

## 【情報 II】

実教出版「数学 II」P.68~95「Python の基礎」  
主要なプログラミング言語である Python の基本的なプログラムを理解する。

### ～ClassPad.net の各種ふせん・授業支援機能を活用する～

Python を使い実践的なプログラミングを行う授業  
ゲーム形式で Python を活用することで、プログラミングに対する理解を深める。

#### 【本授業の目的・狙い・到達目標】

教師向けの目標：Python によるプログラミングを生徒が主体的に学習できるように指導をする。  
生徒向けの目標：Python によるプログラミングを通して、プログラムを記述するときの基礎的な関数や、プログラミングによる効率的な情報処理を主体的に学習する。

#### 【ClassPad.net 活用によるメリット】

- ・ **板書時間削減** : 必要な情報はすべて、事前に作成しておいたテキストふせん・ファイルふせんで投影するだけで済む。
- ・ **生徒の理解促進** : ファイルふせんによるプログラミングコードの共有や、授業支援機能による一斉学習により、Python について効果的に学習できる。
- ・ **協働学習促進** : グループワーク機能で、他の人の実行結果や考察を知ることができ、協働学習をサポートできる。

### 授業の流れ

### ClassPad.net での操作

#### step1

##### 【概要】

〈Pythonの特徴〉  
・文法がシンプルで読みやすい  
・AI活用やデータ分析など、できることが多い

〈今回の授業〉  
・簡単なゲームを通して、実際にPythonでプログラムを作成してみよう

#### 概要の説明

近年、シンプルで汎用性が高い Python が AI 開発など幅広い分野で利用されていることを紹介する。その上で、Python を使った簡単なゲームで実際にプログラムを作成することを伝える。

#### step2

##### 【基礎的な関数の意味と使い方】

print()  
かっこの中に文字や数字を入れて、画面に表示する。

〈例〉 print("こんにちは")  
→画面に「こんにちは」と表示

#### 重要事項の解説①

print(), input(), if, while といった基礎的な関数の意味と使い方について説明する。また、Python においては#の後ろの記述はプログラムとして認識されないため、メモや説明書きに利用できることを説明する。

それぞれの関数について、テキストふせんを用いて説明を行う。

## step3

### 【実際にプログラムを実行しよう】

1. print()で好きな文言を表示させる
  2. input()を使って、自分の名前を打ち込むと「〇〇さん、よろしくね!」と表示させる
  3. if, elseを使って、18以上の数を打ち込むと「成人」、18未満だと「未成年」と表示させる
- まずは自分で考えて、次で答え合わせ

## 重要事項の解説②

step2 で説明した関数を使った、ごく簡単なプログラム3つを、実際に生徒個人が実行する。

### 【プログラム A】

1. print()で好きな文言を表示させる
2. input()を使って、自分の名前を打ち込むと「〇〇さん、よろしくね!」と表示させる
3. if, else を使って、18 以上の数を打ち込むと「成人」、18 未満だと「未成年」と表示させる

プログラム A コードはあらかじめテキストふせんに記述しておき、生徒がそれを見ながらコードをコピー&ペーストまたは手入力で行えるようにする。

## step4

### ✓ ペアワーク：数字当てゲーム①

ペアで出題者と回答者に分かれる。出題者はプログラム上で1~100の数字を設定する。回答者はその数字を予想して入力し、何回で正解できるかやってみよう。→結果はふせんに記入

## ペアワーク①

ペアで出題者と回答者に分かれる。出題者側は1~100の中で好きな整数を設定する。回答者はその数字を予想して入力して、何回の入力で正解できるかを試行する。

### 【プログラム B】

while と if,else を用いて、出題者があらかじめ設定した1~100の任意の整数について、回答者が入力した数字に応じて「もっと大きいよ!」「もっと小さいよ!」「当たり!おめでとう!」と表示させる。当たったら break で繰り返し終了。

プログラム B コードはあらかじめテキストふせんに記述しておき、生徒がそれを見ながらコードをコピー&ペーストまたは手入力で行えるようにする。

生徒は、何回で正解できたかの回数をふせんに記入し共有する。

## step5

### 【振り返り】

- ・他のペアとの結果を比べてみよう
- ・どんな工夫をしたら当てやすい?

## ペアワーク①の振り返り

各ペアの結果を全体で確認した上で、どうすれば確実に早く当てられるかのアイデアを生徒に考察してもらい、共有する。

生徒はそれぞれのアイデアをふせんに記入し共有する。

## step6

### 【二分探索】

数字を1つ入力しその数字が  
正解よりも大きい  
→それ以下の数字は全て正解の候補から外す  
正解よりも小さい  
→それ以上の数字は全て正解の候補から外す  
正解を探す数字の範囲が絞られるので、次は  
その範囲の中央値を入力する

以上の繰り返しで、  
範囲がどんどん絞られる！

## ペアワーク②

step5の生徒のアイデアを共有した上で、教師が二分探索によって正解を突きとめる方法を紹介する。そこで、二分探索を利用してペアワーク①と同じゲームを攻守交代して実行する。

### 【プログラムC】

ペアワーク①のプログラムに加え、入力した値に応じて正解の数字がありうる範囲を教えてくれるようにする。

例：出題者が25を設定した場合  
回答者が40と入力した場合、「もっと小さいよ！」とともに「現在の範囲：1から39」と表示される。次に回答者は1と39の中央値である20を入力し、「もっと大きいよ！」とともに「現在の範囲：21から39」と表示され、以降これを繰り返す。

プログラムCコードはあらかじめテキストふせんに記述しておき、生徒がそれを見ながらコードをコピー&ペーストまたは手入力で行えるようにする。

生徒は、何回で正解できたかの回数をふせんに記入し共有する。

## step7

### 【振り返り】

- ・1回目と比べて回数はどうなった？
- ・1から1,000,000でやると？  
→二分探索なら、何回かかる？

## ペアワーク②の振り返り

各ペアの結果を全体で共有し、ペアワーク①の場合よりも早く正解できる傾向にあることを確認する。

教師が「二分探索を利用した場合、1から100ではなく、1から100万では何回かかるか」という問いを提示し、生徒に予想してもらおう。その上で教師がその場で実演または事前に行った結果を共有し、20回以内で当てられることを示す。

生徒は、1から100万の場合は何回かかるかの予想をふせんに記入し共有する。

## step8

### ◎ まとめ

- ・今回の授業では、Pythonによるプログラミングについて学んだ。
- ・Pythonでは、少ない行数でさまざまな処理ができる。
- ・二分探索といったアルゴリズムと組み合わせると、より効率的な処理が可能になる。

## まとめ・宿題

二分探索のアルゴリズムをプログラミングによって行うことにより、効率よく正解にたどり着けることから、プログラミングにより効率的に情報処理ができることを説明する。

宿題として、randomを利用すればコンピュータ側が正解を設定してくれることを利用し、生徒自身でも1~100万の範囲で実際に数字当てに挑戦する。

学習支援機能を用いて、宿題で入力したコードと結果をスクリーンショットしてファイルふせんで送信する。