



Pythonの道

簡単なプログラム
を作ろう④

もくじ

- ・ 虫食いアルファベットゲームを作る



虫食いアルファベットゲームを作る

画面に表示されているアルファベットの中で抜けている文字を探すゲームを作ろう。datetimeモジュールを使って、クリアする時間を計測するゲームにするよ。

まずは、リストで定義したデータを1つずつ出力するプログラムを作ろう。

```
アルファベット = [" A" , " B" , " C" , " D" , " E" , " F" , " G" ]
```

..... リスト アルファベットを定義する

```
for i in アルファベット: .....
```

..... リストの要素を1つずつ、変数 i に代入し、処理を繰り返す

```
    print(i) .....
```

..... 変数 i の値を表示する

※変数は通常、アルファベットで定義する。理解が追いつくまで、かな文字で変数を定義してもOK。ただし、他のプログラミング言語ではアルファベットでしか変数を定義できないことは知っておこう。

虫食いアルファベットゲームを作る

単語帳 ～英語も一緒に覚えてしまおう！～

for:～に向かって、in:～の中で、 print:印刷する

「Run Module」またはF5キーを押してプログラムを実行する。



A
B
C
D
E
F
G
>>>

リストを理解する

データをたくさん使うときは「変数」ではなく、「リスト」を使うよ。



scratchの「リスト」ブロックと同じ

リストとは、「番号付きの棚」のようなもので、「何番の中の値」と番号を指定して、値を入れたり、取り出したりする。「名前」ではなく「番号」で指定できるので、データがたくさんあっても扱いやすい。

リストを理解する

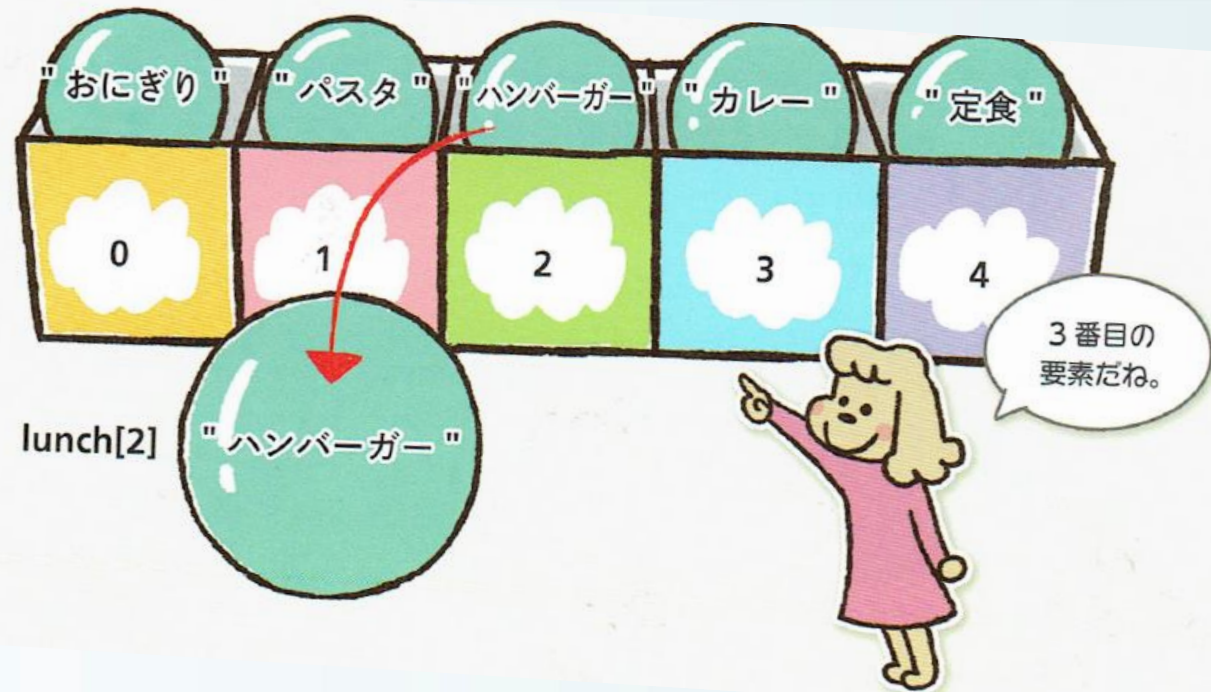
リスト名 = [要素1,要素2,要素3,……] リストを作る
print(リスト名[インデックス]) リストの要素を表示する

```
lunch = ["おにぎり", "パスタ", "ハンバーガー", "カレー", "定食"]  
print(lunch[2])
```

0 1 2 3 4

⇒ ハンバーガー
>>>

ポイント
リストはゼロからカウントすることを覚えておこう。



虫食いアルファベットゲームを作る

AからGまでのいずれか1文字が抜けたアルファベットの文字列を作ろう。プログラムを実行するたびに抜けたアルファベットが変化するよ。

```
import random . . . . ランダム関数を呼び出す
アルファベット = [" A" , " B" , " C" , " D" , " E" , " F" , " G" ]
. . . . リスト 「アルファベット」 を定義する

虫食い = random.choice(アルファベット)
. . . . 変数 「虫食い」 にリスト 「アルファベット」 の中からランダムに選んだ文字を代入する

表示 = " " . . . . 変数 「表示」 を空白にする

for i in アルファベット: . . . . 繰り返し処理 リスト 「アルファベット」 に入っ
    if i != 虫食い: . . . . ている文字を順番に取り出す
        表示 = 表示 + i . . . . もし取り出したアルファベットが変数 「虫食い」
. . . . の値と異なる場合、表示 = 表示 + i の処理をする
print(表示) . . . . 変数 「表示」 に表示 + i を代入して画面に表示させる
```

虫食いアルファベットゲームを作る

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、random:でたらめの、choice:選択、for:～に向
かって、in:～の中に、if:もし～なら、print:印刷する

「Run Module」またはF5キーを押してプログラムを
実行する。

➡ ABDEFG
>>>

if文を理解する

if文を使えば、条件分岐のプログラムを作ることができる。



scratchの「もし〇〇ならXX」ブロックと同じ

```
if i != 0 :      . . . . . もし i が 0 と等しくない場合
    print(" XX" ) . . . . . XX と表示する
```

※「:」(コロン)の後に改行してインデントを付ける。
インデントの範囲内では、条件式は有効とならないことを覚えておこう。

※ != は「等しくない」ときという意味になる。

虫食いアルファベットゲームを作る

虫食いのアルファベットを入力し、答えが正しいかどうかを判定するプログラムを作る。

```
import random . . . . ランダム関数を呼び出す
アルファベット = [ " A" , " B" , " C" , " D" , " E" , " F" , " G" ]
. . . . リスト 「アルファベット」 を定義する
虫食い = random.choice(アルファベット)
. . . 変数 「虫食い」 にリスト 「アルファベット」 の中からランダムに選んだ文字を代入する
表示 = " " . . . . 変数 「表示」 をブランクにする
for i in アルファベット: . . . . 繰り返し処理 リスト 「アルファベット」 に入っ
    if i != 虫食い: . . . . ている文字を順番に取り出す
        表示 = 表示 + i . . . . もし取り出したアルファベットが変数 「虫食い」
. . . . の値と異なる場合、表示 = 表示 + i の処理をする
print(表示) . . . . 変数 「表示」 に表示 + i を代入して画面に表示させる
答え = input(" 虫食いのアルファベットは?" )
. . . . 変数 「答え」 に 「虫食いのアルファベットは？」 の回答を代入する
```

次のページに続く

虫食いアルファベットゲームを作る

```
if 答え == 虫食い: . . . . . もし変数「答え」の値が変数「虫食い」の値と同じなら
    print(" 正解です" ) . . . . . 「正解です」と表示する
else: . . . . . そうでないなら
    print(" 違います" ) . . . . . 「違います」と表示する
```

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、random:でたらめの、choice:選択、for:～に向
かって、in:～の中に、if:もし～なら、print:印刷する、
else:～でなければ

「Run Module」またはF5キーを押してプログラムを実行する。

```
➡ ABDEFG
   虫食いのアルファベットは? C
   正解です
   >>>
```

if文を理解する

if文を使えば、条件分岐のプログラムを作ることができる。



scratchの
「もし〇〇ならXX
でなければ△△」
ブロックと同じ

```
if 条件式(〇〇):      . . . . . もし〇〇なら    (条件分岐)
print(" XX" )      . . . . . 条件を満たす場合、XXと表示する
else:               . . . . . でなければ    (条件分岐)
print(" △△" )    . . . . . 条件を満たさない場合、△△と表示する
```

※「:」(コロン)の後に改行してインデントを付ける。
インデントの範囲内では、条件式は有効とならないことを覚えておこう。

虫食いアルファベットゲームを作る

虫食いアルファベットを見つけるまでの時間を計測できるようにしよう。

```
import random . . . . ランダム関数を呼び出す
import datetime . . . . デイトタイム関数を呼び出す
アルファベット = [" A" , " B" , " C" , " D" , " E" , " F" , " G" ]
                . . . . リスト 「アルファベット」を定義する

虫食い = random.choice(アルファベット)
        . . . . 変数 「虫食い」にリスト「アルファベット」の中からランダムに選んだ文字を代入する

表示 = " " . . . . 変数 「表示」を空白にする

for i in アルファベット:
    . . . . 繰り返し処理 リスト「アルファベット」に入っている文字を順番に取り出す
    if i != 虫食い:
        . . . . もし取り出したアルファベットが変数「虫食い」の値と異なる場合、表示 = 表示 + i の処理をする
        表示 = 表示 + i

print(表示) . . . . 変数 「表示」に表示 + i を代入して画面に表示させる
```

次のページに続く

虫食いアルファベットゲームを作る

```
時間1 = datetime.datetime.now() . . . 変数 「時間1」に現在の時刻を代入する
答え = input(" 虫食いのアルファベットは?" )
. . . . 変数 「答え」に「虫食いのアルファベットは?」の回答を代入する
if 答え == 虫食い: . . . . もし変数 「答え」の値が変数 「虫食い」の値と同じなら
    print(" 正解です" ) . . . . 「正解です」と表示する
    時間2 = datetime.datetime.now()
    . . . . 変数 「時間2」に現在の時刻を代入する
    print((時間2-時間1).seconds)
    . . . . 変数 「時間2」 - 「時間1」の結果を秒数で表示する
else:
    . . . . そうでないなら
    print(" 違います" ) . . . . 「違います」と表示する
```

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、random:でたらしめ、 datetime:日付/時間
、choice:選択、 for:～に向かって、 in:～の中に、 if:もし～なら、
print : 印刷する、 now:今、 else:～でなければ

虫食いアルファベットゲームを作る

「Run Module」またはF5キーを押してプログラムを実行する。



ACDEFG

虫食いのアルファベットは？B

正解です

6

>>>

datetime関数を理解する

datetime関数を使えば、現在の時刻や日時を取得することができる。



scratchの「現在の〇〇」ブロックと同じ

現在日時を表示するときはdatetime.now()と書く。

```
import datetime
```

```
現在日時 = datetime.datetime.now()
```

```
print(現在日時)
```

現在時刻を呼び出すクラス

```
>>> 2020-06-06 13:57:15.279526
```

現在日付を表示するときはdate.today()と書く。

```
import datetime
```

```
現在日付 = datetime.date.today()
```

```
print(現在日時)
```

現在日時を呼び出すクラス

```
>>> 2020-06-06
```


datetime関数を理解する

年月日時分秒を別々に取得したい場合は以下のようにコーディングすることを知っておこう。

```
import datetime
現在日時 = datetime.datetime.now()
print(現在日時.year)  yearは年のこと
print(現在日時.month) monthは月のこと
print(現在日時.day)  dayは日のこと
print(現在日時.hour)  hourは時間のこと
print(現在日時.minute) minutelは分のこと
print(現在日時.second) secondは秒のこと
print(現在日時.microsecond) microsecondは
                               マイクロ秒のこ
                               と
```



```
2020 年
6     月
6     日
15    時
20    分
31    秒
587133 マイクロ秒
>>>
```

虫食いアルファベットゲームを作る

リストにHからZまでを加え、秒数の表示を追加して虫食いアルファベットゲームを完成させよう。

```
import random . . . . ランダム関数を呼び出す
import datetime . . . . デイトタイム関数を呼び出す
アルファベット = [
    " A" , " B" , " C" , " D" , " E" , " F" , " G"
    " H" , " I" , " J" , " K" , " L" , " M" , " N"
    " O" , " P" , " Q" , " R" , " S" , " T" , " U"
    " V" , " W" , " X" , " Y" , " Z"
] . . . . リスト 「アルファベット」 を定義する
虫食い = random.choice(アルファベット)
. . . . 変数 「虫食い」 にリスト 「アルファベット」 の中からランダムに選んだ文字を代入する
表示 = " " . . . . 変数 「表示」 を空白にする
```

次のページに続く

虫食いアルファベットゲームを作る

```
for i in アルファベット:
```

・・・繰り返し処理 リスト「アルファベット」に入っている文字を順番に取り出す

```
    if i != 虫食い: ・・・もし取り出したアルファベットが変数「虫食い」の値と異なる場合、表示 = 表示 + i の処理をする
```

```
        表示 = 表示 + i
```

```
print(表示) ・・・変数「表示」に表示 + i を代入して画面に表示させる
```

```
時間1 = datetime.datetime.now()
```

・・・変数「時間1」に現在の時刻を代入する

```
答え = input(" 虫食いのアルファベットは? ")
```

・・・変数「答え」に「虫食いのアルファベットは?」の回答を代入する

```
if 答え == 虫食い:
```

・・・もし変数「答え」の値が変数「虫食い」の値と同じなら

```
    print(" 正解です" ) ・・・「正解です」と表示する
```

```
    時間2 = datetime.datetime.now()
```

・・・変数「時間2」に現在の時刻を代入する

```
    print(str((時間2-時間1).seconds) + " 秒かかりました。" )
```

・・・変数「時間2」-「時間1」の結果を秒数で表示した上で〇秒掛かりましたと表示する

次のページに続く

虫食いアルファベットゲームを作る

```
else:                . . . . . そうでないなら  
    print(" 違います" ) . . . . . 「違います」と表示する
```

単語帳 ～英語も一緒に覚えてしまおう！～

import:取り込む、random:でたらしめ、datetime:日付/時間
、choice:選択、for:～に向かって、in:～の中に、if:もし～なら、
print : 印刷する、input:入力する、now:今、else:～でなければ

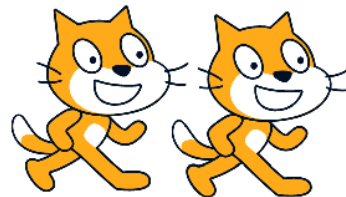
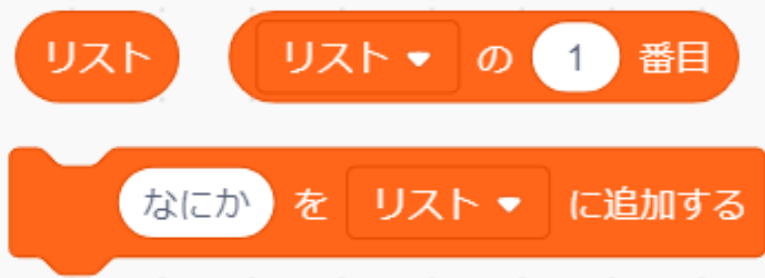
「Run Module」またはF5キーを押してプログラムを実行する。



ABCDEFGHIJKLMNOPQRSTUVWXYZ
抜けているアルファベットは? Y
正解です
20秒かかりました
>>>

復習 & チャレンジ

ここまで習ったことをScratchでもできるかチャレンジしてみよう。
その過程でScratchでできること、Pythonでないといけないことを整理してみよう。
例えば、データの型という概念はscratchにはあったらどうか？



メモ



プログラミング教室の テクノロ

なまえ：