

ステップ30

留^{りゅう}学^{がく}生^{せい}のための

Python [基^き礎^そ編^{へん}]

ワークブック



はじめに

本書はPythonの基礎を習得するための書籍です。

すでに他のPython入門書籍・Webサイト・ビデオなどで学習していて、それらの学習内容の定着具合を確かめたい方にも活用いただける教材です。自身の習熟度合い・理解度を客観的に見ることは難しく、プログラムを書いたり読んだり、レビューをもらったり、それらを繰り返して自分に足りない部分を学んでいくのが一般的です。本書は演習問題が付属したワークブック形式になっており、習熟度・理解度の確認が容易になっています。

最近では機械学習が注目されたこともあり、簡単な画像認識などを扱うPythonの入門書籍もありますが、本書ではPythonの基礎的な内容を中心に学びます。どのような分野に進んでも本書で学んだ内容が土台となるはずですよ。

■ 対象読者

プログラミングを初めて学ぶ高校生を対象読者に想定しています。プログラミングを初めて学ぶ留学生の皆さんにも役に立つと思います。

■ 本書の構成

本書の構成は次のようになっています。

- 要点として、記憶してほしい用語や概念を示します。
- ワーク[基礎]として、簡単な問題を出題します。
- ワーク[応用]として、少し高度な問題を出題します。

ワークの中には、複数の答えがある問題もあります。学校のように多くの方が同時に学習する環境では、皆さんに様々な解答を出していただき、他人の発想を学んだり、その解答が適切かを議論していただければと思います。

Step 01	Python とは	7
	1.1 Python の歴史.....7	1.2 Python の特徴.....7
Step 02	四則演算	9
	2.1 Python での四則演算 + a9	2.2 ZeroDivisionError.....9
	2.3 演算の優先順位.....10	
Step 03	変数	12
	3.1 変数の使い方.....12	3.2 変数名に使える名前.....12
	3.3 変数の応用①.....13	3.4 変数の応用②.....14
Step 04	組み込み型 文字列①	16
	4.1 組み込み型とは.....16	4.2 文字列を扱う.....16
	4.3 組み込み関数.....17	
Step 05	組み込み型 文字列②	20
	5.1 文字列のメソッド.....20	5.2 インデクシングとスライシング.....21
Step 06	組み込み型 リスト①	24
	6.1 リストとは.....24	6.2 リストのメソッドと、組み込み関数.....25
Step 07	組み込み型 リスト②	28
	7.1 ミュータブル.....28	7.2 ミュータブルな型の注意点.....29
Step 08	組み込み型 タプル	31
	8.1 タプルとは.....31	8.2 リストとの違い.....31
	8.3 アンパック.....32	
Step 09	組み込み型 辞書	35
	9.1 辞書とは.....35	9.2 辞書のメソッド.....36
Step 10	組み込み型 集合	39
	10.1 集合とは.....39	10.2 集合のメソッド.....40
Step 11	組み込み型 まとめ	42
	11.1 bool 型.....42	11.2 NoneType 型.....42
	11.3 組み込み型まとめ.....42	

Step 12	じょうけん ぶん き ぶん 条件分岐 if文①	44
	12.1 if文の基本44	12.2 インデント
	4445
Step 13	じょうけん ぶん き ぶん 条件分岐 if文②	48
	13.1 比較演算48	13.2 ブール演算
	4849
	13.3 暗黙のTrue、False50	
Step 14	く かえ ぶん 繰り返し for文①	53
	14.1 for文の基本53	14.2 breakとelse
	5355
Step 15	く かえ ぶん 繰り返し for文②	58
	15.1 指定回数の繰り返し58	15.2 rangeオブジェクト
	5859
	15.3 enumerateとzip60	
Step 16	く かえ ぶん 繰り返し while文	64
	16.1 while文とは64	
Step 17	かんすう 関数①	67
	17.1 関数とは67	17.2 引数
	6768
	17.3 戻り値70	
Step 18	かんすう 関数②	72
	18.1 デフォルト引数72	18.2 可変長位置引数
	7273
	18.3 可変長キーワード引数74	18.4 キーワード専用引数
	7475
Step 19	かんすう 関数③	77
	19.1 スコープ77	19.2 ミュータブルな型の注意点
	77	再び
Step 20	クラス①	82
	20.1 クラスとは82	20.2 self
	8283
Step 21	クラス②	87
	21.1 継承とは87	21.2 オーバーライド
	8788
	21.3 superで親のメソッドを呼ぶ89	
Step 22	クラス③	92
	22.1 クラスの属性92	22.2 インスタンス属性との区別
	9293
Step 23	モジュール	96
	23.1 モジュールとは96	23.2 モジュールの直接実行
	9697
Step 24	パッケージ	100
	24.1 パッケージとは100	24.2 __init__.py
	100101

Step 25	<small>にゆうしゅつりよく</small> 入出力	104
	25.1 ファイルの書き込み.....104	25.2 ファイルの読み込み.....105
	25.3 他のモード.....105	25.4 エンコーディング.....105
Step 26	<small>れいがい</small> 例外	109
	26.1 例外を捕まえる.....109	26.2 finallyとelse.....110
Step 27	ライブラリ	113
	27.1 標準ライブラリとは.....113	27.2 サードパーティ製ライブラリ.....114
Step 28	<small>めいろ</small> 迷路アプリケーション①	116
	28.1 迷路アプリケーションの概要.....116	28.2 プロトタイプを作る.....117
Step 29	<small>めいろ</small> 迷路アプリケーション②	122
	29.1 移動処理を実装する.....122	29.2 座標のチェック処理.....123
Step 30	<small>めいろ</small> 迷路アプリケーション③	128
	30.1 クラスを使う.....128	
<small>ふろく</small> 付録A	Pythonのインストール	133
	A.1 Windows.....133	A.2 Mac.....134
<small>ふろく</small> 付録B	<small>たいわ</small> 対話モードで実行する	135
	B.1 対話モードに入る.....135	B.2 他のバージョンのPythonを使う.....135
	B.3 対話モードを試す.....135	
<small>ふろく</small> 付録C	<small>じっこう</small> Pythonスクリプトの実行	136
	C.1 Pythonスクリプトの実行方法.....136	C.2 エディタ・IDEの紹介.....136
	C.3 IDLEの開き方と使い方.....137	
<small>さくいん</small> 索引		141

※ワークの解答は、以下のWebページからダウンロードできます。

http://www.cutt.jp/books/python_work_837/



Pythonとは

学習のはじめに、Pythonというプログラミング言語について説明します。「Pythonとは何か?」、「Pythonの何がよいのか?」と質問をされたときに答えられるようにしておきましょう。

ようてん

要点

1.1 Pythonの歴史

1989年の12月、オランダ人のガイド・ヴァンロッサム (Guido van Rossum) はクリスマス休暇の暇つぶしとしてプログラミング言語の開発を始めました。これがPythonです。その後1991年2月にalt.sourcesニュースグループ上でバージョン0.9.0が一般公開され、今ではバージョン3の後半です。4大P言語 (Perl, Python, PHP, Ruby) の中では2番目に古く、意外に感じるかもしれませんが、JavaやC#よりも年上になります。1957年のFORTRAN (フォートラン) から始まるプログラミング言語の歴史から見ると、90年前後というのは最近に感じます。しかし、よく使われているメジャーな言語の中ではそれなりに古く、歴史のある言語といえます。

余談ですが、「Python」という名前の由来は、イギリスのコメディ番組「空飛ぶモンティ・パイソン」から取っています。

1.2 Pythonの特徴

Pythonの特徴は多くありますが、代表的なものは可読性・生産性・汎用性の3つです。

■可読性

可読性とはプログラムの読みやすさのことです。Pythonは「実行可能な疑似コード」と表現されるほど、自然言語に近い言語です。他言語でよく使われる丸括弧などの記号は、少なくなるように設計されています。コードが読みやすくと、それだけ学びやすく、覚えやすく、上達が早くなります。他人との共同作業もはかどります。

■生産性

ある時間内や行数で、「どれだけのプログラムや処理が作れるか?」を生産性といいます。Pythonはスクリプト言語と呼ばれるグループの仲間です。スクリプト言語はプログラムの実行が簡単で、少ないコードでたくさんのことができます。CやJavaといった言語に比べるとコードの量が半分以上以下になることも珍しくありません。

■ 汎用性

Pythonは幅広い用途に使える言語であり、汎用的な言語といわれています。Webアプリケーションや科学分野をはじめ、多くの分野で実際に利用されています。汎用目的に作られた言語は数多くありますが、ある分野での開発が現実的ではない言語もあります。たとえば、Webでよく使われているPHPでスマホアプリを開発するのは、(少なくとも今は)現実的ではありません。Pythonにも不得意なこと、または他言語の方が得意な分野はありますが、総合的にはとても汎用的な言語です。

プログラミング言語に最も必要なものは、コミュニティです。どんなに素晴らしい言語であっても、利用者が少なければ今後の発展は見込めず、情報の取得に苦労するでしょう。Pythonはどうでしょうか? ご安心ください、とてもホットな言語です。ある言語がどれだけ人気があるかの指標として、以下の3サイトはよく参考にされます。どのサイトでもPythonは高い位置に存在することがわかります。

<http://pypl.github.io/PYPL.html>

<https://www.tiobe.com/tiobe-index/>

<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017>

ワーク

きそ

基礎①

次のうち、Pythonの説明として正しいものを1つ選びましょう。わからない語句は、Google検索を活用しても構いません。

1. Pythonは生産性が低く、読みづらい言語だ。
2. Pythonはオブジェクト指向をサポートしていない。
3. Pythonは可読性、生産性、汎用性に優れた静的言語だ。
4. Pythonは可読性、生産性、汎用性に優れたスクリプト言語だ。

Step 02

しそくえんざん 四則演算

Pythonに付属している対話モードは、簡単な電卓としても使えます。対話モードで四則演算 + α を試してみましょう。安物の電卓より、Pythonでの数値計算のほうが便利です。なお、Pythonのインストールは付録A、対話モードの使い方は付録Bを見てください。

要点

2.1 Pythonでの四則演算 + α

足し算、引き算、掛け算、割り算のことを四則演算といいます。Pythonは四則演算だけでなく、様々な計算ができます。実行結果2.1.1で確認しましょう。

実行結果2.1.1 対話モードでの実行例

```
>>> 1 + 2
3
>>> 3 - 2
1
>>> 5 * 2
10
>>> 5 ** 2
25
>>> 4 / 2
2.0
>>> 4 // 2
2
>>> 10 % 3
1
```

掛け算の記号は \times ではなく、* (アスタリスク) になります。割り算は \div ではなく、/ (スラッシュ) になります。/ による割り算は、小数点以下も表示されます。小数点部分を切り捨てたい場合は、// とします。*^{*} で指数の計算ができ、% (パーセント) 記号で剰余 (割った余り) を求めることができます。

2.2 ZeroDivisionError

数学では、0で割ることは許されない行為です。Pythonでは0で割ったときに実行結果2.2.1のように表示されます。

じっごうけっか
実行結果2.2.1

```
>>> 1 / 0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
```

これは例外と呼ばれるもので、Pythonで実行中に検出されたエラーのことです。例外を扱う方法は後ほど詳しく説明します。今は「エラーがこのように表示される」と覚えておいてください。

えんざん ゆうせんじゆんい
2.3 演算の優先順位

演算には優先順位があります。それは数学の優先順位と同じです。実行結果2.3.1では、まず100**0の指数が計算され、「掛け算・割り算」の後に「足し算・引き算」が計算されます。通常の数学における優先順位と同じです。

じっごうけっか
実行結果2.3.1

```
>>> 10 + 20 - 10 / 5 * 100 ** 0
28.0
```

実行結果2.3.3では、丸括弧をつけてみました。丸括弧内の演算が優先されます。

じっごうけっか
実行結果2.3.2

```
>>> 10 + (20 - 10) / 5 * 100 ** 0
12.0
```

ワーク

きそ
基礎①

つぎ
次のうち、ZeroDivisionErrorとなるものを1つ選びましょう。

- 1. >>> 5 - 0
- 2. >>> 5 * 0
- 3. >>> 5 / 0
- 4. >>> 6 ** 0

きそ
基礎②

「+」「-」「*」「/」の記号を1つずつ使い、括弧の中を埋めましょう。

```
>>> 10 ( ) 5
15
>>> -10 ( ) 10
-20
>>> 10 ( ) 3 ( ) 5
6.0
```

おうよう
応用

「%」「//」「**」の記号を1つずつ使い、括弧の中を埋めましょう。

```
>>> 10 ( ) 3
3
>>> 10 ( ) 3
1
>>> 5 ( ) 2
25
```