

## はじめに

今、この本を手にとっているあなたは、「ChatGPT を使った社労士業務の自動化」に興味を持たれているのでしょうか。あるいは、単純に「プログラム不要」ってどういうことだろう？と疑問に思われているかもしれませんね。

筆者は 2020 年に Python (パイソン) というプログラム言語に出会い、瞬時にその可能性に魅了されました。また、この Python を使えば PC を自由自在に操ることができることから、日常業務のさまざまな場面で Python を活用できる場面があるのではないかと、そう思うまでに、さほど時間はかかりませんでした。

ところが、です。難しいのです…この「Python」が！

もともと文系人間である筆者にとって、「プログラム言語」に接するのも初めてでしたし、「何をどうすれば今の自分にとって必要な知識を身につけることができるのか」という方策を探るのに四苦八苦する日々が続きました。なぜこんなにも前進できないのか。その理由は、「Python の守備範囲があまりにも広い」ということに尽きます。

「スポーツ万能選手の力が借りたいのではない、私が必要としているのはキャッチボールのレベルなんだ。このギャップをどうすれば埋めることができるのかな…」と頭をひねり続けていましたが、そこに登場したのが ChatGPT です。

これはもう、筆者にとっては「救世主」と呼べるものでした。この ChatGPT を使えば、「私のように **Python の知識がまったくない人でも**、容易に Python のプログラムを作成することができて、**社労士業務の効率化をすぐに実現することができる**」ということに気づいたのです。

あらかじめお断りしておきますが、本書は Python の学習を進めるための書籍ではありません。あくまで、**ChatGPT を使って Python のプログラムを組み、社労士業務の効率化を図ることを目的**としています。Python を学びたければ市販の書籍が山ほどありますし、インターネット上にも情報はあふれています。しかし、本書のように「**プログラムの知識なしで、ちょっとした業務を PC に自動で作業させる**」ということに主眼を置いた書籍はほとんどありませんでした。

本書を通じて、「なるほど、こうすれば ChatGPT と Python を使ってフォルダを自動で作ったり、複数の請求書を一瞬で作成したり、PDF の必要な部分だけをコピーしたりすることができるのか」ということをご自身で体感していただきたいと思います。

ところで、皆さんは RPA（ロボティック・プロセス・オートメーション）という言葉を目にしたことはありますか？ 本書を読み進めると、実は **Python・ChatGPT・PyCharm（パイチャーム）** を使うことで市販の RPA と同じようなことが皆さんにも簡単に実現できてしまいます。

「え、そんなことが本当にできるようになるの？」と思われるかもしれませんが、本書を読み進めていただければ、誰でもできるようになります。さらに、皆さんが作った Python プ

プログラムに「おまじない」をかけることで、これを「アイコン化」することができます。それは、誰にでも使うことができる「ロボット」です。このロボットをダブルクリックすることで、さまざまな社労士業務を瞬時に完遂できるのです。

もちろん、皆さんの事務所のスタッフも、このロボットを使うことができます。スタッフが四苦八苦して時間をかけて行っていた作業がアイコンのダブルクリック一発で終了してしまう感動をぜひ体感していただきたいと思います。

このロボットを作るにはプログラムが必要ですので、本書では Python のプログラムを使います。ただ、繰り返しになりますが、本書は Python の学習を目的とした書籍ではありません。PC で Python のプログラムを動かして業務を効率化することが主な目的です。その観点を忘れず、ぜひご自身の手で PC を操作してみてください。その上で、さらに Python に興味がある方は、本書を読み終えた後に市販の書籍やインターネット上の情報等を通じて学びを深めていただければと思います。

2026年2月

社会保険労務士 井上義教

## 〈本書のご利用に当たって〉

### ◆サンプルプログラム

本書で使用しているサンプルプログラム（DLの表示のあるもの）は、以下の URL からダウンロードできます。

[https://www.horei.co.jp/book/book\\_contents.shtml](https://www.horei.co.jp/book/book_contents.shtml)

なお、上記 URL からダウンロードできるサンプルプログラムに関する権利は、株式会社日本法令および著者が有しています。許可なく転載・配布等を行うことはできません。

### ◆サンプルプログラムの動作確認環境

以下の環境で動作を確認していますが、PC の構成や設定等により、同様に動作しない場合がありますので、あらかじめご了承ください。

OS : Windows11

Python : Python13.9

PyCharm : PyCharm2025.2.4

### ◆免責事項

サンプルプログラムの提供に当たっては、可能な限り正確な記述に努めておりますが、株式会社日本法令および著者は、その内容について一切の保証を行うものではありません。また、実際の動作結果についても責任を負いかねます。

なお、本書に記載されている会社名・商品名・製品名については、一般に各社の商標または登録商標であり、本書では® や™ マークを省略しています。

### ◆著作権について

本書の内容は、著作権法により保護されています。

### ◆ご注意

本書の内容は、2026 年 1 月時点の情報に基づいています。Python や PyCharm のアップデートにより、実際の動作や画面の表示が変更される可能性がありますので、ご注意ください。

(本書のご利用に当たって) — 4

## 序 章 Python を使って 社労士業務を自動化しよう！ — 13

- 1 >>> 定型業務に忙殺される社労士の日々 — 14
- 2 >>> ChatGPT の登場で変わった「自動化の敷居」 — 15
- 3 >>> 社労士の現場に業務の自動化が向いている理由 — 16
- 4 >>> 身近な定型業務を自動化するとどう変わるのか — 17
- 5 >>> 自動化の最初の一步は「身近な不満」から — 18
- 6 >>> 本書のゴールと対象者 — 18
- 7 >>> 自動化の対象となる業務の例 — 19

## 第 1 章 「三種の神器」で 自動化の旅に出よう！ — 23

- 1 >>> Python = プログラム言語 (外国語のようなもの) — 24
- 2 >>> ChatGPT = あなた専属の「翻訳者さん」 — 25
- 3 >>> PyCharm = 翻訳された内容を実際に動かすための  
場所 (作業机) — 26
- 4 >>> 三種の神器が揃えば、もう準備は完了です！ — 27
- 5 >>> ChatGPT と「対話型プログラミング」の未来 — 27
- 6 >>> 「できる人」と「続けられる人」 — 29
- 7 >>> 本書との上手な付き合い方 — 30

## 第 2 章 Python をインストールしよう！ — 31

- 1 >>> まずは公式サイトへ — 32
- 2 >>> Windows を選択 — 33
- 3 >>> インストーラーをダウンロードして起動する — 33
- 4 >>> インストール完了後に「Close」をクリック — 36
- 5 >>> Python が正しくインストールされたかどうかの確認 — 37

### 第 3 章 ChatGPT を 使えるようにしよう！ — 41

- 1 >>> まずは公式サイトへ — 42
- 2 >>> アカウントを作成する — 43
- 3 >>> メールアドレスを登録する — 44
- 4 >>> パスワードを設定する — 45
- 5 >>> メール認証を行う — 46
- 6 >>> 氏名・生年月日を入力する — 47
- 7 >>> ChatGPT の基本的な使い方 — 48

### 第 4 章 PyCharm を インストールしよう！ — 51

- 1 >>> まずは公式サイトへ — 53
- 2 >>> インストーラーをダウンロードする — 54
- 3 >>> インストーラーを起動する — 55
  - (1) 最初の画面 — 55
  - (2) インストール先の指定 — 56
  - (3) インストールオプションの選択 — 56
  - (4) スタートメニューフォルダーの選択 — 57
  - (5) インストールの開始 — 58
  - (6) 終了画面の表示 — 58

### 第 5 章 PyCharm をすぐに 使える状態にしよう！ — 61

- 1 >>> PyCharm の起動 — 63
  - (1) 検索欄から起動する方法 — 63
  - (2) デスクトップアイコンから起動する方法 — 64
- 2 >>> 初回起動時の設定 — 65

- 3 >>> 使用許諾契約の同意 — 66
- 4 >>> データ共有の有無 — 66
- 5 >>> プロジェクトの作成 — 68
- 6 >>> PyCharm を「日本語化」しよう！ — 70

## 第 6 章 Python ファイルを作って 実行してみよう！ — 75

- 1 >>> PyCharm の起動 — 76
- 2 >>> Python ファイルの作成 — 76
- 3 >>> Python ファイルを動かしてみよう！ — 80

## 第 7 章 ChatGPT で作った プログラムを動かしてみよう！ — 83

- 1 >>> ChatGPT へ問いかけてみよう — 84
- 2 >>> ChatGPT の回答をコピーして  
PyCharm に貼り付ける — 85
- 3 >>> PyCharm で実行して結果を確認する — 86
- 4 >>> ショートカットをうまく利用しよう！ — 86
- 5 >>> 後学のために：ChatGPT のさらなる使い方 — 87
- 6 >>> 練習問題をやってみよう！ — 88
  - (1) 複数のサイトを開いてみる — 89
  - (2) URL を自分の好きなサイトに変えて  
動かしてみる — 89
  - (3) 実行した後にメッセージを  
表示してみよう！ — 90

## 第 8 章 「外部ライブラリ」を体験してみよう！ — 91

- 1 >>> ChatGPT に問いかけてみよう！ — 93
- 2 >>> エラーが出た場合は — 95
- 3 >>> 標準ライブラリと外部ライブラリ — 96
- 4 >>> 「ターミナル」を使って外部ライブラリをインストールしてみよう！ — 97
- 5 >>> 再度実行してみよう！ — 99
- 6 >>> 主な標準ライブラリと外部ライブラリ — 100
- 7 >>> 後学のために：エラーの構造について — 101
  - (1) エラーの種類 (Type) — 102
  - (2) 詳細情報 (Message) — 102
  - (3) 発生箇所 (Traceback) — 102
- 8 >>> 練習問題をやってみよう！ — 103
  - (1) 別のファイル名で Excel ファイルを保存してみる — 103
  - (2) 「練習問題 1.xlsx」のシート名 (サンプル 2) を「テストシート」に変更してみる — 104

## 第 9 章 ChatGPT でいろいろなプログラムを作ってみよう！ — 105

- 1 >>> 新しいフォルダを作ってみよう！ — 107
- 2 >>> フォルダに新しい Excel ファイルを作ってみよう！ — 110
- 3 >>> 作成した Excel ファイルに日付や数字、文字を書き込んでみよう！ — 113
- 4 >>> Excel ファイルを複製して、別の名前で保存してみよう！ — 116

- 5 >>> コピーした Excel ファイルを  
PDF に変換して保存してみよう！ — 118
- 6 >>> PDF の内容に応じて  
ファイル名を変更してみよう！ — 122
- 7 >>> 不要になったファイルを削除してみよう！ — 126
- 8 >>> 後学のために：ごみ箱を空にすることもできる！ — 129

## 第 10 章 請求書の作成業務を 自動化してみよう！ — 131

- 1 >>> まずは請求書のフォーマットを Excel で作ろう！ — 132
- 2 >>> 請求書作成の業務フローを思い出そう！ — 134
- 3 >>> ChatGPT に問いかけ →  
プログラムを実行してみよう！ — 135

## 第 11 章 いろいろな社労士業務を 自動化してみよう！ — 141

- 1 >>> (例 1) 人材開発支援助成金の支給申請書の作成 — 142
  - (1) 支給申請書の作成に当たって  
Python の活用を考えてみる — 142
  - (2) 「こんなことができるかな？」を  
ChatGPT に相談してみよう！ — 144
  - (3) PyCharm に貼り付けて実行してみよう！ — 146
- 2 >>> (例 2) 雇用保険の資格取得届の zip ファイルから  
特定の PDF を取り出す — 148
  - (1) e-Gov の電子公文書をダウンロードした  
後の作業で Python の活用を考えてみる！ — 148
  - (2) 「こんなことができるかな？」を  
ChatGPT に相談してみよう！ — 149
  - (3) PyCharm に貼り付けて実行してみよう！ — 152

- 3 >>> (例 3) 社会保険の資格取得届の zip ファイルの xml ファイルを PDF 化する — 153
  - (1) 社会保険の資格取得データを PDF 化する手作業の煩雑さ — 154
  - (2) Python で自動化するとどう変わるか — 155
  - (3) 「こんなことができるかな？」を ChatGPT に相談してみよう！ — 156
  - (4) PyCharm に貼り付けて実行してみよう！ — 160
  - (5) 後学のために — 162

## 第 12 章 **自分が作ったプログラムを周りの人に使ってもらおう！** — 163

- 1 >>> 作ったプログラムを周りの人に使ってもらうことの意味 — 164
- 2 >>> pyinstaller をインストールしよう！ — 165
- 3 >>> Yahoo! Japan のトップページを開くプログラムを実行ファイルにしてみよう！ — 166
- 4 >>> 実行ファイルを周りの人に渡そう！ — 170
  - (1) USB メモリーで渡す — 170
  - (2) Chatwork を利用する — 170
  - (3) クラウドストレージを利用する — 170
  - (4) メールに添付して送る — 171
- 5 >>> ウイルス検知の誤作動が起きた際の対応方法 — 171
  - (1) Windows Defender の設定変更 — 172
  - (2) 市販のウイルス対策ソフト — 173
- 6 >>> 後学のために：こんなことをすると便利かも — 173
  - (1) メモ帳にプログラムをコピーしておく — 173
  - (2) 「実行ファイルを作らない」という選択もありかも — 174
- 7 >>> まとめ — 174

- (1) ChatGPT との向き合い方 — 178
- (2) ChatGPT への問いかけ方 — 178
- (3) Python の基礎操作 — 178
- (4) よく使う標準ライブラリ  
(覚える必要ナシ!) — 178
- (5) よく使う外部ライブラリ  
(「pip install」以外は覚える必要ナシ!) — 179
- (6) pyinstaller で exe ファイル化する方法 — 179
- (7) 実行ファイルの配付方法 — 179
- (8) ウイルス検知の誤作動対策 — 179
- (9) 実務での工夫 — 180
- (10) エラー対応 — 180
- (11) よく使うショートカット — 180
- (12) ファイルパスの確認方法 — 181
- (13) まとめ — 181
- (14) おまけ (さらに Python を学びたい方へ  
書籍のご紹介) — 181

# 序章

Python を使って  
社労士業務を自動化  
しよう！



## 1 定型業務に忙殺される社労士の日々

皆さんは、「定型業務」という言葉を耳にしたことがあるでしょうか？おそらく、社会保険労務士（以下、「社労士」といいます）で、この言葉を一度も耳にしたことのない方はいないと思います。そして、その「定型業務」という言葉には、多くの場合、好ましくないニュアンスが含まれているのではないのでしょうか。

その本質はこうです。「作業内容は単純・単調であるにもかかわらず、マンパワーを割かざるを得ない。そのため、常に誰かが同じ作業を黙々と繰り返している。」…こうした状況が定型業務に潜んでいると考えられます。

しかし、**Python (パイソン) を活用すれば、こうした業務を自動化することが可能です。**例えば、不要なファイルを削除したり、PDFを開いて氏名を確認し、その氏名をファイル名に付与したりする…こうした作業を「定型業務」と呼ぶのであれば、それを今後は事務所のスタッフではなく PC に任せるべきではないのでしょうか。

もちろん、複雑な業務を自動化しようとするれば、それ相応に複雑なプログラムを作成する必要があります。その場合は専門家（プロのプログラマー）に委ねるのが適切です。しかし、私たちに求められるのは、あくまで**「簡単なプログラムを手軽に作り、単純な作業を PC に任せる」**という発想です。その積み重ねによって、社労士事務所の定型業務の自動化は着実に進んでいきます。

定型業務に忙殺されているのは、発想力が鈍り、疲弊を招く恐れがあります。反対に、PC に任せることができる作業を少しずつ増やすことができれば、皆さん（社労士）も、皆さんの事務所のスタッフも、より創造的（クリエイティブ）な業務にマンパワーを振り向けることが可能となるはずで

## 2 ChatGPT の登場で変わった「自動化の敷居」

これまでは、業務の自動化に取り組もうとすると、とにかくある程度のプログラミングの知識を身につけなければなりません。まずは Python の基本的な文法を学び、次に環境を整え、プログラムのエラーと対峙しながら試行錯誤を重ねる…こうしたプロセスに抵抗を覚え、結局手を付けることができなかつた方も少なくないはず。

ところが、ChatGPT の登場によって、その状況は一変しました。今や「こういう作業を自動化したい」と ChatGPT に伝えれば、ChatGPT が瞬時に Python のプログラムを作ってくれるのです。自分でゼロからプログラムを書く必要がないどころか、Python の知識もまったく不要です。今やプログラミング言語に関する知識よりも「**自分がやりたいことを正しく ChatGPT に説明することができるかどうか**」が重要な時代になっているのです。

つまり、ChatGPT の登場によって、業務の自動化のための「プログラミングの壁」は大幅に低くなったのです（というか、もはや撤去されたに等しい）。Python はプログラミング言語ですが、ChatGPT が「翻訳者」として間に立ってくれるおかげで、私たちは新しく言語を学ぶ必要がなくなりました。

この変化は、社労士事務所の業務においても非常に大きな意味を持ちます。これまでは「プログラムが書けないから自動化は難しい」と考えていた方も、ChatGPT を活用することで業務を効率化できるようになったのです。必要なのは複雑なプログラムの知識ではなく、「**自分の業務のどの部分を PC に任せたいのか**」を見極める発想力です。

### 3 社労士の現場に業務の自動化が向いている理由

なぜ、私たち社労士の業務において、PCを使った業務の自動化が特に効果を発揮するのでしょうか。その理由の1つは、業務の中に「**単純だけれども絶対に間違えてはいけない作業**」が数多く存在しているからです。例えば、次のような作業です。

- ・申請書類（Excel、PDF等）に適切なファイル名を付ける
- ・ファイルの保存先を間違わずに選択する
- ・Excelにおけるデータの転記や集計

これらは簡単な作業なのですが、少しでもミスが起きると後々の大きなトラブルにつながりかねません。そのため、事務所のスタッフは、この作業を慎重に何度も確認しながら時間をかけて行わざるを得ないのです。

しかし、こういった作業こそ、PCが最も得意とする分野です。一度手順をプログラム化してしまえば、何百回でも同じ精度で繰り返すことができます（残業代も発生しませんし、文句も言いません。ハラスメントの余地もありませんし、ムチャ優秀ですよ！）。スタッフが集中力を消耗させながらこういった業務に取り組む必要はなくなり、その分、本来注力すべき判断業務や顧客対応に時間を割くことが可能になるのです。

もう1つの理由は、**社労士の業務が「書類とデータ」に直結している点**です。今や紙から電子申請への移行が進み、業務のほとんどの成果物がデータとしてPCの中に保存されるようになりました。つまり、私たちの業務はPCの中で完結できる作業が多く、自動化との相性が非常によいのです。

このように考えると、社労士事務所の業務の自動化は、決して特別

なITスキルを持つ人だけのものではなく、私たち社労士にとっても身近で役立つ手段であることがわかります。

## 4 身近な定型業務を自動化するとどう変わるのか

では、実際に私たちの身近にある定型業務を自動化すると、どのような変化が生まれるのでしょうか。

例えば、行政機関から届いたPDFの内容をスタッフが確認して、被保険者の名前を抜き出し、その名前と作業日の日付をファイル名に付けて保存する…このような作業を思い浮かべてみてください。

これまでは、スタッフが一つひとつのPDFを開いて内容を確認し、名前と日付を手入力した上で保存先を指定する必要がありました。単純な作業ではありますが件数が多いと非常に時間がかかりますし、ミスが起きることも当然に予想されます。

ところが、Pythonを使って自動化すると、この一連の作業はPCが瞬時に行ってくれます。数十件、数百件の作業であっても、作業時間は圧倒的に短縮され、しかもミスはほぼゼロになります。スタッフがやるべきことは「処理の結果を確認するだけ」になる訳です。

Excelを使った単純な転記作業もまったく同様です。別々のシートやファイルに散らばっている数値をコピーして貼り付けるような作業は手作業では疲れやすく、単調な作業のために集中力が途切れると誤入力の原因になります。しかし、Pythonで「このセルの数値をこちらへコピーしてね」と指示してあげると、一瞬で正確に処理してくれます。

こうした小さな効率化が積み重なることで負担が減り、「余裕を持って業務に取り組める」状態へと変わっていきます。結果として、クライアントへの提案や新しいサービス企画といった、より創造的で付加価値の高い仕事に時間を使うことができるようになります。

## 5 自動化の最初の一步は「身近な不満」から

では、実際に自動化を始めるに当たって、何から手をつければよいのでしょうか。

その答えはとてもシンプルです。「日々の業務で、ちょっとした不満を感じる場面」から始めてみることです。

例えば、

- ・毎回ファイル名を手作業で付けるのが面倒だ
- ・Excelの集計作業にすごく時間がかかっている
- ・大量のExcelファイルをPDF化するのに1時間もかかる

といった「ちょっとした不満」は、誰もが経験しているはずです。大きな課題ではなく、ほんの小さな「手間」や「面倒くささ」こそ、自動化にぴったりの題材なのです。

こういったことを考えると、最初の一步は決して「大きなシステムを作ること」ではありません。自分のデスクの上で繰り返されている単純な作業を何でもよいから取り出してみて、「これをPCに任せることができないかな？」と考えることが重要です。その小さな一歩こそが、自動化の世界への最適な入り口なのです。

## 6 本書のゴールと対象者

本書のゴールはとてもシンプルです。

それは、「社労士業務の中に存在する繰り返しの定型作業をPCに肩代わりさせること」です。

最初から大きなシステムを作る必要はまったくありません。身近な

不満を出発点にして、小さな自動化を一つずつ積み重ねていけばそれで十分です。その結果、定型業務に追われる日々から抜け出し、もっとクリエイティブで付加価値の高い業務に時間を使うことができるようになる——これが、本書を通じて目指す姿です。

では、本書を手にとっていただきたいのは誰か。

それは、日々の業務の中で「もう少し業務を効率化できないだろうか」と感じている社労士の皆さんです。Excel や PDF の操作に多くの時間を割いているけれど、本当は顧客に対する提案や新しい挑戦にもっと力を注ぎたい、そんな思いを抱えている方には、本書の内容はきっと役に立ちます。

また、「自動化には興味があるけれど、プログラミングは難しそう」と感じている方にも安心して読み進めていただくことができるよう、本書では、プログラミングの専門知識があることを特に前提とせず、ChatGPT を活用しながら、誰でも自然に自動化の世界へ足を踏み入れることができる方法を紹介しています。

## 7 自動化の対象となる業務の例

### 「いったい、どんな社労士業務が自動化できるのか…」

本書を手にとった皆さんが最も興味をお持ちのポイントは、きっとここだと思います。

もちろん、本書でそのすべてをお伝えすることは難しいのですが、筆者の事務所で実際に大幅な業務の効率化に成功したメニューの一部を紹介します。どれもこれも、特別なプログラミング知識がなくても自動化が可能なものばかりです。

### (1) e-Gov で届いた離職証明書関係のファイルの処理

e-Gov で届いた離職証明書関係のファイルを自動的に判別し、「被保険者へ送付するもの」と「会社へ送付するもの」に分け、それぞれのフォルダへ振り分けることが可能です。

### (2) 顧問先へ発行する請求書の自動作成

事前に Excel シートにデータを入力しておけば、複数の顧問先向けの請求書を一瞬で自動作成することが可能です（第 10 章でご紹介しています）。

### (3) e-Gov で届いた社会保険関連の xml ファイルの処理

e-Gov で届いた社会保険関連の xml ファイルを自動で PDF に変換する（名前を付けて保存する手前まで）ことが可能です（第 11 章でご紹介しています）。

### (4) Excel ファイルのうち必要なページだけを PDF 化し、氏名をファイル名に付与する作業

ある顧客の Excel データでは、1 シートに大量の印刷内容が含まれていました。しかし、必要なのは 1 ページ目だけです。Python を使えば、必要なページだけを自動抽出して PDF に変換し、さらに該当ページ内に記載された「氏名」を読み取り、ファイル名に自動的に付与することが可能です（第 11 章でご紹介しています）。

### (5) 賃金台帳のデータを転記する処理

顧客から預かった賃金台帳をもとに給与計算ソフトへデータ入力する際、月ごとのデータを従業員ごとに転記する作業を手作業で行うのは負担が大きく、ミスも発生しやすい作業です。Python を使えば、これを一瞬で正確に転記することができます。

#### (6) 翌月分の勤怠管理表を更新する処理

顧客の勤怠管理を行っている場合、月末に翌月分の新しい勤怠シートを人数分作成する必要があります。50名分のシートを手作業で作るとなるとかなり面倒ですが、Pythonを使えば一瞬でまとめて作成することができます。

#### (7) Word 文書を PDF 化し、フォルダに保存する処理

例えば、Wordで作った労働条件通知書(20名分)を、一瞬でPDFに変換し、フォルダに整理して保存することが可能です。

#### (8) 複数名の給与明細ファイル(PDF)を人ごとに分割する処理

給与計算ソフトによっては、給与明細が全員分まとめて1つのPDFとして出力されることがあります。Pythonを使えば、このPDFを従業員ごとに自動分割し、一瞬で配付用のデータとして整えることができます。

以上、ほんの一部ですが、社労士業務の中には自動化できる作業が驚くほど多いことを実感していただけたのではないのでしょうか。

改めて整理すると、自動化に向いているのは、主に次のような業務です。

- ・ 繰り返しが多い単調な作業
- ・ 手作業で行うとミスが起きやすい作業
- ・ 複数のファイルを行き来し、手順が複雑な作業

こういったもののすべてが片っ端から自動化の対象となる訳です。

「プログラミングの知識がないと難しいのでは…」もしもそんな不安をお持ちでしたら、どうか忘れてください。本書で扱う方法は、「社

労士実務に特化した、誰でもできる自動化」なのですから！

くじけそうになっても、とにかく「プログラムの知識は不要」ということを思い出して、何とか頑張ってみてください。本書を読み終える頃には、皆さんはきっと立派な「ロボット使い」に育っていることでしょう。社労士事務所の未来を大きく変える技術を、ぜひ実際に体感してみてください。

## 【著者略歴】

〈著者〉 井上義教（いのうえ よしのり）

1964年東京都生まれ。

大阪大学経済学部卒業。銀行勤務を経て2004年に社労士として開業登録、2011年に特定社労士付記。規程の整備と労務相談を中心に業務を展開。著作に「最新IT業界の人事・労務管理と就業規則」（共著、日本法令）、「営業店の年金推進ガイド」（経済法令研究会）等がある。

オスピス社労士事務所：<https://auspice-sr.com/>

〈監修者〉 西野亜紀（にし の あき）

1971年大阪府生まれ。

2006年に社会保険労務士試験に合格。以来、人材開発支援助成金やキャリアアップ助成金の活用を通じ、企業の人材育成をサポートしてきた。オンラインのコミュニケーションツールを積極的に活用し、地域にとらわれず全国の企業と柔軟な連携体制を構築。顧問先からは「実務に強く、伴走型の支援が心強い」との評価を受けている。

アール社会保険労務士事務所：<https://rqmca.com/resr/>

〈校正担当〉 伊藤あかね（いとう あかね）

1992年千葉県生まれ。

東洋大学経営学部を卒業後、複数の企業にて経理・労務・総務をはじめとするバックオフィス業務全般に従事。勤怠管理や給与計算の運用改善、業務フローの再構築に取り組み、現場視点に基づく実務知識を幅広く蓄積してきた。2025年に社会保険労務士試験に合格、2026年に社会保険労務士として登録と同時にオスピス社労士事務所に参加。現在は、バックオフィス業務で培った実務理解と現場感覚を強みとする社会保険労務士として、企業の労務管理や制度運用のサポートに携わっている。