

## Python プログラミングの知識習得度テストの試作と実践結果の検討

山川 広人\*<sup>1</sup> 上野 春毅\*<sup>1</sup> 小松川 浩\*<sup>1</sup> 長谷川 理\*<sup>2</sup>

\*<sup>1</sup> 公立千歳科学技術大学 理工学部

\*<sup>2</sup> 武蔵野大学 データサイエンス学部

yamakawa@photon.chitose.ac.jp

本研究は、小中高の各学校段階のプログラミング教育の全面開始や、そのさらなる発展となる大学初年次段階でのプログラミング教育の接続を意識したレベル別プログラミング教材とその利用モデルの実現を目的としている。高校情報や大学初年次段階ではデータサイエンス教育も重視されていることから、ツールとなる Python プログラミングを学び、その知識習得度を判定する仕組みづくりも重要となる。本稿は受験者の Python の知識習得度の判定を狙ったテスト問題を試作し、その実践の結果について検討を行ったものである。

### 1. はじめに

各学校段階でのプログラミング教育の全面実施が進む中で、特に高校情報 I・II の中では Python などのプログラミング言語を用いてコンピュータの仕組みやアルゴリズム、データの分析・活用の基礎を学ぶ内容が取り上げられる。さらに大学では初年次教育とも連係した数理データサイエンス・AI 教育が重要視されている。こうした中で本研究は、各学校段階のプログラミング教育の接続を意識した教材とその利用モデルの実現を目指している。本稿はその一環として、Python プログラミングの知識習得度テストの実践を通じて得られたアンケート結果と Computer Based Testing (CBT) の結果を関連づけ、受験者の状況と今後の教材整備にむけた検討を行う。

### 2. ベースシステムと試作した教材

本稿の Python プログラミングの知識習得度テストは、筆者らが開発した CBT システム<sup>(1)</sup>をベースとして、学習者の Python プログラミングの習得度をレベル別で判定できるよう狙ったものである。ベースシステムでは、教員が事前に 7 段階の難度にレベル分けされたテスト問題を一定量用意することで CBT を実施できる。受験者が CBT で出題された問題に 1 問ずつ答案を送信するたび、システムはその正否情報から受験者が正答できる確率の高い難度レベルを推定し、そのレベルの問題を新たに出题する。これを繰り返すことで、CBT の終了時に受験者が安定し正答できる難度レベルを判定結果（知識習得度）として示す。

テスト問題は、表 1 に示す 7 つの單元ごとに、7 段階の難度のものを整備した<sup>(2)</sup>。單元ごとに、レベル 1 が最も難度の低い知識を問う設問（例：語句やコード記述方法の確認）、レベル 7 が最も難度の高い知識を問う設問（例：実コードの穴埋め、

処理結果の思索など）となるよう意識して整備を行った。

### 3. 先行事例での実践結果

筆者らの先行事例では、2 大学での実践を通じて CBT 受験者のアンケートを分析した<sup>(2)</sup>。このうち、ある大学（大学 B とする）のデータサイエンス系科目を受講する学部 1 年生を対象とした検証結果を本稿でも紹介する。この科目は Python の基礎を学んでいることを前提に、アルゴリズムやデータ収集の方法を新たに学ぶ。その内 3 回分の授業に対して、Python の知識習得度や事前の補習の必要性の確認のため CBT を受験するよう学生に指示をした。CBT の利用の終了後に「CBT 用の問題の難しさ」へのアンケートを実施した。受験者が自己申告した「CBT 利用前の Python 習得状況」ごとにアンケート結果を整理したものが表 2 である。紙面の都合上割愛しており詳細は参考文献(2)を参照されたいが、表 2 の結果は、他方の大学（大学 A とする）で情報系科目を受講する学部 3 年生に同様の実践とアンケートを行った場合とは以下のような違いがあった。

- 大学 A では Python に初めて触れる学生が大半を占めているが、選択肢イを選択した学生が多い。大学 B は Python の基本的文法を既習した学生が大半と思われる一方で、選択肢ウを選択した学生が多い。
- 「学習を手助けしたもの」を聞いたところ、大学 A では「他のプログラミング言語で学んだ知識」、大学 B は「Web サイトや書籍の情報」を挙げた学生が多い。

本稿が目指す各学校段階のプログラミング教育の接続を意識した教材の用途を考えた場合、事前に他のプログラミング言語等の学習経験の少ない大学 B がより実用環境に近いと考えられる。そこ

で本稿は大学 B のアンケートと CBT 結果とを関連づけ、上記の結果との関係を検討する。

#### 4. アンケートとテスト結果を関連づけた検討

大学 B 受験者の CBT の判定レベルを集計した結果を表 3 に示す。授業 1 を例にすると CBT が課せられた単元は表 1 の単元 1,2,3 である。その中で、Python 学習済・学習中・未学習に分類された学生の CBT 受験結果を基本統計量で示している。本稿は表 2・表 3 の結果を関連づけ、大学 B の学生が CBT 用の問題への難度を感じた要因と課題を検討する。

まず、Python 学習済みの学生は単元によらず、平均 5~6 レベルを記録していることがわかる。その一方で、学習中・未学習の学生は、単元が進むにつれ平均レベルが下がっていく。こうした状況は、表 2 で選択肢ウを選んだ学生が多いことと矛盾しない。一般に、順次・条件・反復など小学校段階のプログラミング教育でも取りあげる概念を用いた単元 1,2,3 よりも、データ構造や処理構造についての概念が含まれた後半の単元が学習者にとって難度が高いことは想像に難くない。データ構造や処理構造の基礎は、特に中学・高校段階のプログラミング教育で取り上げられる範囲と考えており、小中高の接続の中でこうした知識や考え方を学べるような支援や補助教材が必要であろう。

また実践を行った科目の前提上、事前の習得状況を基準に分けた際の人数に差がある点に注意をすべきであるが、学習済みの学生に比べ、学習中・未学習の学生は各単元の平均レベルや中央値も低い。本稿で作成したテスト問題はレベル後半にかけて実コードの問題へと推移していく。つまり、学習中・未学習の学生は、単元 1,2,3 の段階においても、単元で求められる考え方や知識と実コードの対応に苦慮している学生が多いことを示している可能性がある。こうした考え方やコードの対応部分の知識の習得にも支援が必要であろう。

#### 5. おわりに

本稿は Python プログラミングの知識習得度判定を目指した CBT とそのテスト問題を整備し、実践結果となるアンケート・テスト結果を関連づけ課題を検討した。検討結果をもとに、今後は新たな単元のテスト問題や、学習を支援する教材の整備を図っていく。

#### 謝辞

本研究は JSPS 科研費 JP20K03234 の助成を受けたものである。

表 1 Python 知識習得度テストの単元と問題数

単元名	総問題数 (問)
1. Python 入門	54
2. 条件分岐	71
3. 反復処理	57
4. リスト・タプル	98
5. 辞書・集合	68
6. 関数	60
7. クラス	61

注) 1. Python 入門は文法や変数の内容である

表 2 CBT 用の問題の難しさへの回答 (大学 B)

選択肢	Python 学習済 (7名)	Python 学習中 (62名)	Python 未学習 (13名)
ア)簡単に感じる事がほとんどだった	1	0	1
イ)どちらかといえば簡単に感じる事が多かった	4	14	6
ウ)どちらかといえば難しく感じる事が多かった	1	41	6
エ)難しく感じる事がほとんどだった	1	7	0

表 3 大学 B 受験者の CBT 判定レベルの集計

	授業 1	授業 2	授業 3
<b>Python 学習済</b>	単元 1,2,3	単元 4,5	単元 6,7
受験人数	7	5	5
最大	7	7	7
最小	4	3	1
平均(標準偏差)	5.86(1.07)	5.00(1.58)	5.20(2.49)
中央値	6.00	5.00	6.00
<b>Python 学習中</b>	単元 1,2,3	単元 4,5	単元 6,7
受験人数	62	51	49
最大	7	7	6
最小	1	1	1
平均(標準偏差)	4.87(1.44)	4.69(1.73)	3.31(1.46)
中央値	5.00	5.00	3.00
<b>Python 未学習</b>	単元 1,2,3	単元 4,5	単元 6,7
受験人数	11	8	7
最大	6	6	3
最小	2	1	1
平均(標準偏差)	4.09(1.45)	3.75(1.67)	2.14(0.69)
中央値	4.00	3.50	2.00

#### 参考文献

- (1) 上野春毅ら：段階的な学習目標を持つ反転学習モデルのための適応型学習システムの開発、教育システム情報学会紙, Vol.37, No.3, pp.212-217 (2020)
- (2) 山川広人ら：プログラミング教育必修化での利用を狙った Python プログラミングの知識習得度テストの試作, 大学 e ラーニング協議会 / 日本リメディアル教育学会 合同フォーラム 2021, pp.57-60 (2022)