算術演算子：足し算



コード例

```
In [1]: print(1 + 1)
```

```
Out [1]: 2
```

```
In [2]: print(1 - 1)
```

```
Out [2]: 0
```

算術演算子以外の演算子に比較演算子や論理演算子があります（Python文法Ⅱ）。

Pythonでは、足し算や引き算は数学の記号と共通しますが、掛け算や割り算、べき乗は特有の記号となります

算術演算子の記号

算術演算	演算子記号
足し算	+
引き算	-
掛け算	*
割り算	/
割り算（整数部）	//
割り算（余り部）	%
べき乗	**

1×2

$1 \div 2$

2^2

コード例

In [1]: `print(1 * 2)`

Out [1]: 2

In [2]: `print(1 / 2)`

Out [2]: 0.5

In [3]: `print(2 ** 2)`

Out [3]: 4



「割り算（余り部）」は、あるデータがちょうど割り切れるか調べるのに便利。例）「% 2」
▶ 偶数なら0
▶ 奇数なら1

演算子が複数ある場合、優先度の高い演算子順に評価されます
数学のように足し算より掛け算の優先度が高く、「()」により優先度が変更されます

算術演算子の優先度

優先度	演算子記号
高	**
中	*, /, %
低	+, -

(例) 足し算と掛け算の評価順

$1 + 2 * 3$  $1 + 6$  7

$$\begin{array}{c} \text{中} \\ 1 + 2 \times 3 \\ \text{低} \end{array}$$

$$\begin{array}{c} \text{中} \\ (1 + 2) \times 3 \\ \text{高} \end{array}$$

コード例

In [1]: `print(1 + 2 * 3)`

Out [1]: 7

In [2]: `print(1 * 2 + 3)`

Out [2]: 5

In [3]: `print((1 + 2) * 3)`

Out [3]: 9

優先度が同じ場合、プログラムの左から順に評価されます。

データを格納するための箱を**変数**と言い、計算結果を保持します
変数名=値とすることで、左辺の変数に右辺のデータを**代入**します



変数名がyの変数に
2という値が入っている



コード例

```
In [1]: x = 1  
        y = 2  
        print(x+y)
```

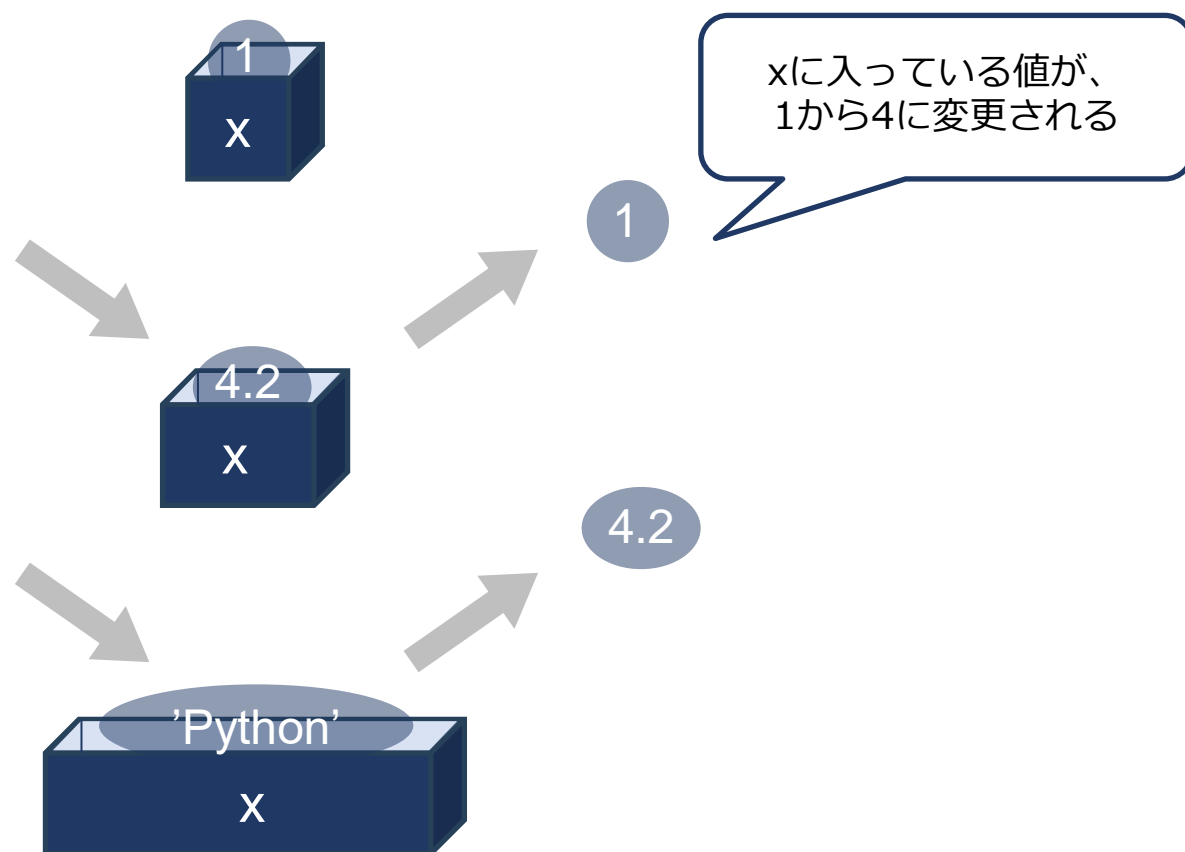
```
Out [1]: 3
```

```
In [2]: word = 'Hello'  
        print(word)
```

```
Out [2]: Hello
```

変数には様々なデータ型の値を代入できる (Python文法 I : データ型)

同じ変数に**再代入**することで、格納されている値が変更されます



コード例

```
In [1]: x = 1  
        print(x)
```

```
Out [1]: 1
```

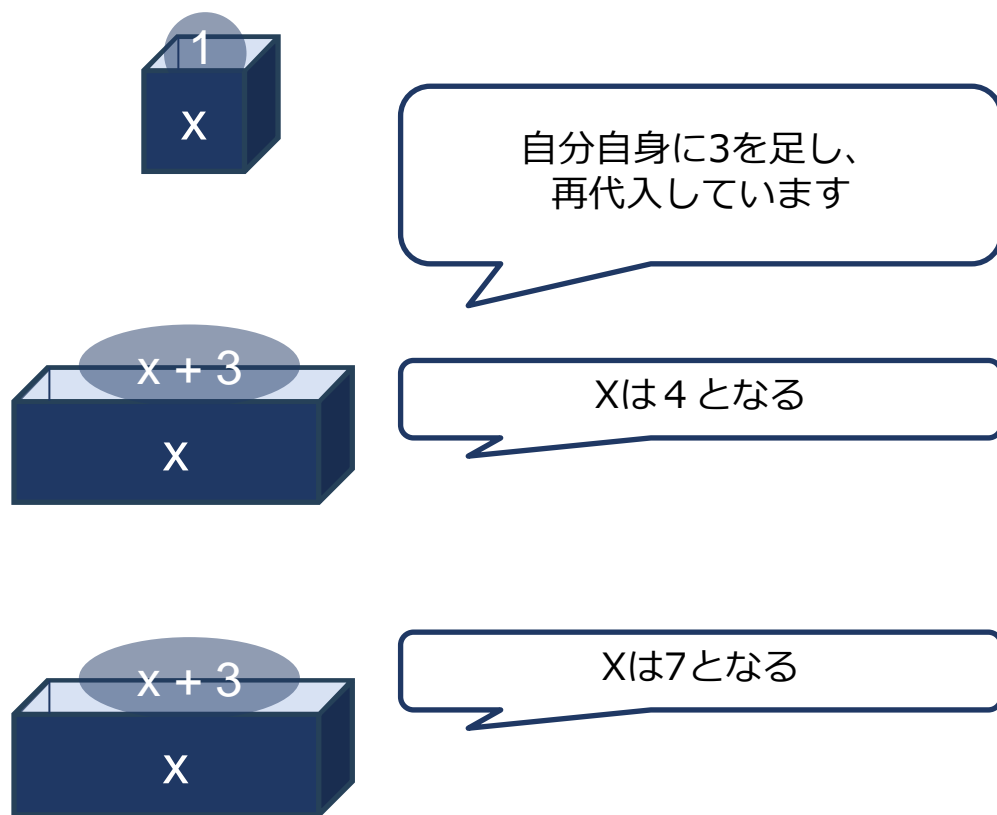
```
In [2]: x = 4.2  
        print(x)
```

```
Out [2]: 4.2
```

```
In [3]: x = 'Python'  
        print(x)
```

```
Out [3]: Python
```

変数は自分自身の値を使って、**再代入**することもできます



コード例

右辺を計算し、左辺に代入

```
In [1]: x = 1  
        x = x + 3  
        print(x)
```

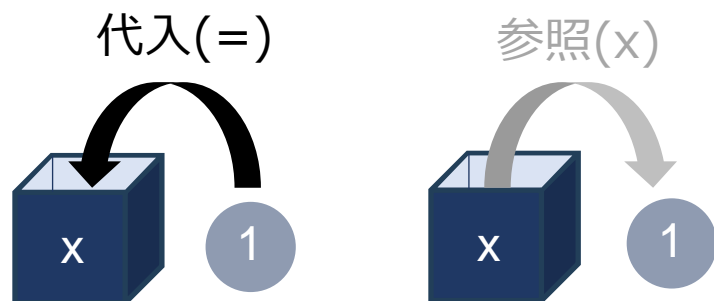
Out [1]: 4

```
In [2]: x = x + 3  
        print(x)
```

Out [2]: 7

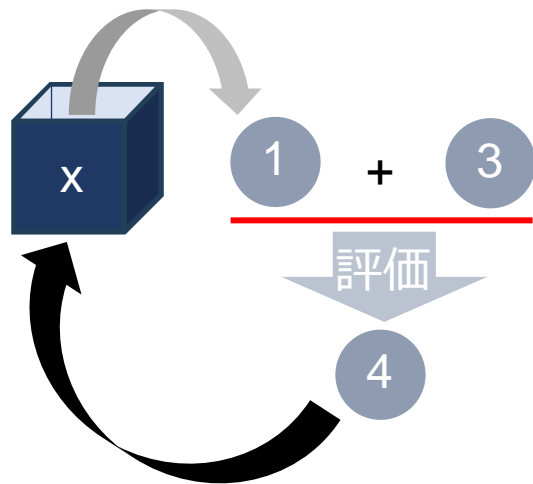
変数の代入、参照についての補足

・ 変数の代入と参照イメージ



・ 変数を用いた計算イメージ

例) $x = x + 3$



コード例

In [1]:

`x = 1`

代入

`print(x)`

参照

printで出力

Out [1]:

1

In [2]:

`x = x + 3`

参照

評価

代入

`print(x)`

Out [2]:

4

プログラムで扱う値の種類を**データ型**と言います
変数に代入した値のデータ型は、type関数で調べられます

主なデータ型

データ型	内容	例
int	整数	1
float	小数	3.14
str	文字列	'Python'
bool	真偽値	True、False

(例) string1 = '文字列1'
string2 = "文字列2"
「'」または「"""」で挟むと文字列
代入時に変数の型がstr型に

※ bool型は「はい」と「いいえ」のような2値で条件確認に利用できます (Python文法 II)

コード例

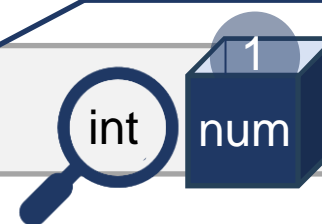
```
In [1]: num = 1  
        print(type(num))
```

```
Out [1]: int
```

```
In [2]: pi = 3.14  
        print(type(pi))
```

```
Out [2]: float
```

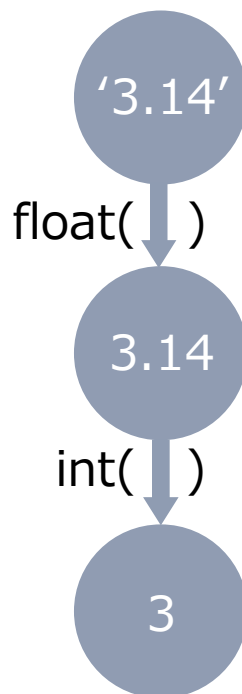
type関数：
かっこ内に変数入力
▶ データ型出力



変数に代入した値のデータ型は、int関数やstr関数などで変更できます

型変換の関数

関数	出力の型
int()	整数のint型へ (※1)
float()	小数のfloat型へ (※2)
str()	文字列のstr型へ
bool()	真偽値のbool型へ



コード例

```
In [1]: x = '3.14'
        print(type(x))
```

```
Out [1]: str
```

```
In [2]: y = float(x)
        print(type(y))
```

```
Out [2]: float
```

```
In [3]: z = int(y)
        print(z)
```

```
Out [3]: 3
```

※1 : 小数をint型に入力すると整数部のみになります。

※2 : 'abc'などの文字列をintやfloat関数に入力するとエラー。

