Mynavi Advanced Library

恋するプログラム

Rubyでつくる人工無脳

秋山智俊 著

ほんとに無脳？

人間とプログラムのコラボレーション。これって恋？

ネットワークにはいろんな人工無脳がいます。
彼らは人間とおしゃべりをして、笑わせたり感心させたりむかつかせたりしています。
お遊びプログラム、人工知能までにはいたらないオモチャ。
ではあるんですが、これがなかなかかわいいオモチャなのです。
本書ではオブジェクト指向スクリプト言語Rubyを使って人工無脳を作っています。
Rubyプログラミングの入門書としても最適です。

マイナビ

恋するプログラム

Rubyでつくる人工無脳

秋山智俊 著

マイナビ

Mynavi Advanced Library

●本書は
「恋するプログラム」（2005年4月刊行）
の電子版/オンデマンド版になります。

●誤植の修正だけを行い、できるだけ当時のまま『**復刻**』しています。本書を読み進めるにあたり、以下の点について留意いただき、ご理解・ご了承いただきますようお願いいたします。

●初版発刊当初付属していたCD-ROMの収録内容をそのままzip形式で圧縮したファイルを、下記の本書サポートサイトからダウンロード可能です。お使いのパソコンで解凍してごらんください。
http://book.mynavi.jp/support/pc/5346/

●Chapter 9で使用する「Google Web API」は復刻版発刊時点（2014年10月）ではサービスの提供が行われておらず、**記事のままでは再現できません**。内容の再現・実装については、ご自身のご判断で行ってください。

●本文中の内容は、Windows XPをベースに95,98,Me,2000で学習できるように書かれていますが、Chapter 9以外の内容はWindows 7/8上でも動作することを確認いたしました（Windows 8にはスタートメニューがありませんためソフトウェア起動方法などの説明箇所は適宜読み替えてください）。

●本書中の情報は、基本的に書籍執筆段階のものです。本書に登場するソフトウェアのバージョン、WebサイトのURL、製品のスペックなどの仕様や情報は、すべてその原稿執筆時点（2005年5月）のものです。執筆以降に変更されている可能性がありますので、ご了承ください。

・本書中に掲載している画面イメージは特定の設定に基づいた環境にて再現される一例です。ハードウェアやソフトウェアの環境によっては必ずしも本書通りの画面にならない場合があります。あらかじめご了承ください。
・本書に記載された内容は、情報の提供のみを目的としております。したがって、本書を用いての運用はすべてお客様自身の責任と判断において行ってください。
・本書の制作にあたっては正確な記述につとめましたが、著者や出版社のいずれも、本書の内容に関してなんらかの保証をするものではなく、内容に関するいかなる運用結果についてもいっさいの責任を負いません。あらかじめご了承ください。
・本書中の会社名や商品名は、該当する各社の商標または登録商標です。本書中では™および®マークは省略させていただいております。

はじめに

「人工無脳」ってご存知ですか？

ネットワークにはいろんな人工無脳がいます。彼らはチャットや掲示板、インスタントメッセンジャーで人間とおしゃべりをして、笑わせたり感心させたりむかつかせたりしています。日記やブログを書いて、トラックバックをしてくる人工無脳もいます。

つまりはお遊びプログラム。人工知能までにはいたらないオモチャ。ではあるんですが、これがなかなかかわいいのです。しかも基本的なしくみは単純なので、それなりのものを思いのほか簡単に作ることができます。

本書は人工無脳を作ってみようという本です。まずは「人工無脳ってなに？」というところから始め、単純なサンプルプログラムを作り、いくつかのテクニックを盛り込みながら、それを次第に複雑なものへと進化させていきます。

紹介するサンプルプログラムについては、ほぼすべての内容を説明しています。手法を羅列しただけのものにならないよう、とくに「なぜそう作るのか？」ということをできるだけ省かずに書いたつもりです。「やりたいこと」から実装への道筋を理解できれば自分なりに改良し、アイディアを試してみることも容易になるでしょう。

執筆しながら感じたのですが、人工無脳の製作はプログラミング入門のテーマとしてかなりナイスなのではないでしょうか。基本的な文字列の入出力から始まり、文字列処理、ファイルアクセス、正規表現によるパターンマッチ、GUI、ネットワークプログラミングと、プログラミングを身に付ける上で「押さえておきたいツボどころ」のかなりの部分をカバーしています。やることが多くて大変そうですが、本書ではまつもとゆきひろ氏開発のオブジェクト指向スクリプト言語Ruby（http://www.ruby-lang.org/ja/）を使っており、エレガントかつパワフルなRubyのおかげで、コンパクトかつわかりやすいコードに収まっています。Rubyプログラミングの入門書としても読んでいただけるのではないでしょうか。

それでも、本書で紹介しているワザの数々はごくごく基本的なものです。中身を自分でイジって、さらにわけのわからないことを言わせて楽しむのが人工無脳のほんとうのおもしろさです。

人工無脳プログラミングをお楽しみください。

本書の構成

1章と2章では「人工無脳ってなに？」ということについて明らかにし、続けてRubyによるプログラミングの準備と基礎知識の解説を行います。Rubyプログラミングの雰囲気をざっくりつかんでください。

3章ではもっとも基本的な人工無脳を作ります。ろくな会話はできませんが、ここで用意するプログラムの枠組が以降の人工無脳でもそのまま使われることになるので、その意味からするともっとも重要な人工無脳でもあります。

4章から具体的な人工無脳製作が始まります。4章では3章で作った人工無脳にGUIの殻をかぶせます。アニメーションもここで導入されます。

5章では辞書を使って応答のバリエーションを増やします。単純にランダム選択するためのランダム辞書と、ユーザ発言のあるパターンに反応して応答するパターン辞書の2種類を作ってみます。

6章では感情を扱います。「感情をプログラムする」とはどういうことでしょうか。ごくシンプルな感情モデルを組み込み、表情のアニメーションパターンを増やします。

7章では人工無脳に学習させてみます。ユーザ発言をそのまま学習するという方法から、ユーザ発言を形態素解析により分析しパターンを取り出したりテンプレートとして学習したり、ということまで挑戦します。

8章で取り上げるマルコフ辞書は、人工無脳に独自の文章を作らせてみようという試みです。形態素解析でバラバラになった状態で学習させ、そこから単語を1つずつつないでいって1つの文章を作ります。

9章ではGoogleを使うことを人工無脳に教えます。これにより、学習対象をインターネット全域にまで広げることができます。また、ここまで作ってきたサンプル人工無脳の今後の課題や、やってみるとおもしろそうなテーマについて少し触れています。

本書ではRubyに馴染みのない方でも読んでいただけるように、Rubyプログラミングについてもできるだけ解説するようにしていますが、カバーしきれない部分については本書の配布ファイルに収録しているRubyのリファレンスマニュアルにお任せしています。

本書の対象読者

本書は人工無脳についての本ですが、「Rubyに興味がある」という方にもRubyプログラミングの入門書として読んでいただけると思います。VisualBasicやPerlなど他のプログラミング言語をちょっとやってみたけどいまいち馴染めなかったという方、Rubyはおすすめですよ。

「プログラミング自体が初めて」という方でもなんとかなるように書いたつもりです。サンプルプログラムはほんとうにシンプルなところから始め、ギアもゆっくり上げていきます。Windowsの基本的な操作がわかっていてコマンドプロンプトにアレルギーがない、という方なら大丈夫です。

そしてもちろん、人工無脳に興味がある方。「あのWebのキャラクターって人工無脳らしいけど、人工無脳ってなに？」という方や、「人工無脳って作ってみたいけどなんか難しそう」という方に本書をおすすめします。

CONTENTS

CHAPTER 1
はじめの2,3歩

1-1 人工無脳ってなに？ 10
1-2 ActiveScriptRubyのインストール 12

CHAPTER 2
湯けむりRubyひとめぐりの旅

2-1 irb 16
2-2 最初のサンプル 18
2-3 オブジェクト 20
2-4 繰り返し 22
2-5 変数と条件 24
2-6 配列 28
2-7 制御構造をもう1つ 31
2-8 ハッシュ 33
2-9 オリジナルメソッド 35
2-10 オリジナルオブジェクト 37
2-11 オリジナルオブジェクトを使う 40

CHAPTER 3
ほんとに無能

3-1 もう一度「人工無脳ってなに？」 44
3-2 とはいえ… ──応答クラスの分離── 46
3-3 最初の人工無脳　proto 48
3-4 別のResponder 55

CHAPTER 4
あこがれのGUI

4-1 VisualuRuby計画（仮称）でGUI.....62
4-2 コントロールとイベントハンドラ.....65
4-3 ノビィ.....68
4-4 Responderをランダムに切り換える.....79
補足 NobyCanvas（nobycanvas.rb）について.....81

CHAPTER 5
辞書を片手に

5-1 辞書はファイルに.....88
5-2 パターンに反応する.....92
5-3 人工無脳のための正規表現.....94
5-4 PatternResponderの作成.....100

CHAPTER 6
感情コントロールの魔術師

6-1 「しゃべる」パターンを追加する.....110
6-2 ノビィの感情モデル.....114
6-3 感情モデルの実装 ――Emotionクラス――.....117
6-4 感情モデルの実装 ――DictionaryクラスとPatternItemクラス――...119
6-5 感情モデルの実装 ――Responderクラス、PatternResponderクラス、Unmoクラス――.....126
6-6 感情モデルの実装 ――表情の変化――.....129

CHAPTER 7
学習のススメ

7-1 まるごと覚える.....134
7-2 形態素解析.....140

7-3　キーワードで覚える……149
7-4　テンプレートとして覚える……158

CHAPTER 8
文章を作り出す

8-1　マルコフモデル……168
8-2　マルコフ辞書の実装……172
8-3　ノビィ、マルコフ辞書を使う……182

CHAPTER 9
ノビィ、ネットワークにつながる

9-1　Googleと仲良くなろう……190
9-2　ググるプログラム……196
9-3　ノビィ、Webにでかける……206
9-4　これからのノビィ……211

付録……215
索引……220

CHAPTER 1
はじめの2,3歩

「人工無脳」ってなんでしょう？　「人工知能」という言葉には馴染みがありますが、それとどう違うのでしょうか？　本書のテーマは「人工無能を作る」ですが、なにを作るかはっきりしないのでは作りようがありません。「人工無能ってなに？」「なにをするためのもの？」ということに対して、まずはここで筆者のスタンスを明らかにしようと思います。そのあとRubyをインストールして、人工無能を作るための準備を始めましょう。

CHAPTER 1－1

人工無脳ってなに？

「人工無脳（無能）」とは人間とおしゃべりをしてくれるプログラム、つまり「会話プログラム」のことです。「会話プログラム」というと人工知能を連想しますが、ではなぜ「知能」ではなく「無脳（無能）」なのでしょうか？

人工知能が目指しているのは（究極的には）「知能」そのものです。人工知能は、その知能の活動の結果として会話を行います。相手の発言内容を理解していなければなりませんし、人工知能自身も自分がなにを言っているかわかった上で発言します。人工知能にとって重要なのは会話という行為そのものではなく、意味の理解や推論などの会話を支える「目に見えない」知的活動です。

一方、人工無脳は「会話」という行為そのものを目指します。それも「おもしろい会話」「ユーザを楽しませる会話」です。そこに「知的活動」は必要ありません。いや、そのほうが「おもしろい会話」ができるのであればやるべきでしょうが、意味の理解や推論ができてもユーザを飽きさせてしまったら、それは人工無脳としては失敗です。人工無脳にとっての問題は知的活動の有無ではなく、いかに「オモシロ発言」をしてユーザを楽しませるかです。

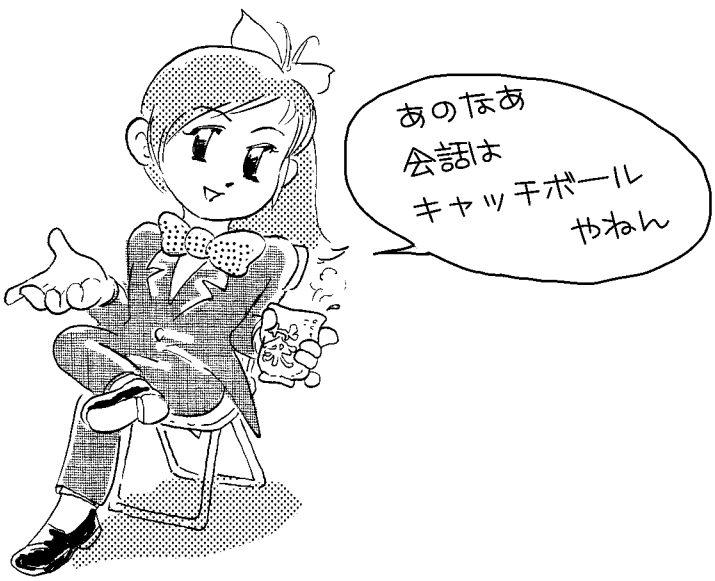
ただ、まったくナンセンスな受け答えしかできないのでは相手をシラケさせるだけですので、人工無脳は巧妙に知能を持っている「ふり」をします。必ずしも論理的である必要はありません。トンチンカンなことも言います。しかし、ユーザがそれを「深読み」してくれるだけの仕掛けがあればそれでよいのです。こうして会話をしている「ふり」をするプログラムは、なかば自虐的に「人工無脳（無能）」と呼ばれるようになりました。

「自分で考えていないのなら、そんなのは見せかけだけだ」と思うでしょうか？　確かに人工無脳は悪く言えばユーザを「だまし」ているわけですが、しかし、だからといってそれが「無意味」だとは筆者は思いません。

例えば、私たちは犬や猫などのペットが「自分を理解してくれている」と感じることがあります。本当に理解しているかどうかは別として、それで「癒され」たりするのは事実でしょう。また、パソコンの調子が悪くなったとき「すねている」ように感じたことはないですか？　あなたの隣にいる人は、ほんとうに「知能」を持った存在でしょうか？　その根拠は？

コミュニケーションを楽しめるかどうかということと、相手に「ほんとうに知能があるのか」ということはあまり関係がありません。単純なしくみしか持たない人工無脳が発した言葉を、私たちはときとしてこちらの内面を見通しているかのように感じ、おもしろがることができます。ここで浮き彫りになるのは人工無脳の「だまし」の巧妙さではなく、このようなトンチンカンな受け答えからも意味を汲み取ろうとする人間の柔軟性です。

会話には共同作業的なところがあります。人工無脳のつたない発言を積極的に理解してあげようとするユーザの前向きな姿勢がセッションを盛り上げます。人工無脳との会話は、人間であるユーザとプログラムとのコラボレーションなのです。人工無脳は「エンターテイメント」であり、漫才でいうところの「相方」でもあるわけです。



CHAPTER 1－2

ActiveScriptRubyのインストール

前節の終盤、少々オーバーヒート気味なテンションとなった人工無脳賛歌に読者もちょっと引き気味でしょうから、地道な作業でクールダウンすることにします。Rubyのインストールを始めましょう。

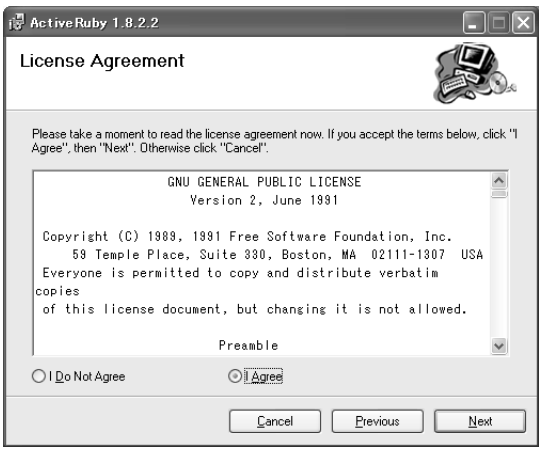
Windowsで動作するRubyにはいくつかの種類がありますが、本書ではarton氏が配布しているActiveScriptRuby[*1]を使います。ActiveScriptRubyは本書配布ファイルのActiveScriptRubyフォルダに収録されています。フォルダ内には1.8.2.2（ActiveRuby18.msi）と1.8.1.3（ActiveRuby.1.8.1.3.msi）の2つのバージョンのインストーラがあります。1.8.2.2のほうがバージョンが新しくRubyの機能も豊富になっていますが、Windows Meや98、95でインストールが失敗することがあるので、1つ前のバージョンである1.8.1.3も収録してあります。1.8.2.2がインストールできない場合、こちらをお試しください。本書で扱う範囲では両者の違いはほとんどありません。

また、インストールにはWindows Installer 1.1以上が必要です。Windows 2000、Windows Me以降のWindowsには標準でインストールずみですが、それ以前のWindowsをお使いの方はマイクロソフトのWebサイトからダウンロードし[*2]、インストールしてください（編注：2014年9月現在、Windows 8/9でのインストールも確認できました）。では、ActiveScriptRubyをインストールしましょう。

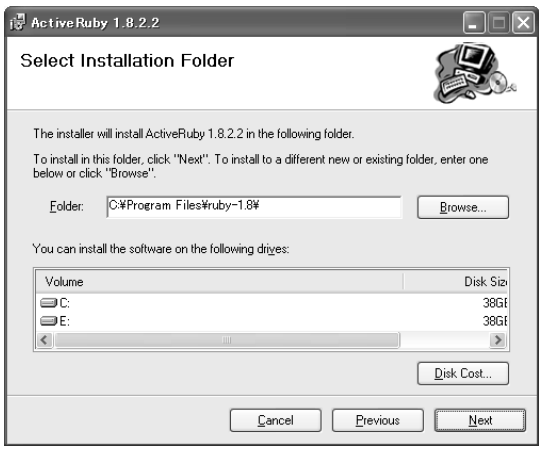
❶ インストーラをダブルクリックしてインストールを開始します。

❷ バージョンを確かめて「Next」ボタンをクリックしてください。

❸ ライセンス（GPL[*3]）が表示されます。ライセンスに同意するならば「I Agree」を選択して「Next」ボタンをクリックしてください。

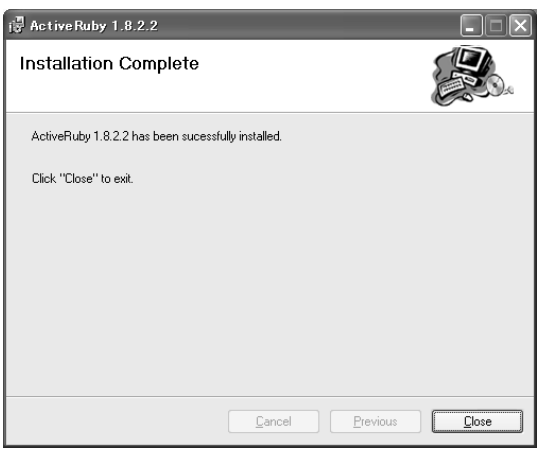


❹ インストール先のフォルダを選択します。デフォルトの設定ではC:¥Program Files¥ruby-1.8¥にインストールされます。



❺ インストールしてよいかどうかの確認で「Next」ボタンをクリックするとインストールが開始されます。しばらくお待ちください。

❻ インストールが完了しました。「Close」ボタンをクリックしてインストーラを終了してください。



なお、Windows 95/98/MeではC:¥Autoexec.batファイルを編集してPATHを追加する必要があります。メモ帳などで開き、ファイルの最後に

```
PATH "%PATH%;C:¥Program Files¥ruby-1.8¥bin"
```

と1行追加してください。インストール先を変更しているときは、「C:¥Program Files¥ruby-1.8」の部分を読み替えてください。また、設定を反映させるためにWindowsを再起動する必要があります。

❼ 以上でインストールは完了です。スタートメニューに「Ruby-1.8」という項目が追加されていますので、その中の「ruby console」を実行してください。

コマンドプロンプトが起動したら、ruby -vというコマンドを実行してみてください。

```
ruby 1.8.2 (2004-12-25) [i386-mswin32]
```

Rubyのバージョン情報が表示されたら準備完了です。

*1 COM Meets Ruby
http://www.geocities.jp/artonx/ruby/
*2 [INFO] MSI : Windowsインストーラエンジンの入手方法
http://support.microsoft.com/default.aspx?scid=kb;ja;292539
*3 GPL (General Public License) についての詳しい情報は、以下のURLを参照してください。
GNU一般公衆利用許諾契約書
http://www.opensource.jp/gpl/gpl.ja.html
GNU's Not Unix! - GNUプロジェクトとフリーソフトウェア財団 (FSF)
http://www.gnu.org/home.ja.html

CHAPTER 2
湯けむりRubyひとめぐりの旅

インストールがすんだところで、Rubyの文法についてざっと見ていきましょう。あまり仔細にこだわりすぎるとページがいくらあっても足りないので、ここでは基本的なランドマークだけを示し、残りは新しい要素が出てきたときにそのつど説明します。

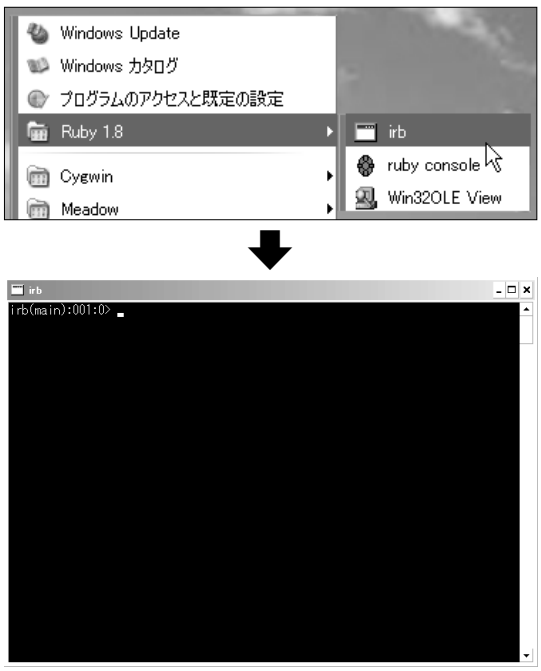
この章の目的はとりあえず人工無脳を作るための準備段階としてRubyの基礎的な知識を大急ぎで身に付けるということと、Rubyのプログラムの雰囲気に慣れていただくことの2点です。そのため記述がちょっぴり正確ではなかったり、意図的に省いたりする部分が多分にあります。非常にざっくりとした解説となりますが、そのくらいのお気楽さでプログラミングを楽しみましょう。

CHAPTER 2－1

irb

本節ではRubyの試し斬りツールとしてirbを使います。irbはInteractive RuByの略で、Rubyのコード（プログラムの断片）を1行ごとに入力・実行するツールです。通常、Rubyのプログラムはテキストファイルとして作成され、コマンドプロンプトからruby hoge.rbなどとして実行するのですが、irbを使うとファイルを作る手間が必要ないので、より手軽にRubyの機能を試すことができます。

それではirbを実行しましょう。ActiveScriptRubyのインストールがすんでいれば、スタートメニューにirbが登録されています。実行すると図のようなウィンドウが表示されます。



ここでの「irb(main):001:0>」はirbが表示しているプロンプトで、irbが入力を待ち構えていることを表します。ここにRubyのコードを入力し、Enterキーを押すことで実行されるわけです。また、irbには非常に強力な入力支援機能があり、キーのタイプ量を軽減することができます。

▌ヒストリ（履歴）機能

過去に入力したコードをもう一度ひっぱり出してくることができます。つまり、同じことを2度入力する必要がありません。

履歴をさかのぼる ……↑キー または Ctrl＋Pキー
履歴を下る ……↓キー または Ctrl＋Nキー
さかのぼる方向に検索 ……Ctrl＋Rキー
下る方向に検索 ……Ctrl＋Sキー

ひっぱり出してきたものを方向キーやDeleteキーなどで再編集することもできます。

▌補完機能

あるキーワードを途中まで入力してTabキーを押すとirbが残りを補ってくれる機能です。候補が複数ある場合は、そのリストが表示されます。例えば、

```
irb(main):029:0> prin
```

まで入力したところでTabキーを押すと、

```
irb(main):029:0> print
```

まで自動的に入力されます。ここでもう一度Tabキーを押すと、

```
irb(main):029:0> print
print   printf
```

と複数ある候補がリストアップされます。この例ではあまりうれしくありませんが、もっと長いキーワードを入力するときや、綴りをど忘れしてしまったときなどはとても役に立つ機能です（リファレンスマニュアル：添付ライブラリ > irb.rbを参照）。

CHAPTER 2－2
最初のサンプル

irbの準備ができたら、いよいよRubyに触れましょう。まずはシンプルな例から。

```
print("ワンツー¥n")
```

これはワンツーパンチを放つプログラムです。irbでの実行結果はこうなります。

```
irb(main):001:0> print("ワンツー¥n")
ワンツー
=> nil
```

「ワンツー」が表示されました。その下の「=> nil」は「実行結果はnilになりました」というirbからのメッセージです。ここで「nilってなんだ？」という疑問はもっともなのですが、ちょっと後回しにさせていただいて話を先に進めます。

「print」はメソッドです。メソッドとは命令、コマンドのことだと考えてください。メソッドを実行する（メソッドを「呼び出す」とも言います）とある決められた働きをし、その結果となるモノを返してきます。この返ってきたモノを「戻り値」と言います。

またメソッドには、その命令を実行するのに必要なモノを渡すことができます。この渡すモノのことを「引数」と言います。printメソッドは引数として与えられたモノを表示し、戻り値としてnilを返す、という働きをします。上の例では引数の「ワンツー」が表示され、戻り値のnilが「=> nil」と表示されている、ということになります。

printメソッドに引数として渡されている「"ワンツー¥n"」は「文字列」と呼ばれる文字の連なりで、このようにダブルクォート（"）やシングルクォート（'）で囲みます。最後に変なもの（¥n）が付いていますが、これは改行を表す特別な記号です。これがないと、「ワンツー=> nil」のように改行されずに表示されてしまいます（リファレンスマニュアル：Rubyの文法 > メソッド呼び出し、および標準ライブラリ > 組み込み関数 > printを参照）。

カッコについて

Rubyではメソッド呼び出しのカッコを省略することができます。つまり、

```
print('hoge')
```

は

```
print 'hoge'
```

と書くことができます。プログラムを書く上でメソッド呼び出しは頻繁に行われるので、このカッコの省略はタイプ量の軽減に大きく貢献するのですが、場合によってはわかりにくいコードになったり、意図しない実行順序になって思わぬ不具合を生み出したりということがあるので、Rubyプログラミングの勘どころをつかむまではきちんとカッコを付けたほうがよいでしょう。
ただ、引数がないときに「()」とだけ書くのは無駄な気がするのも確かですので、本書のサンプルコードでは以下のルールに従うことにします。

・原則、カッコは省略しない。
・ただし、引数なしのメソッド呼び出しでは省略する。

これはスタイルの問題ですので正解があるわけでありませんが、付属の標準ライブラリやネットで配布されているプログラムなどを見てみると、カッコを省略できるところはできるだけ省略するというスタイルが主流のようです。

CHAPTER 2－3
オブジェクト

先ほど「メソッドは結果となるモノを返す」「メソッドにはモノを引数として渡すことができる」と変な言い方をしました。先の例では"ワンツー¥n"という文字列やnilがその「モノ」にあたるわけですが、このような「モノ」のことをRubyでは「オブジェクト」と言います。オブジェクトには様々な種類があり、文字列もその一種（文字列オブジェクト）です。プログラマ自身が新しいオブジェクトを作り出すこともできます。オブジェクトはRubyというプログラミング言語のもっとも重要な基本単位であり、Rubyによるプログラムはこのオブジェクトを中心として構成されます。つまり、Rubyは「オブジェクト指向言語」なのです。

オブジェクトはもちろん文字列や数値などの情報を保持します。が、それだけではなくて、オブジェクトのおもしろさは固有のメソッドをも併せ持つことができるという点にあります。要するに、ある情報と、その情報に関連したメソッドをグルグルッとひとくくりにしたものがオブジェクトです。うーん、むずかしいですか？　"ワンツー¥n"もオブジェクトですのでもちろんメソッドを持ちます。やってみましょう。

```
irb(main):002:0> "ワンツー¥n".size
=> 9
```

このように、オブジェクトのメソッドを呼び出すときは「オブジェクト.メソッド」のようにピリオド（.）でつなぎます。

文字列オブジェクトにはsizeというメソッドがあり、保持する情報である文字列のバイト数を調べて返すという働きをします。9ではちょっと数が合わないような気がしますが、日本語の文字は1文字で2バイト必要ですので「ワンツー」で8バイト、改行の「¥n」で1バイト、合わせて9バイトとなります。

どうでしょう。ある情報に、その情報を操作するためのメソッドが装備されているイメージがつかめたでしょうか？　では、ここで質問です。先ほどの例ではsizeメソッドから9という整数が返されたわけですが、この9はオブジェクトでしょうか？

もちろんオブジェクトです。このことを利用して、次のようなことも可能です。

```
irb(main):003:0> "ワンツー¥n".size.next
=> 10
```

文字列オブジェクトのsizeメソッドはバイト数を整数オブジェクトとして返します。整数オブジェクトにはnextというメソッドがあり、その次の数を持った整数オブジェクトを返します。上の例ではsizeが返した9という整数オブジェクトに対してすぐさまnextメソッドを呼び出し、9の次の整数である10という結果を得ています。言うまでもなく（でも言いますけど）、この10もオブジェクトです。

試し読みはお楽しみ
いただけましたか？

ここからはManatee
おすすめの商品を
ご紹介します。

Tech Book Zone Manatee

3.15
2017

おすすめ電子書籍

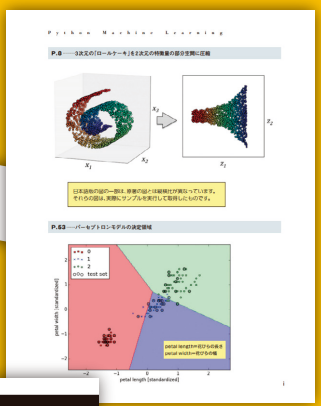
Tech Book Zone Manatee

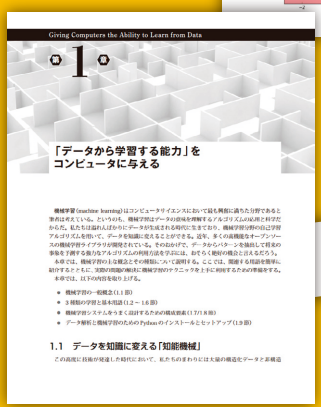
1

AI技術の基礎を習得!
分類/回帰問題や深層学習を知る

機械学習の理論とPython実践法を網羅的に解説した技術書です。機械学習とは、データから学習した結果をもとに判定や予測を行うことです。すでにさまざまな機械学習の方法が開発されています。本書では、それらの方法について背景にある理論や特徴を解説した上で、Pythonによる実装法を説明。初期の機械学習アルゴリズムから取り上げ、前処理や次元削減、Webへの展開のほか、終盤ではディープラーニングについても見ていきます。

機械学習の理論とPythonでの実践法を網羅的に解説





AIプログラミングの第一歩を踏み出すための格好の一冊

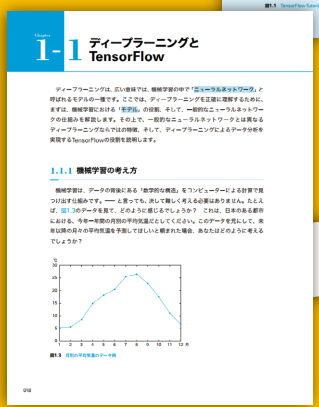
「人工知能・機械学習」

2

TensorFlowを動かしながら
ディープラーニングを理解しよう

機械学習やデータ分析が専門ではない、一般の方が対象の解説書。ディープラーニングの代表とも言える「畳み込みニューラルネットワーク」を例に、その仕組みを根本から理解すること、そしてTensorFlowを用いて実際に動作するコードを作ることが本書の目標です。ディープラーニングの根底にあるのは、古くからある機械学習の仕組みそのものです。簡単な行列計算と微分の基礎がわかっていれば、その仕組みも理解しやすいでしょう。

ディープラーニングの根本原理やTensorFlowのコードの書き方を学習できる



畳み込みニューラルネットワークを構成する1つひとつのパーツの役割を丁寧に解説

人工知能・機械学習

Python
機械学習プログラミング
達人データサイエンティストによる理論と実践

インプレス
SebastianRaschka（著者）・
株式会社クイープ（翻訳）・
福島真太朗（監訳）464 ページ
価格：4,320 円（PDF）

人工知能・機械学習

TensorFlow で学ぶ
ディープラーニング入門
畳み込みニューラルネットワーク徹底解説

マイナビ出版
中井悦司（著者） 264 ページ
価格：2,905 円（PDF）