

研究 主題	ICTを効果的に活用した問題発見・解決能力の育成を目指して ー1人1台端末を活かしたスモールステップによるプログラミング教育ー
----------	--

第1学年情報科学習指導案

指導月日 令和4年10月17日
所属校名 宮城県石巻西高等学校
氏名 尾形 大

1 単元名 第3章「コンピュータとプログラミング」

2. アルゴリズムとプログラム（日本文教出版 情報Ⅰ）

2 単元の目標

- (1) アルゴリズムを表現する手段，プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に付けることができる。〔知識及び技能〕ア(イ)
- (2) 目的に応じたアルゴリズムを考え適切な方法で表現し，プログラミングによりコンピュータや情報通信ネットワークを活用するとともに，その過程を評価し改善することができる。〔思考力，判断力，表現力等〕イ(イ)
- (3) 情報や情報技術を適切かつ効果的に活用して情報社会に主体的に参画し，その発展に寄与しようとする態度を身に付けようとする。〔学びに向かう力，人間性等〕

3 単元観

本単元は，高等学校指導要領情報 情報Ⅰ2内容(3)コンピュータとプログラミング〔知識及び技能〕ア(イ)，及び〔思考力，判断力，表現力等〕イ(イ)に基づいている。プログラムの構成や基本的な命令文などを取り扱うとともに，生徒の実態に応じたプログラム作成の実習などを通して，基礎的なプログラミングの知識・技能を習得させることをねらいとしている。

まずはビジュアルプログラミング言語を利用した視覚や表現から直感的にプログラムを作成させ，プログラミングが苦手な生徒でも興味・関心を持ちながら，学習に取り組むことができるよう工夫する。

また，プログラミング言語Python^{バイソン}を活用し，プログラミング的思考を身に付け，自分の意図したとおりにプログラムを作成するには，一連の動作手順を考えながらプログラムを検討し，動作が誤っている場合は，その原因を特定し，問題点を見付けて修正する，という問題発見と問題解決のプロセスが必要になる。プログラミングによる試行錯誤を重ねることで，論理的思考力や問題解決能力などを育むことにつながると考える。

4 生徒の実態〔第1学年1組40名〕

生徒は，教科情報の学習に対して興味・関心が高いが，学習に対する自信が持てないことで，積極的に活動できないことや物事に主体的に取り組めない，自己肯定感が低いなどの傾向がある。クラスの約2割がプログラミング未経験者であるため，プログラミング活動に自信を持てない生徒も見受けられた。しかし，8月のアンケートの結果によると，「これまでの『アルゴリズム』，『ドリトル』，ロボット『梵天丸』などの学習活動を通して，あなたの抵抗感や苦手意識などのプログラミングへの意識を教えてください」という質問では，「抵抗感などはなくなった」「少し抵抗感などはなくなった」と答えた生徒が約90%いた。ここからプログラミング学習への抵抗感や苦手意識などが減少したことやプログラミングの思考を働かせ，学習に取り組むことができる生徒が増えてきたことが分かる。一方，教科情報の学習が日常生活にどのように結び付いているのかを理解しておらず，興味・関心を持つことができない生徒も見受けられる。

したがって、これまでの生徒の学習活動の様子を踏まえ、日常生活に関わるプログラムと関連付けながら、プログラミング学習に興味・関心を喚起できるように働き掛けたい。

5 指導観

導入段階では、プログラミングに対する興味・関心を高め、今後の学習に主体的に取り組ませることができるよう、基本的なプログラムの作成から始めていき、様々な問題に対して、調整をしながら問題発見と問題解決ができる力を身に付けさせたい。さらに、プログラムの一つ一つの命令がステップで動いていることを理解させ、プログラミング言語 Python の文法・構文について基礎基本を身に付けさせる。プログラミングを経験することによって、問題解決能力と新たな価値ある情報を作り出す力を養いたい。また、主体的に学習に取り組み、気付きや振り返りにつなげられるように工夫する。

6 研究主題との関連

本研究のテーマは、端末やOSに関係なく全ての生徒が同じ学習ができる環境で、1人1台端末を活用したプログラミング教育の実践を通して、情報活用能力やプログラミング的思考などを育成することである。生徒に物事の目的を的確に捉えさせ、柔軟な思考と想像力を養うことができる授業を目指したい。

(1) 苦手意識があるプログラミング学習に対する配慮

生徒がプログラミングの理解を深め、興味・関心を持つために、ビジュアル型プログラミング言語や日本語ベースによるプログラミング言語などを用いた導入教育を実施する。また、1人1台端末環境の利点を生かして、休み時間や放課後、家庭学習などの授業以外で学習者のペースでビジュアル型プログラミング言語や日本語ベースによるプログラミング言語などに触れてもらうことで、プログラミング言語 Python の学習に円滑な移行ができるようにする。

(2) Google Workspace for Education などの教育支援ツールの活用

将来の予測が難しい社会において、ICTを効果的に活用し、情報を主体的に捉えながら、何が重要かを他者と協働しながら考え、新たな価値を見出していくには、情報活用能力の育成が重要となる。また、情報技術は人々の生活にますます身近なものとなっていくと考えられ、これからの情報社会において、必要に応じて情報技術を手段として、学習や日常生活に活用できるようにしていくことも重要となる。そのため、ClassroomやColaboratory、Forms、Meet、DriveなどのGoogle Workspace for Educationを有効活用し、場所を問わずに学習できる環境を目指すことで、コロナ禍による学びの保障や個別最適な学びを実現した深まる学習につなげたい。

(3) 生徒の問題発見・問題解決を助けるファシリテーター型教材の活用

一方的に教える教師中心の授業スタイルから、生徒が主体的に課題に取り組みかつ問題解決を意識し、課題を自らのものとするすることで、解決したいという意欲が生まれ、主体的な学びへとつながるような授業に変えていきたい。また、プログラミングに慣れていない生徒の戸惑いや苦手意識などにも配慮し、生徒自身が学習理解度を把握するためのFormsによる学習教材、主体的にプログラミング学習を深めるためのColaboratoryによるスモールステップの学習教材など、ICTを効果的に活用してファシリテーター型のプログラミング教材を授業や自学自習などで活用することで、個別最適な学習を視野に入れた問題発見能力や問題解決能力が育成できるプログラミング学習を行う。

7 単元の指導と評価の計画

(1) 単元の評価規準

知識・技能	思考・判断・表現	主体的に学習に取り組む態度
①コンピュータや外部装置の	①コンピュータで扱われる情	①問題解決にコンピュータを

仕組みや特徴，コンピュータでの情報の内部表現と計算に関する限界について理解している。((3)ア(ア)) ②アルゴリズムを表現する手段，プログラミングによってコンピュータや情報通信ネットワークを活用する方法について理解し技能を身に付けている。((3)ア(イ)) ③社会や自然などにおける事象をモデル化する方法，シミュレーションを通してモデルを評価し改善する方法について理解している。 ((3)ア(ウ))	報の特徴とコンピュータの能力との関係について考察することができる。 ((3)イ(ア)) ②目的に応じたアルゴリズムを考え適切な方法で表現し，プログラミングによりコンピュータや情報通信ネットワークを活用するとともに，その過程を評価し改善することができる。 ((3)イ(イ)) ③目的に応じたモデル化やシミュレーションを適切に行うとともに，その結果を踏まえて問題の適切な解決方法を考えることができる。 ((3)イ(ウ))	積極的に活用し，自ら結果を振り返って改善しようとしている。 ②生活の中で使われているプログラムを見いだして改善しようとするなどを通じて情報社会に主体的に参画しようとしている。
---	---	---

(2) 単元の全体計画（14 時間扱い 本時 9 / 14）

時	主たる学習活動	評価規準	評価方法
1 ・ 2	アルゴリズムの基本と表現方法 ・課題解決型ソフトウェア「アルゴロジック」にフローチャート作成及び課題解決活動	アルゴリズムの意味や簡単なフローチャートの意味を理解している。【知技②】 アルゴリズムの意味を理解して，簡単なフローチャートで表現できる。【思判表②】	行動観察 ワークシート 課題作成
3	フローチャートとプログラミング① ・教育用プログラミングロボット「梵天丸」によるプログラミング	ロボットの動きからプログラムを主体的に考え，プログラミングしようとしている。【知技①】 ロボットとプログラムの関係性を理解している。 【思判表①】	行動観察 課題作成
4	フローチャートとプログラミング② ・教育用プログラミング言語「ドリトル」によるプログラミング	処理の手順や命令の表し方を理解している。【知技①】 目的に応じてプログラムの問題点を考え，解決しようとしている。【思判表③】	行動観察 ワークシート 課題作成
5	Python プログラミング① ・プログラムの構成要素と関数の概念について	関数の概念について理解している。【知技①】	行動観察 ワークシート

	・ print 関数について ・ input 関数について	print 関数や input 関数を用いた処理について理解している。【知技②】 print 関数や input 関数を用いた処理が日常生活の中でどのように使われているかを考えている。 【態度②】	課題作成
6	Python プログラミング② ・ 変数の使い方について	変数を用いた処理について理解している。【知技②】	行動観察 ワークシート 課題作成
7	Python プログラミング③ ・ if 文について ・ 比較演算子について ・ 入力された答えを数値に変換	if 文や比較演算子などを用いた処理について理解している。【知技②】 アルゴリズムを理解して、簡単なフローチャートで表現できる。【思判表②】	行動観察 ワークシート 課題作成
8	Python プログラミング④ ・ for 文について	for 文を用いた処理について理解している。【知技②】 アルゴリズムを理解して、簡単なフローチャートで表現できる。【思判表②】	行動観察 ワークシート 課題作成
9	Python プログラミング⑤ ・ random 関数について	random 関数を用いた処理について理解している。 【知技②】 random 関数を用いてプログラムの問題点を考え、解決しようとしている。 【思判表②】	行動観察 ワークシート 課題作成
10	Python プログラミング⑥ ・ リストについて	リストを用いた処理について理解している。【知技②】	行動観察 課題作成
11 ・ 12 ・ 13	Python プログラミング⑦ ・ 制作するアプリの検討及びアプリ制作	アプリ制作を通して、これまで学習した内容への理解を深めようとしている。【知技①②③思判表①②③】 評価シートへの記入を通して、学習の成果と課題を振り返ることができる。【態度①②】	行動観察 ワークシート 課題作成 評価シート

14	Python プログラミング⑧ ・制作したアプリの発表及び自己評価	発表会を通して、これまで学習した内容への理解しようとしている。 【知技①②③思判表①②③】 評価シートへの記入を通して、学習の成果と課題を振り返ることができる。【態度①②】	行動観察 ワークシート 課題作成 評価シート
----	---	--	---------------------------------

8 本時の計画

(1) 目標

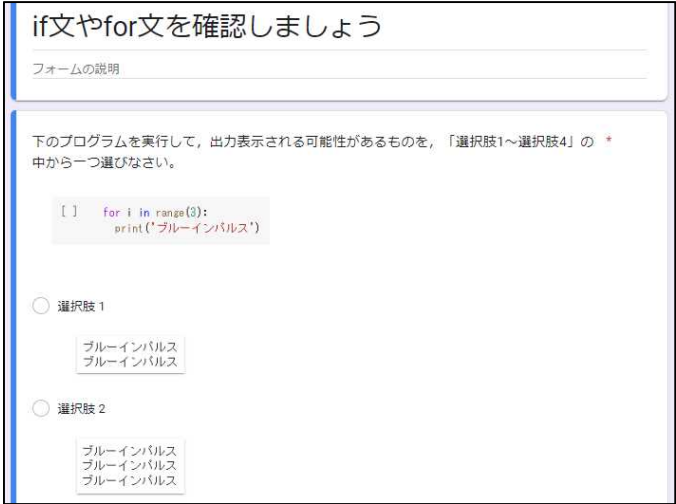
- ・ random 関数を用いた処理について理解し、自分の意図したフローチャートの作成や簡単なプログラムを作成・実行することができる。
- ・ 乱数（ランダム生成）を利用し、日常生活の事象と関連付けながら、プログラムを考えることができる。

(2) 本時の指導に当たって

本時は、プログラムの流れを理解しながら、作成するプログラムを考えていくことができるようにしていく。また、実行結果を予測しながらプログラムを作成できるようにする。

様々なプログラムを発想しやすいように、グループ活動によるプログラム作成やサンプルプログラムの提示など、生徒の興味・関心を引き出すとともに、容易にプログラムの流れが理解できる工夫をしたい。また、random 関数の要素が日常生活の様々な場面で活用されていることを意識させたい。

(3) 指導過程

段階	学習活動 ○主な発問・指示 ◆予想される生徒の反応	形態	指導上の留意点	評価
導入 5分	1 前時までの学習内容を確認する。 ・ if 文や比較演算子, for 文などについて振り返る。 ○Classroom に提示された前時までの学習内容の復習ファイル (Forms ファイル) を開き, 学習内容を振り返りましょう。 <u>復習</u> 「これまでの学習内容を確認しよう」 	一斉	◎これまで学習してきた内容をプロジェクタや Google Meet の画面共有機能などで提示する。 ◎前時までの学習内容の復習ファイルを Classroom で提示し, これまでの学習内容を確認するように促す。 ◎前時までの学習内容の復習ファイルの正答率をプロジェクタや Google Meet の画面共有機能などで共有し, 理解不足が見受けられる問題に関しては補足を行う。	

展開 42分	2 本時の学習内容を確認して、使用するプログラムファイルを準備する。（5分）			
	乱数（ランダム生成）を利用したプログラムを作成しよう！			
	○身の回りにおける日常生活の中で、乱数（ランダム生成）を利用した仕組みやシステムがあるのかを考えてみましょう。 ◆さいころアプリ ◆ビンゴアプリ ◆ゲームにおける乱数発生（パズルゲームやカードゲーム、シューティングゲームなど） など ○Classroomに提示されたプログラムファイルを開いてみましょう。	一 斉	◎身の回りの日常生活の中で、乱数（ランダム生成）を利用した仕組みやシステムについて、グループで考えるように促す。 ◎ClassroomにアップロードされているColaboratoryの学習教材を使用するように指示する。	
	3 プログラムの構成要素とrandom関数の概念について説明する。（12分） 演習1 「1～10の数値をランダムに表示しよう」 ○次のプログラムがどのような動きをするのかを予想して、実行してみましょう。 演習① 下のプログラムを実行してみましょう。 また、どのようなプログラムなのか、予想してみましょう！ <pre>[] import random kotae = random.randint(1,10) print(kotae)</pre> ◆グループで迷いながらも、プログラムを予測して実行しようとしている。 ・実行した演習1のプログラムを基に、学習するプログラムの構成要素と関数の概念について確認する。 課題1 「random関数の練習問題を解こう」 ○グループで気付きなどを共有しながら、練習問題1から練習問題3までのプログラムを作成しよう。	小 集 団	◎グループで相談しながらプログラムの動きを予測して、プログラムを実行するように指示する。 ◎繰り返しプログラムを実行して、ランダムに表示される数値にはどのような特徴があるのか、また、どのような範囲の整数が表示されているのかを意識させる。 ◎教科書を参考にしながら、プログラムの構成要素と関数の概念について、説明する。 ◎1～100, 20～1000, 500～745の範囲の整数をランダムに表示されるプログラムを作成できたら、その他の条件でランダム	【思判表②】 行動観察

練習問題① 下のプログラムは、このままだとエラーが表示されます。
1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
print(kotae)
```

練習問題② 下のプログラムは、このままだとエラーが表示されます。
20～1000までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
print(kotae)
```

練習問題③ 下のプログラムは、このままだとエラーが表示されます。
500～745までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
```

- ◆**演習1**のプログラムを参考にしながら、プログラムを作成しようとしている。
- ◆失敗しながら、迷いながらもプログラムを作成しようとしている。
- ◆教科書に提示されている「プログラムの構成要素」などを参考にしながら、プログラムを作成しようとしている。

- ・**課題1**の作成例を提示して、全体で共有する。

演習2 「ミニ数当てゲームを作成しよう」

○グループでワークシートを活用してプログラムとフローチャートの関連を意識しながら、演習2のプログラムを作成しましょう。

[Step2] 「ミニ数当てゲーム」を作ってみよう！

演習② 下のプログラムは、このままだとエラーが表示されます。
1～10までの数値を当てる「ミニ数当てゲーム」を作成してください。

```
[ ] #ミニ数当てゲーム①
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

に数値が表示されるプログラムを作成するように促す。

- ◎これまでの学習で、全角入力によるプログラムエラーが多かったことを生徒に伝え、正しくプログラムを作成できるように促す。
- ◎教科書を参考にしながら、
random.randint(a,b)
がa～bの範囲の整数をランダムに返す関数であることを正しく理解できるようにする。

- ◎プロジェクタやGoogle Meetの画面共有機能などで、作成例を提示してプログラムを共有する。

- ◎ワークシートを配布して、プログラムとフローチャートの関係を意識させる。

◆既習のプログラムを参考にしながら、プログラムを作成しようとしている。 ・演習2で作成したプログラムを全体で共有する。 4 学習したプログラムを発展させる。(15分) ○演習2を改良して、3回答えることができるミニ数当てゲームを作成しましょう。また、ワークシートを活用して、プログラムとフローチャートを作成しましょう。 課題2 「演習2を改良して、3回答えることができるミニ数当てゲームを作成しよう」 練習問題④ 下のプログラムは、このままだとエラーが表示されます。 (1) 数値を3回答えられる 1～20までの数値を当てる「ミニ数当てゲーム」を作成してください。 (2) (1)の課題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。 <pre> [6] #ミニ数当てゲーム④（3回答えられるようにしよう！） import random kotae = random.randint() print('あなたの予想は?') answer = int(input()) if kotae == answer: print('正解!') else: print('不正解') print('答えは...') print(kotae) </pre> ◆random関数の特性を意識しながら、プログラムやフローチャートを作成しようとしている。	◎プロジェクタやGoogle Meetの画面共有機能などで、生徒が作成したプログラムを共有する。 ◎プログラムとフローチャートを意識しながら、ワークシートを記入するように指示する。 ◎for文のインデント（字下げ）していない記述などによるプログラムエラーのつまりきに対して、生徒がプログラムを試行錯誤しながら作成して、少しずつ正解に近づき、学習を深められるように支援する。 ◎生徒が探究プロセスにおいて、何らかのつまりきや困り感を感じている場合は、生徒の話を傾聴し、問い掛けることによって、生徒の意見を引き出し、問題解決を促す。特に、(2)の課題に取り組んでいる生徒には、「forループから抜ける」や「答えに関するヒントを表示する」などのように、各生徒の探究プロセスの特性に応じて、問題解決ができるように支援する。	【思判表②】 ワークシート
---	---	--------------------------

	・課題2で作成したプログラムを全体で共有する。 5 本時の学習内容の振り返りをする。(10分) ○Classroomに提示された本時学習した random 関数のプログラムについて復習ファイル (Forms ファイル) を開き, 学習内容を復習しましょう。 random関数のプログラムを復習しましょう 下に書かれたプログラムを実行して, 出力表示される可能性があるもの * 1ポイントを選びなさい。 <pre>[] 1 import random 2 3 kotae = random.randint(1,10) 4 print(kotae)</pre> ☐ 0 ☐ 10 ☐ 100 ☐ 1000 ・本時で使用したワークシートを回収する。 ・本時で使用したプログラミング実習用ファイルを共有フォルダに提出させる。	一斉	◎プロジェクタや Google Meet の画面共有機能などで, 生徒が作成したプログラムを共有する。 ◎様々なプログラムがあることを理解し, 学習内容が深められるように促す。 ◎本時の学習内容のポイントを提示する。 ◎復習用の課題ファイルを Classroom で提示し, 本時の学習内容の振り返りを促す。 ◎課題の正答率をプロジェクタや Google Meet の画面共有機能などで共有し, 理解不足が見受けられる問題に関しては解説を行う。 ◎本時で使用したプログラミング実習用ファイル (Colaborator ファイル) を Google ドライブ内にある共有フォルダに提出するように促す。	【知技②】課題作成
まとめ 3分	6 家庭学習の指示と次時の予告をする。 ○Classroomに提示された家庭学習用の課題ファイル (Colaborator ファイル) を開き, 確認しましょう。	一斉	◎家庭学習用の課題ファイルを Classroom で提示し, 学習内容を確認するように促す。	

▼ random関数 random関数を使用することで、乱数を作成することができます。 ゲームやシミュレーションなどのプログラムで利用されます。 ▼ random関数の使用方法 random.randint(1,10)は、1から10までの値の表示 <pre>[] import random kotae = random.randint(1,10) print(kotae)</pre> 宿題① 下のプログラムは、このままだとエラーが表示されます。 1～100までの数値がランダムに出力表示されるように修正してください。 <pre>[] import random kotae = random.randint(100) print(kotae)</pre> ・次時の予告をする。		
--	--	--

(4) 本時の評価

評価の観点	評価規準	十分満足できる(A)	努力を要する生徒(C) への手立て
知識・技能	プログラムやフローチャートを正しく作成し、実行することができる。	プログラムの流れを正しく把握し、random関数のプログラムを適切に作成し、実行することができる。また、フローチャートを適正に作成することができる。	random関数のプログラムを正しく作成できない。各演習のプログラムやフローチャートを用いて助言する。
思考・判断・表現	プログラムやフローチャートを読み解き、その手順や仕組みを説明できる。また、プログラムやフローチャートの改良点を考えることができる。	random関数のプログラムを読み解き、その手順や仕組みを説明できる。また、プログラムやフローチャートの改良点を考え、正しく作成できる。	random関数のプログラムやフローチャートの手順や仕組みが理解できていない。どのような手順や仕組みなのかを一つ一つ整理して検討させる。

(5) 準備物

- ①教師：教科書、タブレット端末、タブレット端末用キーボード、パソコン、プロジェクタ、授業用学習課題ファイル・ワークシート（別紙1）
- ②生徒：教科書、タブレット端末、タブレット端末用キーボード、筆記用具

授業用学習課題ファイル・ワークシート等

①家庭学習用の課題ファイル（Forms ファイル／復 習）

if文やfor文を確認しましょう

下のプログラムを実行して、出力表示される可能性があるものを、「選 * 1ポイント
択肢1～選択肢4」の中から一つ選びなさい。

```
[ ] for i in range(3):  
    print('ブルーインパルス')
```

ブルーインパルス
ブルーインパルス

☐ 選択肢 1

ブルーインパルス
ブルーインパルス
ブルーインパルス

☐ 選択肢 2

ブルーインパルス
ブルーインパルス
ブルーインパルス
ブルーインパルス

☐ 選択肢 3

ブルーインパルス
ブルーインパルス
ブルーインパルス
ブルーインパルス
ブルーインパルス

☐ 選択肢 4

下のプログラムを実行して、出力表示されるものを選びなさい。 * 1ポイント

```
[ ] answer = 5  
  
if answer == 1:  
    print('好き')  
else:  
    print('嫌い')
```

☐ 好き
☐ 嫌い
☐ 5
☐ 1

下のプログラムを実行して、出力表示されるものを選びなさい。 * 1ポイント

```
[ ] answer = 5  
  
if answer >= 5:  
    print('好き')  
else:  
    print('嫌い')
```

☐ 好き
☐ 嫌い
☐ 0
☐ 5

送信

フォームをクリア

②プログラミング実習用ファイル（Colaborator ファイル／[演習 1](#)・[課題 1](#)・[演習 2](#)・[課題 2](#)）

▼ 05_random関数

【Step1】 本日学ぶプログラムを理解しよう！

演習① 下のプログラムを実行してみましょう。
また、どのようなプログラムなのか、予想してみましょう！

```
[ ] import random

kotae = random.randint(1,10)
print(kotae)
```

練習問題① 下のプログラムは、このままだとエラーが表示されます。
1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
print(kotae)
```

練習問題② 下のプログラムは、このままだとエラーが表示されます。
20～1000までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
print(kotae)
```

練習問題③ 下のプログラムは、このままだとエラーが表示されます。
500～745までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
```

【Step2】 「ミニ数当てゲーム」を作ってみよう！

演習② 下のプログラムは、このままだとエラーが表示されます。

1～10までの数値を当てる「ミニ数当てゲーム」を作成してください。

```
[ ] #ミニ数当てゲーム①
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

練習問題④ 下のプログラムは、このままだとエラーが表示されます。

- (1) 数値を3回答えられる 1～20までの数値を当てる「ミニ数当てゲーム」を作成してください。
- (2) (1)の課題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。

```
[6] #ミニ数当てゲーム②（3回答えられるようにしよう!）
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

③ワークシート（配付プリント／**演習2**・**課題2**で使用）

プログラムとフローチャート
乱数（ランダム生成）を利用した
プログラムを作成しよう！

1年（ ）組（ ）番
名前（ ）

【例】
演習②のプログラムをフローチャートで表現しましょう。

演習② 下のプログラムは、このままだとエラーが表示されます。
1～10までの数値を当てる「ミニ数当てゲーム」を作成してください。

```
[ ]
#ミニ数当てゲーム①
import random

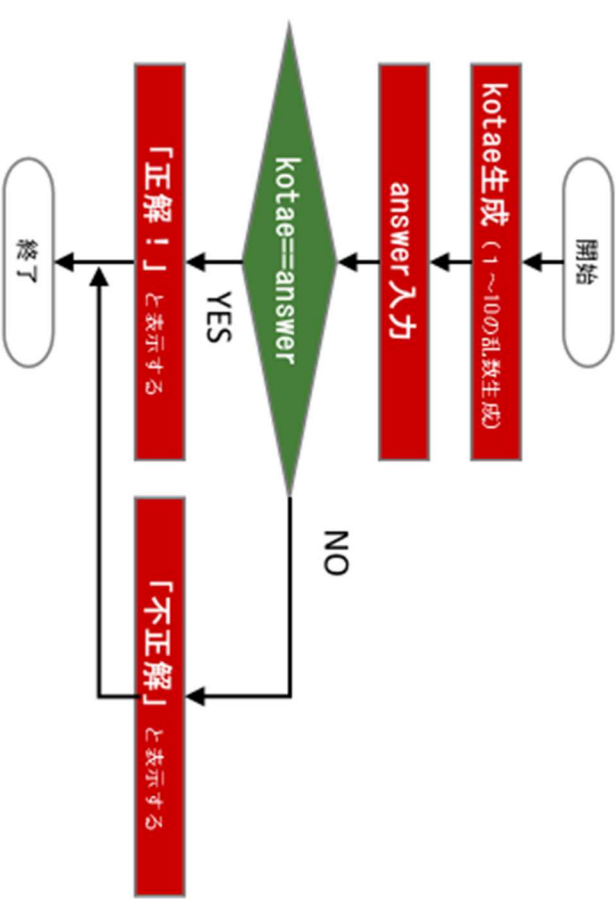
kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

【例】演習②のプログラムをフローチャートで表現しましょう！



【練習問題④】

(1) のプログラムをフローチャートで表現しましょう。

練習問題④

下のプログラムは、このままだとエラーが表示されます。

- (1) 数値を3回答えられる 1～20までの数値を当てる「三三数当てゲーム」を作成してください。
 (2) (1)の課題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。

#三三数当てゲーム (3回答えられるようにしよう！)

import random

kotae = random.randint()

print('あなたの予想は?')

answer = int(input())

if kotae == answer:

print('正解!')

else:

print('不正解')

print('答えは...')

print(kotae)

【練習問題④】

(2) のプログラムをフローチャートで表現しましょう。

練習問題④

下のプログラムは、このままだとエラーが表示されます。

- (1) 数値を3回答えられる 1～20までの数値を当てる「三三数当てゲーム」を作成してください。
 (2) (1)の課題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。

#三三数当てゲーム (3回答えられるようにしよう！)

import random

kotae = random.randint()

print('あなたの予想は?')

answer = int(input())

if kotae == answer:

print('正解!')

else:

print('不正解')

print('答えは...')

print(kotae)

【練習問題④】(1)のプログラムをフローチャートで表現しましょう！

開始

終了

【練習問題④】(2)のプログラムをフローチャートで表現しましょう！

開始

終了

④ 振り返り用課題ファイル（Forms ファイル）

random関数のプログラムを復習しましょう

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 kotae = random.randint(1,10)
    4 print(kotae)
```

- ☐ 0
- ☐ 10
- ☐ 100
- ☐ 1000

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 kotae = random.randint(5,10)
    4 print(kotae)
```

- ☐ 1
- ☐ 10
- ☐ 100
- ☐ 510

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 kotae = random.randint(10,100)
    4 print('オガタが',kotae,'人登場！')
```

- ☐ オガタが 6 人登場！
- ☐ オガタが 55 人登場！
- ☐ オガタが 101 人登場！
- ☐ オガタが 1000 人登場！

下のプログラムを実行して、出力表示される可能性があるものを、「選 * 1ポイント
択肢1～選択肢4」の中から一つ選びなさい。

```
[ ] 1 import random
    2
    3 for i in range(3):
    4     kotae = random.randint(10,30)
    5     print(kotae)
```

6
2
7

☐ 選択肢 1

12
19
10
9

☐ 選択肢 2

19
29
14

☐ 選択肢 3

18
13
18
15
14

☐ 選択肢 4

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 number1 = random.randint(1,10)
    4 number2 = random.randint(1,10)
    5
    6 kotae = number1+number2
    7 print(kotae)
```

- ☐ 1
☐ 16
☐ 0
☐ 22

送信

フォームをクリア

⑤家庭学習用の課題ファイル（Colaborator ファイル）

▼ random関数

random関数を使用することで、乱数を作成することができます。
ゲームやシミュレーションなどのプログラムで利用されます。

▼ random関数の使用方法

random.randint(1,10)は、1から10までの値の表示

```
[ ] import random

kotae = random.randint(1,10)
print(kotae)
```

宿題① 下のプログラムは、このままだとエラーが表示されます。

1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint(100)
print(kotae)
```

宿題② 下のプログラムは、このままだとエラーが表示されます。

正しく実行できるように修正してください。

○のところには、1～100までの数値がランダムに出力表示されるように作成してください。

＜期待する出力表示＞

マリモが ○ 匹あらわれた

```
[ ]

kotae = random.randint(10,100)
print('マリモが',kotae,'匹あらわれた')
```

宿題③ 下のコードエリアに書かれたプログラムは、このままだとエラーの表示がでます。
1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint(1,100)
print(kotae)
```

宿題④ random関数を用いて、「ねこが○匹あらわれた」と表示するようなプログラムを完成させてください。
○のところには、4～18までの数値がランダムに出力表示されるように作成してください。

<期待する出力表示>
ねこが ○ 匹あらわれた

```
[ ] import random

kotae =
print('ねこが',kotae,'匹あらわれた')
```

授業用学習課題ファイル・ワークシート等

①家庭学習用の課題ファイル（Forms ファイル／復 習）

if文やfor文を確認しましょう

下のプログラムを実行して、出力表示される可能性があるものを、「選 * 1ポイント
択肢1～選択肢4」の中から一つ選びなさい。

```
[ ] for i in range(3):  
    print('ブルーインパルス')
```

ブルーインパルス
ブルーインパルス

ブルーインパルス
ブルーインパルス
ブルーインパルス

☐ 選択肢 1

☐ 選択肢 2

ブルーインパルス
ブルーインパルス
ブルーインパルス
ブルーインパルス

ブルーインパルス
ブルーインパルス
ブルーインパルス
ブルーインパルス
ブルーインパルス

☐ 選択肢 3

☐ 選択肢 4

下のプログラムを実行して、出力表示されるものを選びなさい。 * 1ポイント

```
[ ] answer = 5  
  
if answer == 1:  
    print('好き')  
else:  
    print('嫌い')
```

☐ 好き

☐ 嫌い

☐ 5

☐ 1

下のプログラムを実行して、出力表示されるものを選びなさい。 * 1ポイント

```
[ ] answer = 5  
  
if answer >= 5:  
    print('好き')  
else:  
    print('嫌い')
```

☐ 好き

☐ 嫌い

☐ 0

☐ 5

送信

フォームをクリア

②プログラミング実習用ファイル（Colaborator ファイル／[演習 1](#)・[課題 1](#)・[演習 2](#)・[課題 2](#)）

▼ 05_random関数

【Step1】 本日学ぶプログラムを理解しよう！

演習① 下のプログラムを実行してみましょう。
また、どのようなプログラムなのか、予想してみましょう！

```
[ ] import random

kotae = random.randint(1,10)
print(kotae)
```

練習問題① 下のプログラムは、このままだとエラーが表示されます。
1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
print(kotae)
```

練習問題② 下のプログラムは、このままだとエラーが表示されます。
20～1000までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
print(kotae)
```

練習問題③ 下のプログラムは、このままだとエラーが表示されます。
500～745までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint()
```

【Step2】 「ミニ数当てゲーム」を作ってみよう！

演習② 下のプログラムは、このままだとエラーが表示されます。

1～10までの数値を当てる「ミニ数当てゲーム」を作成してください。

```
[ ] #ミニ数当てゲーム①
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

練習問題④ 下のプログラムは、このままだとエラーが表示されます。

(1) 数値を3回答えられる 1～20までの数値を当てる「ミニ数当てゲーム」を作成してください。

(2) (1)の課題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。

```
[6] #ミニ数当てゲーム②（3回答えられるようにしよう!）
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

③ワークシート（配付プリント／**演習2**・**課題2**で使用）

プログラムとフローチャート
乱数（ランダム生成）を利用した
プログラムを作成しよう！

1年（ ）組（ ）番
名前（ ）

【例】
演習②のプログラムをフローチャートで表現しましょう。

演習② 下のプログラムは、このままだとエラーが表示されます。
1～10までの数値を当てる「ミニ数当てゲーム」を作成してください。

```
[ ]
#ミニ数当てゲーム①
import random

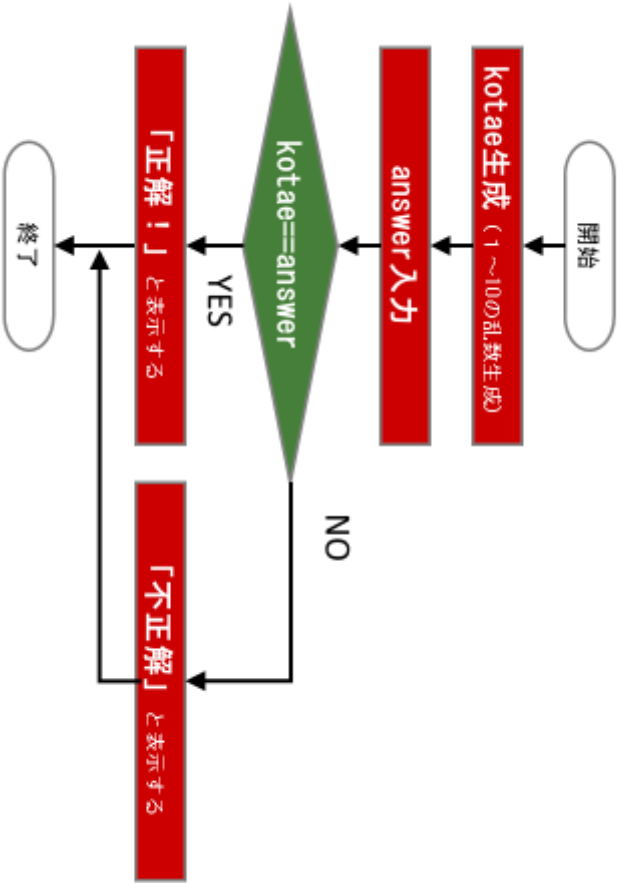
kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

【例】演習②のプログラムをフローチャートで表現しましょう！



【練習問題④】

(1) のプログラムをフローチャートで表現しましょう。

練習問題③

下のプログラムは、このままだとエラーが表示されます。

- (1) 数値を3回答えられる 1~20までの数値を当てる「三三数当てゲーム」を作成してください。
- (2) (1)の問題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。

```
#三三数当てゲーム④ (3回答えられるようにしよう！)
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

【練習問題④】

(2) のプログラムをフローチャートで表現しましょう。

練習問題③

下のプログラムは、このままだとエラーが表示されます。

- (1) 数値を3回答えられる 1~20までの数値を当てる「三三数当てゲーム」を作成してください。
- (2) (1)の問題を終えたら、さらにゲームが遊びやすくなるように改良してみましょう。

```
#三三数当てゲーム④ (3回答えられるようにしよう！)
import random

kotae = random.randint()

print('あなたの予想は?')
answer = int(input())

if kotae == answer:
    print('正解!')
else:
    print('不正解')

print('答えは...')
print(kotae)
```

【練習問題④】(1)のプログラムをフローチャートで表現しましょう！

開始

終了

【練習問題④】(2)のプログラムをフローチャートで表現しましょう！

開始

終了

④振り返り用課題ファイル（Forms ファイル）

random関数のプログラムを復習しましょう

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 kotae = random.randint(1,10)
    4 print(kotae)
```

- ☐ 0
- ☐ 10
- ☐ 100
- ☐ 1000

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 kotae = random.randint(5,10)
    4 print(kotae)
```

- ☐ 1
- ☐ 10
- ☐ 100
- ☐ 510

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 kotae = random.randint(10,100)
    4 print('オガタが',kotae,'人登場！')
```

- ☐ オガタが6人登場！
- ☐ オガタが55人登場！
- ☐ オガタが101人登場！
- ☐ オガタが1000人登場！

下のプログラムを実行して、出力表示される可能性があるものを、「選 * 1ポイント
択肢1～選択肢4」の中から一つ選びなさい。

```
[ ] 1 import random
    2
    3 for i in range(3):
    4     kotae = random.randint(10,30)
    5     print(kotae)
```

6
2
7

☐ 選択肢 1

12
19
10
9

☐ 選択肢 2

19
29
14

☐ 選択肢 3

18
13
18
15
14

☐ 選択肢 4

下のプログラムを実行して、出力表示される可能性があるものを一つ選 * 1ポイント
びなさい。

```
[ ] 1 import random
    2
    3 number1 = random.randint(1,10)
    4 number2 = random.randint(1,10)
    5
    6 kotae = number1+number2
    7 print(kotae)
```

☐ 1

☐ 16

☐ 0

☐ 22

送信

フォームをクリア

⑤家庭学習用の課題ファイル（Colaborator ファイル）

▼ random関数

random関数を使用することで、乱数を作成することができます。
ゲームやシミュレーションなどのプログラムで利用されます。

▼ random関数の使用方法

random.randint(1,10)は、1から10までの値の表示

```
[ ] import random

kotae = random.randint(1,10)
print(kotae)
```

宿題① 下のプログラムは、このままだとエラーが表示されます。

1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint(100)
print(kotae)
```

宿題② 下のプログラムは、このままだとエラーが表示されます。

正しく実行できるように修正してください。

○のところには、1～100までの数値がランダムに出力表示されるように作成してください。

＜期待する出力表示＞

マリモが ○ 匹あらわれた

```
[ ]

kotae = random.randint(10,100)
print('マリモが',kotae,'匹あらわれた')
```

宿題③ 下のコードエリアに書かれたプログラムは、このままだとエラーの表示がでます。

1～100までの数値がランダムに出力表示されるように修正してください。

```
[ ] import random

kotae = random.randint(1,100)
print(kotae)
```

宿題④ random関数を用いて、「ねこが○匹あらわれた」と表示するようなプログラムを完成させてください。

○のところには、4～18までの数値がランダムに出力表示されるように作成してください。

<期待する出力表示>

ねこが ○ 匹あらわれた

```
[ ] import random

kotae =
print('ねこが',kotae,'匹あらわれた')
```