大学入学共通テスト「情報Ⅰ」に対する EduBlocks の適合度分析

内田保雄[1]　玉城龍洋[2]　松島拓路[3]　坂本眞人[4]

概要：平成 30 年度高等学校学習指導要領改訂により，共通教科情報科は必履修科目の「情報 I」が令和 4 年度から実施されている．これを受けて，令和 7 年度大学入学者選抜に係る大学入学共通テストから「情報Ⅰ」が出題科目となった．その中では，プログラミングに関する問題も出題されるが，「出題する際のプログラム表記は受験者が初見でも理解できる大学入試センター独自の表記を用いる」となっている．一方高校の授業においては，多くの場合，Python，JavaScript，VBA，Scratch などのプログラミング言語を使用して授業がなされている．本研究は，それらに対してブロックプログラミング言語 EduBlocks を用いた場合の，大学入学共通テスト「情報Ⅰ」に対する適合度の分析を試みようとするものである．

キーワード：大学入学共通テスト，情報Ⅰ，EduBlocks，ブロックプログラミング言語

Analysis of EduBlocks' Suitability for “Informatics I” of the Common Test for University Admissions

YASUO UCHIDA[†1]　TATSUHIRO TAMAKI[†2]
TAKUMI MATSUSHIMA[†3]　MAKOTO SAKAMOTO[†4]

Abstract: According to the revision of the Guidelines for the Course of Study for Senior High School Students in 2008, “Informatics I” a required subject for the common subject of Information Science, has been implemented since the 2022 academic year. In response to the revision, “Information I” became a subject on the 2025 Common Test for University Admissions for University Entrance Examinations. In this test, questions related to programming are included, but the Center for University Entrance Examinations uses its own notation for the program so that examinees can understand it even if they see it for the first time. On the other hand, most high school classes use programming languages such as Python, JavaScript, VBA, and Scratch. This study attempts to analyze the suitability of EduBlocks, a block programming language, for the “Information Study I” section of the Common Test for University Admissions.

Keywords: Test for university admissions, Informatics I, EduBlocks, Block programming language

1. はじめに

平成 30 年度高等学校学習指導要領改訂により，共通教科情報科は必履修科目の「情報 I」が，令和 4 年度から実施されている．これを受けて，令和 7 年度大学入学者選抜に係る大学入学共通テストから「情報Ⅰ」が出題科目となった．その中では，プログラミングに関する問題も出題されるが,「出題する際のプログラム表記は受験者が初見でも理解できる大学入試センター独自の表記を用いる」となっている．一方高校の授業においては，多くの場合，Python，JavaScript，VBA，Scratch などのプログラミング言語を使用して授業がなされている．それらはテキスト型の言語であり，習熟すれば効率的にプログラミングが可能となるが，初学者にとってはやや敷居が高い．それらに対して，Edublocks[1]はブロックを組み合わせることでプログラムの作成が可能であり，入門レベルに適している．Edublocks は Anaconda 社によって無料で提供されている教育用のビジュアル型ツールであり，Python3 のコア部分が実装されている．本研究は，ブロックプログラミング言語 EduBlocks を用いた場合の，大学入学共通テスト「情報Ⅰ」に対する適合度の分析を試みようとするものである．

2. 先行研究・関連研究

ブロックプログラミング言語 EduBlocks を，Scratch から Python への移行におけるプログラミング教育の継続へのアプローチとして捉えた提案がある[2]．すなわち，Scratch は，その単純さにもかかわらず，最も単純なプログラムだけでなく，変数，ループ，分岐などのプログラミングの基本原理を反映した非常に複雑なプログラムも書くことができると指摘している．現在では，プログラミングを学ぶことは，大人にとっても子どもにとっても重要な活動となっている，と述べている．そして，実際には，ビジュアルなイベント駆動型言語 Scratch [3]が，7〜12 歳の子どもたちにプログラミングを教える最初のステップとなることが多いと説明している．一方，高級言語である Python プログラミ

1 宮崎産業経営大学
Miyazaki Sangyo-keiei University
2 沖縄工業高等専門学校
National Institute of Technology，Okinawa College
3 福岡県立明善高等学校
Fukuoka Prefectural Meizen High School
4 宮崎大学
University of Miyazaki

ング言語は,最近多くの人気を集めている,としながらも,テキスト言語への移行には困難が伴うと述べている．すべてのコードを独立して入力しなければならず，生徒にとって課題があまり面白くなくなり，すぐに結果が出ず，インデントで行き詰まり，多くの場合，この時点で生徒のプログラミングに対する興味は急激に失われるか，著しく低下する，と指摘している．そこで，継続的な学習の実施の一環として Scratch と Python プログラミングをよりスムーズに移行させるためには，中間段階を加える必要があり，そのために EduBlocks を用いることを提案している．

文献[4]では，フィリピンの数学教室において 7 年生を対象に，学校の幾何学のカリキュラムの中で，生徒が Python プログラミング，特に Edublocks をどのように活用しているかを探っている．その目的は，コンピュータ・コーディングの使用を通じて，数学的な概念を遊び感覚で，かつ非公式に習得する可能性を示すことである,と報告している．そこでは，生徒が Python プログラミングの活動の中で，幾何学における特殊な角度の組，特に補角と補助角の概念をどのように探求し，内面化したかに焦点を当てている．そして，Python プログラミングと Edublocks を併用することで，生徒たちは即座にフィードバックを得ることができ，数学の領域における問題解決への試行錯誤的なアプローチを育むことができた，とまとめている．なお，生徒は基本的なタートルの描画，タートルを使った角度の描画，タートルを使った図形の描画というプログラミング活動を行った．

文献[5]では，大学の初年次生を対象としたアルゴリズム学習科目において，フローチャート学習と EduBlocks を併用したプログラミング教育について紹介している.そして，テキスト型プログラミング言語を学ぶ前段階での，アルゴリズム教育におけるブロックプログラミング言語を利用した理解促進の可能性について述べている．

情報 I におけるプログラミング言語の選択が大学入学共通テストの解答に及ぼす影響を調べた取り組みがある[6]．まず,令和 4 年度から始まった「情報 I」の教科書に Python，JavaScript，VBA，Scratch の 4 つのプログラミング言語が使用されたことから，多くの高等学校がいずれかのプログラミング言語を使用して授業を行うことになる．そこで，クラスごとに 4 つのプログラミング言語を使い分けて同じ授業内容でプログラミング教育を実践し，最後に大学入学共通テスト「情報」サンプル問題（第 2 問）を全員に解答させた．そして，サンプル問題の解答結果や事後アンケートの結果から，どのプログラミング言語を授業で使用していても，大学入学共通テストの解答に大きな影響を及ぼす可能性が低いが，各プログラミング言語が持つ特有の表記や仕様により，少なからず解答に影響が生じていることが明らかとなった，と報告している．

共通テスト「情報」プログラミング問題の解決に必要な知識について検討した研究がある[7]．それは，共通テスト「試作問題」として公表されたプログラミング問題の認知的課題分析と，この問題を解決した大学生の発話プロトコル分析により，共通テストでのプログラミング問題で要求されている知識を明らかにしようとするものである．認知的課題分析では,「試作問題」を初見で解いた大学教員の発話プロトコルをもとに，この問題の解決に必要な知識をプロダクション・ルールとして明らかにしている．プロダクション･ルールは,手続き的知識の表現であり,IF-THEN 形式のルールである．条件（IF）節は問題の状態を，行為節（THEN）はその状態で行うべき操作を表している．大学生の発話プロトコル分析では,「試作問題」の解決を試みた大学生の発話プロトコルに基づき，認知的課題分析の妥当性を確認する．そして，大学生の問題解決過程は認知的課題分析によって提示されたプロダクション・ルールの使用を示唆するものであるか，また認知的課題分析でのプロダクション･ルールは教示に利用できるか,を検討している．

3. 試作問題について

3.1 令和 7 年度共通テストの出題教科・科目の問題作成方針に関する検討の方向性

大学入試センターにより令和 7 年度共通テストの出題教科・科目の問題作成方針に関する検討の方向性が示されている[8].

○ 新学習指導要領で示されている「情報I」で育成を目指すこととされている資質・能力を重視したものとなるよう検討する.
○ 今回公表する試作問題は以下の考えの下で作成した.
・日常的な事象や社会的な事象と情報との結び付き，情報と情報技術を活用した問題の発見・解決に向けての探究的な活動の過程，及び情報社会と人の関わりを重視する.
・社会や身近な生活の中の題材や受験者にとって既知ではないものも含めた 資料等に示された事例や事象について，情報社会と人との関わりや情報の科学的な理解を基に考察する力を問う問題などとともに，問題の発見・解決に向けて考察する力を問う問題も含めて検討する.
○ 試作問題の中にあるプログラム表記は，授業で多様なプログラミング言語が利用される可能性があることから，受験者が初見でも理解できる大学入試センター独自の日本語でのプログラム表記を用いた．令和 7 年度試験問題も同様の方向性で検討する.

したがって，その方針を確認しておく必要がある．

まず 1 つ目の項目においては,「情報I」で育成を目指すこととされている資質・能力が指摘されている．3 つの柱のうち「学びに向かう力，人間性等」の評価については「主体的に学習に取り組む態度」の観点が対応するが，ペーパ

ーテストでは評価が難しいと考えられる．また，「知識・技能」のうち知識については，単純に知識を求める問題は方針から外れることとなる．「技能」に関しては，プログラムの読解力や構成力などとして測ることができる．また「思考・判断・表現」については，論理的な分析力・考察力などとして評価できると考えられる．

2 つ目の項目においては，文献[9]のような解説が公開されている．

3 つ目の項目については，公平性の観点からも大学入試センター独自の日本語でのプログラム表記を採用することが示されている．

3.2 試作問題の概要

試作問題『情報 I 』は，図 1 のような構成となっており，実質的に，シミュレーション，プログラミング，データの活用に多くの配点がなされていることがわかる．

問題番号		選択方法	出題内容（平成 30 年告示高等学校学習指導要領との対応）	配点
第 1 問	問 1 ※1	全問必答	(1) 情報社会の問題解決	4
	問 2 ※2		(4) 情報通信ネットワークとデータの活用	6
	問 3		(3) コンピュータとプログラミング	6
	問 4		(2) コミュニケーションと情報デザイン	4
第 2 問	A ※3		(1) 情報社会の問題解決 (2) コミュニケーションと情報デザイン	15
	B ※4		(3) コンピュータとプログラミング	15
第 3 問 ※5			(3) コンピュータとプログラミング	25
第 4 問			(4) 情報通信ネットワークとデータの活用	25
			合計	100

図 1　試作問題『情報 I 』の構成

3.3 共通テスト用プログラム表記

高等学校の「情報 I 」の授業で使用するプログラミング言語は多様であることから，共通テスト『情報 I 』の試作問題作成にあたり，共通テスト用のプログラム表記を使用するとされた．そして，参考のためにその基本が例示されている．比較対象として示されているのは，フローチャート，Python3，JavaScript，VBA，Scratch，となっている．

そこで，本研究の前段として，令和 7 年度大学入学共通テスト試作問題『情報 I 』第 3 問・問 2 を対象として，共通テスト用プログラム表記および Python3 ならびに EduBlocks について類似度を考察した結果を図 2 に示す．類似度の欄における記号は，◎は表記がほぼ同じ，○は表記に差異があるが容易に理解できる，△は表記が異なるが動作は似ていることを示している．なお，EduBlocks は Python3 のコア部分が実装されているため，ほぼ同等と見なすことができるので，EduBlocks と共通テスト用プログラム表記との親和性は高いと判断できる．Python のコードの行と EduBlocks のブロックを比較すると，ほぼ対応していることが分かる．

4. 令和 7 年度本試験問題との適合度

ブロックプログラミング言語 EduBlocks を用いた場合の，大学入学共通テスト「情報 I 」に対する適合度の分析を試みる．

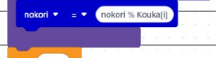
4.1 シミュレーション問題

令和 7 年度大学入学共通テスト情報 I 第 2 問 B は，表計算ソフトウェアの使用を想定したシミュレーションに関する問題となっている．

そこで，この問題の内容を EduBlocks で実装することにより，EduBlocks による学習が適応できることを確認する．実装で使用する主なプログラミング要素はリストである．

共通テスト用プログラム表記、Python、EduBlocks の類似度

（例）令和 7 年度大学入学共通テスト試作問題『情報 I 』第 3 問・問 2

共通テスト用プログラム表記	類似度	Python	EduBlocks
(1) Kouka = [1,5,10,50,100]	◎	Kouka = [1, 5, 10, 50, 100]	
(2) kingaku = 46	◎	kingaku = 46	
(3) maisu = 0, nokori = kingaku	○	maisu = 0; nokori = kingaku	
(4) iを4から0まで1ずつ減らしながら繰り返す：	△	for i in range(4, -1, -1):	
(5) ｜ maisu = maisu + nokori ÷ Kouka[i]	○	maisu = maisu + nokori // Kouka[i]	
(6) ｜_ nokori = nokori % Kouka[i]	◎	nokori = nokori % Kouka[i]	
(7) 表示する (maisu)	◎	print(maisu)	

共通テスト用プログラム表記と Python との類似度の説明
(3) 1 行に 2 つの命令文を書く時の記号（用法が異なる）
(4) 共通テスト用プログラム表記の繰り返し条件の「0 まで」はその値を含んでいるが、range(start, stop[, step])関数の stop はその値を含まないので、この場合は stop に「-1」を指定する必要がある。
(5) 共通テスト用プログラム表記「÷」は整数除算対応であるため、Python でも整数除算の演算子「//」を用いる。

図 2　共通テスト用プログラム表記，Python，EduBlocks の類似度

なお，この例では出力の表の体裁を整えるためのエスケープシーケンス（¥t）を用いているが，問題の本質に関わるものではない．

EduBlocks による実行例を図 3 に，EduBlocks のコードを図 4 に示す．

```
[6, 9, 1, 2, 6, 4, 7, 10, 9, 8]
        乱数r   一万円   千円
初期値    ー      0       0
1 人目    6       1       -4
2 人目    9       2       -8
3 人目    1       2       -2
4 人目    2       2       4
5 人目    6       3       0
6 人目    4       4       -4
7 人目    7       5       -8
8 人目    10      6       -12
9 人目    9       7       -16
10 人目   8       8       -20
```

図 3　表 1 出力の実行例

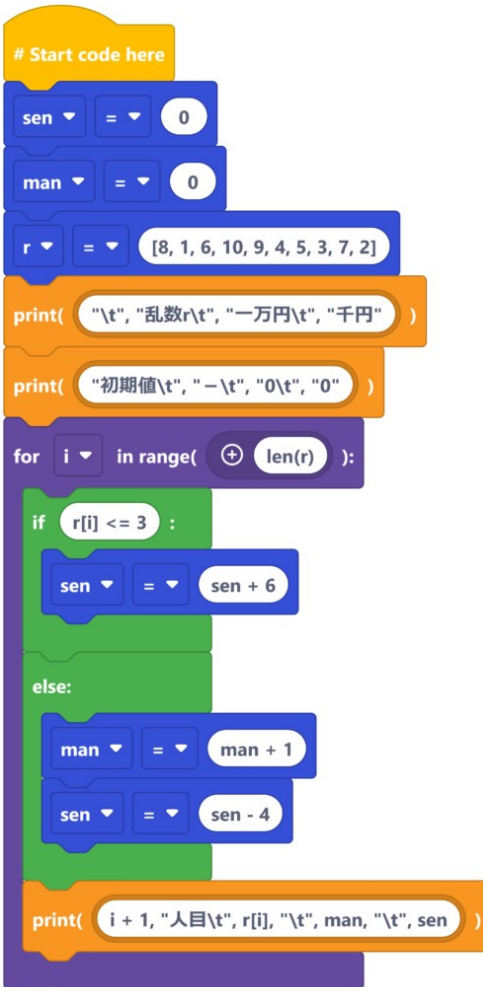


図 4　表 1 を出力する EduBlocks のコード例

なお，EduBlocks ではブロックでコードを作成すると，リアルタイムで Python コードに変換され表示される（図 5）．

これを見ると，EduBlocks では字下げがブロックにより自動的に実現されるため，Python で直接実装するよりコーディングの負荷が軽減され，よりアルゴリズムやデータ構造の理解に集中できる可能性がうかがえる．

Code

```python
#Start code here
sen = 0
man = 0
r = [8, 1, 6, 10, 9, 4, 5, 3, 7, 2]
print("¥t", "乱数r¥t", "一万円¥t", "千円")
print("初期値¥t", "ー¥t", "0¥t", "0")
for i in range(len(r)):
  if r[i] <= 3:
    sen = sen + 6
  else:
    man = man + 1
    sen = sen - 4
  print(i + 1, "人目¥t", r[i], "¥t", man, "¥t", sen)

```

図 5　変換された Python コード

4.2　プログラミング問題

第 3 問のプログラミングの問題については，前述のとおり共通テスト用プログラム表記との親和性が高い．ただし，今回の問題については，配列の添え字が 1 から始まると指定されたため注意が必要となる．Python や EduBlocks を用いた学習に慣れている場合には，添え字が 0 から始まる場合との差異に留意して問題に当たることが求められる．なお，実装して確認する場合には配列をリストに置き換えるとともに，0 番目の要素にダミーの値を設定しておく．ダミーの値としては，EduBlocks では Python と同じ「None」も利用できる．

たとえば，共通テスト用プログラム表記における配列の表現は，図 6 のように EduBlocks のブロックへ機械的に置き換えることができる．

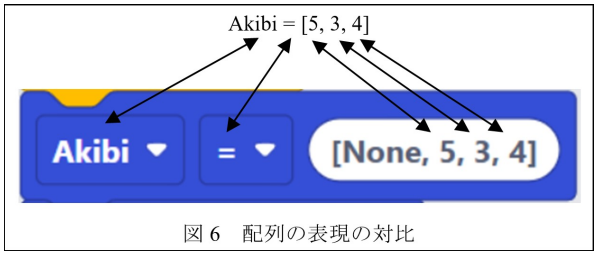


図 6　配列の表現の対比

共通テスト用プログラム表記における繰返しの指定は「m から n まで」のように表現されているため値 n を含むので，EduBlocks あるいは Python で range 関数を用いる場

合には stop 引数の値に 1 を加え必要がある．

たとえば，図 7 のように修正すれば容易に置換できる．

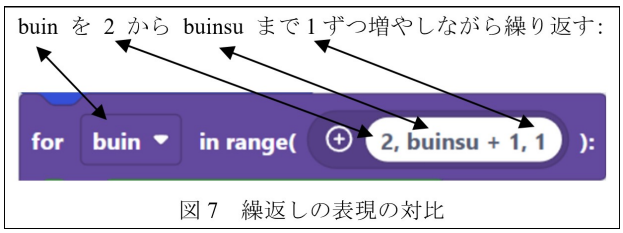


図 7　繰返しの表現の対比

条件分岐は，図 8 のようにほとんどそのままの表現である．

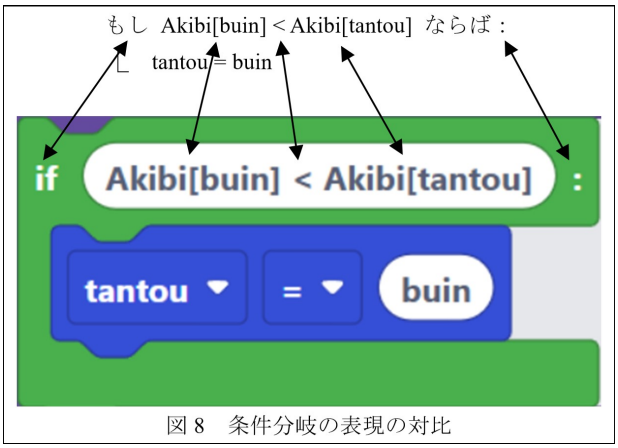


図 8　条件分岐の表現の対比

4.3　データの活用問題

第 4 問については，今回の問題に関して言えば箱ひげ図の表示など EduBlocks による実装では難しい部分もあるが，棒グラフ，帯グラフ，散布図については，容易に作成できるため基礎的な学習においても活用が期待できる．

たとえば，問題中の「図 1　表 1 のデータに基づいて作成した棒グラフと帯グラフ」の棒グラフは図 9 のように，図 10 に示すプログラムにより得ることができる．

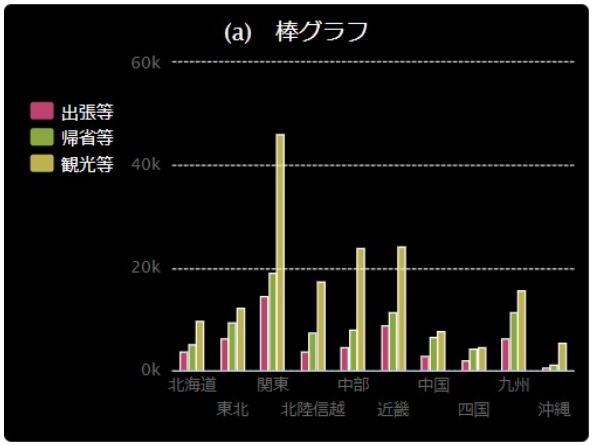


図 9　問題中の図 1 (a) 棒グラフ相当の出力

図 10　問題中の図 1(a)を出力するプログラム

このように，EduBlocks では描画パッケージである「Pygal」[10]が利用できるので何種類かのグラフ表現が可能である．

また，Python の matplotlib.pyplot モジュール[11]も準備されているため散布図も容易に出力できる．

たとえば，問題中の「図 3　出張等と観光等の旅行者数の組み合わせの散布図」は図 11 のように，図 12 に示すプログラムにより得ることができる．

相関係数の算出に関しては，簡単な関数などは利用できないため，実装する場合には煩雑となる．しかしながら，相関係数そのものを求めるような問題は出題されないと想定されるので問題はないと考えられる．なお，基本的な方法による実装は可能であり，アルゴリズムの学習にも併用できる[12].

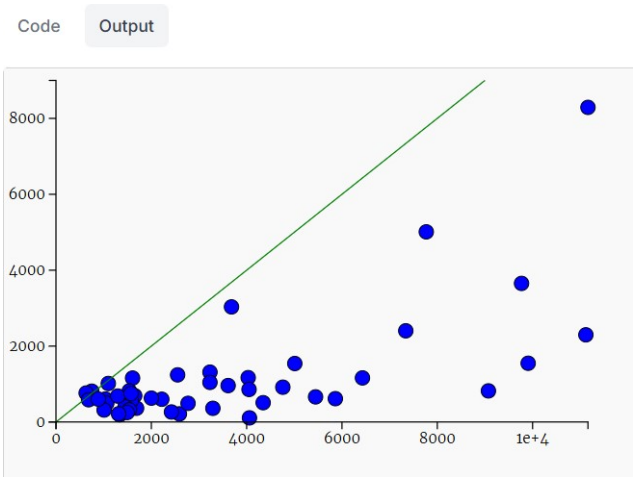


図 11　EduBlocks による問題中図 3 相当の出力

図 12　問題中の図 3 を出力するプログラム

5. おわりに

令和 7 年度大学入学共通テスト情報 I 第 2 問 B から第 4 問を対象として，EduBlocks による学習が大学入学共通テストにも適合できるかどうかについて検討した. その結果，アルゴリズムやデータ構造といったプログラミングの基礎学習については他の Python 環境と遜色なく，むしろコーディング負荷の観点からは効率的な学習につながることが期待できると想定できる.

今後の課題としては，実際に高校での学習に取り入れる実践を通じて，その有効性を検証することが求められる.

参考文献

[1] Anaconda: EduBlocks （オンライン），入手先 〈https://edublocks.org/〉（参照 2025-02-08）.
[2] Belokonova, S. S. and Plotnikova, M. S.: Block Python as a transition stage in learning programming, Informatics in school, (7), pp. 57-62 (2021). (In Russ.)
[3] Massachusetts Institute of Technology: Scratch （オンライン），入手先 〈https://scratch.mit.edu/〉（参照 2025-02-08）.
[4] Francis, S. M.: EXPLORING GEOMETRIC CONCEPTS THROUGH PYTHON PROGRAMMING ACTIVITIES, 2024 International Conference on UN SDGs and Social Innovation, pp. 1-15 (2024).
[5] 内田保雄，玉城龍洋，大西淳：ぺた語義：ブロックプログラミング言語を援用したアルゴリズム教育 —EduBlocks を用いた授業実践—，情報処理, Vol. 63, No. 11, pp. 612-615 (2022).
[6] 井手広康：情報 I におけるプログラミング言語の選択が大学入学共通テストの解答に及ぼす影響，情報処理学会論文誌教育とコンピュータ，Vol. 9, No. 1, pp. 1-10 (2023).
[7] 寺尾敦，秋岡敬子，中村優花：共通テスト「情報」プログラミング問題の解決に必要な知識—発話プロトコル分析—，2024PC カンファレンス，4 pages (2024).
[8] 大学入試センター：令和 7 年度大学入学者選抜に係る大学入学共通テストの出題教科・科目の問題作成方針に関する検討の方向性について （オンライン），入手先 〈https://www.dnc.ac.jp/albums/abm.php?d=511&f=abm00003128.pdf&n=3_令和 7 年度共通テストの出題教科・科目の問題作成方針に関する検討の方向性について.pdf〉（参照 2025-02-08）.
[9] 大学入試センター: 令和 7 年度大学入学共通テスト 試作問題「情報」の概要 （オンライン），入手先 〈https://www.dnc.ac.jp/albums/abm.php?d=511&f=abm00003141.pdf&n=6-1_概要「情報」.pdf〉（参照 2025-02-08）.
[10] Florian Mounier: Pygal （オンライン），入手先 〈https://www.pygal.org/〉（参照 2025-02-08）.
[11] The Matplotlib development team: Matplotlib （オンライン），入手先 〈https://matplotlib.org/〉（参照 2025-02-08）.
[12] 内田保雄，坂本眞人：IT リテラシーとしてのコンパクトなプログラミング入門教材の提案，令和 6 年度私情協教育イノベーション大会抄録, p. 206 (2024).