

Pythonによるプログラミング導入教育と学習者の理解度評価

辻, 康孝
九州大学大学院工学研究院

<https://hdl.handle.net/2324/7172568>

出版情報：情報処理学会全国大会講演論文集. 82, pp.4-307-4-308, 2020. 情報処理学会
バージョン：
権利関係：(c) 2020 Information Processing Society of Japan



Python によるプログラミング導入教育と学習者の理解度評価

辻 康[†]

九州大学大学院工学研究院[†]

1. はじめに

非情報系・工学系専攻でもプログラミング教育が重要になりつつある。初学習者に対するプログラミング導入教育において、どのプログラミング言語が導入教育の学習言語として適切かは、学習内容や期待される学習成果とともに、プログラミング教育における基本的な研究課題となっている。最近では、基礎的構文の簡素さ、ソースコードを簡潔に記述できる点などから、Python が採用される場合が増えている（例えば [1] の調査では 1 位、[2] では 2 位で 24%）。

また、学習者の関心や理解を高めるため、プログラミングの基礎学習だけでなく、専攻分野での課題テーマと組み合わせた導入教育も実践されており、そのような指向の導入教育においても、その用途の幅広さから Python が採用されている。一方、適切なテーマ（課題）の設定に対しては、学習者の Python の各学習項目の理解度を適切に把握する必要がある。特にリスト、ファイル・文字列処理は、実践的なデータ処理に必要な学習項目であり、クラスの概念を学習する前にオブジェクト操作を行うメソッドも学ぶ必要がある。初学者がこれらの項目をどの程度理解あるいは使いこなせるかが重要となる。

本稿では、著者が本学経済学部経済工学科を対象に行っているプログラミング導入教育 [3] において、学習者の Python の各項目の理解度と学習者が選択した総合課題との関連について調査を行い、より効果的なプログラミング導入教育のための基礎的な検討を行う。

2. Python によるプログラミング導入教育

本学では、全学 1 年生に対して「プログラミング演習（1 単位、15 回）」の授業がプログラミング導入教育の授業として開講されている。2018 年度より、一部の工学系専攻を除き Python が学習言語として採用されており、演習環境として、LMS : Moodle, 教材配信システム : BookRoof,

作業環境 : JupyterHub を用いている。共通教材は準備されているが、授業の実施及び学習内容の選定等は、担当教員の裁量に任されている。

著者は 2014 年より経済学部経済工学科のプログラミング演習（選択必修・前期）を担当してきた。同学科は、数理的知識・思考を必要とする授業が多いため、数学を得意とする学生も比較的多く、文理融合の特徴を持っている。

各授業では Python の基本文法（変数と型、標準入出力、数学関数、各制御構造、リスト、文字列処理、ファイル処理、ユーザ定義関数等）を学習し、授業トピックに関連した演習課題に取り組む。最後（15 回目）に一定期間をかけて取り組む総合課題を課し、成績評価は各演習と総合課題で行っている。なお著者の授業では、独自のスライド教材を開発し用いている。演習課題や授業設計の詳細は既報 [3] を参照されたい。

3. Python に関する学習者の理解度調査

3.1 理解度アンケートによる調査

学習者が Python の基本文法を実際にどの程度理解できたか理解度調査を行った。調査はアンケート形式で行い、各文法項目 Q1~Q9（図 1 参照）に対する学習者の理解度（自信度）を以下の 4 段階で自己評価してもらった。実施は 15 回目授業で行った。

[理解度 H]（教材を見ながら）そのトピックに関連した“簡単な”プログラムを一から作成することができる。

[理解度 M]（教材を見ながら）そのトピックに“関連したサンプルコードを改良して解ける程度”の問題ならプログラムを作成できる。

[理解度 L+] そのトピックの内容は理解できたが、プログラムを組めるまでの自信はない。

[理解度 L-] そのトピックの内容を理解できなかった。

なお Q9 : 標準関数とメソッドの項目に関しては、その理解度（自信度）を、プログラムソースを読んで両者を区別できるかどうかで尋ねている。M : 簡単なプログラム H : 少し複雑なプログラム

3.2 理解度（自信度）に関する結果

図 1 (a) に 2018 年度実施アンケート [3],

Teaching an introductory programming in Python and its assessment of learner's comprehension

[†]Faculty of Engineering, Kyushu University

(b) に 2019 年実施の結果を示す。まず Q9 を除き、ほとんどの学生が各授業トピックの授業の内容は理解できたようだが、自身でプログラムを作成できるまで理解できたという学生は 6~7 割程度であった。最も基本的な「Q1 変数と型」でさえも、3 割程度が理解できてもプログラムを組む自信がないと回答しており、このような学生にいかに関与するかが今後の課題といえる。

初学者にとっては一次元配列及び配列操作は理解しづらいトピックである。Python では「Q6 リスト型」が配列に相当するが、アンケートからは、理解できなかったと回答した学生は非常に少ない結果となった。リストでは柔軟な配列操作が可能なので、学生には理解しやすかったものと思われる。またリストは for ループ、内包表記、matplotlib に関連するため、本授業ではリストの関連する内容を細かく分けて指導しており、このような授業設計も有効だったと考える。一方、「Q9 標準関数とメソッドの違い」を、プログラムソースを読んだときに区別できると回答した学生は半数を切る結果となった。ファイル処理および文字列処理に関する一連のメソッドは、やはり初学者には複雑な内容だったようである。

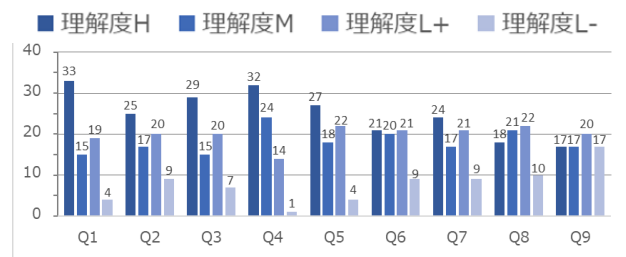
4 理解度（自信度）と総合課題（実践力）

授業では受講学生に自身のレベルに合った総合課題を選択してもらい、2 週間程度で提出してもらっている。ここでは前章の理解度を用いて各学生の自信度を算出し、学生（アンケート未回答者は除く）が選択した総合課題との関連を調べる。自信度は各項目に以下のようなポイントを与え、それらを合計して算出した。Q0 は標準入出力である。

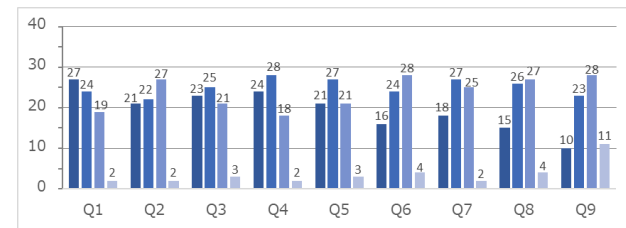
Q0~Q5, Q7 : H(3) M(2) L+(1) L-(0)

Q6, Q8, Q9 : H(5) M(3) L+(1) L-(0)

結果を表 1 に示す。表から、自信が高いグループほど、難易度の高い問題に挑戦する割合が高くなっていることが分かる。授業や演習を通してプログラミングに自信を付けた学生ほど、良い成績を得るため、あるいは力試しで難しい課題に挑戦していた。一方で、自信度が低いグループほど N の未提出者等の割合が高い。特に 2019 年度では、L の問題を少し複雑なものにしたため、前年に比べその数が増加した。未提出者等のアンケート結果を調べたところ、Q6, Q8, Q9 を L-, L+ と回答したものが多かった。特に「Q9 標準関数とメソッドの違い」では、2018 年度 12 名、2019 年度 15 名であった。



(a) 2018 年実施 (回答数 71 / 受講者数 83) [3]



(b) 2019 年実施 (回答数 72 / 受講者数 93)

図 1 Python に関する理解度アンケート結果 (Q1:変数と型 Q2:数学関数 Q3:順次構造 Q4:条件分岐 Q5:定数反復 Q6:リスト Q7:不定反復 Q8:ファイル入出力 Q9:標準関数とメソッド)

表 1 理解度と総合課題の関係 (H: 難易度が高い課題, M: 中程度の課題, L: 低い課題, N: 未提出・不完全・剽窃)

自身度	2018 年度 (71)	2019 年度 (72)
≥23	H 7 / M 13 / L 1 / N 1	H 12 / M 4 / L 11 / N 3
13~22	H 1 / M 2 / L 8 / N 4	H 1 / M 5 / L 11 / N 6
≤12	H 1 / M 1 / L 13 / N 8	H 0 / M 2 / L 6 / N 11

5. おわりに

非情報系の学生を対象とした Python によるプログラミング導入教育において、学習者がどの程度 Python の基本項目を習得できたかについて、アンケートと総合課題の取り組みから調査を行った。内容は理解できてもプログラムを組む自身がない学習者の支援とともに、メソッドをいかに理解させるかが課題として挙げられる。

参考文献

- [1] Philip Guo, "Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities", <http://cacm.acm.org/blogs/blog-cacm/176450-python-isnow-the-most-popular-introductory-teaching-language-at-top-us-universities>, 2014.
- [2] Brett A. Becker and Thomas Fitzpatrick, "What Do CS1 Syllabi Reveal About Our Expectations of Introductory Programming Students?", SIGCSE '19: Proc. of the 50th ACM Technical Symposium on Computer Science Education 2019 pp.1011-1017.
- [3] 辻 康孝, "Python によるプログラミング導入教育の実践とその学習効果", 基幹教育紀要, Vol.5, 2019, pp.43-55.