

本事例の基礎データ

カテゴリ	プログラミングに関する指導方法		
学校種	高等学校	事例提供者	都立雪谷高等学校
学年	1年生	教科等	情報科
単元名	問題の解決と処理手順の自動化		
主な ICT 機器	コンピュータ、タブレット PC、micro:bit、プログラミングカー		
授業の概要	<p>さまざまなアルゴリズムについて理解する。</p> <p>プログラミングを問題解決に生かす方法を考える。</p>		
「情報活用能力 #東京モデル」の位置付け	情報活用	STEP5	<ul style="list-style-type: none"> 問題解決に向け、計画を最適化し、評価、改善しながら実行できる 問題解決のための安全・適切なプログラムの制作や、そのモデル化ができる

本事例における教育の情報化について

【ポイント1】	<p>アルゴリズムの基本の理解する</p> <p>コンピュータを用いたデータ処理がどのように行われているかに関心をもたせ、アルゴリズムの考え方を理解させる。</p>
【ポイント2】	<p>プログラム作成手順を理解する</p> <p>順次処理、繰り返し、判断分岐の3つの基本構造を学び、アルゴリズムをプログラムであらわすための知識・技能を身に付ける。</p>
【ポイント3】	<p>問題解決の手段・手法を身に付ける</p> <p>問題解決の流れを PDCA サイクルの中でとらえ、さまざまなツールを用いて効果的で効率のよい活動を行うための知識を養う。</p>

本単元（題材）における指導の流れ

時間	●主な学習活動 ・ 児童の活動	○支援・留意点 ☆評価
1 ～ 4	<ul style="list-style-type: none"> ●【問題解決の考え方と手法】 ・「問題」とは何か理解する。 ・問題解決のプロセスを理解する。 ・提示した問題に対して、グループで解決する方法を考える。 	<ul style="list-style-type: none"> ○ワークシートを用いて問題を解決するアイデアの考え方や、話し合いの方法について指導する。 ☆問題解決で利用する様々な手法や考え方に関心をもつことができる。 【関心・意欲・態度】
5 ～ 8	<ul style="list-style-type: none"> ●【プログラミングの基礎知識】 ・アルゴリズムの考え方と表し方、プログラム言語の関係を理解する。 ・アルゴリズムの基本構造とプログラム作成方法を理解する。 ・サンプルプログラムを作成しながら、コーディングについての知識を養う。 	<ul style="list-style-type: none"> ○プログラムを作成する環境を整え、うまく実行できないときの対処方法を指導する。 ☆コンピュータによるデータの処理とアルゴリズムとの関係を理解する。 【知識・技能】
9 ～ 12	<ul style="list-style-type: none"> ●【プログラミングの実践】 ・判断分岐や繰り返しの仕組みを理解し、例題のプログラムを作成し、実行する。 ・応用問題に取り組み、正しい実行結果に導くための方法を考える。 	<ul style="list-style-type: none"> ○ワークシートを確認しながら、コードの書き方・読み方について指導する。 ☆基本的なアルゴリズムを簡単なプログラムであらわすことができる。【知識・技能】
13 ～ 20 (本時)	<ul style="list-style-type: none"> ●【実機を使った実習】 ・micro:bit を使用し、作成したプログラムの転送、実行を行うための技術を身に付ける。 ・ライトレースカーを使った課題に対し、計画的にグループワークを進めるために活動する方法を考える。 ・課題を通して自分たちの活動と考えをパワーポイントで作成し、発表する。 	<ul style="list-style-type: none"> ○課題を解決するために必要な計画を立て、必要な知識・技能を考えるように促す。 ○発表を行うための資料として、個々の端末を用いて静止画や動画など記録を行わせる。

本時の流れ

段階	● 主な学習活動・児童の活動	○ 支援・留意点 ☆ 評価
導入	<ul style="list-style-type: none"> ● 【課題と授業予定の確認する】 ・ 端末を起動し、Teams 上に配布された資料を確認し、必要な技術を学ぶ。 ・ グループでプログラム提出までの計画を確認し、作業について分担をする。 	<ul style="list-style-type: none"> ○ プログラミングカーによるライントレースを行う技術を学ぶために、学習する内容をまとめた資料を用意する。
展開	<ul style="list-style-type: none"> ● 【プログラムの実験】 ・ コースを走行させるためのプログラムを作成し、与えられたテストコースで走行実験を行う。 ・ 実験結果から修正案を検討し、繰り返し実験を行う。 ● 【活動の記録】 ・ 活動を文章で記録したり、より具体的な情報としてカメラを用いて記録する。 	<ul style="list-style-type: none"> ○ 毎時、作業の進捗を確認し、遅れているグループには助言を行う。 ○ うまくコースを走行できないときに、どのような状態であるが、どうしたらよいか、モデルを考え、整理させる。 ☆ 状況を判断し、グループでアイデアを出し合い、解決に向けて活動することができる。 【思考力・判断力・表現力等】
まとめ	<ul style="list-style-type: none"> ● 【次回の計画】 ・ 教材の状態を確認し、故障や異常があれば報告する。 ・ 次回の実験に向けて内容をグループで検討し、計画変更があるか確認する。 	<ul style="list-style-type: none"> ○ P D C A サイクルを基に活動を振り返り、問題点の洗い出しと今後の計画の見直しをする。

授業の実際

【ポイント1】 ●理解を深めるプログラミング学習の資料

<情報の科学>

授業プリント No. 31

◇プログラム言語「Python」

[19]繰り返し「while」

条件によって処理を繰り返す関数は、2つあります。今回の while 関数は for 関数とは違った働きをします。

○ while 継続条件： (説) (式や関数の)継続条件が真である限り、処理を繰り返す

~~~~~繰り返す処理

for 関数は繰り返す回数を指定できる。 while 関数は繰り返すための条件を指定できる。

例1) <ファイル名 while\_001> while 関数を動かしてみよう。

|              |                                |              |
|--------------|--------------------------------|--------------|
| a = 0        | (読)変数 a に数値 0 を入れる             | 戻って<br>繰り返す! |
| while a < 5: | (読)変数 a が5より小さいならば、処理を繰り返す     |              |
| print(a)     | (読)変数 a を表示しろ                  |              |
| a = a + 1    | (読)変数 a に変数 a と数値 1 を足した結果を入れる |              |
| print(a)     | (読)変数 a を表示しろ                  |              |

※このプログラムを実行すると終了するための条件が発生しない「無限ループ」が発生します。

例2) <ファイル名 while\_002> 変数aを1ずつ増加させて、10まで表示させてみよう。

|               |                                |              |
|---------------|--------------------------------|--------------|
| a = 0         | (読)変数 a に数値 0 を入れる             | 戻って<br>繰り返す! |
| while a < 10: | (読)変数 a が10より小さいならば、処理を繰り返す    |              |
| a = a + 1     | (読)変数 a に変数 a と数値 1 を足した結果を入れる |              |
| print(a)      | (読)変数 a を表示しろ                  |              |
| print(a)      | (読)変数 a を表示しろ                  |              |

必ず while 関数の繰り返しを終了させるための条件を、発生させるようにコーディングします。

【練習問題】<ファイル名 while\_002.EX> 変数 c を増加させて、「0から20までの奇数が表示される」ようにしない。

|                |                                 |            |              |
|----------------|---------------------------------|------------|--------------|
| 書き換えるポイントは     | ※繰り返す回数                         | ※変数aの増え方   | 戻って<br>繰り返す! |
| a = 1          | (読)変数 a に数値 1 を入れる              |            |              |
| c = 0          | (読)変数 c に数値 0 を入れる              | →回数カウンタ用変数 |              |
| while c < 20:  | (読)変数 c が20より小さいならば、処理を繰り返す     |            |              |
| print( a + c ) | (読)変数 a と変数 c を足した結果を表示しろ       |            |              |
| c = c + 2      | (読)変数 c に、変数 c と数値 2 を足した結果を入れる |            |              |

【応用問題】<ファイル名 while\_003> 文字列「\*」が1行ごとに1つ増え、5行まで表示されるプログラムを作成した。

|              |                                 |              |
|--------------|---------------------------------|--------------|
| a = *        | (読)変数 a に文字列「*」を入れる             | 戻って<br>繰り返す! |
| b = a        | (読)変数 b に変数 a を入れる              |              |
| c = 0        | (読)変数 c に数値 0 を入れる              |              |
| while c < 5: | (読)変数 c が5より小さいならば、処理を繰り返す      |              |
| print( a )   | (読)変数 a を表示しろ                   |              |
| a = a + b    | (読)変数 a に、変数 a と変数 b を足した結果を入れる |              |
| c = c + 1    | (読)変数 c に、変数 c と数値 1 を足した結果を入れる |              |

プログラム言語は Python を使用する。基本構造を学習した後に、判断分岐や繰り返しの記述について学習する。記述されている内容を理解するため、プログラムの「読み下し文」を併せて記載する。また、変数の値の変化を確認しながら、コンピュータが処理している内容を時系列で理解する。

### 【ポイント2】 ●作成したプログラムで実際に動かす



micro:bit とプログラミングカーを用いて、作成したプログラムが想定した通りに動くかを実感できるようにしている。プログラミングカーの扱い方に注意しながら、実験を繰り返し、課題を解決するためのプログラムを模索する。

### 【ポイント3】 ●問題解決のためのグループワーク



少人数のグループで活動し、P D C A サイクルを意識しながら、限られた実験の時間の中で問題を解決するためのアイデアを出し合うようにしている。また、実験の内容を記録し、パワーポイントによる発表の素材を用意させる。その際に I C T 機器を利用して伝わりやすい内容にできるよう、活用方法を学習させる。

## 今後に向けて

- 与えられた課題に対する解決方法について、優先順位を考えながら考えるためのワークシートを工夫する。その中で、実験の進捗を確認するための目安を作り、より多くの情報を処理できるようにする。今回はブロックコーディングがメインだったが、今後は学習したプログラム言語を用いてコーディングできるように指導する。