

高校情報科のプログラミング学習に関する生成 AI 活用の研究

A Study on Programming Learning in High School 'Informatics'
through Use of Generative AI

市川隆司

大阪信愛学院大学 教育学部

要旨

2022 年度より実施の高校学習指導要領で情報科ではプログラミングが必須となった。2022 年に公開された生成 AI は社会の様々な分野に影響を及ぼしており、教育分野もその例外ではない。本研究はプログラミング学習に生成 AI をどのように活用できるか、教科書の例題からその可能性を探るものである。プログラミング言語 Python を対象としてプロンプトの設計によって目的のコードを出力できることが確認できた。学習者にとって出力コードとともに表示される解説はコード内の設定内容のみならず、プログラムの改良や発展の理解につながる。

キーワード：プログラミング, AI, 情報科, 高校

1. はじめに

2018 年の高等学校学習指導要領改訂により科目編成された情報科では、情報技術の急激な進展から「情報の科学的な理解」に基づく情報活用能力の育成が重要視されることになった。情報の科学的な理解において、情報技術の適切かつ効果的な活用としてプログラミングが取り上げられている¹⁾。そのため共通必修科目として新設された「情報 I」では、すべての生徒がプログラミングを学習することとなった。プログラミング教育で育む資質・能力は、全ての学習の基盤となる資質・能力である情報活用能力の一部であり、全ての学校段階の学習指導要領の総則において、情報活用能力を育成することと規定されている²⁾。

プログラミング教育の必要性が求められる一方で情報科を担当する教員に対する課題も指摘されている。文部科学省の調査によれば、全国の公立高校で教諭臨時免許状（情報）の授与を受けた者又は情報の免許外教科担任の許可を受けた者が情報科を担当している割合が 2022 年に約 16%となっている³⁾。2023 年の調査では全国の公立高校で情報科を担当する教員の免許取得状況は改善されているが、依然としてさらなる改善が必要であることを指摘されている⁴⁾。また担当教員に対して一層の指導力向上の取り組みを求めている⁴⁾。

2022 年度から実施された学習指導要領における教育課程で学習を進めた高校生は、2025 年度大学入学共通テストで出題科目として追加された教科「情報」を受験することになる。大学入試センターが提供するサンプル問題において、プログラミングに関する内容が出題されており、実際の試験においても出題の可能性が高い。そのためプログラミングは受験生にとって習得が必要な学習内容となっている。

生成 AI が社会的に注目されるようになったのは、2022 年 11 月に公開された ChatGPT(GPT-3.5)の登場である。生成 AI は大規模言語モデル(LLM)に基づいて構築されており、対話形式で自然言語に対応して回答するのでさまざまな分野に影響を及ぼすことになった。その後、Google の Gemini をはじめとして次々と同様のサービスが提供されてきた。対話によって生成される回答は、文章の要約や校正、企画やアイデア、翻訳など幅広い分野に及ぶが、プログラミング言語のコードの生成が可能である。

生成 AI の回答には、ハルシネーション（幻覚）と呼ばれる誤った回答を出力する現象があることを認めなければならないが、

初学者に向けたプログラムコード生成の活用から得られる学習効果にもその可能性が期待される。

2. 研究の目的

高校の情報科「情報Ⅰ」において必修となっているプログラミング学習に関して、生成 AI は初学者の学習目的に応じたプログラムコードを的確に出力する可能性をもつ。本研究の目的は、プログラミング学習に生成 AI を活用してプログラムコードを出力する方法について検討することである。また特定非営利活動法人みんなのコードによる全国の高校情報科教員を対象とした調査において、19.6%がプログラミング分野を苦手と回答している⁵⁾。本研究では、プログラミング指導に不安を抱える情報科教員がプログラミング指導を行う際に有用となる生成 AI の活用方法を検討する。

3. 研究の方法

本研究は以下の利用環境下において 2024 年 9 月時点の生成 AI を活用して行った。

2019 年より文部科学省の GIGA スクール構想に基づき児童生徒に一人 1 台のデジタル端末を活用した学習が進められている。MM 総研による調査では、2023 年「小中 GIGA スクールにおける ICT 環境のベンダーシェア分析」から一人 1 台端末の OS および汎用クラウドツールのシェアで Google が 42%と最大である⁶⁾。またそれ以前に MM 総研による「都道府県立高校における端末配備状況調査」においても Google の Chrome OS の割合が 30%ある⁷⁾。そのため本研究においては生徒の利用率が高いと推定される Google Workspace for Education を念頭に提供可能な Google の Gemini を生成 AI のサービス対象とした。

Google の生成 AI である Gemini を使用するには、Google アカウントが必要となる。Google アカウントは、13 歳以上の利用者自身で作成管理ができる。しかし本研究時点では教育機関向け Google Workspace for Education ではこの年齢条件が適用されず、Gemini の利用は学校の管理者にアクセスが一任されており、年齢条件も 18 歳以上となっている。従って高校生が個人で Gemini を利用することは可能であるが、現時点では学校用ユーザーを利用した機能については制限される場合がある。

高校の情報科「情報Ⅰ」検定教科書は、6社 12種類（2編合本の教科書は 1種類とした）が発行されている。教科書本文中で取り扱われるプログラミング言語は表 1 に示す通りであり、最も多く採用されているプログラミング言語は Python になる。そこで本研究の対象プログラミング言語は Python とした。

表 1 「情報Ⅰ」教科書取扱プログラミング言語

プログラミング言語	採用教科書数
Python	6
表計算ソフト・マクロ	5
JavaScript	4
Scratch	3

※教科書に複数のプログラミング言語が併記されている場合、各プログラミング言語を算入している。

表 2 対象教科書のプログラミングコードに関するページ数

「情報Ⅰ」教科書	A1	A2	B	C
プログラムコード記載ページ数	9	12	20	12

※巻末資料等を除く

同一のプログラミング言語を取り扱っていても、教科書本文中でプログラムコードを取り扱うページ数には差があり、プログラミングの処理内容についてアルゴリズムの複雑さや配列データの種類といった処理の難易度も異なっている。表 2 に示す本文中に Python を中心とした記載がある教科書 4 種類を研究対象とした。表中のアルファベットは出版社を示し、数字は別種類の教科書であることを示す。

教科書で取り扱われているプログラミングの内容は、順次、反復、分岐、配列、演算、関数等目的とする処理に分類されている。そこで教科書の分類に応じて生成 AI にプログラムコードを出力させた。出力した Python のプログラムコードの実行結果の確認には、Google Collaboratory を用いた。

4. 結果

処理内容をプログラムの動作別に整理して教科書別に Python のプログラムコードを生成する Gemini の出力結果を以下に示す。

4.1 教科書 A1

(1) 順次

2つの単語を単語ごとに改行して表示する順次処理のプログラムでは、教科書の指示通りに Gemini に質問した場合、教科書通りのプログラムコードが出力された。ただし改行の指示を与えない場合、改行しないコードで出力された。

(2) 演算

四則演算の結果を表示するプログラムでは、表示する演算結果は教科書と同一であったが、変数を一旦定義したのち計算式を代入して結果を表示させる点が教科書とは異なった。変数に値を代入し、さらにその変数に別の値を加えて演算結果を表示させるプログラムでは、教科書通りのコードが出力された。

(3) 分岐

代入した値が偶数か奇数かを判定するプログラムでは、代入する値を事前に決めた定数として取り扱うか、実行時に入力させるかによって出力コードが異なった。Gemini に対する質問で「xに代入した」という文言を含めると実行時に値の入力を求めるプログラムコードが出力された。教科書 A1 では学習者が値を入力する input 関数をプログラムの基本構造としては扱っておらず、発展的なプログラムの中で扱われる。偶数か奇数かを判定する if~else 文の処理に関して教科書通りのコードが出力された。

(4) 反復

単語2語のフレーズを3回表示させるプログラムでは、教科書通りの指示で教科書通りのプログラムが出力された。1桁の値を1から順番に表示させるプログラムでは、最初に for 文を使用したコードを出力した。そこで教科書記載の while 文を使用するよう指示したところ、教科書通りのコードが出力された。for と while の使用方法に関して相違点を教科書と同様に Gemini は出力した。

反復と分岐を組み合わせた処理で0から99までの値のうち定数で割り切れる値を表示させるプログラムは、教科書通りの指示で教科書通りのコードが出力された。

(5) リストと配列

値のリストから配列の要素を特定して表示するプログラムでは、Gemini への質問で指定した要素番号を解釈してプログラム上で置き換えてコードを出力した。Python のリストのインデックスは0からはじめるので、3番目と指定した場合インデックスでは2となる。Gemini はその解釈を反映させてコードを出力した。

4つの単語をリストにして単語数に応じた値を入力させ、その値から単語を表示するプログラムでは、Gemini では範囲外の値を入力した際の処理も出力される。教科書ではエラー処理に関するコードは記載されていない。

(6) 乱数

サイコロを振った際の目の数を表示するプログラムでは、「import random」を使用することに教科書と Gemini の出力コードに相違はなかったが、Gemini では「def」を使用して関数を定義し、目の数を表示するコードを出力した。「def」を使用しないよう制約を加えると教科書と同様のプログラムコードが出力された。

じゃんけんの手をランダムに表示するプログラムでも、Gemini ではサイコロのプログラムと同様に「def」を使用するプログラムが出力されるが、「def」を使用しないよう制約を加えて出力することができた。出力されたコードではリストの選択にランダムな整数を生成する「random.randint」ではなく、じゃんけんの手を定義した変数からランダムに要素を選択する「random.choice」が使用された。

(7) 関数

半径を入力させて円の面積を求めるプログラムでは、Gemini では math ライブラリを使用するコードを出力した。教科書では円の面積の計算に「半径×半径×3.14」という計算式を提示しているのに対して、Gemini の出力では円の面積の計算に「半径×半径×math.pi」と math.pi 定数を使用するコードとなった。教科書では「def」を使用するコードとなっていたので、「def」を使用するよう指示するとそれに応じたコードを出力した。出力中の解説で Python では2乗の計算を「**」で実行できることが説明され、コードの簡略化につながることが示された。

4.2 教科書 A2

演算、反復、分岐、リストの処理は、教科書 A1 と同様のプログラムであり、Gemini の出力コードとコードの内容に差はなかった。

(1) 配列要素の演算

配列の要素で合計と平均を計算するプログラムで最初に出力されたコードは、教科書と同様「for」を使用してリストの要素を順番に合計するコードであった。また平均の計算結果を変数に一旦代入して表示するコードが出力された。同じ内容のプログラムコードをより簡潔な記述 (Python らしい書き方) として図1を出力した。出力中の解説で合計の計算に sum 関数を使用した方がより簡潔な記述となることが示された。NumPy ライブラリを使用するコードも出力されるが、Python のライブラリの使用にはライブラリのインストールが必要となる記述は出力されなかった。

```
numbers = [1, 2, 3, 4, 5]

# 内包表記と sum 関数を使って合計を計算
sum_numbers = sum(numbers)

# 平均を計算
average = sum_numbers / len(numbers)

print("合計:", sum_numbers)
print("平均:", average)
```

図1 配列要素の合計と平均を表示するプログラムコード

(2) 反復

サイコロを 1000 回振ったときに 1～6 の目がそれぞれ何回出たか表示するプログラムでは Gemini の出力は「for」を使用するコードであった。教科書では「while」の使用が記載されているが、Gemini では反復に使用する while 文と for 文の違いと使い分けが出力された。より簡潔なプログラムコードの出力を求めると Counter クラスを使用したコードが出力された。さらに初学者向けのコードを求めると教科書で取り扱われている関数を使用したコードが出力された。

4.3 教科書 B

(1) 順次

順次処理のプログラムは、変数名に相違はあるが、Gemini の出力コードは教科書同様の結果であった。

(2) 分岐

分岐処理のプログラムは、変数名に相違はあるが、Gemini の出力コードは教科書同様の結果であった。

(3) 反復

反復構造のプログラムは、Gemini では最初に for 文を使用するコードが出力された。while 文を使用してコードの作成を指示すると応じたコードが出力された。出力されたコードでは計算式に累算代入演算子「+=」を使用して総和の計算を「sum += i」として簡略化された。教科書では取り扱われていないが、Gemini の出力コードでは記述を簡潔にする方向が示された。

(4) 配列

for 文の中で配列の要素数分繰り返す処理において、Gemini の出力コードでは range 関数で定数として繰り返す回数を指定するのではなく、図2に示すリストを定義した変数で繰り返して処理する点が教科書の記載とは異なった。リストの要素数が変更になった場合でも柔軟に対応できるプログラムコードとなった。

```
numbers = [3, 63, 27]

for number in numbers:
    print(number)
```

図2 for 文の繰り返し回数の処理

二次元配列に関するプログラムで教科書の記載に沿った質問を Gemini に行ったが、出力されたコードの実行結果が目的の表示結果とならず、表示を制御することが困難であった。ひらがなのあ行、か行、さ行を1行ずつ表示するプログラムを Gemini に指示するには、図3に示すプロンプトの設計⁸⁾を活用する必要があった。

配列と演算を組み合わせたプログラムで入力した値を配列の要素として、その和を表示するプログラムでは、教科書の指示通りの設問で Gemini に問うた場合、求めるコードが出力されなかった。図4に示すプロンプトで Gemini に指示したところ目的とするプログラムコードを出力した。教科書記載のプログラムコードでは入力の要素数を len 関数で取得するが、Gemini では len 関数を使用せず、リストの変数を使用された。

```
###指示:
Python でプログラムを作成してください
###手順:
行ごとに分けて表示する
###リスト:
あ い う え お
か き く け こ
さ し す せ そ
###条件:
二次元配列を使用する
```

図3 二次元配列でひらがなを行ごとに表示するコード出力に対するプロンプト

(5) 関数

三角形の面積を求めるプログラムを「def」で関数を定義して処理する内容

は、教科書の設問内容を Gemini に問うた場合、変数名は異なるが教科書通りのコードが出力された。

(6) 線形探索

5個の整数から探索値を入力して要素番号を表示するプログラムにおいて、教科書記載の指示通りに Gemini に回答させたところ関数を定義したプログラムコードを出力した。そこで関数を定義しないでプログラムを作成するよう指示したところ、教科書とは異なる点として index 関数を使用した。また探索値がリストになかった場合の else 処理が出力された。if 文に対する else は省略可能であるが、リストにない探索値を入力した場合、何も実行されないため処理の確認には else 文は有効である。

教科書記載コードを改善するよう Gemini に指示すると、さらに改善したコードが出力された。そのコードには探索値が見つからない場合の処理と入力値のエラー処理が含まれた。

(7) 二分探索

前述の線形探索と同内容のプログラムについて関数を定義しないよう制約を与えて Gemini に質問した場合、教科書記載と同様のプログラムコードが出力された。

平均探索回数と最大探索回数を求めるプログラムでは、図5に示すプロンプトを設計すると Gemini は目的とするプログラムコードを

出力した。データ数を 30 として線形探索と二分探索の最大探索回数を比較するグラフを出力するコードの作成を指示すると教科書と同様のグラフを作成するコードが出力された。プログラムの実行でそのグラフが確認できた。

(8) 整列

4個の整数を配列に入れて交換法で昇順に表示するプログラムでは、図6に示すプロンプトを設計すると目的に沿うコードが Gemini で出力された。また交換法を選択法に変更する指示で目的に沿うコードが出力された。その中でこの2方法についての比較が出力され、次の内容が説明された。

- 1) 交換回数が少ない方が効率が良いと考える場合は、選択法が適している。
- 2) 交換法の方が実装が簡単である。

4.4 教科書 C

教科書 C は、プログラム構造の順次、分岐、反復に関して具体的なプログラムを示して処理内容を分類して説明されていない。最初の例は約数を出力するプログラムであるが、関数を定義しないプログラムを Gemini に出力させたところ、教科書同様のプログラムコードが出力された。

(1) 変数・配列・演算

鉄道路線の駅番号から駅名を検索するプログラムでは、リストと分岐の処理が含まれるため、単純な指示で Gemini にプログラムコードの出力を求めることはできなかった。図7に示すプロンプトを設計したところ、目的に応じたコードが出力された。while 文を使用して、リストの要素数の範囲に当てはまらない場合のエラー処理を含むコードが出力された。if 文を使用したプログラムコードに修正を指示したところ、同じ目的に応じたコードが出力された。コードの説明において while 文と if 文の使用の違いについて出力された。また教科書記載のコードを評価させたところ、プログラミング初学者に理解しやすい構造であるが、入力値に文字列や小数が入力された場合のチェック処理が不十分であることが出力された。

(2) 関数

関数を定義して素数を判定させるプログラムでは、教科書記載の指示で目的に応じたプログラムコードが出力された。教科書記

```
###指示:
Python でプログラムを作成してください
###手順:
整数をキーボードから半角カンマで区切りながら複数入力する
入力された数値の和を表示する
```

図4 処理手順を指示するコード出力に対するプロンプト

```
###指示:
Python でプログラムを作成してください
###質問:
線形探索と二分探索で算出した平均探索回数と最大探索回数を求める
###条件:
データ個数 10
```

図5 平均探索回数と最大探索回数を求めるコード出力に対するプロンプト

```
###指示:
Python でプログラムを作成してください
###手順:
交換法でリストを昇順に整列する
要素番号とデータを表示する
###リスト:
88,43,27,65
```

図6 交換法でリストの表示を指示するコード出力に対するプロンプト

```

###指示:
Python でプログラムを作成してください
###手順:
配列番号を入力する
駅名を検索して表示する
###リスト:
天王寺 寺田町 桃谷 鶴橋 玉造 森ノ宮 大阪城公園 京橋 桜ノ宮 天満 大阪 福島 野田 西九条 弁天町
大正 芦原橋 今宮 新今宮

```

図7 配列の要素番号で検索するコード出力に対するプロンプト

載のコードではライブラリとして `import.math` が使用されているが、Gemini の出力では計算が簡略化される `is_prime` 関数を使用した。Gemini に `math` モジュールを使用するよう指示すると応じたプログラムコードが出力され、`math` モジュールは素数判定に限らず数学的な計算に応用できることが説明された。

(3) データと配列

駅名と所要時間の2つのリストからその対応を表示するプログラムでは、Gemini に対する指示が複雑となるためプロンプトを設計して出力させた。Gemini の最初の出力では、リストを駅名と所要時間を辞書として1変数で定義したコードが出力された。そこで図8に示すプロンプトを設計して出力させた。駅名と所要時間の2つのリストを定義するか、辞書で定義するかの違い以外に出力されたプログラムコードに大きな相違はなかった。

```

###指示:
Python でプログラムを作成してください
###手順:
駅名を入力する
東京メトロ銀座線の渋谷駅から各駅までの所要時間を調べて表示する
###リスト:
駅名 渋谷 表参道 外苑前 青山一丁目 赤坂見附 溜池山王 虎ノ門 新橋 銀座 京橋 日本橋 三越前 神田
末広町 上野広小路 上野 稲荷町 田原町 浅草
所要時間 0 2 4 5 7 9 11 13 15 17 18 20 22 23 25 26 28 30 31
###条件:
リストの変数は別々に定義する

```

図8 2つの配列で変数を定義するコード出力に対するプロンプト

5. 考察

以下で Gemini に対する質問から出力されたコードについてプロンプトの内容から考察し、生成 AI を効果的に活用する方法を検討する。

5.1 プロンプトを設計しないコードの出力

文字の表示、四則演算、反復処理、入力値による条件分岐といった単純な処理について、教科書例題の設問文に沿って Gemini に質問すると、プロンプトを設計せずとも目的に適したプログラムコードが出力された。コード各行の解説に加え、プログラムの特徴や改善点なども提示されるため、教科書の説明を補足し、理解を深めるのに有益である。

ただし、単語の表示を求めるプログラムの実行結果から、表示様式の制御が難しいことが確認できた。目的とする表示様式を得るには、改行位置などの条件や制約を含めた指示が必要である。複数の出力コードを比較することで表示制御に関する理解がより深まると考えられる。

また、意図したコードが得られない場合でも、条件や制約を追加して問い直すことで、適切なプログラムコードを得ることが可

能である。さらに、使用する関数やライブラリを明示的に指定すると、それに応じたコードが出力される。このことから、同様の処理を異なる制御構文で表現する方法や、特定の制約を課すことで意図したコードを生成させる手法を学ぶことができる。Gemini は、処理を関数として定義する場合が多く、これによりコードの保守性が向上し、汎用性の理解を深めるのに役立つ。

Gemini を活用することで、プロンプトを詳細に設計しない場合も、よりよいプログラムがどのような観点で構成されるかを理解できる。

5.2 プロンプトを設計したコードの出力

教科書の設問文に沿って Gemini に質問した際目的のプログラムコードを出力しない場合は、プロンプトの設計が有効であることが理解できた。表3に示す通り教科書記載の処理内容が複雑になると、プロンプトを設計して質問することによって目的とするプログラムコードが出力された。特に配列のデータを適切に処理する場合、プロンプトの設計が有効であった。

表3 プロンプトを設計した処理内容

教科書	A1	A2	B	C
プロンプト設計	なし	なし	配列 探索 整列	配列

5.3 プログラムコードの評価と修正に関する活用

プログラミングに対する学習者の知識やスキルに応じて、一度出力されたプログラムコードを生成 AI に評価させ、学習者がその内容を判断できる。また学習者が出力されたコードの理解に困難を感じた場合、より簡潔なプログラムに修正するよう指示を与えて再度出力させることが可能である。

出力されたプログラムコードが必ずしも正しいとは限らないため、動作環境を整えて実行し、結果を確認することが必要である。しかし初学者レベルのプログラムコードは、実行結果の確認が短時間でできるため、生成 AI の出力結果を有効活用できる。

生成 AI を活用してプログラミングの習熟度を高める上で、適切なプロンプト設計により的確なプログラムコードが出力されれば、生成 AI との対話により追加の情報が容易に得られる。「出力コード各行に対するコード内コメントによる説明」「目的の処理を行うためにコード内で定義される関数についての解説」「プログラムの改良点やプログラムを発展させるために提供される情報」は初学者や特にプログラミングに習熟していない教員にとって技術向上につながる有益な情報である。

5. まとめ

高校「情報 I」のプログラミング学習において教科書の処理内容に関するプログラムコードは、Gemini を活用して出力可能であった。文字の表示や四則演算を処理するプログラムは設問の指示で目的に沿ったコードが出力され、処理内容がより複雑となっても手順、条件や制約、リストを含むプロンプトを設計して指示すると目的に沿ったコードが出力されることが確認できた。

より効果的に生成 AI を活用するためには、プロンプトの設計が重要であり、その内容によって有効な回答を得ることができるに大きく関わる。作成するプログラムの難度が上がるとさらにプロンプト設計の重要度が増す。生成 AI を有効活用する上でプロンプトの基本的な設計を習得することはプログラミング学習に非常に役立つものであり、適切なプロンプトの設計を理解することが必要である。

このことから高校生がプログラミングに興味関心をもって取り組む場合に、生成 AI のプロンプトの設計を理解して活用すれば、特に初学者にとって生成 AI はよきサポーターとなる可能性を有する。また高校「情報 I」の授業において、現在の担当教員の体制からどの程度プログラミング学習に取り組むかは差があることが予想される。その状況で教員のプログラミング指導力の向上に生成 AI の活用は効果を発揮する可能性を持つと期待される。

なお、本論文に関して、開示すべき利益相反関連事項はない。

参考文献

- 1) 文部科学省. 高等学校学習指導要領解説 情報編. 開隆館出版販売, p.6-7, 2019.
- 2) 文部科学省. 高等学校学習指導要領(平成 30 年告示)解説 総則編. 東洋館出版社, p.55, 2019.

- 3) 文部科学省. 高等学校情報科に係る指導体制の一層の充実について. 2022. https://www.mext.go.jp/content/20221124-mxt_jogai02-000021518_001.pdf 2024年9月1日参照
- 4) 文部科学省. 高等学校情報科に係る指導体制の一層の充実について. 2023. https://www.mext.go.jp/content/20231227-mxt_jogai01-000021518_1.pdf 2024年9月1日参照
- 5) 特定非営利活動法人みんなのコード. 2022年度プログラミング教育・高校「情報I」実態調査報告書. 2023. <https://speakerdeck.com/codeforeveryone/programmingeducationreport2022-high> 2024年11月22日参照
- 6) MM 総研. 「小中 GIGA スクールにおける ICT 環境のベンダーシェア分析」. 2023. <https://www.m2ri.jp/release/detail.html?id=599> 2024年9月3日参照
- 7) MM 総研. 「公立高校の生徒 1 人 1 台端末配備」は 4 割 都道府県立高校における端末配備状況調査. 2021. <https://www.m2ri.jp/release/detail.html?id=483> 2024年9月3日参照
- 8) クジラ飛行機. 大規模言語モデルを使いこなすためのプロンプトエンジニアリングの教科書. マイナビ出版. 2024.

受理 2025年2月19日

公開 2025年4月1日

<連絡先>

氏名 市川隆司

宛先 〒536-8585 大阪府大阪市城東区古市2丁目7番30号

電話番号 06-6939-4391 (代表)

E-mail t-ichi@osaka-shinai.ac.jp