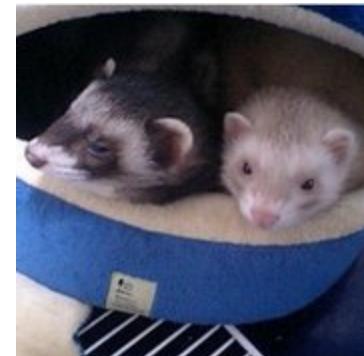


いちばんやさしい Pythonの学び方の作り 方

2017 Nov 8 / みんなのPython勉強会#30
Takanori Suzuki

Who am I?(お前誰よ)

- 鈴木たかのり / Takanori Suzuki
- Twitter: [@takanory](https://twitter.com/takanory)
- Python mini Hack-a-thon(#pyhack)主催
- Pythonボルダリング部(#kabepy)部長



株式会社ビープラウド所属

- <https://www.beproud.jp/>



PyQ

- Pythonに特化したオンライン学習サービス
- <https://pyq.jp/>

The screenshot shows the homepage of the PyQ website. At the top, there is a navigation bar with links for 'PyQ' logo, '特徴・機能' (Features), '料金・プラン' (Pricing), '学習コンテンツ一覧' (Learning Content Catalog), and '企業・学校のお客様へ' (For Enterprises and Schools). On the right side of the header are 'ログイン' (Login) and '学習を始める' (Start Learning) buttons.

The main headline reads '本気でプログラミングを学びたいあなたへ' (For those who seriously want to learn programming) followed by 'Python'.

A central feature is a screenshot of a computer monitor displaying a Python development environment. It shows code in a code editor, a terminal window with command-line output, and a browser window showing a Python-related application interface.

To the right of the screenshot, there is a bulleted list of features:

- ✓ 人気の言語Pythonで、初心者から実務レベルまで
- ✓ ブラウザだけですぐに始められる
- ✓ Web開発、データ分析、機械学習を深く学べる

Below this list, it says 'ライトプラン2,980円/月・スタンダードプラン7,980円/月' (Light Plan 2,980 yen/month - Standard Plan 7,980 yen/month).

At the bottom of the page, there are social sharing buttons for Twitter ('ツイート'), Facebook ('いいね！ 807'), and LinkedIn ('B!ブックマーク').

At the very bottom, there are three footer sections: a satisfaction survey chart ('お客様満足度' 94%), a '推薦教材' (Recommended Materials) section featuring 'PythonED' and '一般社団法人Pythonエンジニア育成推進協会 推薦', and a 'connpass開発・運営' (connpass Development and Operation) section featuring the 'connpass' logo.

connpass

- IT勉強会支援プラットフォーム
- <https://connpass.com/>

The screenshot shows the homepage of connpass, a platform for IT study groups. The top navigation bar includes the connpass logo, event search, category list, new events, and login/signup links. The main title is "エンジニアをつなぐ IT勉強会支援プラットフォーム". Below it are three main sections: "見逃さない、集まりやすい" (Don't miss, easy to gather), "イベントを育てよう" (Grow events), and "IT勉強会にピッタリ" (Perfect for IT study groups). The bottom section highlights "事前決済の手数料は業界最安値" (Lowest industry-wide processing fees), "参加者のためのiPhoneアプリ" (iPhone app for participants), and "多彩なイベント告知" (Multi-channel event notifications). Call-to-action buttons include "connpassをもっと詳しく" and "イベントを作成する".

おすすめグループ



新着イベント

Facebookで最新情報を配信中！

いいね！することで、最新情報を受け取れます



「いいね！」した友達はまだいません

一般社団法人PyCon JP理事

- <https://www.pycon.jp/>



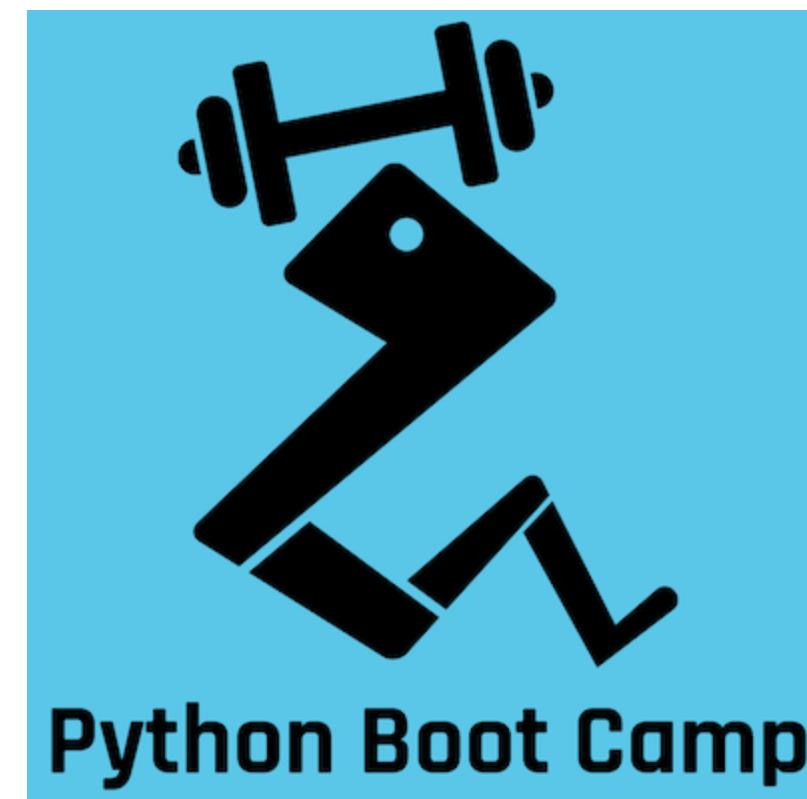
PyCon JP 2017

- 国内最大のPythonイベント
- <https://pycon.jp/2017/ja/>



Python Boot Camp

- 初心者向けPythonチュートリアル
- <https://www.pycon.jp/support/bootcamp.html>



はなすこと

- 「いちばんやさしいPythonの教本」の紹介
 - 執筆チームの紹介
 - この本のつくり方
- プログラミングを学ぶときのポイント

いちばんやさしいPythonの教本



いちばんやさしいPythonの教本

- 272ページ / 2,376円(税込)
- 著者: 鈴木 たかのり、杉谷 弥月
- 2017年8月10日発売
- <https://book.impress.co.jp/books/1116101151>

「いちばんやさしい教本」シリーズ

- <https://book.impress.co.jp/category/series/easybook/>
- 「絶対に挫折しない」をコンセプトとした実用書
- 第一線で活躍する講師のセミナーを受けているように学べる

いちばんやさしい教本の書籍一覧



いちばんやさしいデジタルマーケティングの教本 人気講師が教えるメディアと販促の新しい基礎

ISBN : 9784295002307
発売日 : 2017/9/15
¥1,980+税



いちばんやさしいJimdoの教本 人気講師が教える本格ホームページの作り方

ISBN : 9784295002260
発売日 : 2017/8/25
¥2,200+税



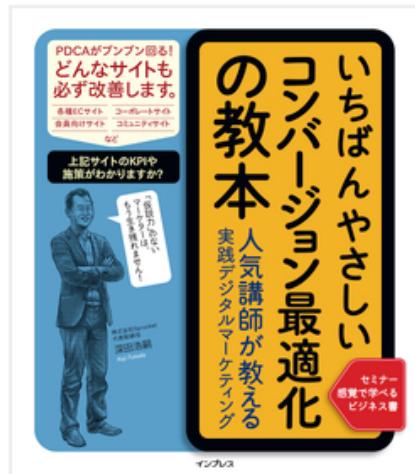
いちばんやさしいPythonの教本 人気講師が教える基礎からサーバサイド開発まで

ISBN : 9784295002086
発売日 : 2017/8/10
¥2,200+税



いちばんやさしいPHPの教本 第2版 人気講師が教える実践Webプログラミング

ISBN : 9784295001249
発売日 : 2017/5/25
¥1,980+税



「解説パート」 + 「手順パート」



このレッスンの ポイント

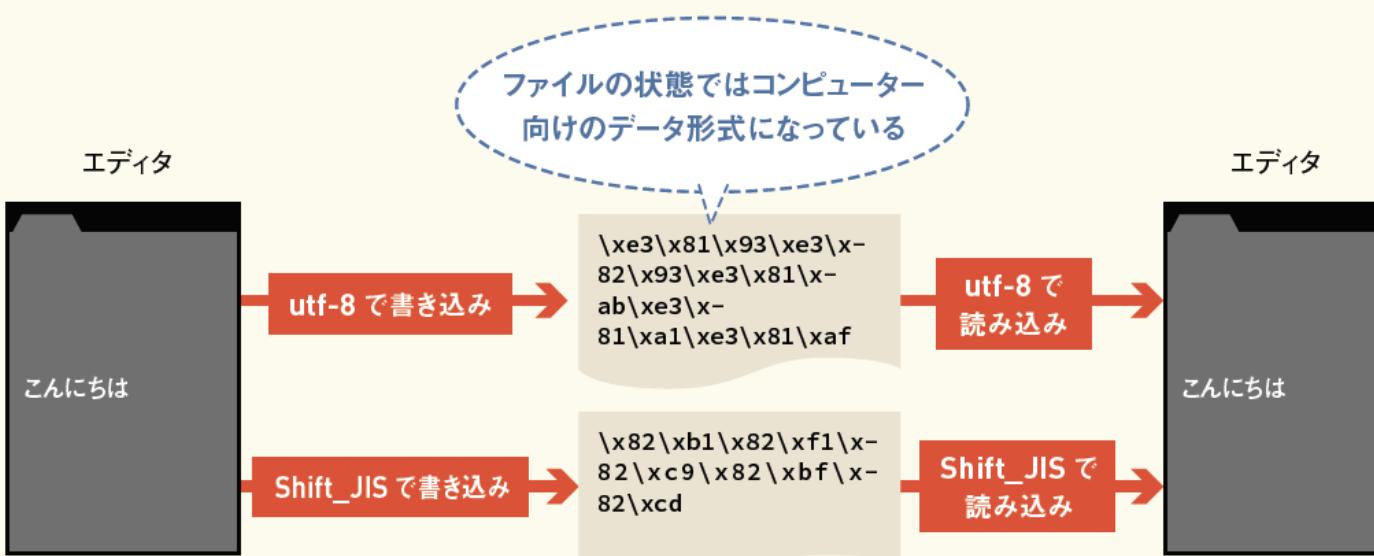
辞書形式で定義したあいさつのデータを、ファイルから読み込むようにします。データ定義を別ファイルとすることで、プログラム自身を書き替えずプログラムの動作が変更できるようにしましょう。日本語のテキストファイルを読み込むときは文字コードに気を付けましょう。

→ テキストファイルの文字コードに注意が必要

日本語を含むテキストファイルから読み込み、書き込みをするときには文字コードに注意が必要です。日本語にはutf-8やWindowsでよく使われるShift_JISなど、いくつかの文字コードが存在します。誤つ

た文字コードを指定してファイルを読み込むと、プログラムでエラーが発生します。本書では文字コードとしてutf-8を使用します。

▶ ファイルと文字コードの関係

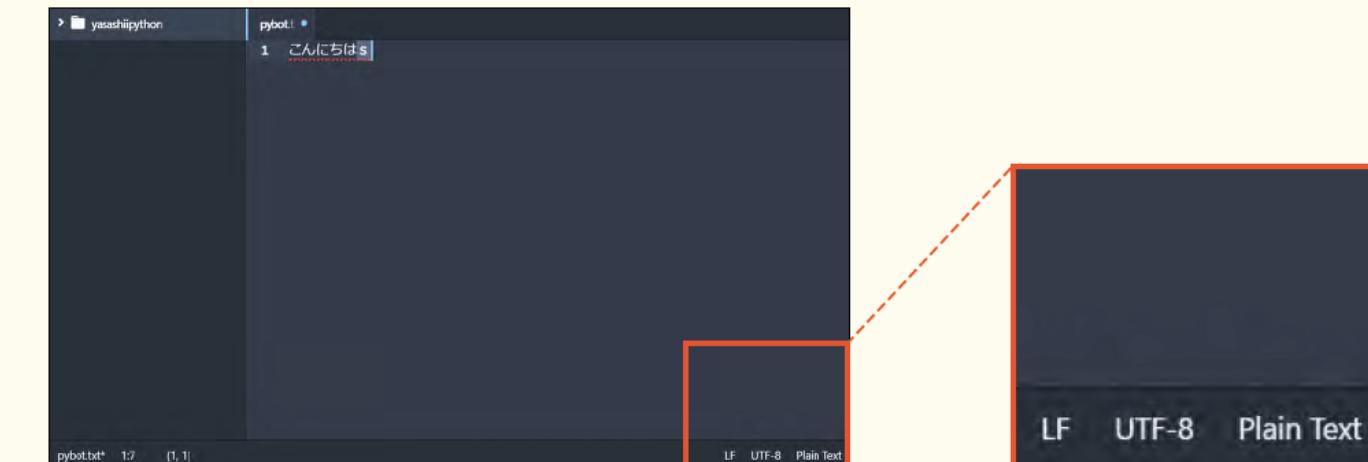


→ 日本語のデータを保存する

日本語の含まれたテキストファイル pybot.txtを作成します。ファイルの中には「こんにちは」とだけ記述します。ファイルの文字コードはutf-8で保存して

ください。Atomエディタでは画面の右下にある文字コード表示で確認できます。標準で utf-8が選ばれているはずです。

▶ Atomエディタの文字コード表示



→ 日本語のデータを読み込む

次にファイルの内容を読み込みます。ファイルを `open()` で開くときに、`encoding='utf-8'` と指定します。日本語の含まれるファイルを読み込む場合は、**文字コード (encoding)** を **ファイル保存時の文字コード** と合わせる必要があります。文字コードを正しく

指定しないと、読み込み時にエラー（UnicodeDecodeError）が発生してプログラムが停止します。そのため、ファイルを開くときにはファイルの文字コードと、open()で指定する文字コードを合わせてください。

▶ 日本語のファイルを読み込む

```
text_file_=open('pybot.txt',_encoding='utf-8')… ファイルの文字コードを指定  
raw_data_=text_file.read()  
text_file.close()  
print(raw_data)……「こんにちは」と表示される
```

日本語のファイルを読み込む、書き込む場合にはencodingを必ず指定しましょう。



○ あいさつデータをファイルから取得する会話botを作る

1 あいさつの定義ファイルを作成する

pybot.txt

あいさつとそれに対する応答のセットとなるテキスト1 さつ、次に応答の文字列を記述します。ファイルを作成します。カンマ区切りで最初にあい

```
001 こんにちは,コンニチハ
002 ありがとう,ドウイタシマシテ
003 さようなら,サヨウナラ
```

1 あいさつと応答のセットを入力

2 あいさつの定義ファイルデータを行ごとの文字列データにする

pybot.py

あいさつの定義ファイルからデータを読み込んで、1 ルの中身をすべて読み込み、改行(\n)で分割して行ごとの文字列データにします。文字コード(utf-8)2 行ごとの文字列のリストを作成します2。

を指定してファイルを開きます1。read()でファイル

```
001 command_file_=open('pybot.txt',_encoding='utf-8')
002 raw_data_=command_file.read()
003 command_file.close()
004 lines_=raw_data.splitlines()
```

1 文字コードを指定

2 行ごとの文字列に分割

3 あいさつの辞書データを生成する

あいさつの辞書データを作成します。最初に空の1 分割します2。分割した文字列の0番目を辞書のキー辞書データの変数を用意します1。1行分の文字列一に、1番目をバリューにして、辞書データに値を(こんにちは,コンニチハ)をカンマで2つの文字列に追加します3。

```
001 command_file_=open('pybot.txt',_encoding='utf-8')
002 raw_data_=command_file.read()
003 command_file.close()
004 lines_=raw_data.splitlines()
005
006 bot_dict_={}
007 for_line_in_lines:
```

1 空の辞書を作成

```
008     word_list_=line.split(',')
009     key_=word_list[0]
010     response_=word_list[1]
011     bot_dict[key]=response
```

- 2 カンマで2つの文字列に分割
- 3 2つの文字列をキーとバリューとして、辞書にセット

3 pybotを実行する

辞書データができれば以降の処理はLesson 34と同じです1。動作も同じです。このプログラムではあいさつの定義ファイルを書き替えることによって、

あいさつのパターンを変更できます。定義ファイルを書き替えて試してみてください。

```
001 command_file_=open('pybot.txt',_encoding='utf-8')
002 raw_data_=command_file.read()
003 command_file.close()
004 lines_=raw_data.splitlines()
005
006 bot_dict_={}
007 for_line_in_lines:
008     word_list_=line.split(',')
009     key_=word_list[0]
010     response_=word_list[1]
011     bot_dict[key]=response
012
013 while_True:
014     command_=input('pybot> ')
015     response_=''
016     for_key_in_bot_dict:
017         if_key_in_command:
018             response_=bot_dict[key]
019             break
020
021     if_not_response:
022         response_='何ヲ言ッテルカ、ワカラナイ'
023     print(response)
024
025     if 'さようなら' in_command:
026         break
```

- 1 Lesson 34と同じプログラムを追加

執筆チーム

- 著者: 鈴木 たかのり(@takanory)、杉谷 弥月
- 社内レビュワー
 - @shimizukawa
 - @haru860
 - @okusama27
- ヘルプコラミスト
 - @hirokiy

書籍の内容

目次

- 1章 Pythonを学ぶ準備をしよう
- 2章 コマンドプロンプトに慣れよう
- 3章 基礎を学びながらプログラムを作成しよう
- 4章 繰り返しと条件分岐を学ぼう
- 5章 辞書とファイルの扱いを学ぼう
- 6章 会話botを作ろう
- 7章 ライブラリを使いこなそう
- 8章 サードパーティ製パッケージを使いこなそう
- 9章 Webアプリケーションを作成しよう
- 10章 さらに知識を身に付けるための学び方

プログラミング入門

プログラムを書いたことがない人向けの内容

- 1章 Pythonを学ぶ準備をしよう
 - コンピューターの仕組み
 - Python、Atomインストール
- 2章 コマンドプロンプトに慣れよう
 - コマンドでのファイル操作
- 3章 基礎を学びながらプログラムを作成しよう
 - 演算、print()関数
 - 変数、str型、int型
 - リスト、タプル

Python言語入門

プログラムを作る上で欠かせない文法を解説

- 4章 繰り返しと条件分岐を学ぼう
 - for文、if文
- 5章 辞書とファイルの扱いを学ぼう
 - 辞書
 - open()関数

会話botによるアプリケーション開発

会話botの開発を通してPythonの機能を使いこなす

- 6章 会話botを作ろう
 - in演算子、break文
 - 関数、組込関数
- 7章 ライブラリを使いこなそう
 - モジュール、標準ライブラリ
 - 例外処理
- 8章 サードパーティ製パッケージを使いこなそう
 - pip、venv
 - requests

はじめてのWebアプリケーション開発

会話botをWebアプリケーション化する。

- 9章 Webアプリケーションを作成しよう
 - Webアプリケーションとは
 - Bottle
 - テンプレート
 - フォーム

次のステップ

さらに学ぶための情報源を紹介。

- 10章 さらに知識を身に付けるための学び方
 - 書籍
 - Webサイト
 - コミュニティ

この本のつくり方

「いちばんやさしい教本」シリーズ

- 「絶対に挫折しない」をコンセプトとした実用書

素早く理解できるように

- 説明していない用語を使わない
- 徐々にできることが増えていく
- 最初にコードとストーリーをレビュー

読みやすい日本語

- できるだけ短い表現にする
- 曖昧な用語を使わない、主語をはっきり
- 代名詞をできるだけ使わない
- 表記ゆれをなくす

次のステップをサポート

- botプログラムに徐々にコマンドを追加
- botアプリケーションをWeb化

この本でできないこと

- リファレンス的な使い方

研修講師の経験をフィードバック

自分の場所がわからない問題

コマンドプロンプトで操作する フォルダを移動しましょう



このレッスンの
ポイント

マウスでのフォルダ操作の際はあまり意識しませんが、コマンドプロンプトで操作をするときはフォルダの構造を正しく理解する必要があります。ここではフォルダ構造と、コマンドプロンプト上で「現在のフォルダ」を移動するためのcdコマンドについて解説します。

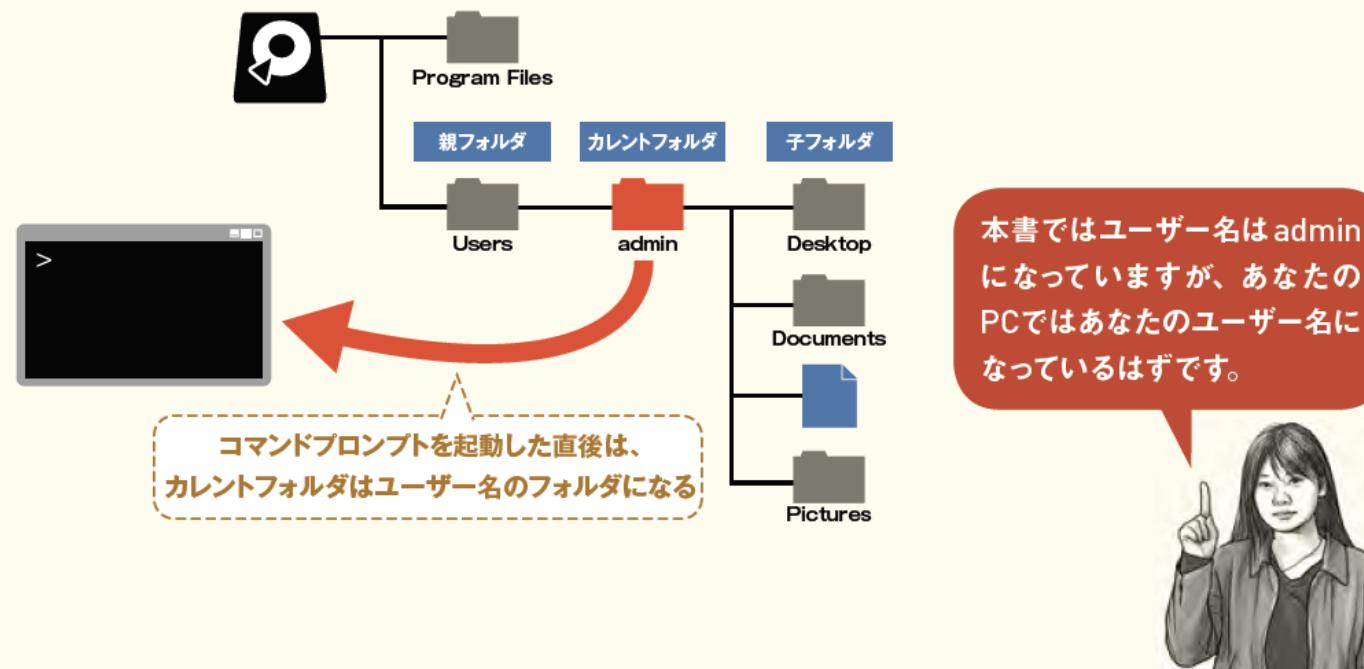
→ フォルダの構造とカレントフォルダ

フォルダとは、PC上でのファイルやフォルダを格納する部屋のようなものです。

フォルダは木の枝のような階層構造になっています。階層構造の中で、上位に位置するフォルダを「親フォルダ」、下位に位置するフォルダを「子フォルダ」といい、現在操作対象のフォルダをカレントフォルダといいます。

Windowsでコマンドプロンプトを起動したときのカレントフォルダは、デフォルトではユーザー名のフォルダ (C:\Users\ユーザー名)になります (macOSでは /Users/ユーザー名)。コマンドプロンプトでファイルを参照したり、操作したりするコマンドを入力するときは、カレントフォルダがどこかを把握している必要があります。

▶ フォルダ概要図

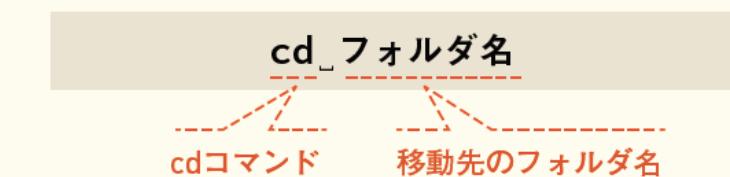


→ カレントフォルダを移動するコマンド

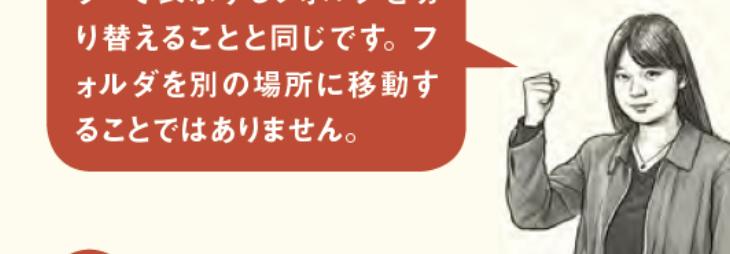
Pythonのプログラムを実行するためには、プログラムのファイルが配置してあるフォルダまで移動する必要があります。

フォルダの階層を移動するには cdコマンドを使用します。cdは Change Directory (ディレクトリを変更

▶ cdコマンド



cdコマンドは、エクスプローラーで表示するフォルダを切り替えることと同じです。フォルダを別の場所に移動することではありません。



→ 相対パスと絶対パス

パスとは、特定のフォルダの位置を表す文字列のことです。パスには「相対パス」と「絶対パス」の2種類があります。相対パスは、カレントフォルダから目的のフォルダやファイルまでの位置を表現します。絶対パスは、一番上のフォルダからの位置を

する)の略です。フォルダのことはOSによってはディレクトリとも呼ばれるため、このような名前になっています。cdコマンドの使い方は以下のように移動したいフォルダ名を指定します。

▶ cdコマンドの実行例

```
カレントフォルダはC:\Users\admin
コマンドプロンプト
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

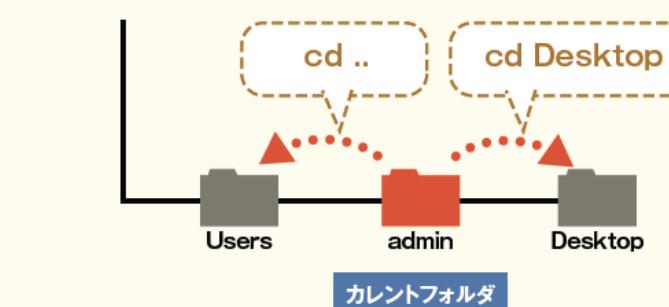
C:\Users\admin>cd Desktop
C:\Users\admin\Desktop>
```

cdコマンドでDesktop
フォルダに移動

カレントフォルダが
C:\Users\admin\Desktop に変わった

表現します。親フォルダに移動したい場合には相対パスで「cd ..」と入力します。1つ前は「..」(Mac OSの場合は「.../」)と表し、たどる階層が増えるたびに「..」と「..」というように増やしていきます。

▶ 相対パスでの移動



機能分割できない問題

→ 命令を組み合わせた手順をアルゴリズムという

コンピューターにしてほしいことを達成させるための一連の手順をアルゴリズムといいます。先ほどの解説で分解した「生まれた年を入力させる」→「生まれ年をもとに干支を特定する計算をする」→「計算結果がxの場合yという干支が選ばれる(x=0のとき

はy=子、x=1のときはy=丑……)」→「干支の名前を表示する」という動作も、アルゴリズムの1つなのです。プログラムを書くということは、これらアルゴリズムを一連の手順にまとめる作業なのです。

「どう組み合わせれば目的を達成できるのか」という問題の解き方をアルゴリズムといいます。

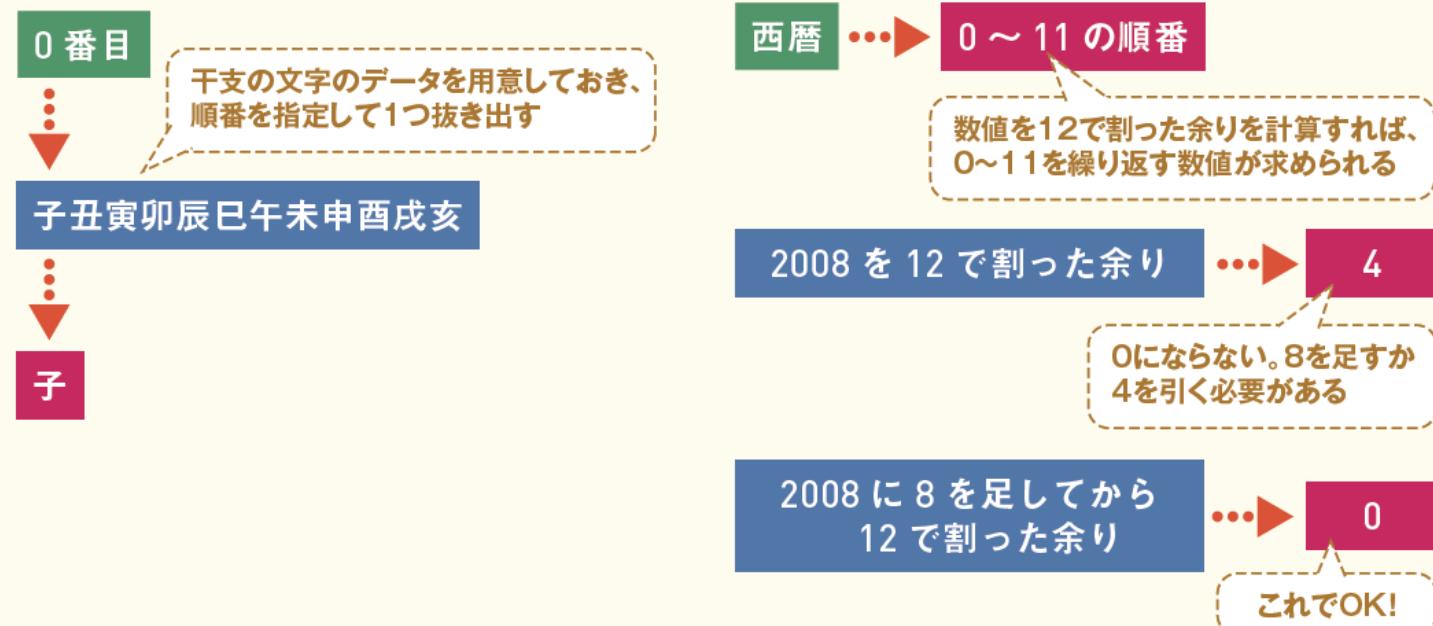


→ 西暦という「数値」から干支という「文字」を求めるには？

このプログラムの中心となる、西暦から干支を求める部分を考えてみましょう。具体的にいうと、例えば1996年、2008年、2020年のいずれかが入力されたら「子」年と表示する方法です。そのためにはまず数値から文字を求める方法を知らないといけませんね。プログラムで、数値から対応する文字を見つければいいのです。

求める場合、「文字に0から番号を割り当てて、番号で文字を取り出す」のが一般的な手法です。今回の場合は、干支の12文字それぞれに0から11までの番号を割り当てるので、1996、2008、2020などの数値が入力されたときに0という順番を計算する方法を見つければいいのです。

▶ 数値から対応する文字を求める

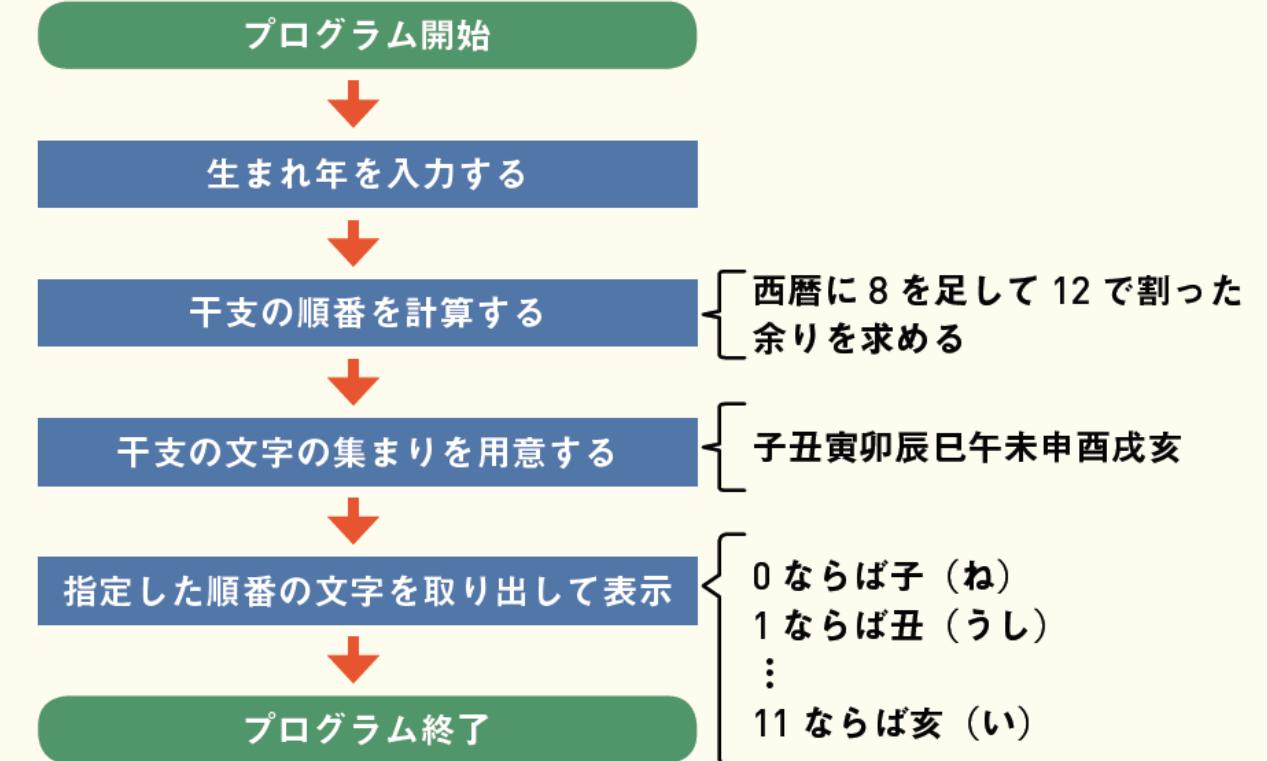


→ アルゴリズムを図で表す

頭の中でプログラムがどういう動きをするのかを想像することは、ある程度プログラムを書くことに慣れていないとすぐにはできないものです。そういうときは、アルゴリズム(プログラム)の流れを図

に起こしましょう。以下のような流れを表す図を「フローチャート」といいます。図を書くことは、これからプログラムを書くにあたっての、考えるときのよき助っ人となってくれるでしょう。

▶ フローチャートの例



この例は流れが一本道でシンプルですが、プログラムの多くはもっと複雑です。悩んだときは一度フローチャートに描き出して整理してみましょう。

👉 ワンポイント フローチャートやアルゴリズムを書くツールは？

プログラムを書きはじめる前に、コンピューターにやらせたいことやアルゴリズムを事前にまとめておくとよいとお話ししました。では、フローチャートやアルゴリズムを書くにはどんなツールを使えばいいのでしょうか？

実はもっともおすすめのツールは紙とペンです。大きなツールは必要ありません。紙とペンでまとめた図はカメラで撮影して保存しましょう。まずは思考をまとめるために、書き出して整理することが重要です。

コードどこ書き換えるのかわからない問題

```

001 open_file_=open('point.txt')
002 raw_data_=open_file.read()
003 open_file.close()
004 point_data_=raw_data.splitlines()
005
006 point_dict_={}
007 for line_in_point_data:
008     student_name_, points_str_=line.split(':')
009     point_dict[student_name]=points_str
    
```

1 空の辞書
2 1行ずつ処理する
3 コロンで分割
4 辞書にデータを追加

4 合計点と平均点を算出する

次はpoint_dictの中にある点数から合計点と平均点を算出します。

合計点と平均点を保存しておくための辞書としてscore_dictを作成します①。for文でpoint_dictを繰り返し処理します。変数student_nameにはキーとなっている氏名が入ります②。point_dict[student_

name]で点数の文字列が取得できるので、カンマで区切って点数のリストを作成します③。

点数のリストから、教科数(subject_number)、合計点(total)と平均点(average)を算出します④。最後に計算した合計点、平均点、教科数をscore_dictに保存します⑤。

```

006 point_dict_={}
007 for line_in_point_data:
008     student_name_, points_str_=line.split(':')
009     point_dict[student_name]=points_str
010
011 score_dict_={}
012 for student_name_in_point_dict:
013     point_list_=point_dict[student_name].split(',')
014     subject_number_=len(point_list)
015     total_=0
016     for point_in_point_list:
017         total_=total+_int(point)
018     average_=total/_subject_number
019     score_dict[student_name]=(total, average, subject_number)
    
```

1 空の辞書
2 点数データを繰り返し処理
3 点数リストを作成
4 点数を計算
5 辞書にデータを追加

5 評価結果を作成する

合計点をもとに、条件分岐で評価結果となる文字列を取得します。

評価を保存するための空の辞書のevaluation_dictを作成します①。直前のfor文で作成した合計点、平均点などが入った辞書を繰り返し処理します②。辞書のバリューにはタプルで合計点、平均点、教

科数が入っているので、必要なデータを取り出します③。教科数から評価のための基準となる点数を計算し、8割以上なら評価を'Excellent!'、6.5割以上なら'Good'とします。どちらの条件も満たさない場合は'Bad'とします④。最後に評価結果を辞書に追加します⑤。

```

016     for point_in_point_list:
017         total_=total+_int(point)
018         average_=total/_subject_number
019         score_dict[student_name]=(total, average)
020
021 evaluation_dict_={}
022 for student_name_in_score_dict:
023     score_data_=score_dict[student_name]
024     total_=score_data[0]
025     average_=score_data[1]
026     subject_number_=score_data[2]
    
```

1 空の辞書を作成
2 合計点データを繰り返し処理
3 合計点などを取り出す
4 評価文字列を作成
5 辞書にデータを追加

```

027
028     excellent_=subject_number_*100*_0.8
029     good_=subject_number_*100*_0.65
030     if total_>=excellent:
031         evaluation_='Excellent!'
032     elif total_>=good:
033         evaluation_='Good'
034     else:
035         evaluation_='Bad'
036     evaluation_dict[student_name]=evaluation
    
```

スライスわかりにくくい問題

リスト、タプル、文字列からまとめてデータを取り出しましょう



このレッスンの
ポイント

randomモジュールを使った「選ぶ」コマンドでは「選ぶ」というコマンド自体も抽出の対象になっていました。対象とする範囲を絞り込むには、スライスという機能を使います。インデックス指定と似た書き方で、リストなどから指定した範囲を取り出せます。

→ インデックス指定では1つのデータしか取り出せない

Lesson 40で追加した「選ぶ」コマンドでは、「選ぶカレー ラーメン」と書いた場合に「選ぶ」という言葉自体もランダムに選択される対象となることには気づいていたでしょうか? 「選ぶ」というキーワードをランダムに選択する対象からはずす必要があります。Pythonには「リストの1番目から最後まで」を取り出すといったときに便利なスライスという機能が

あります。ここではその使い方を学びましょう。スライスの説明の前に、Chapter 3で説明した、インデックスによる値の取り出し方をおさらいしておきましょう。先頭を0番目とした順番を指定すると、リスト、タプル、文字列などから特定の値を取得できるのでしたね。

▶ インデックス指定による値の取り出し

```
eto_list = ['子', '丑', '寅', '卯', '辰', '巳', '午', '未', '申', '酉', '戌', '亥']
print(eto_list[0]) ..... 0番目の要素の「子」が表示される
print(eto_list[1]) ..... 「丑」が表示される
print(eto_list[11]) ..... 「亥」が表示される
```

```
blood_type = ('A', 'B', 'O', 'AB') ..... 血液型のタプルを定義
print(blood_type[3]) ..... 「AB」が表示される
```

```
weekday_text = '月火水木金土日'
print(weekday_text[1]) ..... 「火」が表示される
```



スライスを使って範囲内のデータを取り出す

スライスは、文字列、リスト、タプルなどの連続したデータから、まとまった範囲のデータを取り出す方法です。角カッコの中に[開始位置:終了位置]の形式でインデックスを指定します。取り出されるのは、終了位置の1つ前までという点に注意してください。

なお、開始位置と終了位置は省略が可能で、省略時は、最初からまたは最後までが取り出されます。リストからスライスで値を取り出した場合、取り出した値もリストとなります。

▶ スライスの書き方



▶ スライスの実行例

```
eto_list = ['子', '丑', '寅', '卯', '辰', '巳', '午', '未', '申', '酉', '戌', '亥']
slice1 = eto_list[2:5] ..... ['寅', '卯', '辰']を取得する
slice2 = eto_list[:4] ..... 0番目(子)から3番目(卯)までのリストを取得する
slice3 = eto_list[6:] ..... 6番目(午)から最後(亥)までのリストを取得する
```

▶ スライス指定のイメージ



スライスは最初は少しわかりにくいかかもしれません。図の矢印で示したところが、スライスでデータが取り出される範囲です。



プログラミングを学ぶときのポイント

書籍は自分に合うものを

- 大きい本屋に行ってパラパラとめくってみる
- 自分がなんとなく気に入ったものを買う
- リファレンスとしては使えないと割り切る

使うものを作ろう

- 繰り返している作業
- 面倒なことをプログラムにやらせよう

オレ凄えって思う

- 誰かに褒められるともっとうれしい



Takanori Suzuki 16:37

\$manual



PyCon JP Bot BOT 16:37

PyCon JP 運営マニュアル <http://manual.pycon.jp/>



Takanori Suzuki 16:37

SAR-700



PyCon JP Bot BOT 16:37

[SAR-700](#) bot: plugin作成

担当者

Takanori Suzuki

ステータス

クローズ



Takanori Suzuki 16:37

\$jira 検索 bot



PyCon JP Bot BOT 16:37

「bot」の検索結果(オープンのみ)([JIRAで見る](#))

- [SAR-691](#) bot: drive plugin 作成(進行中)
- [SAR-716](#) bot: plusplus plugin実装(進行中)



Takanori Suzuki 16:37

2 ** 16



PyCon JP Bot BOT 16:37

65536



Takanori Suzuki 16:38

\$天気 銚路



PyCon JP Bot BOT 16:38

[銚路の天気予報](#)

今日は 曇り

明日は 曇時々雨 最低気温6°C 最高気温12°C



Takanori Suzuki 16:38

\$choice spam ham eggs



PyCon JP Bot BOT 16:38

eggs

PyQもよろしくね!

- Pythonに特化したオンライン学習サービス
- <https://pyq.jp/>

The screenshot shows the homepage of the PyQ website. At the top, there is a navigation bar with the PyQ logo, links for '特徴・機能', '料金・プラン', '学習コンテンツ一覧', and '企業・学校のお客様へ', and buttons for 'ログイン' and '学習を始める'. The main heading is '本気でプログラミングを学びたいあなたへ' with a 'Python' subtitle. Below the heading is a large image of a person's head with a brain-like hexagonal pattern. In the center, there is a screenshot of a computer monitor displaying a Python code editor with code related to Django and PostgreSQL. To the right of the monitor, there is a list of features: '人気の言語Pythonで、初心者から実務レベルまで', 'ブラウザだけですぐに始められる', and 'Web開発、データ分析、機械学習を深く学べる'. Below this is a section for the 'ライトプラン' (Light Plan) at 2,980円/月 and the 'スタンダードプラン' (Standard Plan) at 7,980円/月. A note states '※キーボードが必要です。また、スマートフォンには対応していません。'. At the bottom, there are social sharing buttons for Twitter, Facebook, and LinkedIn, along with a 'B!ブックマーク' button. The footer contains a satisfaction survey chart showing 94% satisfaction, logos for 'PythonED' and 'connpass', and text indicating connpass is the developer and operator.

お客様満足度
期待通り以上 94%

推薦教材
一般社団法人Pythonエンジニア育成推進協会 推薦

connpass開発・運営
connpassを開発・運営しているBeProudのノウハウ

ありがとうございました

Question?