



# 生成 AI を活用したプログラミング教育の実践



東京学芸大学附属高等学校  
飯田秀延

# 本校の概要

- ・東京都世田谷区(渋谷からバスで20分弱)
- ・1954年創立(71期生)
- ・東京学芸大学の附属高校(教育研究校)
- ・各学年8クラス(一般120名, 附属中200名, 帰国15名)
- ・教員数82名(情報科は1名)
- ・ほとんどの生徒が4年制大学へ進学



東京府青山師範学校(1936年)

東京学芸大学大学史資料室 より  
<https://www.u-gakugei.ac.jp/shiryoshitsu/>



現在

「プログラミング」の授業はこの程度(年間で10時間前後)

## Contents

### 1章 Theory

#### 情報社会

01 情報とその特性	2
02 メディアとその特性	4
03 問題を解決する方法	6
04 情報の収集と分析	8
05 解決方法の考案	10
06 知的財産	12
07 個人情報	14
08 情報セキュリティ	16
09 情報モラルと個人の責任	18
10 情報技術の進歩と役割	20
11 情報技術が社会に与える光と影	22
章末資料	24
1 検索の順位付け	
2 SNSへの投稿	
3 サイバー犯罪	
4 ICTの普及と識字の関係	
5 人工知能と人間の仕事	
6 インターネットと選挙	
私の問題解決	27
1章のまとめ	28
章末問題	29

### 2章 Theory

#### 情報デザイン

12 コミュニケーションとメディア	32
13 情報のデジタル化	34
14 数値の表現	36
15 2進法の計算	38
16 文字のデジタル表現	40
17 音のデジタル表現	42
18 画像のデジタル表現	44
19 データの圧縮	46
20 デジタルデータの特徴	48
21 メディアと文化の発展	50
22 ネットコミュニケーションの特徴	52
23 情報デザイン	54
24 操作性の向上と情報技術	56
25 全ての人に伝わるデザイン	58
26 コンテンツ設計	60
章末資料	62
1 機軸依存文字	
2 色の表現	
3 ラスタ形式とベクトル形式	
4 ピクトグラムと東京オリンピック	
5 デザインで問題解決	
6 色の識別性	
私の問題解決	65
2章のまとめ	66
章末問題	67

### 3章 Theory

#### プログラミング

27 コンピュータの構成	70
28 ソフトウェア	72
29 処理の仕組み	74
30 論理回路	76
31 アルゴリズムの表現	78
32 プログラムの表現	80
33 プログラムの仕組み	82
34 プログラミング入門	84
35 プログラムの応用	86
36 問題のモデル化	88
37 モデル化の活用	90
38 シミュレーション	92
章末資料	96
1 補助記憶装置	
2 時間を主役とした地図	
3 固定小数点表現と浮動小数点表現	
4 プログラミング言語の年表	
私の問題解決	99
3章のまとめ	100
章末問題	101

### 4章 Theory

#### ネットワークの活用

40 情報通信ネットワーク	104
41 デジタル通信の仕組み	106
42 インターネットの利用	108
43 安全安心を守る仕組み	110
44 情報システム	112
45 さまざまな情報システム	114
46 情報システムの信頼性	116
47 データの活用とデータベース	118
48 データの管理	120
49 データの収集と種類	122
50 データの分析	124
51 不確実な事象の解釈	126
52 2つのデータの関係	128
章末資料	130
1 通信規格の推移	
2 海底ケーブル	
3 ルートDNSサーバ	
4 生体認証	
5 グラフを疑う	
6 ガンブラーの誤算	
私の問題解決	133
4章のまとめ	134
章末問題	135

### 5章 Action

#### 問題解決

53 検索のコツ	138
54 仕事の研究	139
55 アイディアの大量生産	140
56 光の三原色体験	141
57 データ量の見積もり	142
58 図解表現	143
59 ピクトグラム	144
60 部活紹介CM	145
61 Webニュースページ	146
62 作図しよう	148
63 プログラムで動きを再現	149
64 お知らせセンサ	150
65 気まぐれAI	152
66 左右限定あっち向いてホイ	154
67 プログラムの改善	156
68 Myお天気キャスター	158
69 災害時帰宅マップ	160
70 シミュレーションをしよう	161
71 誕生日シミュレーション	162
72 高校生の実態調査	163
73 コンビニデータベース	164
章末資料	166
1 意思伝達装置を活用した物理学者	
2 SDGs	
3 プレゼンテーション	

Python      6時間  
Processing    4時間



Processingは「楽しい」  
しかし・・・

- ・新たな言語習得は大きな負担
- ・プログラミングスキルがないと「やりたいこと」が実現できない

## 生成AIで補助

新たなプログラミング言語を習得する負担を軽減しつつ、「やりたいこと」を実現するためのスキルを補うことによって、プログラミング教育の学習効果を高めることが期待できる。

制作4時間(Processingの学習含む) + 相互評価1時間

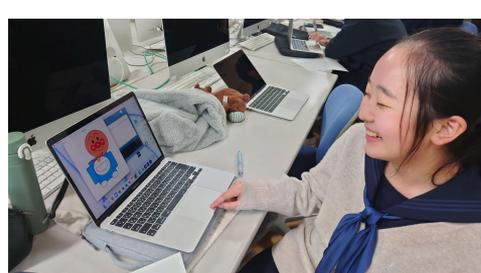
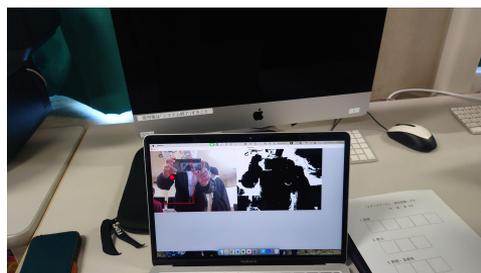
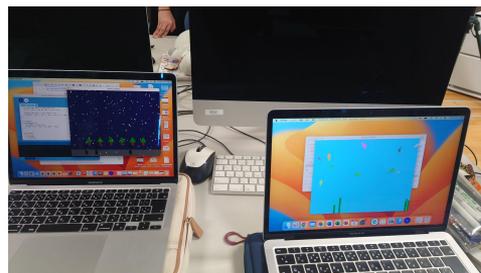
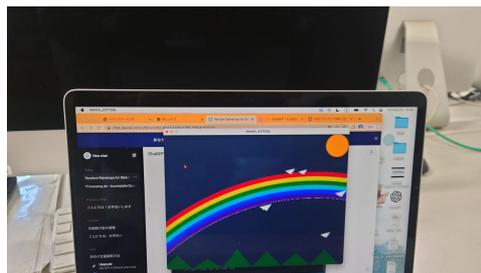
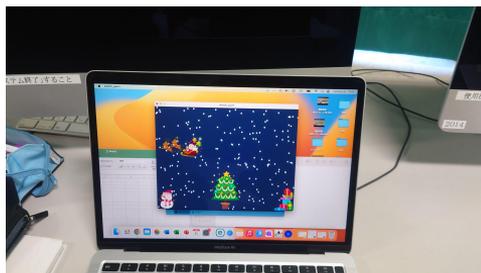
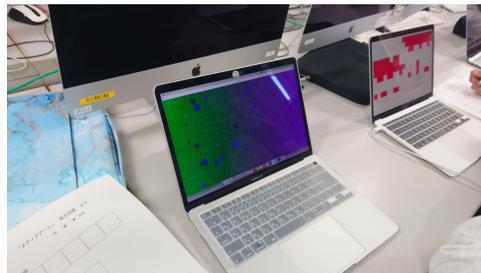
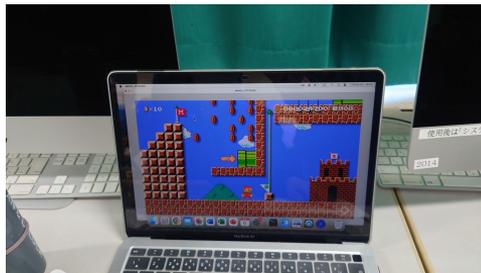
評価の観点「技術」「努力」「芸術性」

制作過程で何を調べても良い

生成AIの使用は自由(使っても使わなくてもいい)

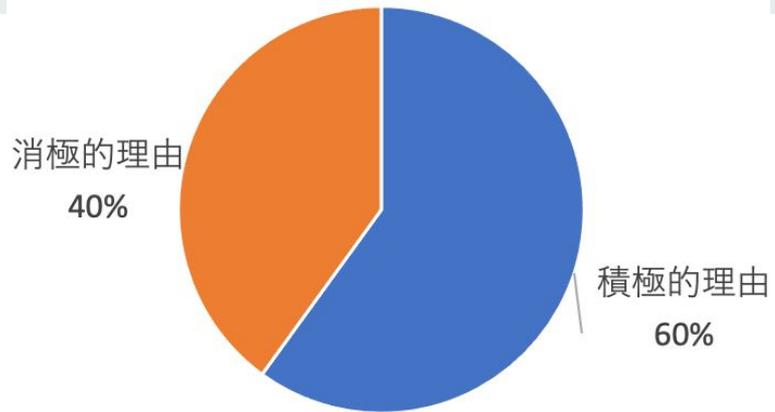


# 作品例



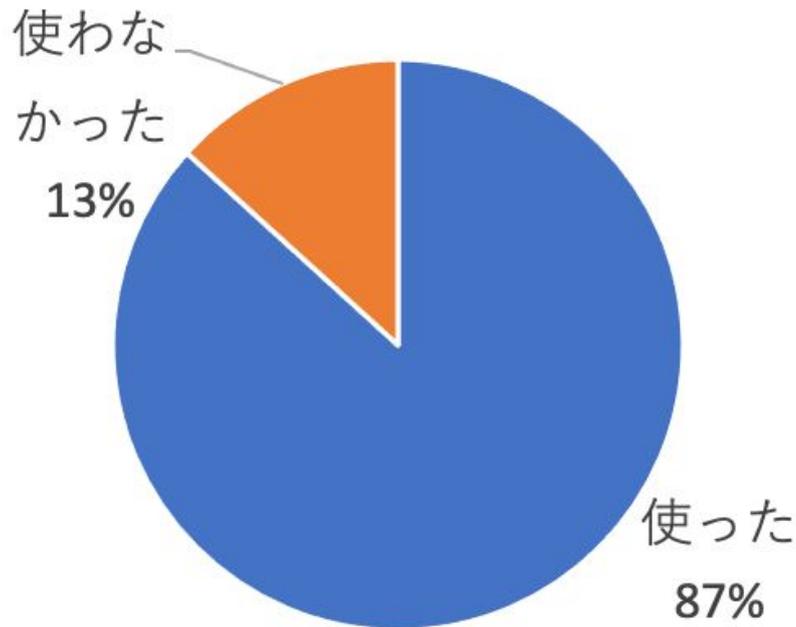
## 相互評価(ギャラリーウォーク)の様子





生成AIを使用しなかった理由

「消極的な理由」で使用しなかった生徒は少数(5.2%)



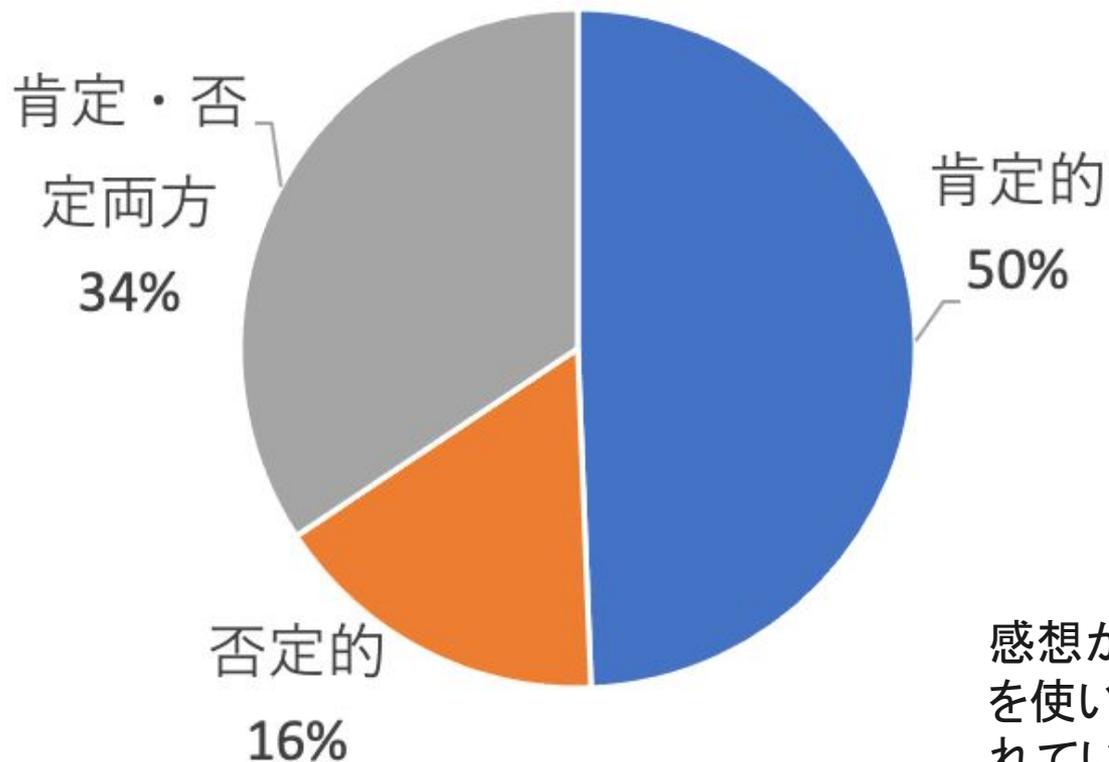
生成AIの使用の有無

## 「使った理由」

コードの法則性や使い方が完全には理解できていないため、作りたいものがあるならChatGPTに作ってもらったコードを少し弄った方が発想が広がるだろうと思ったから。

まだprocessingの記号や関数を理解できていないため、ヒントが欲しかったため。でも、自分がこだわりのかったものは自力で頑張りました。

試しに使ってみたら、言うことを全て聞いてくれて嬉しかったから。



感想からは、多く(84%)の生徒がAIを使いながら自分の知識を深められていたと考えられる。

## 生成AIに対する感想

## 「使ってみて良かった点」

使ってみて、殆どは対応してくれたように思います。しかし、固有名詞や、流行りのものを実現してもらうのが、難しかったように思います。

自分の思い描くものを作らせるのが難しかった。ただ思いもしなかったアイデアもそこから出てくるので作品の質が向上したと思う。

使ってみて良かった点は、結構抽象的な要望をしてもしっかり答えてくれるので、コードをうまく作ることができた。よくなかった点はあまり同じタブを使いすぎると、回答が前に答えたコードによっておかしくなるのでAIの学習機能に注意が必要だった。

## 「使ってみて良かった点」

かなり想像した通りのコードを書ってくれたのでよかった。本格的に生成AIを使ったのが初めてだったので特に驚いたのは、プロンプトが以前の文脈も読み込んだ上でコードを書ってくれたのが楽だった。

元の土台から自分のしたいようにどんどん変えていけるのはやっていて楽しいし適度に難しいからいいバランスだと思った。めっちゃこの授業楽しかったです！！

出力されたコードを読み解くことでとても力がついた(気がした)し、**出てきたコードを実行する瞬間がワクワク、とても楽しかった。**

8割以上の生徒の感想から

- ・自分の考えを優先しながらAIを活用できる
- ・AIを使いながらも自分の知識を深めることができる



## プログラミングの授業に生成AIを活用すると・・・



新たなプログラミング言語を習得する負担を軽減しつつ、「やりたいこと」を実現するためのスキルを補うことによって、プログラミング教育の学習効果を高めることが期待できる。

**※ただし、基本的なコーディングの知識は必要**



キーワードを入力



トップ

速報

ライブ

エキスパート

オリジナル

みんなの意見

ランキング

主要

国内

国際

経済

エンタメ

スポーツ

IT

科学

ライフ

地域

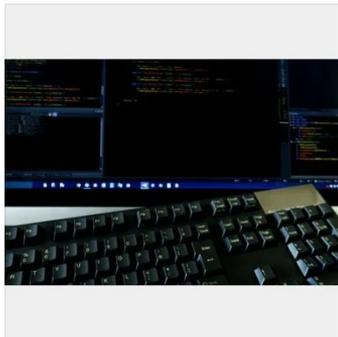
## 生成AIに依存し過ぎると人間はバカになるのか？実験が明らかにした、生成AIを賢く使える人、使えない人

6/8(土) 17:21 配信

39



JBpress



簡単なプログラミングなどは生成AIが軽くやってくれる時代だが……

(写真：Wakko/イメージアート)

生成AIを活用すれば、プログラミング言語やアルゴリズムなどの知識やスキルがない人でもコーディングが可能な時代になった。

だが、最新の研究結果によれば、コーディングが得意な学生は生成AIを活用することで効率的にタスクを処理したが、苦手な学生は生成AIを活用することで、逆にスキルの習得から遠ざかる結果になった。

生成AIによって何でも効率的にできると思いがちだが、AIに頼るにしても、賢くなったのはAIであり、自分自身の継続的なスキルアップが不可欠だ。

(小林 啓倫:経営コンサルタント)

6/8(土) Yahoo!  
ニュース

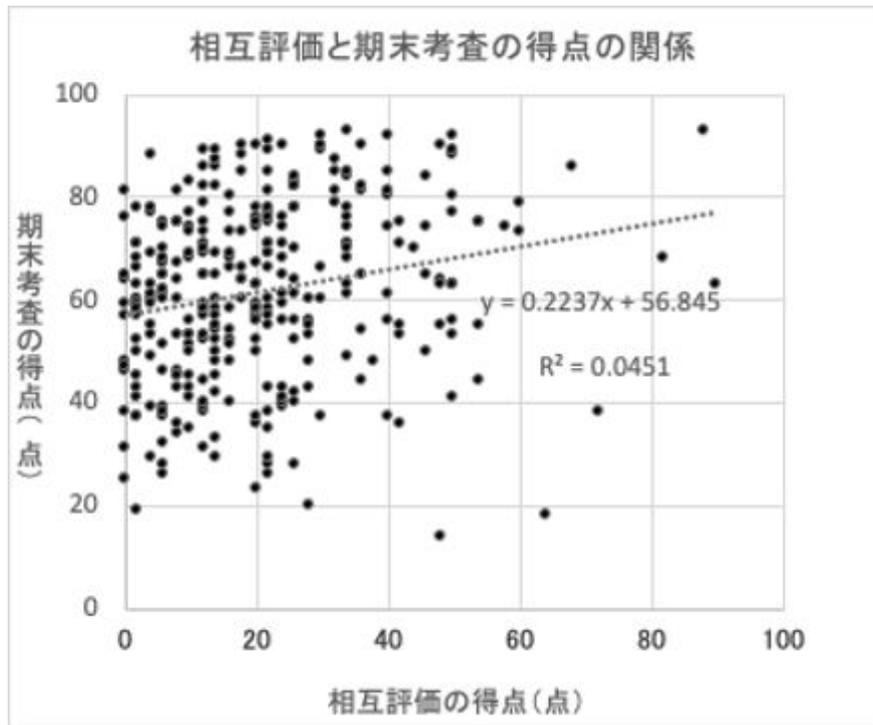
えたのかを確認したのである。

結果はどうだったか。結論から言うと、もともとコーディングが得意な学生については、AIの提案を適切に取捨選択しながら効率的に課題を解くことができた一方で、コーディングが苦手な学生は、AIの提案に振り回されて本質的な理解が追いつかず、結果的に自分の力でプログラムが書けなくなってしまう傾向が見られたそうである。

もともとコーディングの授業において良い成績を収めていた学生は、コード生成AIの提案を吟味し、適切なものを選択した。その結果、効率的に課題に取り組むことができ、自分の考えを優先しながらAIを活用する姿勢が見られた。そうした学生は、AIを使いながらも自分の知識を深めることができたと研究者らは結論付けた。

「プログラミングが得意な生徒ほど生成AIをうまく使いこなせる」は本当か？

うだ。



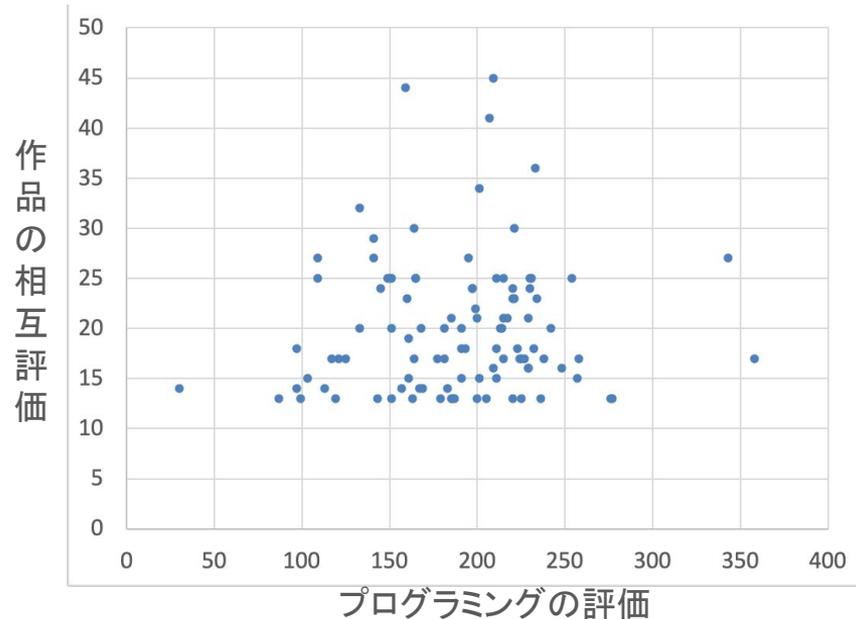
相互評価(生成AIをうまく使いこなせていたか)と期末考査の得点の相関係数は0.21

相関係数の目安

- 0.0~0.2 ほとんど相関関係がない
- 0.2~0.4 やや相関関係がある
- 0.4~0.7 かなり相関関係がある
- 0.7~1.0 強い相関関係がある

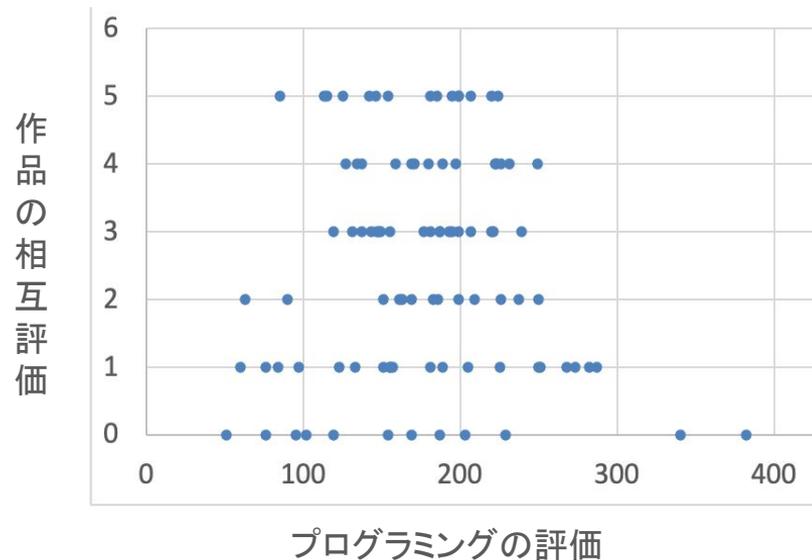
作品の相互評価 **上位100人**の  
プログラミングの評価と作品の相互評価の関係

相関係数 **0.068**



作品の相互評価 **下位100人**の  
プログラミングの評価と作品の相互評価の関係

相関係数 **-0.029**



プログラミングの評価 **上位100人**の  
プログラミングの評価と作品の相互評価の関係

相関係数 **-0.22**

プログラミングの評価 **下位100人**の  
プログラミングの評価と作品の相互評価の関係

相関係数 **0.17**

あくまで今回の授業に関してですが



「プログラミングが得意であることと生成AIをうまく使いこなせることはあまり関係がない」？！

「情報」以外の教科と関係がある？（国語力？）

表1 作品の評価と期末考査の得点の関係(情報以外)

科目	現代の国語	言語文化	地理総合	歴史総合	数学 I	数学 A
相関係数	<b>0.10</b>	<b>0.16</b>	<b>0.16</b>	<b>0.20</b>	<b>0.18</b>	<b>0.17</b>

科目	生物基礎	地学基礎	保健	英コミュI	論理・表現I	SSH 探究
相関係数	<b>0.16</b>	<b>0.20</b>	<b>0.13</b>	<b>0.12</b>	<b>0.13</b>	<b>0.08</b>

## (生成AIによる)評価の難しさの要因5つ

### 1. 学習アプローチの違い

学習アプローチやスタイルが異なるため、一部の生徒は伝統的な教科の成績が高くても、生成AIのような新しい分野に適応するのが苦手なのかもしれない。

### 2. 興味や動機の違い

生徒が興味を持っている分野や、将来の職業への動機が生成AIとは無関係であるのかもしれない。

### 3. 成績評価の偏り

学校の成績評価体系が、生成AIでのスキルや知識を正確に測定できないのかもしれない。

### 4. 教育カリキュラムの不足

学校の教育カリキュラムがまだ生成AIや先端技術に焦点を当てていないのかもしれない。

### 5. 個々の適性や才能

生成AIを使いこなすための適性や才能が、他の教科の成績と直接関連しないのかもしれない。

→要するに、今回の評価規準(基準)がよくなかった





## 「評価」をどうするか？

作品(アウトプット)を評価することはできるが、  
「生成AIをうまく活用できたこと」(過程)をどのように評価  
するのか？

## 結論

基本的なコーディングの知識を取得している生徒がプログラミングの授業に生成AIを活用すると自分の考えを優先しながらAIを活用することやAIを使いながらも自分の知識を深めることができ、プログラミング教育の学習効果を高めることが期待できる。



「評価」をどのようにするのが今後の課題