

教科・科目	総合的な探究の時間(2年普通科Future Vision II講座にて)
時期	7月～10月
単元	IoTプログラミング研修：円周率 π をプログラミングで求めよう～モンテカルロ法を用いて～
指導区分 本時の位置	1 Pythonプログラミングの基礎を学ぼう (2時間) 2 じゃんけんプログラムをつくろう (2時間) 3 プログラミングで絵を描こう (2時間) 4 matplotlibを用いてグラフをかこう (2時間) 5 円周率πをプログラミングで求めよう～モンテカルロ法を用いて～ (2時間) 6 「素数」か否かを判定しよう (2時間) 7 ラズベリーパイにLED、温湿度センサーを接続しよう (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
5分	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。
10分	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。
75分 ※10分休憩含む	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。	・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。

時間	内容	授業者	生徒
		(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。 ■plt.show()	→変数totalount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。
10分	共有	・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。
10分	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。