

学 習 指 導 案

教 科 ・ 科 目	FV・Ⅱ（プログラミングで身近な課題を解決しよう！）
授 業 者 名	堀内、村田
日 時	令和6年7月下旬～11月中旬
実 施 学 級	2年生普通科・FV選択者（16名）
単 元	プログラミングの基礎を学ぶ、IoTプログラミングの手法を学ぶ
指 導 区 分 本 時 の 位 置	1 Pythonプログラミングの基礎を学ぶ（2時間） 2 じゃんけんプログラムをつくる（A I ?）（2時間） 3 プログラミングの基本構造を絵を書きながら学ぶ（2時間） 4 matplotlibを用いてグラフを表示させてみよう！（2時間） 5 円周率 π をプログラミングで求める！～モンテカルロ法を用いて～（2時間） 6 「素数」か否かを判定するコードを書く！（2時間） 7 日本IBM職員による講義「JAVAによるじゃんけんプログラム」（4時間） 8 ラズベリーパイにLED、温湿度センサーを接続する（2時間）
使 用 教 科 書	配布プリント、啓成高校オリジナルテキスト

1. Python プログラミングの基礎を学ぶ(2時間)

本 時 の 目 標		1 Pythonプログラミングを動作させる環境（特にエディタ）を理解する。 2 データの型について理解する。	
時間	内容	授業者	生徒
事前準備	事前連絡	・当日にノートPCを持参するように伝える。	・ノートPC、配布物を保管するファイルを持参する。
14:25	着席指示	・二人一組の班ごとの着席を指示する。 ・第1回のプリント配布する。	・班ごとに着席する。
14:30	導入	・本時の目的と内容を説明する。 ・オードリータンの執筆資料を紹介し、模倣や変更から学んでいく道筋について説明する。 ・Python言語の特徴と引用例を紹介する。	・説明を聞く。
14:40 14:40	展開	・各自のGoogleアカウントを元に、エディタ Google Colaboratoryへの入り方を説明する。	・説明を聞く。 ・Google Colaboratoryを起動する。
		(1)自分の出席番号と氏名、身長を変数に入力して表示させるコードの説明。 ・具体的には、エディタ上で、x、yなどの変数に、input関数を用いてint（整数）型 float（浮動小数）型、str（文字列）型を代入し、データの値と型をprint関数、type関数にて表示させるコードの説明。	・配布資料や大型モニタを見ながら、コードを書き実行する。
		(2)秒数を入力して、何時間、何分、何秒かを表示させるコードの説明。 ・(1)の内容に四則演算の要素を加えれば解決できることに気づかせる。	time=int(input(“input second>>”))と入力し、int型timeの演算を行う。
		(3)配列（個々のデータの集合体）listを紹介する。一例として、Greetingリストを作成し、要素の取り出し、書き換え、削除、追加などを行う。 ・個々のデータGreeting[1]を表示させる。 ・append関数やremove関数を用いて、データの追加、削除の仕方を説明する。	・Greeting = [‘Good morning’ , ‘Good afternoon’ , ‘ Good bye’] のlistを作成し、新たに要素を追加したり、既存の要素を削除する操作を行う。
		(4)配列をNumpyモジュールを用いると容易に表現できることを紹介する。	・Greeting =

15:55		・ import文でライブラリnumpyを導入する。 ・ 科学計算に強いライブラリ（小プログラムの集合）であるNumpyには、ndarrayという配列がある。理数学的には、ベクトルや行列を意味していることを説明する。	・ import numpy as np ・ a = np.array([1, 2, 3])
		・ 数班指名し、2人で話し合って書いたコードを大型モニタに表示させて、コードの書き方について全体に向けて発表させる。	・ 数班発表する。
16:05	まとめ	・ 今回学んだことを応用するとどのようなコードを書くことができるかを発表させる。	・ 数班発表する。

2. じゃんけんプログラムをつくる(AI?)(2時間)

本 時 の 目 標		1 じゃんけんの「出し手」を配列で表し、ランダムで「出し手」を決めて取り出すための手法を学ぶ。 2 自分の「出し手」とPCによる「出し手」によって生じる勝敗を、条件分岐で表現し、場合の数ができるだけ生じないように工夫するコードを書く。	
時間	内容	授業者	生徒
事前準備	事前連絡	・ 当日にノートPCを持参するように伝える。	・ ノートPC、配布物を保管するファイルを持参する。
14:25	着席指示	・ 二人一組の班ごとの着席を指示する。 ・ 第2回のプリント配布する。	・ 班ごとに着席する。
14:30	導入	・ 本時の目的と内容を説明する。 ・ 配列の復習を行う。	・ 説明を聞く。
14:40	展開	・ エディタGoogle Colaboratoryを起動するように指示する。	・ Google Colaboratoryを起動する。
		(1)人間（自分）が出す手を決めるコードの説明。 <pre> ■Hands = [“GU” , “CHOKI” , “PA”] ■print(Hands[1]) </pre> で何が出力されるかを質問する。 (2) 相手の出す手をコンピュータに決めてもらうコードの説明。 <pre> ・ randomモジュールをインポートする。 ■Choice_PC = random.randint(0, 2) # 0～2 をランダムに選ぶ。 ■print(Hands[Choice_PC]) </pre> (3)人間（自分）とPCのじゃんけんの勝敗を判定するコードの説明。 ・ すべての場合の数（9通り）において、条件分岐if文、elif文を用いて、勝ち、負け引き分けを判定するコードを書く。 (4) (3)にて、場合の数を減らすためには、どのような解決策があるかを2人で話し合うように指示する。 ・ 1つの解決策案として、 <pre> Hands = [“GU” , “CHOKI” , “PA”] </pre> 各要素の配列順を利用する例を紹介する。 この場合、3通りに減る。	→ “CHOKI”が出力される。 ・ 配列の要素を取り出す。 → ■import random → “GU” , “CHOKI” , “PA” の中からランダムに出力されることを確認する。 →Hands[-1] = “PA” Hands[0] = “GU” Hands[1] = “CHOKI” Hands[2] = “PA” の中から、2つ連番で取り出した時、配列順が若い方が、勝ちとなるコードを書く。
15:55		・ 数班指名し、2人で話し合って書いたコードを大型モニタに表示させて、コードの書き方について全体に向けて発表させる。	・ 数班発表する。
16:05	まとめ	・ 今回学んだことを応用するとどのようなコードを書くことができるかを発表させる。	・ 数班発表する。

3. プログラミングの基本構造を絵を書きながら学ぶ～Turtleを使った実習～(2時間)

本時の目標		1 外部ライブラリをインストールする方法を学ぶ。本授業では、Turtle（タートル）をインストールする。 2 簡単な図形を描く手法を学ぶ。複雑な図形は、繰り返し文を利用して簡便化できることを学ぶ。	
時間	内容	授業者	生徒
事前準備	事前連絡	・当日にノートPCを持参するように伝える。	・ノートPC、配布物を保管するファイルを持参する。
14:25	着席指示	・二人一組の班ごとの着席を指示する。 ・第3回のプリント配布する。	・班ごとに着席する。
14:30	導入	・本時の目的と内容を説明する。 ・基本構造である順次進行と繰り返しを扱うことを説明する。	・説明を聞く。
14:40	展開	・エディタGoogle Colaboratoryを起動するように指示する。 (1) Google Colaboratoryに外部ライブラリTurtleをwebブラウザ上で、pipインストールを行う。 ■! pip3 install ColabTurtle ■from ColabTurtle.Turtle import * (2) 亀の初期化を行うと出力先に赤色の亀のポインタが表示される。 ■initializeTurtle(initial_speed =7) ■pencolor('red') (3) Turtleの基本的なコマンドを使う。 ■turtle.forward(50) # 前に50進ませる ■turtle.right(100) #亀の角度を100° 右に回転させる ■turtle.left(60) #亀の角度を60° 左に回転させる (4) (3)の応用として、正多角形を図示するコードを書くには、どうしたらよいかを質問する。 (5) (4)の課題を表現するコードを、よりシンプルに書くには、どうしたら良いかを質問する。例えば、正三角形であれば、 ■ for i in range(4): turtle.forward(50) turtle.right(120) ・for文のブロック中には、インデントに気をつけないといけないことを説明する。	・Google Colaboratoryを起動する。 ・大型モニタを見ながら、コードを書き実行する。 →外部ライブラリTurtleのインストールする。 →亀のポインタが表示されることを確認する。 →亀のポインタが50前進させるforwardメソッドを用いる。 →亀のポインタが100° 右回りに回転するrightメソッドを用いる。 →(3)の各操作を順次進行に従って、丁寧にコードを書く。 →正多角形を書くコードを分解して、ひとつのブロックを繰り返すコードを書く。
15:55		・数班指名し、2人で話し合って書いたコードを大型モニタに表示させて、コードの書き方について全体に向けて発表させる。	・数班発表する。
16:05	まとめ	・今回学んだことを応用するとどのようなコードを書くことができるかを発表させる。	・数班発表する。

4. matplotlib (マットプロットライブラリ) を用いてグラフを表示させてみよう! (2時間)

本 時 の 目 標		1 matplotlib (マットプロットライブラリ) は、グラフを表示する便利なライブラリであり、見やすいグラフを描く操作を通して、情報デザインの考え方を身に付ける。 2 2軸のグラフを描くための、各軸の配列の中身を理解し、今後の授業で扱うデータのインプットにつながることを理解する。	
時間	内容	授業者	生徒
事前準備	事前連絡	・当日にノートPCを持参するように伝える。	・ノートPC、配布物を保管するファイルを持参する。
14:25	着席指示	・二人一組の班ごとの着席を指示する。 ・第4回のプリント配布する。	・班ごとに着席する。
14:30	導入	・本時の目的と内容を説明する。 ・配列リストの復習をする。	・説明を聞く。
14:40	展開	・エディタGoogle Colaboratoryを起動するように指示する。 ・起動できない場合は、Jupyter notebook をダウンロードしてから起動する。 (1) x軸、y軸の各軸 (リスト) に要素値を手入力し、グラフを表示させるコードを書く <pre> ■import matplotlib.pyplot as plt ■x = [1 , 3 , 5 , 7, 9] ■y = [2 , 4 , 6 , 8 , 10] ■plt.plot(x , y) ■plt.show() </pre> (2) 手入力では面倒なので、規則性のあるリスト (配列) を「繰り返し文」にて作成する方法を紹介する。 <pre> ■x = [] ■a = 0 ■for i in range(10): a = a + 1 x.append(a) ■print(x) </pre> (3) ライブラリnumpyを用いると、より簡便に x 軸の配列を作成できることを紹介する。 <pre> ■import numpy as np ■x = np.arange(1, 10, 2) ■print(x) </pre> (4) $y = \sin(x)$ のグラフを描くコードを考えさせる。 <pre> ■import matplotlib.pyplot as plt ■import numpy as np ■x = np.arange(0 , 10 , 0.5) ■y = np.sin(x) ■plt.plot(x, y) ■plt.show() </pre>	・Google Colaboratoryを起動する。 ・大型モニタを見ながら、コードを書き実行する。 →matplotlibをpltで表す。 →xのlistにx軸の要素を、yのリストにy軸の要素を手動により入力。 →グラフ表示させる。 →入れ子をつくる。 →変数a の初期設定を行う。 →繰り返し文にて、x軸を生成する。 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10] と表示されることを確認する。 →初項1, 公差2の等差数列で [1, 3, 5, 7, 9] と表示されることを確認する。 →色々な書き方があるが、個々で書いたコードを見せあって学習する。
15:55		・数班指名し、2人で話し合って書いたコードを大型モニタに表示させて、コードの書き方について全体に向けて発表させる。	・数班発表する。
16:05	まとめ	・今回学んだことを応用するとどのようなコードを書くことができるかを発表させる。	・数班発表する。

5. 円周率 π をプログラミングで求める！～モンテカルロ法を用いて～（2時間）

本 時 の 目 標		1 円周率 π を求める手法は様々あるが、モンテカルロ法という確率を通して求められる手法を学ぶ。 2 試行回数が増えていくと、円周率 π の値の精度が向上することを学ぶ。 3 pythonは、オブジェクト指向言語であり、関数がメソッドになることを理解する。	
時間	内容	授業者	生徒
事前準備	事前連絡	・当日にノートPCを持参するように伝える。	・ノートPC、配布物を保管するファイルを持参する。
14:25	着席指示	・二人一組の班ごとの着席を指示する。 ・第5回のプリント配布する。	・班ごとに着席する。
14:30	導入	・本時の目的と内容を説明する。 ・半径1の円のグラフの書き方を復習する。	・説明を聞く。
14:30	展開	・エディタGoogle Colaboratoryを起動するように指示する。 (1) モジュールと変数の初期設定を行う。 <pre> ■import random ■import matplotlib.pyplot as plt ■totalcount = 100 ■incount = 0 </pre> (2) x軸、y軸の第1象限にランダム関数の(x, y)の座標をプロットする。 <pre> ■for i in range(totalcount): x = random.random() y = random.random() if x**2 + y**2 <= 1.0: incount = incount + 1 plt.scatter(x, y, c = 'red') else: plt.scatter(x, y, c = 'blue') </pre> (3) 円周率 π を求める演算式を書くコードを紹介する。 <pre> ■print('pi :{ }'.format(?)) </pre> ・?の内容を初めは示さず、円周率 π を求めるための演算式を書くには、どうしたら良いかを考えさせる。 ・'pi :{ }'がオブジェクトになり、format関数がメソッドになることを説明する。 (4) 試行回数であるtotalcountを増やしていくと、円周率 π 3.14の値に近づいていく操作をするように指示する。 <pre> ■plt.show() </pre>	・Google Colaboratoryを起動する。 →random, matplotlibモジュールをインポートする。 →試行回数を設定する。 →半径1の円の内側に収まる回数を設定する。 →0から順に整数値をiに100回代入する。 →0から1までの小数(float型)をランダム関数で生成し、x, yに代入する。 →半径1の円の内側の時、incountの値を随時更新し、赤色の点をプロットする。 →半径1の円の外側の時、青色の点をプロットする。 plt.scatterは、散布図を書くコマンドとなる。 →4 * incount / totalcount ・ incount / totalcountでは第1象限(円の面積の1/4)での確率になり、4倍しなければならないことを確認する。 →変数totalcountを増やしていくとPiの値が収束していく過程を確認する。 →グラフ上のプロットが第1象限の全領域に埋め尽くすことを確認する。
15:55		・数班指名し、2人で話し合って書いたコードを大型モニタに表示させて、コードの書き方について全体に向けて発表させる。	・数班発表する。
16:05	まとめ	・今回学んだことを応用するとどのようなコードを書くことができるかを発表させる。	・数班発表する。

6. 「素数」か否かを判定するコードを書く！

本 時 の 目 標		素数かどうかを判定する手法として、「エラトステネスのふるい」という手間がかかる操作を行う必要があるが、このふるいを行う操作をプログラミングのコードに置き換える手法を学ぶ。	
時間	内容	授業者	生徒
事前準備	事前連絡	・当日にノートPCを持参するように伝える。	・ノートPC、配布物を保管するファイルを持参する。
14:25	着席指示	・二人一組の班ごとの着席を指示する。 ・第6回のプリント配布する。	・班ごとに着席する。
14:30	導入	・素数の定義を確認する。 ・四則演算の演算子の確認をする。特に%の意味を確認する。	・説明を聞く。
14:30	展開	・エディタGoogle Colaboratoryを起動するように指示する。 (1) 自分で入力した整数が「素数」かどうかを判定するプログラムを書く！ ・お好みの整数 a を入力するコードを書く。 ■ a = int (input(‘お好みの整数を入力してください>>’)) ・変数 b を初めは素数として設定する。 ■ b = '{ }' は、素数である'.format(a)' ・「エラトステネスのふるい」の操作を繰り返すコードを書く。 ・aが素数ではない条件を記述するには、 if a の後に何を書けば良いかを発問する。 ■ for i in range(?, ?, 1): if a % i == 0: b = '{ } は素数ではない'.format(a) print(b) ・例えば a = 17 であれば、繰り返し文の中で 17/2, 17/3, 17/4, … 17/15, 17/16 が (a % i) の値となる。 ・素数でない場合は、b の値が上書きされ、print文で出力されることを説明する。 (2) 1～999までの数字をすべて、素数か否かを判定するコードを書く！ ・(1)のコードに何を付け足すとできるのかを発問する。 ■ <u>for a in range(1000):</u> b = '{ } は、素数である'.format(a) for i in range(2, a, 1): if a % i == 0: b = '{ } は、素数ではない'.format(a) print(b)	→str型のaをint型に変換 →int型の a を文字列に変換し {} に代入 ・模範例として →for i in range(2, a, 1): → if a % i == 0 : ※range関数の特徴として、 2, 3, 4, … (a-1) を i に代入することになる。 a ÷ i = 0 つまり割り切れる場合は、a は素数ではなくなることを理解する。 →アンダーラインの箇所での追加で(2)の課題が達成できることを確認する。 →2は素数である 3は素数である 4は素数ではない ・ ・ 997は素数である 998は素数ではない 999は素数ではない と出力されることを確認する。
15:55		・数班指名し、2人で話し合っ書いたコードを大型モニタに表示させて、コードの書き方について全体に向けて発表させる。	・数班発表する。
16:05	まとめ	・今回学んだことを応用するとどのようなコードを書くことができるかを発表させる。	・数班発表する。

