

プログラミング生成 AI の衝撃

～プログラミング教育に AI は活用できるのか～

ライフデザイン研究部 主席研究員 柏村 祐

1.生成AIはプログラミング教育に活用できるのか

情報技術の進化に伴い、デジタル化が加速し、多くの産業分野で技術革新が進んでいる。そのため、プログラミングスキルは、現代社会で生き抜くための基本的な能力となっており、教育現場でもその重要度は以前よりも増している。

2020年度より小学校でのプログラミング教育が必修化されたが、プログラミングのスキルを身に付けるだけでなく、小学校段階における論理的思考力や創造性、問題解決能力等の育成も目的とされている。また、中学校では2021年度から、高校では2022年度から、プログラミング教育が導入されている。さらに、2025年以降の大学入学共通テストの出題科目に「情報」が追加されることから、プログラミング教育に対する注目度はさらに高まるだろう。

プログラミング教育のあり方に関する文部科学省の有識者会議のとりまとめによれば、プログラミングに必要なコーディング（プログラミング言語を用いた記述方法）が時代と共に変化するため、単にコーディングを覚えるのではなく、独自の思考を形にするプログラミング的思考力や行動力の育成が重要だとされている。柔軟に時代の変化に対応できるよう、普遍的に求められる資質・能力を身に付けることが最大の目的とされる（注1）。

このプログラミング教育に対して、近年の AI 技術の進歩に伴い、新たな形が求められている。生成 AI は、近年進歩している AI 技術の中でも特に教育分野において活用の幅が広がる可能性がある。生成 AI は、データを学習し、新たな情報やコンテンツを生成する AI 技術である。テキスト、画像、音声などのさまざまな形式の生成が可能であり、生成物は人間が作成したものと見分けがつかないほど高品質な場合もある。

本レポートでは、プログラミング生成 AI の能力について概観し、プログラミング教育における活用方法について解説する。

2.生成 AI がもつプログラミング能力

プログラミング生成 AI（以下生成 AI）がもつプログラミング能力は、プログラム生成・プログラム解説生成・プログラム問題生成に分類される。ここからは、これらの生成能力について具体事例を確認しながらその内容について確認していく。

まず、プログラム生成とは、自然言語で記述された文章からプログラムのコードを生成することを指す。これにより、学習者は実際のプログラム例を参考にしながら、

理解を深められる。実際、生成 AI に対して「Python でジャンケンゲームを作成してください」と要件を出してみると、生成 AI は、一瞬にして Python で動作するジャンケンゲームのプログラムを生成する（図表 1）。Python は、オープンソースの高水準プログラミング言語であり、広く使用されている。

図表 1 生成 AI がジャンケンゲームのプログラムを生成する様子



Pythonでジャンケンゲーム（Rock-Paper-Scissors）を作成する方法を以下に示します。このコードは、ユーザーがコンピュータと対戦するように設定されています。

```
python Copy code

import random

def janken_game():
    print("ジャンケンゲームへようこそ!")
    print("0: グー, 1: チョキ, 2: パー")

    choices = ["グー", "チョキ", "パー"]

    while True:
        user_input = input("あなたの手を選んでください (0, 1, 2, or 'q'で終了):")
        if user_input == 'q':
            print("ゲーム終了。また遊んでね!")
            break
```

資料: ChatGPT より筆者作成

次に、プログラム解説生成に関する生成 AI の対応について確認してみよう。生成 AI は、コードの解説文を自動生成することができる。これにより、学習者はコードの機能や目的を理解しやすくなり、効率的な学習が可能になる。たとえば、世界で最も有名なプログラムとして、プログラムを実行すると画面に「Hello, World!」と出力されるプログラムがある。Pythonでこのプログラムを書くと「print("Hello, World!")」という記述が必要となるが、この内容を生成 AI に読み込ませ「何を意味しているか解説してください」と質問してみた。その結果、生成 AI は、このプログラムの意味について、プログラム初心者が理解できるレベルの内容を順序立ててわかりやすく解説してくれる（図表 2）。

図表 2 生成 AI が「Hello, World!」プログラムを解説する様子



このプログラムは、非常にシンプルなPythonプログラムで、「Hello, World!」というメッセージを画面に出力することが目的です。このプログラムは、プログラミング言語を学ぶ際の最初のステップとしてよく使われます。具体的には、次のように動作します。

1. `print` 関数を呼び出しています。この関数は、引数として与えられた値を画面に出力する役割を果たします。
2. `print` 関数に `"Hello, World!"` という文字列を引数として渡しています。この文字列は、ダブルクォートで囲まれたテキストです。
3. プログラムが実行されると、`print` 関数が呼び出され、引数の `"Hello, World!"` が画面に表示されます。

要するに、このプログラムは「Hello, World!」というメッセージを出力するだけの単純なものですが、プログラミングの基本概念を理解するための良い出発点となります。


資料: ChatGPT より筆者作成

最後に、プログラム問題生成に関する生成 AI の対応について確認してみよう。生成 AI は、学習者の状況に応じて適切な指導を提供することができる。これにより、個別指導の効率化が図られる。実際に生成 AI に対して「中学生向けのプログラミング問題を 2 つ作成してください」と文章を入力してみた。その結果、生成 AI は、瞬時に 2 つのプログラミング問題を生成した（図表 3）。

以上みてきたように、生成 AI は、プログラミング教育に関連するプログラム生成・プログラム解説生成・プログラム問題生成について、自然言語で記述された文章さえ入力すれば、文章の内容に応じた的確な回答を提示してくれる。

この事実は既に生成 AI が、プログラミング教育の中核機能を代替できることを示唆している。このような生成 AI の進化を踏まえれば、現在教師が生徒にプログラミングを教えるという仕組みに加えて、プログラミング教育の現場に生成 AI を導入し、活用できるのではないだろうか。

図表 3 生成 AI が中学生向けのプログラム問題を生成する様子

 **問題1: 簡単な電卓プログラム**

目的: 2つの整数を入力として受け取り、それらの和、差、積、商を出力する簡単な電卓プログラムを作成する。

要件:

- ユーザーから2つの整数を入力として受け取る。
- 入力された2つの整数の和、差、積、商を計算する。
- 計算結果を表示する。

問題2: フィボナッチ数列

目的: ユーザーが指定した長さのフィボナッチ数列を生成するプログラムを作成する。

要件:

- ユーザーから数列の長さ（整数）を入力として受け取る。
- 入力された長さのフィボナッチ数列を生成する。
- 生成された数列を表示する。

資料: ChatGPT より筆者作成

3.生成 AI を活用するプログラミング教育の未来

生成 AI の活用により、プログラミング教育において以下のような革新が期待される。

まず、生成 AI は学習者の過去の学習履歴や理解度を分析し、個別化されたカリキュラムや教材を提供することができる。これにより、学習者は自身に適した速度で学習を進めることができ、効果的な学習を実現できることにつながる。また、生成 AI はプログラミングの課題に対して自動的にプログラムを生成し、学習者に提案できる。さらに、学習者が書いたプログラムを分析し、エラーやバグを指摘するだけでなく、最適なプログラムの提案も行える。これにより、学習者はより効率的にプログラムの理解と改善を行える。

次に、生成 AI を活用したオンライン教育プラットフォームでは、インタラクティブな学習環境を提供することが可能となる。たとえば、チュートリアルや課題の進行に合わせてリアルタイムで質問に回答し、必要に応じて追加の説明を提供することで、学習者の理解を深められる。また、多言語対応やグローバルな協力学習が容易になり、プログラミング教育のアクセシビリティが向上することが期待される。

一方で、生成 AI をプログラミング教育に適用する際には、以下のような懸念事項や

課題がある。まず、生成 AI の応用によって、学習者が不正行為を行いやすくなる。たとえば、自動コード生成機能を利用して、課題の解答を AI に生成させることで、自らの理解を深めずに単に成績を得ることができる場合が生じる。これに対処するためには、AI の活用範囲を適切に制限し、教育者が適切な評価方法を検討する必要がある。また、生成 AI に依存しすぎることで、学習者のプログラミングスキルが低下する恐れがある。学習者が生成 AI の提案に過度に頼ることで、自らの問題解決能力やコードの読解力が向上しない場合が生じる。これを防ぐためには、生成 AI の適切な利用方法を指導し、学習者が自ら考える機会を確保することが重要となる。

生成 AI を活用することで、プログラミング教育において個別化された学習体験、自動コード生成とフィードバック、オンライン教育プラットフォームの発展が期待される。しかし、倫理的な問題やプログラミングスキルの低下といった課題も存在する。今後は、生成 AI の適切な活用方法を模索し、これらの課題に対処しながら、効果的で革新的なプログラミング教育の実現を目指すことが重要となる。教育者と開発者が連携し、生成 AI の利点を最大限に活用しつつ、潜在的なリスクを最小限に抑える教育プログラムの開発が求められる。

加えて、生成 AI と学習者が共同で学習プロセスに参加し、互いに向上し合うことで、より高度なプログラミング教育が実現できるだろう。これにより、教育者はより高度なスキルや知識を学習者に伝えることができ、学習者は自らの理解を深め、独自の創造性を発揮することにつながる。生成 AI がプログラミングに必要なコーディング作業を代替できる現在、求められる能力開発は、オリジナリティのある考えを形にする思考力や行動力ではないだろうか。激しい変化の時代に柔軟に対応できるよう、普遍的に求められる資質・能力を身に付けることが必要だろう。

生成 AI の発展により、より多くの人々がプログラミング教育にアクセスできるようになり、その結果、社会全体がデジタル時代に適応しやすくなることが期待される。

【注釈】

1) 文部科学省 HP より

https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm