第 4 章
プログラミング

2025 年 11 月 24 日

学習目標

(1) プログラミングとはどのようなものか体験する．
(2) プログラミングの基本的な構成要素である，順次・分岐・反復について理解する．
(3) 変数，関数の利用，比較演算子，論理演算子について理解する．
(4) バブルソート，線型探索などのアルゴリズムの考え方の概要を理解する．

本章は，専修大学商学部の高萩栄一郎の著作である．

1　プログラミング・準備

1.1　プログラミング言語

プログラミングとは，コンピューターにどのように作業をさせるのかを記述した作業業指示書を作成することです．作業指示書 (プログラム) をどのように記述するのかという書き方は沢山あり，大きな違いに，プログラミング言語の違いがあります．

作成しようとするプログラムやシステムによって，適切なプログラミング言語を選ぶことになるのですが，流行や顧客意向などさまざまな要因で選ばれます．さまざまなプログラミング言語を学習しなくてはいけないのか，というとそうではなく，1 つのプログラミング言語の考え方をマスターすることが重要です．最近よく使われるプログラミング言語同士は，かなり似通っており，「A 言語のあるの文は，B 言語ではどう書くのか」は，命令の単語の違いなど小さな違いであることが多いです．そこで，あるプログラミング言語で基本的なプログラムの書き方を修得すれば，他のプログラミング言語の修得は比較的容易です．

1.2　Python

本授業では，Python（パイソン）というプログラミング言語を使って学習します．Python は，AI 関連やビッグデータの処理のプログラミングに適していると言われています．また，他の最近メジャーなプログラミング言語に近い表現ですので，他の言語への移行も容易です．本授業では，AI やビックデータ処理のプログラムを学習するのではなく，プログラミングのごく基本的な考え方（アルゴリズム）やその記述法を学修します．

本授業では，Google 社が提供している Google Colaboratory というクラウド上で動作する Python の実行環境を利用します．この授業では，専修大学 Gmail の ID でログオンして利用します．

1.3　起動

Google Chrome[*1]で，専修大学 Gmail にログオンし，次の URL に移動します．

Colaboratory へようこそ (https://colab.research.google.com/?hl=ja)

図 1 のような画面が表示されますので，＋ノートブックを新規作成というボタンをクリックします．すると図 1 のような編集画面が表示されます．　動画:Google Colaboratory の起動

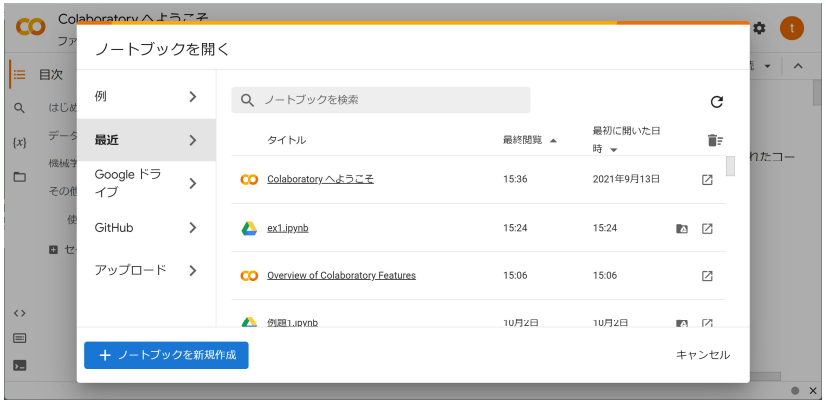

図 1　Google Colaboratory の起動画面

NoteBook 名　適切な名前を付けます．ここでは，ex1.ipynb にしましょう．

[*1] 他のブラウザでも同様に利用できると思いますが，本テキストの記述は Chrome を前提にしています

セル　この矩形の中にプログラムを記述していきます．

実行ボタン　このボタンをクリックするとプログラムが実行されます．Ctrol+Enterでも実行します．

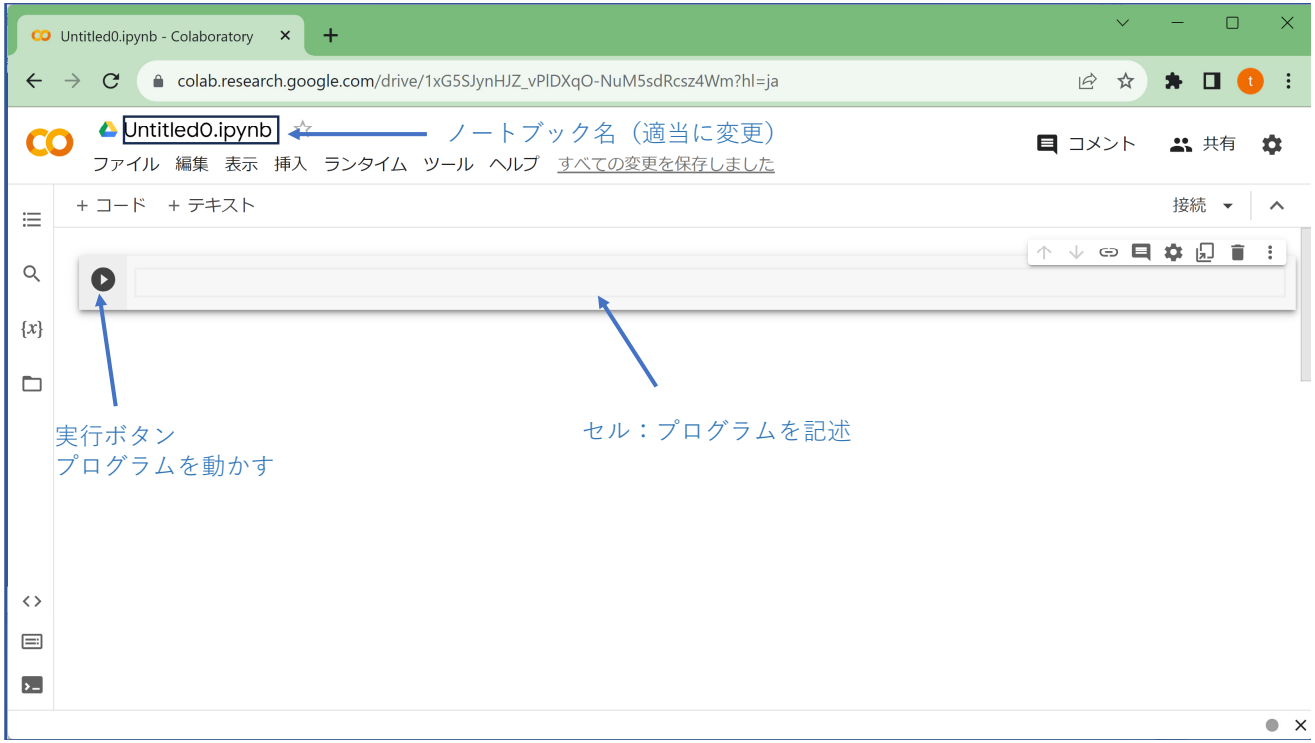

図 2　編集画面

2　順次実行

2.1　例題 1(順次実行)

例題 1 から始めましょう.

例題 1　ex1 (順次実行)

```
1 #ex1 順次実行
2 a = 5
3 b = 4
4 c = a + b
5 print (c)
```

ex1 を Colaboratory のセルに入力しましょう．練習のためコピーアンドペーストを使わずにキーボードから入力しましょう.

注意:

- 囲みの左横の数字は行番号です. 行番号は入力しません.
- プログラムは，基本的に半角で入力します．空白も基本的には半角空白「 」を利用します.
- 1 行目は，#で始まっています．#より右は，コメントというプログラムの注釈で, 基本的に人間のために書きます（コンピュータは無視します）
- 文頭には空白やタブを入力しません (Python では，文頭の空白等には役割があります (後述)).

図 3　ex1 の実行

実行結果は，図 3 のようになります．

■**変数:**　ex1 では，a,b,c という 3 つの変数を使っています．変数は，Excel のセルのように，値を入力（代入）したり，値を参照（計算式で利用）したり，画面表示したりすることができます．

■**プログラムの説明:**　プログラムは，上から下へ 1 行，1 行，順番に実行（順次実行）されます．

a = 5　a という変数に 5 という値が代入されます．
b = 4　b という変数に 4 という値が代入されます．
c = a + b　a という変数から 5 だ取り出され，b という変数から 4 だ取り出され，5 + 4 の計算が実行され, その計算結果 9

が，c という変数に代入されます．

= は，Python などの多くのプログラミング言語では，代入を意味します（等号ではありません）．右辺の計算結果が左辺の変数に代入されます．

print (c) `print（ ）`の括弧内に出力したい変数を書きます．複数あるときは，「,」で区切ります．ここでは，変数 c は，9 が入っているので 9 が出力されます．

図 3 の実行結果に表示されているようになります．

■**プログラムの実行:** 実行ボタンを押すか，Ctrl+Enterで実行します．うまく実行されない場合，プログラムと突き合わせましょう．

動画:ex1 の入力・実行

■**エラー例:**

SyntaxError: invalid non-printable character ... 全角の空白文字などの全角文字を使っていませんか？

SyntaxError: invalid syntax 文法的なミスです．改行位置を間違えたり，print のスペルを間違えていませんか．1 行目を「#」で始めていますか？

IndentationError: unexpected indent 各文を 1 文字目から書いていますか？ 文の最初を空白もしくはタブで始めていませんか？

■**セッションの再起動，課題提出時の確認:** Google Colaboratory の 1 つのノートブック内の変数の値は保持されており，他のセルからも参照できます．

(1) + コード のボタンをクリックします．新しいセルが表示されます．

(2) その新しいセルに，

print (a)

と入力します．

(3) 実行ボタンを押すと，例題のプログラムのセルで設定した a の値 5 が表示されます．

このように，プログラムの作成途中で，設定した変数の値が残り続けます．変数の値が残り続けると，プログラムで設定していなくても，他のセルや作成途中で設定した値により，うまく動作するプログラムのように見えることがあります．

提出前に，一度，変数の値などをクリアして，初期の状態（まっさらな状態）からプログラムを動かし，プログラムの動作を確認します．

(1) セッションを再起動させます．

メニューの[ランタイム] → [セッションを再起動する]

(2) プログラムが書かれた実行ボタンを押して，実行が，想定通りに行われるかを確認します．

動画: ランタイムの再起動

■保存: 保存するには，次のようにします．

メニューの[ファイル] → [保存]

■練習問題 Q1: 新しいノートブックを作成し (メニューの[ファイル] → [ノートブックを新規作成])，Q1.ipynb という名前にします．

三角形の面積を計算します．変数 x に底辺の長さ，変数 y に高さを代入し，変数 z に三角形の面積を計算して代入し，その

値を表示します．変数 x,y に代入する値は，適当に決めなさい．整数でも小数でも構いません．

2.2 例題 2（変数の型，演算子，Input）

例題 2 は，まず，名前と 整数 1,2 を入力して，その商，余りを計算します．次に，実数 1,2 を入力して，割り算の計算結果を表示するプログラムです (実行時の画面出力例を図 4 にあげます)．

例題 2　ex2 (順次実行，剰余など)

```
1  # ex2: 変数の型，演算子とInput
2  s = input('名前(文字列を入力):')   #inputは，文字列で入力される
3  print (s,'さん，こんにちは')
4  a1 = int(input('整数1:'))          #入力した整数1の文字列を整数型に変換して，a1に代入
5  a2 = int(input('整数2:'))
6  b1 = a1 // a2                       # 整数同士の割り算→小数点以下は，切り捨て
7  b2 = a1 % a2                        # 余り（剰余）を計算
8  b3 = a1 / a2                       #計算結果を浮動小数点型に代入
9  print (a1, '÷', a2 ,'=', b1, 'あまり', b2)
10 print (a1, '÷', a2 ,'=', b3)
11 f1 = float(input('実数1:'))         #実数1の文字列を浮動小数点型に変換しf1に代入
12 f2 = float(input('実数2:'))
13 g1 = f1 / f2                        #浮動小数点型同士の計算
14 print (f1, '÷', f2 ,'=', g1)
```

■変数の型: 変数には「型」があり，このテキストでは次の 3 つを使います.

整数型 (int) 整数が入ります．例: 123

浮動小数点型 (float) 実数を扱いますので，小数を扱うことができます．ただし，実数の範囲，精度に限界があります．例: 5.67

文字列型 (str) 文字列を扱います．代入するときは，「’」(Shift+7) または，「”」(Shift+2) で囲みます.

例 1：’senshu’ 例 2：**"専修大学"**

■キーボードからの入力 input(): input 関数は，キーボードからの入力を返す関数です．ただし，入力した文字列は，文字列型として返します.

s = input(’ 名前 (文字列を入力):’) のように，() 内に文字列をかけば，その文字列をプロンプトとして表示し，入力を促します．また，入力した結果を文字列として変数 s に格納しています.

print (s,’ さん，こんにちは’)

print 文の中で，カンマ区切りで，変数 s の値と「さん，こんにちは」を出力しています.

■変数の型の変換:

a1 = int(input(’ 整数 1:’))

input(’ 整数 1:’) で入力した値は，文字列型です．文字列型だと計算できません．例えば，足し算は文字列の連結になります．r = ”20” + ”30” とすると，r には，「2030」が代入されます.

整数として，計算するには，整数型に変換する必要があります．int 関数により，整数型に変換して，変換した整数型の値を a1 に代入します.

■整数の除算:

整数の除算では，小学校低学年で学修した「○○あまり△△」と小学校高学年での「2.66...」のような計算法の両方あります．プログラミングでは，前者は番地や何番目 (index) の計算などで使われる重要な概念です．

b1 = a1 // a2

「//」は，小学校低学年方式の商を求める演算子です．

b2 = a1 % a2

「%」は，小学校低学年方式の余り（剰余）を求める演算子です．

b3 = a1 / a2

「/」は，小学校高学年方式の除算の演算子です．計算結果は浮動小数点型です．

f1 = float(input(' 実数 1:'))

入力した文字列を浮動小数点型に変換し，その後，演算に利用しています．

```
名前(文字列を入力):生田太郎
生田太郎 さん，こんにちは
整数1:10
整数2:3
10 ÷ 3 = 3 あまり 1
10 ÷ 3 = 3.3333333333333335
実数1:5.2
実数2:3
5.2 ÷ 3.0 = 1.7333333333333334
```

図 4　ex2 の実行結果例

■プログラムの実行

例題 2 のプログラムを入力してみましょう．「#」より右はコメントです．必要に応じて入力してください（入力しなくても動作します)．

プログラムを実行して見てください (図 4)．うまくいったら，次の実験をして見ください．

実験 1: 整数 1 に，文字列を入力してみてください．

例 1：専修大学

例 2：百三十六

「専修大学」「百三十六」は，整数に変換できない単なる全角の文字列です．”百三十六”などを整数に変換しようとして失敗し，そのエラー (ValueError) が表示されます．

実験 2: 整数 1 に，小数点付きの数値の文字列を入力してみてください．

例：567.89

「百三十六」と同様にエラーが表示されます．整数型に変換するには，文字列は整数の文字列である必要があります．

実験 3: 整数 2 に，0 を入力してください．例：0

「ZeroDivisionError」という 0 で割り算をしたというエラーメッセージが表示されます．

実験 4: 実数 2 に，整数を入力してください．例：3

入力した値が整数の文字でも，浮動小数点型に変換されて計算されます．画面表示では，実数 1:18.2

実数 2:3

18.2 ÷ 3.0 = 6.066666666666666

の「÷ 3.0」のように，f2 の表示は，浮動小数点型であるので，「.0」がつきます．また，整数型の表示では，小数点が付いていないことを確認しましょう．

動画:ex2 の実行例

■練習問題 Q2:

台形の面積を計算するプログラムを作成しましょう．上底 (変数名 a), 下底 (変数名 b), 高さ (変数名 c) をキーボードから float として入力し，台形の面積を表示しなさい．

新規ノートブック名：Q2.ipynb

注意：演算の優先順位には，Excel と同様に，( ) を使います．台形の面積 = (上底 + 下底) × 高さ / 2

3　分岐

3.1　例題 3 (分岐・比較演算・論理演算)

分岐は，条件により，処理を変更することです．例題 3 では，入力した 2 つの整数の差を求める問題で，正解のときと不正解のときで，処理を変更します．

例題 3　ex3(分岐・比較演算・論理演算)

```
#ex3　分岐・真偽の判定
a = int(input('整数1:'))
b = int(input('整数2:'))
print (a , '-', b , 'はいくつですか')
ans_inp = int(input('答え:'))
ans_cor = a - b
if (ans_inp == ans_cor):
  print ('正解です')
  print (a , '-', b, '=', ans_inp)
else:
  print ('不正解です')
  print (a , '-', b, 'の計算結果と解答した', ans_inp, 'は異なります')
print ('プログラム終了')
```

■プログラムの説明

図 5 に，説明の図を用意しました．

2 行目から 3 行目　変数 a,b に整数を代入して，問題とします．

4 行目　問題文を表示しています．

5 行目　問題の解答を入力し， ans_inp という変数に代入しています．

6 行目　問題の正解を計算して， ans_cor という変数に代入しています．

7〜17 行目　if 文を用いて，正解と不正解のとき，実行する文を変更してます (図 5 参照)．

7 行目:if (ans_inp == ans_cor):　正解 ans_cor の変数値と，入力した解答の ans_cor の値が等しいかどうかチェックしています (等しければ真，等しくなければ偽)．

- ==は，等しいかどうかの比較演算子です．「=」が 1 個だと代入の意味になり，ここでは，比較しているので「==」とします．
- 他の比較演算子があります．<, <=, >, >=, 等しくない（不等号）は，！= と記述します．
- if 文の最後には「:」を付けます．

8〜9 行目　7 行目の if 文で，条件が真の時, 実行されます．

- 実行する範囲は，インデント (段下げ) で示されます．
- インデントは，Tabキー (キーボード左上）を使うと便利です．
- 実行する文は，2 行でもなくても 1 行でも 3 行以上でも構いません．

10 行目 **else:**　7 行目の if 文で，条件が偽の時の実行する文を次の行から書きます．else 文の最後にも「:」を付けます．

11〜12 行目　7 行目の if 文で，条件が偽の時, 実行されます．ここでは 2 行ですが，1 行でも 3 行以上でもかまいません．

真の時と同様に偽の時実行する範囲は，インデントで示します．

13 行目　インデントしていませんので，すべての場合で実行されます．

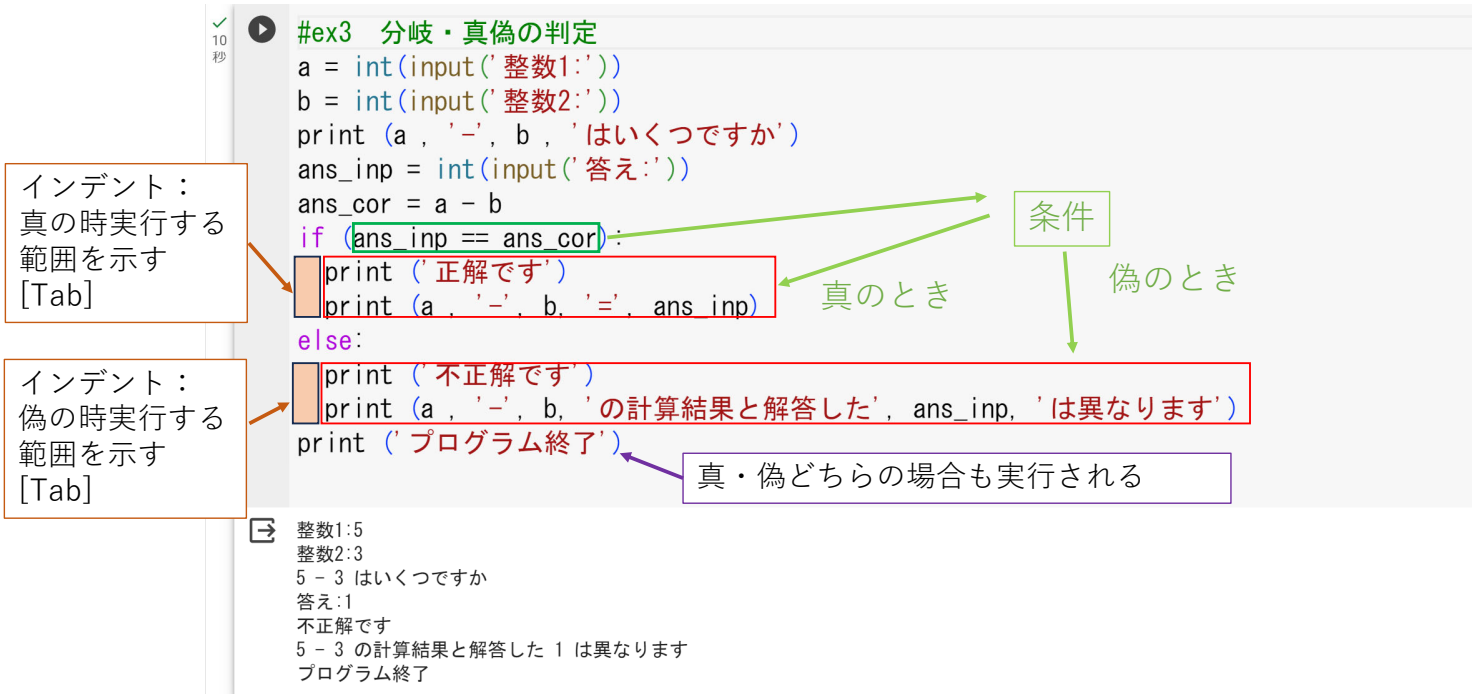

図 5　ex3 の説明と実行結果例

■プログラムの実行

例題 3 のプログラムを入力してみましょう．プログラムを実行して見みましょう (図 5)．正解のパターンと不正解のパター

ンの両方実行してみてください．両パターンで，「プログラム終了」が表示されることを確認してください．
代表的なエラーと対処法

SyntaxError: expected ':' if 文の最後　または　else 文の最後に「:」を付け忘れいませんか？
IndentationError: expected an indented block after 'if' statement
IndentationError: expected an indented block after 'else' statement　if 文や else 文の下の　インデントを忘れていませんか？

■練習問題 Q3:　2 つの整数を入力してもらい，その和を計算し，その和が 21 以下だったら，その和を点数としてその点数を表示し，21 より大きかったら（21 以下ではなかったら）「ドボーン」と表示するプログラムを作成しなさい．

3.2　例題 4 (分岐・比較演算・論理演算)

例題 4 は，例題 3 と同じように，2 つの整数を入力し，その商と余りを入力してもらいます．正解の時，「正解です」と表示し，不正解の時，「不正解です」と表示し，商と余りの正解を表示します．

例題 4　ex4(分岐・比較演算)

```
1 #ex4　分岐・論理演算子
2 a = int(input('整数1:'))
3 b = int(input('整数2:'))
4 print (a , 'わる', b , 'の商とあまりはいくつですか')
5 ansQ_inp = int(input('商:'))
6 ansR_inp = int(input('余り:'))
```

```
 7 ansQ_cor = a // b
 8 ansR_cor = a % b
 9 if ((ansQ_inp == ansQ_cor) and (ansR_inp == ansR_cor)):
10   print ('正解です')
11 else:
12   print ('不正解です')
13   print ('商の正解:',ansQ_cor)
14   print ('余りの正解:',ansR_cor)
```

■プログラムの説明

2〜3 行目: 2 つの整数を入力します.

4 行目: 問題文の提示

5〜6 行目: 商と余りの解答を入力しています.

変数名には，あとでプログラムを見て分かりやすい名前を付けます.

ここでは，商には ans の後ろに Q, 余りには ans の後ろに R をつけ,

入力した値には，その後ろに _inp を，正解には _cor を付けています.

7 行目　商の正解を計算しています．商の計算の演算子は「//」です.

8 行目　余り（剰余）の正解を計算しています．剰余計算の演算子は「%」です.

9 行目　if 文で，2 つの条件が成立していつか，調べています．1 つめの条件と 2 つめの条件をそれぞれ ( ) で囲み，両方の条件が成立するとき真にするので,「and」でつなぎます (片方または両方の条件が成立するとき真にするには「or」でつなぎます).

これらの and とかは or は，論理演算子と呼ばれています．

10 行目　条件が成立したとき (商，余り　両方とも入力した値と計算した値が一致したとき) 実行される文

11〜14 行目　条件が成立 (商，余り　少なくても片方が入力した値と計算した値が不一致のとき) 実行される文

12〜14 行目がインデントで表示されているので，この 3 行は，9 行目の条件が偽の時のみ実行されます．

■練習問題 Q4:　商と余りの問題で，商もしくは余りの少なくともどちらかが正解の時,「商または余りが，両方または片方が正解です」と表示し，そうではないとき,「商と余り両方が不正解です」と表示しなさい．どちらの場合でも，正解の商と余りを表示しなさい．

4　反復

4.1　例題 5 (反復 (for ループ)，関数)

コンピュータでは，処理を繰り返すことで大量の計算をすることが多々あります．本小節ではもっとも単純な n 回 (固定回）の繰り返しを学修します．

例題 5　ex5(反復 (For ループ)・関数)

```
1 #ex5　ループ，関数
2 n =int(input("繰り返す回数(非負の整数)"))
3 a = float(input("aの値(正の実数)"))
4 for i in range(n):
5   x = float(i)
6   y = pow(a,x)
7   print (x,"\t",y)
8 print ("--------------------")
```

\ は，半角の￥マーク（Back Space キー左）を入力します．

■関数:　関数は，引数 (0 個以上) を与えると，その関数名で定義されているプログラムで，戻り値を返すものです．引数は，「,」で区切って指定します．例として，$y = a^x$ を計算する関数 pow は次のようになります．

関数名:　pow

機能: $y = a^x$ を計算する関数
引数 1: a の値
引数 2: x の値
戻り値: y の値

※ 本節で使用する機能等に限って記述します（記載した機能以外にも機能はあります）.

■例題 5 で使われいる関数

Input (2 行目): 機能: キーボードから文字列を入力する関数.
　　引数 1: プロンプトで表示する文字列, 　戻り値：入力した文字列
int (2 行目): 機能: 引数 1 の文字列または数値を整数型に変換する.
　　引数 1: 文字列または数値　戻り値：変換された整数型の値
float (3・5 行目): 機能: 引数 1 の文字列または数値を浮動小数点型に変換する.
　　引数 1: 文字列または数値　戻り値：変換された浮動小数点型の値
range (3 行目): 機能: リスト $[0, \ldots, n-1]$ を返します．実際の利用法は，for 文の説明を参照.
　　引数 1: n , 戻り値: リスト $[0, \ldots, n-1]$
pow (6 行目)　上記参照
print(7・8 行目): 機能：引数で指定した値または変数・関数の値を画面出力します．次の引数との間は「 」(半角空白) が出力されます.

■for 文　for 文は，固定回の繰り返しなどに使われます．図 6 を参照してください.

```
for i in range(n):
```

で，「:」の次の行から，インデントしている範囲を n 回繰り返します．その繰り返しの際，変数 i の値は, 0 から n-1 まで，１ずつ増えながら（i=0,1,...,n-1) 実行します．ここで，i は，0 から始まるので，n 回繰り返しは，n-1 までです．i は，何回目かを数えているので，カウンタともよばれます．

参考:range(n) は，$[0, 1, \ldots, n-1]$ というリスト (次節で説明) で，for i in range(n) は，そのリストから，順番に 1 つずつ，取りだして i に代入して繰り返せとう命令です．range(10) は，[0,1,2,3,4,5,6,7,8,9] となります．

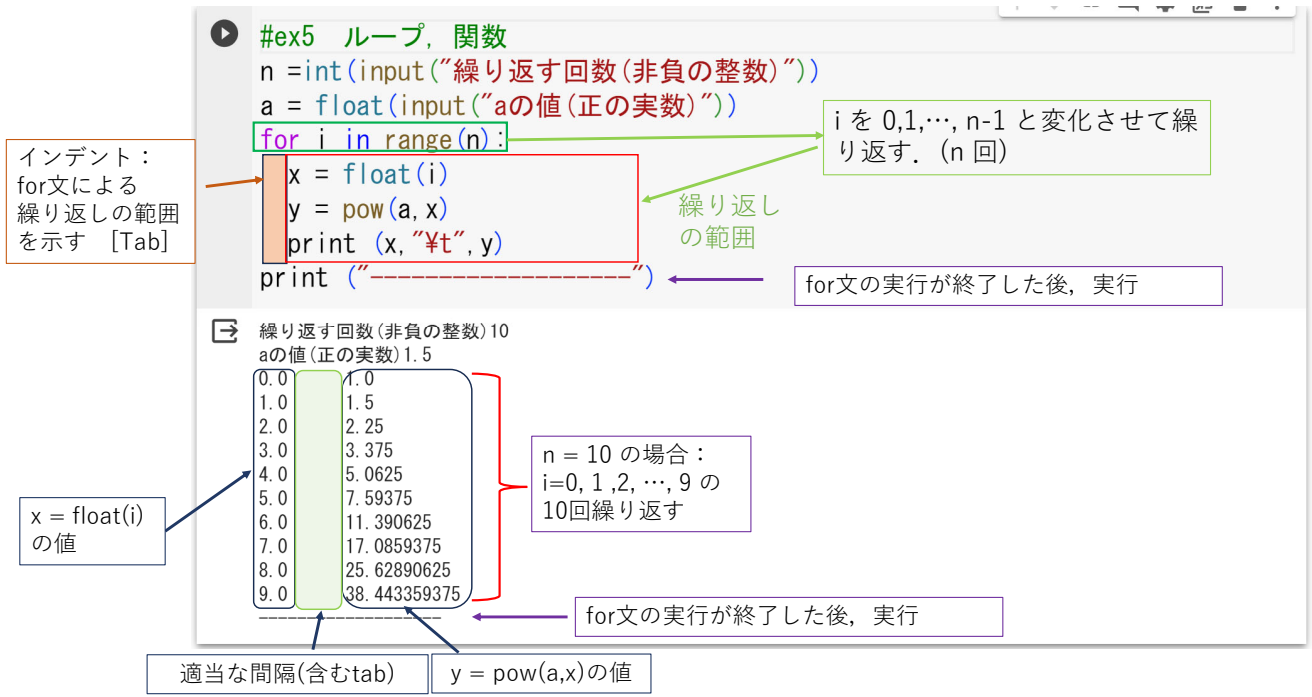

図 6　ex5 説明図

■練習問題 Q5:(2025/11/24 追加)　例題 5 を変更して，x を 0 から 9.9 までの 0.1 刻みで変化する値として，$y = ax + b$ で，各 x と y の値を表示するプログラムを作成しなさい（図 7 参照）.

a　　2 <= 入力した値
b　　3 <= 入力した値

0 から 99までの 100回

i(カウンタ)	0.1 倍	x	a	b	y=ax+b	y
0	----->	0.0	2	3	----->	3.0
1	----->	0.1	2	3	----->	3.2
2	----->	0.2	2	3	----->	3.4
3	----->	0.3	2	3	----->	3.6
:		:	:	:		:
98	----->	9.8	2	3	----->	22.6
99	----->	9.9	2	3	----->	22.8

図 7　ex5 説明図

n　　: 100 とします (固定，定数とする).(ループは，i をカウンタとして，0 から 99 までの 100 回）
a　　: 入力してもらいます (float)
b　　: 入力してもらいます (float)
x　　: for ループの中で，$x = i/10.0$ で計算します.
y　　: $y = ax + b$ で計算します.
表示　: 例題のように，各 i での x の値と y の値を表示します.

5　アルゴリズム・リスト・探索・ソート

アルゴリズムとは，これまで述べた，順次・分岐・反復といった処理手順を組み合わせて，問題を解くための処理手順のことを言います．本節では，探索・ソート・処理手順を学修します．

5.1　リスト (配列)

リストは，いくつかの変数をまとめて処理するときに使います．まとめて，処理するため，リスト名と何番目かの番号 (index) で表します．図 8 は，リストの図解で，例題 7 で，使用するリストの初期値が入っています．リスト名は，「a」で，要素数（小さな箱の数）は 5 です．要素の番号 (index) は，0 から数え始め，4 までです．

リストは，次のように定義します．

```
a = [4,1,5,7,2]
```

これで，図 8 のリストが定義されます．

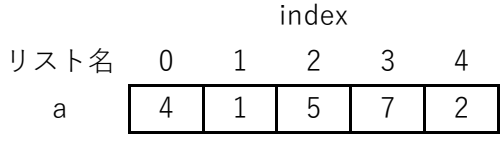

図 8　ex7 リスト

使い方は，a[2] のように，a の何番目かの index かを指定します．ここでは，2 は固定されていますが，a[i] のように，index を変数 i にして，「for i in range(n)」のように，for 文繰り返しで，カウンタ i を順次変化させて処理することに使われ

ます．また，print (a) のように，リスト a 全体を表示することもできます．

リストは，他のプログラミング言語では，配列と呼ばれることが多いです．

5.2 例題 6 探索

例題 6 は，学部コードを入力すると，専修大学の対応する学部学科名を表示するプログラムです．1 つの学部には複数の学科が設置されていることがあるので，複数個の学部学科名が表示されることがあります．

例題 6　ex6 （リスト探索)

```
#ex6 リスト探索
gm = ["経済学部現代経済学科","経済学部生活環境経済学科","経済学部国際経済学科",\
"法学部法律学科","法学部政治学科","経営学部経営学科",\
"経営学部ビジネスデザイン学科","商学部マーケティング学科","商学部会計学科",\
"文学部日本文学文化学科","文学部英語英米文学科","文学部哲学科","文学部歴史学科",\
"文学部環境地理学科","文学部ジャーナリズム学科",\
"ネットワーク情報学部ネットワーク情報学科","人間科学部心理学科",\
"人間科学部社会学科","国際コミュニケーション学部日本語学科",\
"国際コミュニケーション学部異文化コミュニケーション学科"]
gc = ["E","E","E","J","J","M","M","C","C","L","L","L",\
"L","L","L","N","H","H","G","G"]
n = len(gc)
inpstr = input("学部記号（1文字） ")
for i in range(n):
```

```
15    if (gc[i] == inpstr):
16      print (gc[i], gm[i])
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
gm	経済学部現代経済学科	経済学部生活環境経済学科	経済学部国際経済学科	法学部法律学科	法学部政治学科	経営学部経営学科	経営学部ビジネスデザイン学科	商学部マーケティング学科	商学部会計学科	文学部日本文学文化学科	文学部英語英米文学科	文学部哲学科	文学部歴史学科	文学部環境地理学科	文学部ジャーナリズム学科	ネットワーク情報学部ネットワーク情報学科
gc	E	E	E	J	J	M	M	C	C	L	L	L	L	L	L	N

図 9　ex6 2 つのリストの関係の図解

■各変数の役割

gm　学部学科名のリストです.

gc　学部コードのリストです．gm の学部学科名に対応しています (図 9 参照)

n　学部コードのリストの要素数．学部学科名のリストの要素数と同一と仮定します.

inpstr　表示したい学部のコード (1 文字) を入れる変数．キーボード入力します.

i　探索するときのカウンタ．この i を変えていき，探索をします.

図 9 のように，gm の index で示される学部学科名と gc の index で示される学部コードは対応しています．gm[1] の経済学部生活環境経済学科の学部コードは，gv[1] の E であることを示しています.

■リストの定義　2 行目から 11 行目がリストの定義です．学部記号，学部学科名は，文字列なので，「”」(または「’」) で囲みます．文末の \ は，リストの定義が次の行まで続くことを示しています．\ は，半角の ¥ マークを入力します（設定のよって，半角の ¥ マークまたは \ が表示されます).

では，新しいノートブックにプログラムを入力してみましょう．リストの部分は，以下の囲みからコピーアンドペーストするとよいでしょう．

```
gm = ["経済学部現代経済学科","経済学部生活環境経済学科","経済学部国際経済学科",\
"法学部法律学科","法学部政治学科","経営学部経営学科",\
"経営学部ビジネスデザイン学科","商学部マーケティング学科","商学部会計学科",\
"文学部日本文学文化学科","文学部英語英米文学科","文学部哲学科","文学部歴史学科",\
"文学部環境地理学科","文学部ジャーナリズム学科",\
"ネットワーク情報学部ネットワーク情報学科","人間科学部心理学科",\
"人間科学部社会学科","国際コミュニケーション学部日本語学科",\
"国際コミュニケーション学部異文化コミュニケーション学科"]
gc = ["E","E","E","J","J","M","M","C","C","L","L","L",\
"L","L","L","N","H","H","G","G"]
```

■リストの要素数

```
n = len(gc)
```

len(リスト名) は，リストの要素数を求める関数です．

■探索する値の入力

```
inpstr = input("学部記号（1 文字）")
```

inpstr に学部記号（1 文字）を入力してもらいます.

■線型探索　探索のアルゴリズムでもっとも単純なものが，線型探索です.
探索する値（例題 6 では，変数 inpstr に格納された値）をリストの 0 番目の値 (例題 6 では，gc[0]) と比較し，等しければ，0 番目が探索する値とし，何らかの処理（例題 6 では，gc[0] と gm[0] を表示）します.
次に，同様に，探索する値とリストの 1 番目の値と比較し，等しければ，1 番目が探索する値とし，何らかの処理をします.
この手順をすべてのリストの要素（例題 6 では，n-1 番目まで）について行います.
例題 6 のアルゴリズムは，gc[i] と inpstr を比較し，等しければ，gc[i] と gm[i] を画面出力する．この作業を i の 0 から n-1 まで繰り返すということになります.
　このアルゴリズムは，Python のプログラムで，分かりやすく表現でき，例題 6 の以下の部分です.

```
for i in range(n):
  if (gc[i] == inpstr):
    print (gc[i], gm[i])
```

■練習問題 Q6　下記のような 得点のリスト score と対応する学生番号のリスト gno が与えられています．探索最小値（整数型）を入力すると，探索最小値以上の学生の得点と学生番号を表示するプログラムを作成しましょう.
ヒント：例題 4 の 9 行目の if を参考にしよう.

```
score = [68,54,81,72,81,78,77,55,72,66,60,52,81,66,79,74,55,\
89,52,60]
gno=["SU01","SU02","SU03","SU04","SU05","SU06","SU07","SU08",\
"SU09","SU10","SU11","SU12","SU13","SU14","SU15","SU16",\
"SU17","SU18","SU19","SU20"]
```

ヒント：2 つの条件が成立するかの判定は，例題 4 を参照.

※ この小節で学修したことは，Python の辞書という機能で簡単に実現できます．ここでは，線型探索のアルゴリズムの学修という観点から，辞書機能は使いません.

5.3 例題 7 バブルソート

本小節では，ソートのアルゴリズムのうち，単純なバブルソートを学修します．2 重の for ループを使うなど，わかりにくい部分がありますが，おおまかにこのような処理をしているということを理解するようにしてください．リスト a を小さい順 (昇順) にソートするプログラムを考えてみます.

プログラムは，複数のセルに分割して作成していき，最後にまとめます．新しいノートブックを作成してください.

■リストの定義 新しいセルに，次のプログラムを入力します．このセルをセル A とします.

セル A:

```
a = [4,1,5,7,2]
n = len(a)
print ("n=",n,"a=",a)
```

「a = [4,1,5,7,2]」で，リスト a を定義しています．「n = len(a) 」で，リスト a の要素数を変数 n に代入しています．「print (”n=”,n,”a=”,a)」で，n と a の値を表示しています．

■隣の要素を比較　バブルソートの基本的なアルゴリズムは，index を使って隣同士 (j 番目と j+1 番目) の値を比較していき，昇順（または降順）になっていなければ，入れ換えるというものです．

最初は，0 番目と 1 番目，次は，1 番目と 2 番目，... 最後，n-2 番目 (例題では 3 番目）と n-1 番目 (例題では 4 番目）と比較，必要に応じて入れ換えを行っていきます．それを図解したものが，図 9 です．

(1) j=0 は，初期の状態 a= [4, 1, 5, 7, 2] です．
(2) 最初 (j=0) に，a[0] の 4 と a[1] の 1 を比較します．小さい順ですので, 順番が逆で，a[0] と a[1] を入れ換えます．j=0 比較入れ換え後のようになります．
(3) j=1 比較入れ換え前で，a[1] の 4 と a[2] の 5 を比較します．小さい順ですので入れ換えをしません．
(4) j=2 比較入れ換え前で，a[2] の 5 と a[3] の 7 を比較します．小さい順ですので入れ換えをしません．
(5) j=3 比較入れ換え前で，a[3] の 7 と a[4] の 2 を比較します．小さい順ですので，順番が逆で，a[3] と a[4] を入れ換えます．j=3 比較入れ換え後のようになります．

大きな数値が，index が大きな要素に，小さな数値が，index が小さな要素に格納されていく様子がわかります．
※ 図 10 ではまだ，完全には 小さい順に並べ変わっていません．

この部分のプログラムは，次のようになります．新しいセルを作成（+ コード で作成）し，次のプログラムを入力します．このセルをセル B とします．
セル B:

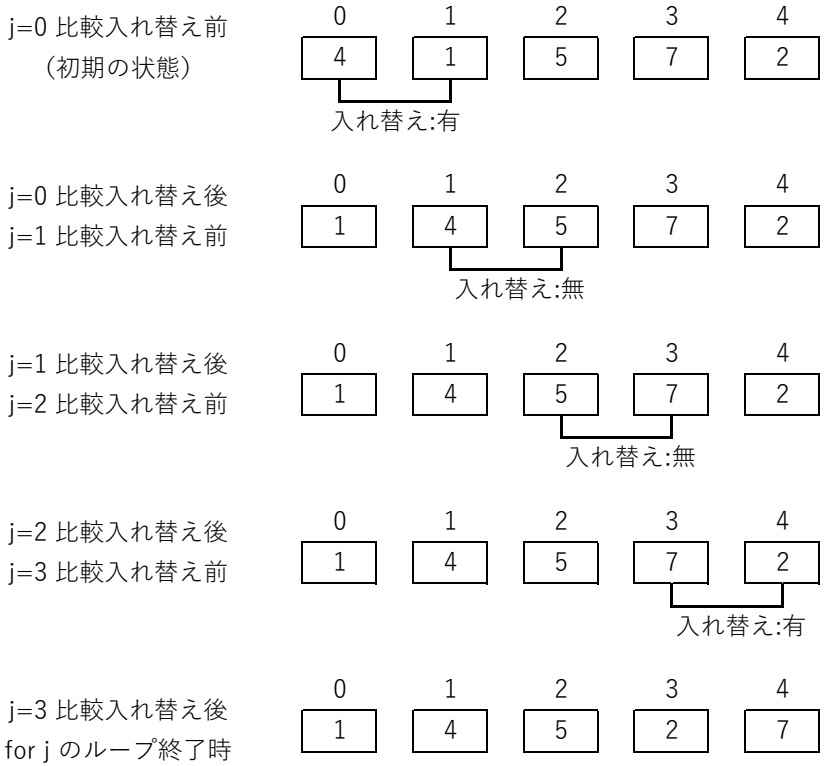

図 10　ex7　隣同士を比較

```
for j in range(n-1):
  if a[j] > a[j+1]:
    a[j], a[j+1] = a[j+1], a[j]
  print ("j=", j , "a=",a)
```

セル B のプログラムを 1 回実行すると，a= [1, 4, 5, 2, 7] となります（図 10 と同じ）.

1 行目: `for j in range(n-1):`

隣同士を比較するために，繰り返します．リストは n 個の要素なので，最後は，n-2 番目 (3 番目) と n-1 番目 (4 番目) です．したがって，j は，n-2 番目まで，繰り返しの回数は，n-1 回です.

2 行目: `if a[j] > a[j+1]:`

j 番目と j+1 番目を比較します．j 番目の方が大きい時は，小さい順ではないので，入れ換えます.

3 行目: `a[j], a[j+1] = a[j+1], a[j]`

for で 1 回インデント，if でもう 1 回インデントするので，tab キーで 2 回分インデントさせます.

a[j] の値と a[j+1] の値との入れ換えの文です．左辺の 1 個目の変数に右辺の 1 個目の値，左辺の 2 個目の変数に右辺の 2 個目の値が代入されます.

4 行目: `print ("j=", j , "a=",a)`

途中経過を表示します．if 文の真偽にかかわらず，実行します.

では，実行して，リスト a を小さい順に並べ換えてみましょう.

(1) ランタイムを再起動します．メニューの ランタイム → ランタイムを再起動
(2) セル A を実行します．リスト a が初期の状態，a= [4, 1, 5, 7, 2] に設定されたと思います.
(3) セル B を実行します．リスト a が並べ換わり，a= [1, 4, 5, 2, 7] となったと思います（出力の最下行).
この段階では，リストは小さい順に並べ換わっていません.
(4) セル B を再度，実行します．リスト a が並べ換わり，a= [1, 4, 2, 5, 7] となったと思いますが，まだ，リストは小さい順に並べ換わっていません.
(5) セル B を再度，実行します．リスト a が並べ換わり，a= [1, 2, 4, 5, 7] となったと思います．これは，小さい順に並べ

換わっています.

3 回セル B を実行すると，小さい順に並べ換わりました．いつも 3 回セル B を実行すれば良いのでしょうか？ セル A のリスト a の設定で，a = [4,7,2,7,1] としてみしょう．セル B を 4 回実行すると，小さい順に並び換わっています．この様子 (出力の最下行) は，次のようになります.

```
 初期     n= 5 a= [4, 7, 2, 5, 1]
1 回目     j= 3 a= [4, 2, 5, 1, 7]
2 回目     j= 3 a= [2, 4, 1, 5, 7]
3 回目     j= 3 a= [2, 1, 4, 5, 7]
4 回目     j= 3 a= [1, 2, 4, 5, 7]
```

ここで，値 1 （最小値）に注目します．初期では a[4]，1 回目では a[3], 2 回目では a[2], 3 回目では a[1], 4 回目では a[0] と 1 つずつ，index が小さくなっています (このように，値がだんだん泡のように浮かび上がってくることから，バブルソートと呼ばれています).

最小値など小さな値が，index の大きな要素から index の小さな要素へ，リスト B の実行で 1 つずつ移動しているので，最大 n 回（リスト a の大きさ）実行すれば，ソートは完成すると考えます．そこで，次のようにプログラムを組もうとおいます.

```
リスト A のリスト a と n の設定．リスト a は，適当に変更可

for i in range(n):
    リスト B
```

このプログラムは，リスト A の初期設定をした後，リスト B を n 回繰り返し実行するものです．例題 6 は，上記のプログラムをまとめたものです．ただし．9 行目では，i の値も出力しています．リスト a の設定を適当に変更してみて，どのように変化するか確認してください．毎回，ランタイムを再起動して実行してみてください．

例題 7　ex7（アルゴリズム，バブルソート，リスト)

```
1  #ex7:  バブルソート
2  a = [4,1,5,7,2]
3  n = len(a)
4  print ("n=",n,"a=",a)
5  for i in range(n):
6    for j in range(n-1):
7      if a[j] > a[j+1]:
8        a[j], a[j+1] = a[j+1], a[j]
9      print ("i=", i,"j=", j , "a=",a)
```

動画:ex7, バブルソートのアルゴリズム（音声付き）

動画:ex7, バブルソートのアルゴリズム（音声無し）

■練習問題 Q7:　ex7　を基にして，大きい順（降順）に並べ換えるプログラムにしなさい．また，リスト a の要素数を 10 個にしなさい．a の要素の値は適当に定めなさい．