

2022年度・鼓山塾

K-15 デジタル社会の技術（プログラミング）

1. 逐次処理・変数・文字列

伊東栄典

九州大学情報基盤研究開発センター

ito.eisuke.523@m.kyushu-u.ac.jp

参考資料



Python ゼロからは始めるプログラミング

著者 : 三谷純

出版社 : 翔泳社

発売日 : 2021/5/24

ISBN : 9784798169460

講義用のスライドも提供

https://mitani.cs.tsukuba.ac.jp/book_support/python/

本資料も、上記のスライドを援用しています。

K-15 デジタル社会の技術（プログラミング）

1. 逐次処理・変数・文字列

1. はじめの1歩

1. はじめの1歩
2. 変数
3. 型と算術演算子
4. 文字列

1. はじめの1歩

- print関数で文字列を出力
 - 以下のように文字列を出力できる。
`print('出力する内容')`
- 文字列の前後を, 以下のどちらかで囲む
 - シングルクォーテーション ('),
 - ダブルクォーテーション (")

print 関数

```
print('Hello World!')
```

関数

引数



```
print('Hello World!')
```

Hello World!

コードセルの直下に出カ

print 関数は、**引数**で与えられたものを標準出力に**出力**する働きをする組み込み関数

※ print 関数のように、はじめから使える関数を**組み込み関数**という（自分で作った関数は**ユーザ定義関数**）

※ 標準出力はプログラム実行環境によって異なる

プログラムを書く際のルール

- 半角英数字と半角記号を使用して記述する
 - (クォーテーションで囲んだ**文字列**は例外)
 - 全角空白は見えないので注意
- 大文字と小文字は区別される
 - `print('Hello')`
 - ✗ `Print('Hello')`
- 単語や数字、記号の前後には、半角の空白文字を入れても入れなくてもよい
 - `3+4`
 - `3 + 4`

K-15 デジタル社会の技術（プログラミング）

1. 逐次処理・変数・文字列

2. 変数

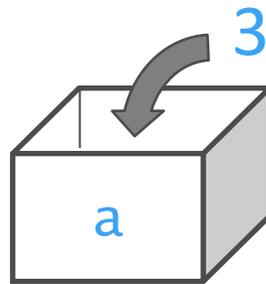
1. はじめの1歩
2. 変数
3. 型と算術演算子
4. 文字列

2. 変数 (Variable)

- 「**変数**」とは、値を入れておく**入れ物**



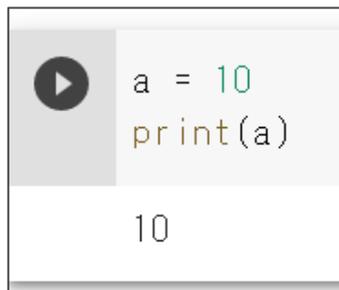
「代入」の一般的なイメージ



a という名前が付いた箱に 3 を入れる

代入した値を確認する

print 関数で、変数に代入されている値を出力できる。

A screenshot of a code execution environment. The code input area contains two lines: `a = 10` and `print(a)`. The output area below shows the result `10`.

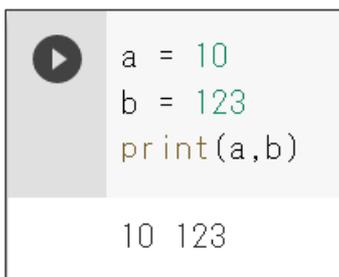
```
a = 10
print(a)

10
```

変数 a に 10 を代入
print関数で a の値を出力

※ 変数の値を見ることを
「参照する」という

複数の変数の値をいっぺんに出力できる

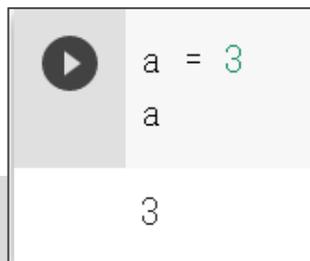
A screenshot of a code execution environment. The code input area contains three lines: `a = 10`, `b = 123`, and `print(a,b)`. The output area below shows the result `10 123`.

```
a = 10
b = 123
print(a,b)

10 123
```

print関数に a と b の値を渡して出力

Google Colab (Jupyter notebook) では print 関数を省略できる

A screenshot of a code execution environment. The code input area contains two lines: `a = 3` and `a`. The output area below shows the result `3`.

```
a = 3
a

3
```

変数名だけ書くと、変数の値が出力される

3つのキーワード

- 変数
- 代入
- 参照

変数の値を上書き

```
 a = 3  
print(a)  
a = 5  
print(a)  
a = 'Hello'  
print(a)  
  
 3  
5  
Hello
```

変数 a に 3 を代入

変数 a に 5 を代入 (上書きされる)

変数 a に文字列 Hello を代入 (上書きされる)

問題 1

- 次の文章のうち正しいものには○を、正しくないものには×をつけてください。
- (1) コンピュータは、Pythonのプログラムコードを直接理解して処理を行う。
 - (2) Pythonのプログラムコードは、大文字と小文字の違いを区別しない。
 - (3) Pythonには、1行ずつプログラムコードを入力して、そのつど実行する方法がある。
 - (4) 「`print(こんにちは)`」と記述すると、「こんにちは」という文字列が出力される。
 - (5) 変数には後から異なる値を代入できる。

問題 1 (解答)

次の文章のうち正しいものには○を、正しくないものには×をつけてください。

- × (1) コンピュータは、Pythonのプログラムコードを直接理解して処理を行う。
- × (2) Pythonのプログラムコードは、大文字と小文字の違いを区別しない。
- (3) Pythonには、1行ずつプログラムコードを入力して、そのつど実行する方法がある。
- × (4) 「`print(こんにちは)`」と記述すると、「こんにちは」という文字列が出力される。
- (5) 変数には後から異なる値を代入できる。

問題 2

- 次の文章の空欄に入れるべき語句を、選択肢から選んでください。
 - コンピュータが値を記憶しておくための入れ物のことを[(1)]という。
 - [(1)] に値を格納することを[(2)]という。
 - [(2)] を行うには、記号[(3)]を使用する。
 - [(1)]に[(2)]された値は print 関数を用いて画面に[(4)]できる。
- 【選択肢】
代入 変数 オブジェクト 出力 >>> =

問題 2 (回答)

- 次の文章の空欄に入れるべき語句を、選択肢から選んでください。
 - コンピュータが値を記憶しておくための入れ物のことを[**変数**]という。
 - [**変数**] に値を格納することを[**代入**]という。
 - [**代入**] を行うには、記号[**=**]を使用する。
 - [**代入**]に[**代入**]された値は print 関数を用いて画面に[**出力**]できる。
- 【選択肢】
代入 変数 オブジェクト 出力 >>> =

問題 3

- 次の計算を実行して結果を確認しましょう。

(1) $1 + 2 + 3 + 4$

(2) $2 + 3 * 2$

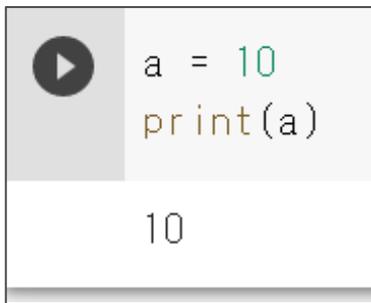
(3) $(2 + 3) * 2$

(4) $10 / 2.5$

(5) $3 / 0$

問題 4

- 以下に，変数 `a` に値 `10` を代入し，`print`関数で `a` の値を出力するプログラムを示します。



```
▶ a = 10
print(a)

10
```

- (1) 変数 `b` に値 `5` を代入し，`print`関数で変数`b`に代入された値を出力してください。
- (2) 変数 `c` に「Python」という文字列を代入し，`print`関数で変数`c`に代入された値を出力してください。

K-15 デジタル社会の技術（プログラミング）

1. 逐次処理・変数・文字列

3. 型と算術演算子

1. はじめの1歩
2. 変数
3. 型と算術演算子
4. 文字列

3. 型と算術演算子

- 型 (Type)

- 数値や文字列などのデータの種類のことを「型」とよぶ
- 数値や文字列を計算機内で「2進数で表現するか」の違い

- 算術演算子 (Arithmetic operator)

- 難しい名称だけど, 実は簡単
- 小学生の頃から習い覚えたアレ

- 足し算 +
- 引き算 –
- 掛け算 * ×ではなく, *アスタリスクを使う (アメリカ式)
- 割り算 / ÷ではなく, %でもなく, /スラッシュを使う

3.1 型 (Type)

- データの種類のことを「型」とよぶ
- はじめから準備されている型「組み込み型」

型	型名(日本語)	型名(英語)	値の例
int	整数型	Integer	-1, 0, 1, 2, 10, 100
float	浮動小数点型	Floating Point	少数を含む数 -0.12, 0.0, 0.5, 3.1415
str	文字列型	String	"Hello", 'こんにちは'
bool	真偽値型	Boolean	True, False

type関数による型の確認

- type(値) または type(変数名) で型を確認できる。

```
[1] type(10)
```

```
int
```

10は int 型
(整数型)

```
▶ type(0.5)
```

```
float
```

0.5は float 型
(浮動小数点型)

```
▶ type('こんにちは')
```

```
str
```

'こんにちは' は str 型
(文字列型)

```
▶ type(True)
```

```
↳ bool
```

True は bool 型
(真偽値型)

```
▶ a = 10  
type(a)
```

```
int
```

変数 a に 10 を代入
a の型は int 型
(整数型)

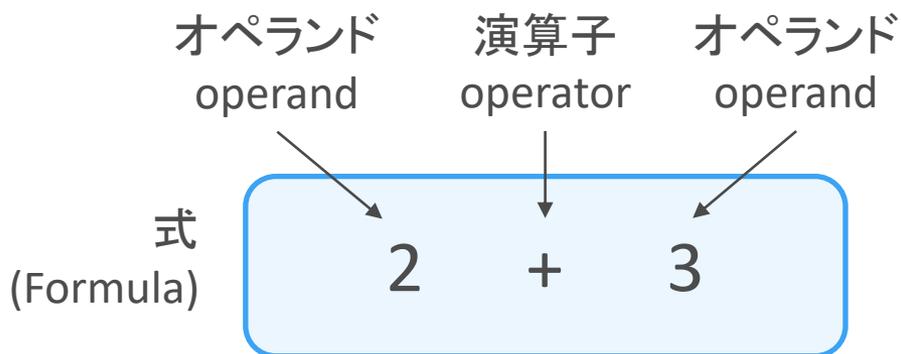
```
[7] b = 'Hello'  
type(b)
```

```
str
```

変数 b に 'Hello' を代入
b の型は str 型
(文字列型)

3.2 算術演算子

- 覚えるべき用語



演算子	演算の内容
+	加算(足し算)
-	減算(引き算)
*	乗算(掛け算)
/	除算(割り算)
//	商
%	剰余(余り)
**	べき乗

除算

```
a = 10/3
print(a)
```

3.3333333333333335

商

```
a = 10//3
print(a)
```

3

剰余(余り)

```
a = 10 % 3
print(a)
```

1

べき乗

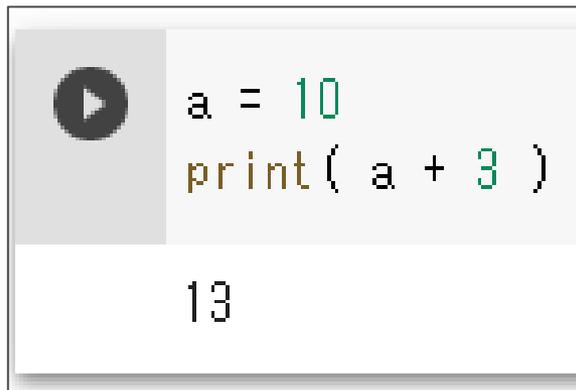
```
a = 2 ** 3
print(a)
```

8

2³を計算

変数を含む算術演算

- 式に変数名が含まれる場合、変数に代入されている値で計算する。

A screenshot of a Python code editor showing a code cell with a play button icon. The code cell contains two lines: `a = 10` and `print(a + 3)`. Below the code cell, the output `13` is displayed.

```
a = 10
print( a + 3 )

13
```

変数 `a` に 10 を代入
`print`関数で表示されるのは、
`a + 3` の計算結果 (10 + 3 の結果)

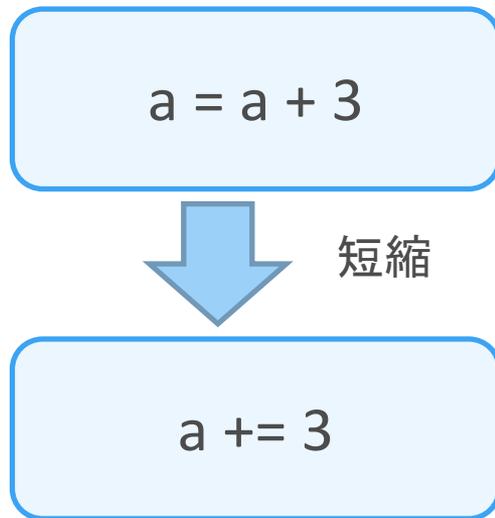
`b = a + 3`

(変数 `a` の値) + 3 が変数 `b` に代入される

`a = a + 3`

(変数 `a` の値) + 3 が変数 `a` に代入される
つまり `a` の値が 3 増える

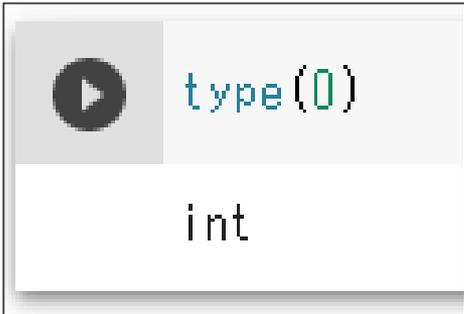
算術演算の短縮表現



演算子	使用例	説明
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code> と同じ
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code> と同じ
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code> と同じ
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code> と同じ
<code>//=</code>	<code>a //= b</code>	<code>a = a // b</code> と同じ
<code>%=</code>	<code>a %= b</code>	<code>a = a % b</code> と同じ
<code>**=</code>	<code>a **= b</code>	<code>a = a ** b</code> と同じ

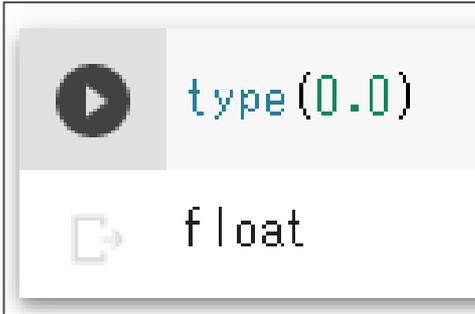
数値の型

- プログラムコードへの記述のしかたで、型が異なる。
 - 小数点を含まない → int型
 - 小数点を含む → float型



```
▶ type(0)  
int
```

0だとint型になる。



```
▶ type(0.0)  
↳ float
```

0.0だとfloat型になる。

演算と数値の型

- int型どうしの加算・減算・乗算 → int型
- int型どうしの除算 → float型
- float型を含む演算 → float型

```
▶ type(3+2)  
int
```

```
▶ type(3-2)  
int
```

```
▶ type(3*2)  
int
```

int 型どうしの加算, 減算, 乗算の結果は int 型になる。

```
▶ type( 3 / 2 )  
float
```

int 型どうしの除算の結果は float 型になる。

```
▶ type( 4 / 2 )  
float
```

割り切れる場合でも float 型になる。

数値の型変換

- int 型 → float 型

```
a = float( 100 )
```

変数 a は float 型になる。値は 100.0 。

- float 型 → int 型

```
a = int( 9.6 )
```

変数 a は int 型になる。
値は 9 (小数点以下を切り捨て)。

数値の指数表現

- すごく大きな数, 小さな数は, 指数表現で記述すると便利

$$2.5e-4 \rightarrow 2.5 \times 10^{-4} \rightarrow 0.00025$$

$$2.5e4 \rightarrow 2.5 \times 10^4 \rightarrow 25000.0$$

2.5e-4

仮数部 指数部

- 例

– アボガドロ数: $6.02214076 \times 10^{23}$

6.02214076e23

– プランク定数: $6.62607015 \times 10^{-34}$

6.62607015e-34

```
▶ a = 6.02214076e23  
print(a)  
print(type(a))
```

```
↳ 6.02214076e+23  
<class 'float'>
```

問題 5

- 以下の記述について、正しいものには○を、誤りのあるものには×をつけてください。
 - (1) 一度int型の値を代入した変数aに対して、後から文字列を代入することはできない。
 - (2) int型の値とfloat型の値を加算するときには、その前にint型の値をfloat型に型変換しておく必要がある。
 - (3) int型とfloat型の値を含む算術演算の結果はfloat型になる。
 - (4) `a = int(3.8)`と記述した場合、変数aの値は4になる。

問題 5 (解答)

• 以下の記述について、正しいものには○を、誤りのあるものには×をつけてください。

- × (1) 一度int型の値を代入した変数aに対して、後から文字列を代入することはできない。
- × (2) int型の値とfloat型の値を加算するときには、その前にint型の値をfloat型に型変換しておく必要がある。
- (3) int型とfloat型の値を含む算術演算の結果はfloat型になる。
- × (4) $a = \text{int}(3.8)$ と記述した場合、変数aの値は4になる。
(3になる)

問題 6

• 次の値を求める式を書いてください。

(1) 100を9で割った商

(2) 1000を7で割った余り

(3) 3の5乗

問題 6 (解答)

• 次の値を求める式を書いてください。

(1) 100を9で割った商 $100 // 9$

(2) 1000を7で割った余り $1000 \% 7$

(3) 3の5乗 $3 ** 5$

問題 7

- 次の命令文を、加算代入 ($+=$)、減算代入 ($-=$)、乗算代入 ($*=$)、除算代入 ($/=$)、剰余代入 ($\%=$) の演算子を使って、短い表現に書き換えてください。

$$(1) a = a + 5$$

$$(2) b = b - 6$$

$$(3) c = c * a$$

$$(4) d = d / 3$$

$$(5) e = e \% 2$$

問題 7 (解答)

- 次の命令文を、加算代入 (`+=`)、減算代入 (`-=`)、乗算代入 (`*=`)、除算代入 (`/=`)、剰余代入 (`%=`) の演算子を使って、短い表現に書き換えてください。

$$(1) a = a + 5 \qquad a += 5$$

$$(2) b = b - 6 \qquad b -= 6$$

$$(3) c = c * a \qquad c *= a$$

$$(4) d = d / 3 \qquad d /= 3$$

$$(5) e = e \% 2 \qquad e \% = 2$$

問題 8

- 次のプログラムコードを実行した後の変数aの値を教えてください
(対話モードで実行するときに表示されるプロンプト「>>>」は省略しています)。

(1) `a = 3`
`a *= 3`

(2) `b = 2`
`a = b * b`

(3) `a = int(1.9)`

(4) `x = 'XXX'`
`y = 'YYY'`
`a = x + y`

問題 8 (解答)

- 次のプログラムコードを実行した後の変数aの値を教えてください
(対話モードで実行するときに表示されるプロンプト「>>>」は省略しています)。

(1) a = 3

a *= 3

9

3*3 なので

(2) b = 2

a = b * b

4

2*2 なので

(3) a = int(1.9)

1

1.9 の小数点以下を切り捨て

(4) x = 'XXX'

y = 'YYY'

a = x + y

XXXYYY

K-15 デジタル社会の技術（プログラミング）

1. 逐次処理・変数・文字列

4. 文字列

1. はじめの1歩
2. 変数
3. 型と算術演算子
4. 文字列

4. 文字列

- 文字列は, シングルクォート (') かダブルクォート (") で囲む。
- 2つの文字列を, + 演算子で連結できる。

```
a = 'AAA' + 'BBB'
```

変数 a の値は 'AAABBB' になる

```
a = 'AAA'  
b = 'BBB'  
c = a + b
```

変数 c の値は 'AAABBB' になる

- *演算子で連結の繰り返し

```
a = 'ABC' * 3
```

aの値は 'ABCABCABC' になる

数値→文字列の変換

数値を文字列のようには扱えない



```
a = 500 + '円'
```



str(数値)で文字列に変換する



```
a = str(500) + '円'
```



```
year = 2022  
print( str(year) + '年' )
```

2022年

year の値 (int型) を文字列型 (str型) に変換してから、連結している。

変数の値の埋め込み

数値を文字列に変換してから連結

```
▶ price = 550  
print('この商品は' + str(price) + '円です')
```

この商品は550円です



フォーマット文字列のを使用して
簡潔に記述できる

```
▶ price = 550  
print(f'この商品は{price}円です')
```

この商品は550円です

f'文字列' とすると、文字列に含まれる {変数名} 部分が
変数の値に置き換わる

フォーマット文字列の活用

フォーマット文字列

f'文字列' とすると、文字列に含まれる {変数名} 部分が
変数の値に置き換わる



{変数名} 部分に式を入れることもできる



```
a = 5  
b = 550  
print(f'1つ{a}円です。{b}個で{a*b}円です')
```

1つ5円です。550個で2750円です

input 関数による文字列入力

- input関数を使うと、プログラム利用者が入力した文字列を得ることができる。
- 例 1：入力された文字列を変数 x に代入

```
x = input()
```

- 例 2：文字列を表示して入力を促す。

```
x = input("文字を入力してenterキーを押せ:")  
print(x)
```

```
x = input("文字を入力してenterキーを押せ:")  
print(f"入力は{x}")
```

文字を入力してenterキーを押せ:

```
x = input("文字を入力してenterキーを押せ:")  
print(f"入力は{x}")
```

文字を入力してenterキーを押せ: test
入力: test

文字列→数値の変換

文字列を数値のように扱えない

```
a = '500'      ← 文字列  
b = a * 2     ← bの値は'500500'になる
```



int(文字列)で整数に変換する

```
a = '500'  
b = int(a) * 2 ← bの値は1000になる
```

※ 小数点を含む数値に変換するときは float(文字列)

len 関数による文字列の長さの取得

- len(文字列) で文字列の長さ（文字数）を取得できる

```
▶ len('Hello')
```

5

```
▶ a = 'Python'  
len(a)
```

6

```
▶ a = '東京都'  
len(a)
```

3

len は、長さを意味する Length の略。

問題 9

- 「私は21歳です。」という文字列が出力されるように作成した次のプログラムコードは、実行するとエラーが発生します。適切に動作するように修正してください。

```
age = 21
print('私は' + age + '歳です。')
```

問題 9 (解答)

- 「私は21歳です。」という文字列が出力されるように作成した次のプログラムコードは、実行するとエラーが発生します。適切に動作するように修正してください。

```
age = 21
print('私は' + age + '歳です。')
```

整数型を、文字列型に変換して修正

```
age = 21
print('私は' + str(age) + '歳です。')
```

フォーマット文字列を用いて修正

```
age = 21
print(f'私は{age}歳です。')
```

問題

- input関数を用いて、入力された文字列を3回表示する@うプログラムを作成せよ。
- なお、入力を受け付ける際、「Input a string:」と表示して、入力を促すようにせよ。

問題

- input関数を用いて、自分の名前を入力してもらい、名前の文字列と、その長さを表示するプログラムを作成せよ。
- 入力を受け付ける際、「Input your name:」と表示して、入力を促すようにせよ。

問題

- input関数を用いて、2つの整数 m, n を入力し、 m と n の割り算の「商」と「余り」を計算するプログラムを作成せよ。
 - ヒント： $m // n$ で商、 $m \% n$ で余りが算出される

問題

- input 関数を用いて、秒数 x を入力してもらい、それを「h 時間 m 分 s 秒」に変換して出力するプログラムを作成せよ。
- 入力を受け付ける際、「Input seconds:」と表示して、入力を促すようにせよ。