群馬高専

高専は、中学校卒業後に進学する際の選択肢のひとつ。

5年制。卒業後は就職したり大学３年に編入したりする。

実験や実習が多い（提出物は大事）。所属学科に無関係に学生ならだれでも利用できる「アントレプレナーシップ教育工房」では、３Ｄプリンタやレーザーカッター、工作機器なども備えている。

→ 手をうごかしたり自分の力で何かをつくってみたいというひとにオススメ。

機械工学科、電子メディア工学科、電子情報工学科、物質工学科、環境都市工学科の ５学科があり、１学年200人（構成は、高専により異なる）。

プログラミングってなに

・ 目的を達成するために、なにをどうすればよいのかを事前に指示したもの。

・ 具体的にどのような手順で実行するのかを指示する内容（= アルゴリズム）を具体的に記述したもの。

・（主に）PCやスマホなどを動かすための指示。PCが理解できるために、特別な文法の言葉（= プログラミング言語）を使うことが多い。例：パイソン、C++、BASIC

象を冷蔵庫に入れる？

・ たしかグーグル社の入社試験だったとか？
・ 象を冷蔵庫にいれるための３つのステップを教えてください。

別解：レイゾウコには、すでにゾウが入っている…
　たしかに。でも今は別の答を探してみましょう。

［考えてみましょう］

象を冷蔵庫に入れる？

- 冷蔵庫のドアを開く
- 中に象を入れる
- ドアを閉める

→ では、今度はキリンさんの登場です。この冷蔵庫にキリンを入れるための４つのステップを教えてください。

キリンを冷蔵庫に入れる？

- 冷蔵庫のドアを開く
- 象を外に出す
- 代わりにキリンを入れる
- ドアを閉める

→ 象が入ったままだと、キリンをいれることができないみたい。

冷蔵庫へ象を代入する？

だまされたみたいな気持ちになっていませんか？

実は、これは、変数（PCなどの上で数値等を記憶しておくための領域）に、数値を代入する（保存するため値を変数名などと紐づける）操作を、「冷蔵庫」に「象」を入れるという具体的な操作で書いているだけなのです。こじつけ、というより情報処理的な考え方をすると、こうなるよ、ということなのです。

やることが同じでも、プログラム言語の仕様によっては、単純に「冷蔵庫に象を入れる」と書けばよいだけのものもあります。

プログラム（アルゴリズム）の練習

・目的を達成する（コロッケが食べたい！）とき、なにをどうすればよいのでしょうか。

［考えてみましょう］

プログラム（アルゴリズム）の練習

・目的を達成する（コロッケが食べたい！）とき、なにをどうすればよいのでしょうか。

→ 我慢する！（目的が達成できません）

→ コロッケを買ってくる / 買ってきてもらう

→ 自分でコロッケを作る / 作ってもらう

→ そして食べる！

同じ目的であっても、手段（アルゴリズム）はいろいろあります。

プログラム（アルゴリズム）の練習

・目的を達成する（コロッケが食べたい！）とき、なにをどうすればよいのでしょうか。

→ コロッケを何らかの方法で用意する

→ コロッケにかじりつく

→ 咀嚼して飲み込む

→ etc

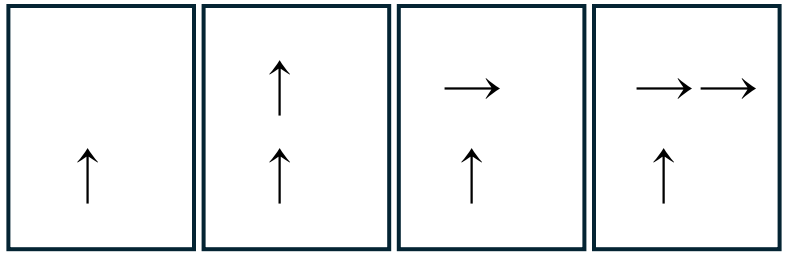
どこまで細かく記述するのかも、いろいろあります。

プログラム

・PCへの指示を出すとき次の３つの構造を覚えておきましょう
ロボットをPC内で動かすつもりになってみましょう。

順次構造：命令をひとつずつ積み上げます。
上（はじめ）から順に実行します。

例 ：・前へ進め。
・前へ進め。
・右を向け。
・前へ進め。



プログラム

・ PCへの指示を出すとき次の３つの構造を覚えておきましょう

反復構造：命令を決まった回数（または条件を満たすまで）
繰り返します。

例 ：「７回」の間（次の内容を）繰り返せ。
｜　・前へ進め。
└（繰り返す内容はここまで）

→ ７歩前へ進むことができます

プログラム

・ PCへの指示を出すとき次の３つの構造を覚えておきましょう

反復構造：命令を決まった回数（または条件を満たすまで）
繰り返します。

例 ：「壁につきあたるまで」（次の内容を）繰り返せ。
｜　・前へ進め。
└（繰り返す内容はここまで）

→ 壁際まで前へ進むことができます
→ 壁がない場合、どこまででも前へ進みます

プログラム

・ PCへの指示を出すとき次の３つの構造を覚えておきましょう

条件分岐：条件を満たすかどうかで、実行する内容を選択します

例 ：「目の前が壁」ならば。
```
|    ・とまれ（移動をしない）
├（そうでないならば）
|    ・前へ進め
└（条件分岐終了）
```

どちらかを選択
一方が空でもOK

条件分岐の例

条件分岐：条件を満たすかどうかで、実行する内容を変えます

例 ：「雨が降っていない」ならば。
　　・外に出て遊ぶ。
（そうでないならば）
　　・教室の中で遊ぶ。
（条件分岐終了）

→ みなさんも普段ふつうにやっていることです。
→ でもこの条件分岐や反復を上手に組み合わせてやると、どんな複雑な処理も書き表すことができることが知られています。

目の前に壁？

・ PCの中で設定されている地形情報（マップ）に、壁が設定されている場合、位置情報から壁があるとわかります。

・ 実在する壁の場合、「センサー」を用いて、壁があったら何かの信号（反射する光や音など）が帰ってくることで検出できます。

・地面に描いた線をセンサーで追跡すると、その線に沿って動くようにプログラムすることもできます。

長い間生きていると、目の前に壁が迫ったように感じることもありますが、すぐにあきらてしまうのではなく、回り込んだり、工夫したりして乗り越えることもできるでしょう。進路変更して別の方向に進む方が良い場合もあります。

正しい（PCに伝わる）書き方

「リンゴを１個買ってきて。タマゴがあったら10個買ってきて」

さて条件分岐（「タマゴがあるかどうか」で行動が変わる）の問題です。上のようにお使いを頼まれました。スーパーに行ったら、タマゴがありました（売られていました）。さあ、何をいくつ買ってきますか？

［考えてみましょう］

正しい（PCに伝わる）書き方

「リンゴを１個買ってきて。タマゴがあったら10個買ってきて」

Ａくん：リンゴを11個買ってきたよ。タマゴがあったから。
Ｂくん：リンゴを10個買ってきたよ。タマゴがあったから。
Ｃくん：え？リンゴ１個とタマゴ10個を買ってきたんだけど。

たぶん、みんな正しいのです。
でも、お使いをたのむ方はどういう気持ちだったのでしょう。
実はPCは、情緒や「普通はこうでしょう」とか通じません。
自分の意図がつたわるように命令をかいてやる必要があります。

正しい（PCに伝わる）書き方

「リンゴを１個買ってきて。タマゴがあったら10個買ってきて」

Aくんの考え方

・ リンゴを１個買う
タマゴがあったら
　┃ ・ リンゴを10個買う
　┣ タマゴが無いとき
　┃ ・ 追加ではなにもしない
　┗ 条件分岐終わり

Bくんの考え方

タマゴがあったら
　┃ ・ リンゴを10個買う
　┣ タマゴが無いとき
　┃ ・ リンゴを１個買う
　┗ 条件分岐終わり

正しい（PCに伝わる）書き方

「リンゴを１個買ってきて。タマゴがあったら10個買ってきて」

Cくんの考え方

・ リンゴを１個買う
タマゴがあったら
　├ ・ タマゴを10個買う　　← 感覚的にはこれが正解？
　├ タマゴが無いとき　　「でもタマゴを10個買うなんて
　│ ・ 追加ではなにもしない　　誰にも言われてないよ！」
　└ 条件分岐終わり　　屁理屈でしょ、って怒られる？

プログラミングってそんなもん

Ｃくんのような応答を期待して頼んだのに、ＡくんやＢくんのような応答が却ってくると…

頼んだ方は、なんで自分の言いたいことが伝わらないのだろうとなやんだり、怒ったりする。でもPC相手に怒っても仕方ない。

PCって、プログラミングって、そんなもんだよね、って諦めながら頼み方（プログラミングの書き方）の方を合わせていく。

「リンゴを１個買ってきて。
　それからタマゴがあったら、タマゴも10個買ってきて」
　こう頼んだのにリンゴを11個買ってきたら怒ってもOK

プログラミングってそんなもん

他にも、変数（数値や文字列などを扱う）だったり、関数（どのボタンを押せばどの飲み物がでてくるか決まっている自動販売機のように、入力に対する出力がなんらかの規則で決まっているもの）だったり、プログラミングをする上では、いろいろと面倒くさいものもある。

まあ、それはまだまだ先の話。

プログラミングってそんなもん

プログラムが指示する先が、ただ一つの対象（オブジェクト）である場合（ロボットを画面上で動かす、だとか）もあるし、プログラムの中で対象とするもの（オブジェクト）ごと、それぞれ別に決められたルールで動いたり対応したりするようにあらかじめ指示されているような「オブジェクト志向の」プログラム（RPGで、村人A, 村人B, 村人C がそれぞれに決まった別の挨拶を返してくるようになっているもの）なんていう違いもある。

まあ、それもまだまだ先の話。

まずは気軽にできることから楽しんでみよう。

プログラミング言語の例（中級）

十進BASIC

https://decimalbasic.web.fc2.com/

テキストベース
英単語が分かれば意味がなんとなく通じる。

数学の専門家が作って管理している真面目な（数値計算やグラフ作成がちゃんとしている）フリーウェア。

慣れてくると、ビジュアル（ブロック）系の言語より、
細かいことを丁寧に伝えることができるようになる。

十進BASICでロボットを動かすときの命令の例

```
! 繰返し
FOR k = 1 TO 3                  ! 3回繰り返します。
   CALL MoveForward             !   ・ 前へ進みます。
NEXT k                          ! 繰返しここまで

CALL SearchFront                ! 地図で自分の前が何か調べます。

! 条件分岐
IF SearchResult = "0" THEN      ! 調べた結果何もない("0")ならば
   CALL MoveForward             !   ・ 前へ進みます。
ELSE                            ! そうでない（前になにかある）なら
   CALL TurnRight               !   ・ 右を向きます。
END IF                          ! 条件分岐ここまで
```

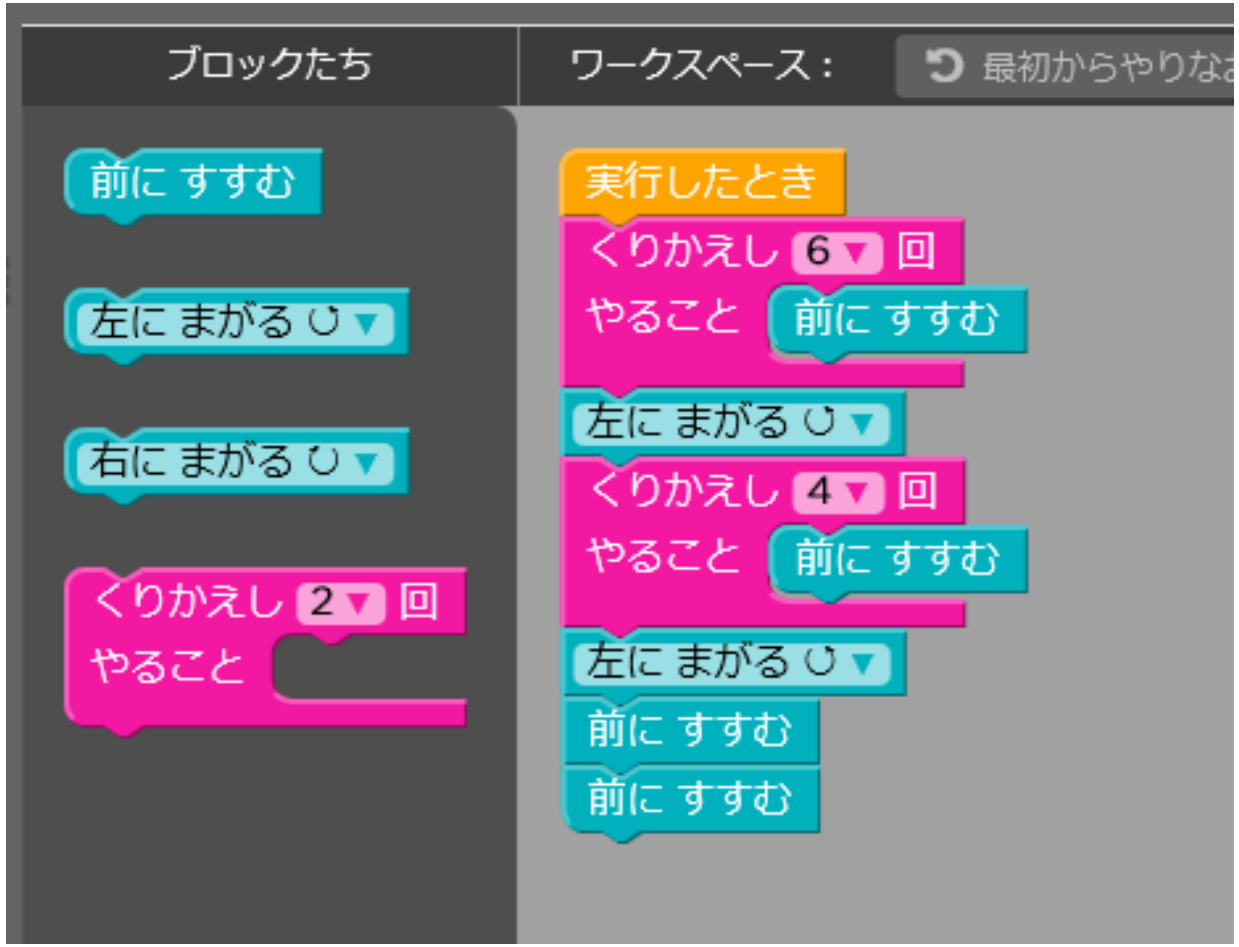
十進BASICで遊んでみるために

この命令を動かしてみたいなら

1) 自分のPCに「十進BASIC」フリーウェアをDLする
2) 次のページから、「プログラミング導入学習　ロボット操縦ゲーム (RoboSteering.txt)」をDLし、コピー、ペーストして実行する。
3) 上記プログラムの中の「WorkSpace」のところに、自分なりのロボットの動きを設計して書いてみる。思った通りに動くか確認する。

群馬高専 物質工学科のHP https://www.chem.gunma-ct.ac.jp/
→ スタッフ紹介
→ 中島 （の名前をクリック） : 中島のホームページ
https://www9.gunma-ct.ac.jp/staff/nakajima/

プログラミング言語の例（初級）



画像は、教材のウェブページ（http://hourofcode.jp/）からのスクリーンショットです。

プログラミング言語の例

Hour of Code （１時間でプログラム）

http://hourofcode.jp/

ビジュアルなブロックベース / 日本語表示にも対応

Code.orgというプログラミングを推進するアメリカの非営利団体が運営するプログラミング学習サイト。無料のチュートリアルがたくさんあり、プログラミングの構造の取っ掛かりとして、グラフィカルで分かりやすいです。特定非営利活動法人みんなのコードが日本国内の展開を推進しています。

→「アクティビティの探索」

https://hourofcode.com/au/ja/learn

→ 言語切替えは、ページ右上にあります

「ブロック」言語で遊べるアクティビティを選ぶ（例）

→ マインクラフト（Minecraft） の Hour of Code

→ マインクラフト・ヴォヤージュ・アクアティック

→ マインクラフト・ヒーローズ・ジャーニー

→ マインクラフト・アドベンチャラー

→ マインクラフト・デザイナー　　etc.

→ フラッピーゲームを造る　　（ミニゲーム作成）

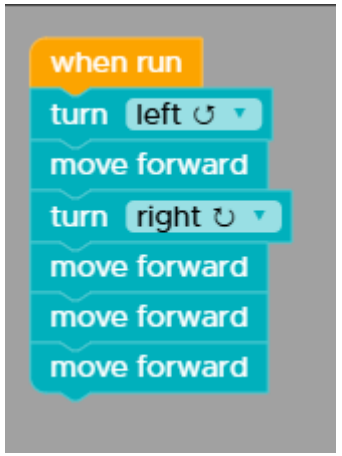
→ アナとエルサとコードを書く（図形の描画）　　etc.

画像は、教材のウェブページ（http://hourofcode.jp/）からのスクリーンショットです。

How To Play

好きなアクティビティを選んで、「GetStarted / 開始する」。
キャラクターを選んで「Select / 選ぶ」で始まります。
動画は見ても見なくてもOK（右上の×で閉じます）
アクティビティを選んだあとからでも、言語の選択ができます。
音がでるアクティビティもあります。音量に配慮してください。

同じ目的を解決するためのプログラム（コード）でも、長い（行数の多い）ものも、短い（行数の少ない）ものもありますが、どちらが正しいというわけではありません。強いていうなら、他人が読みやすく、意図がすっきりとわかりやすいプログラムの方がよいです。



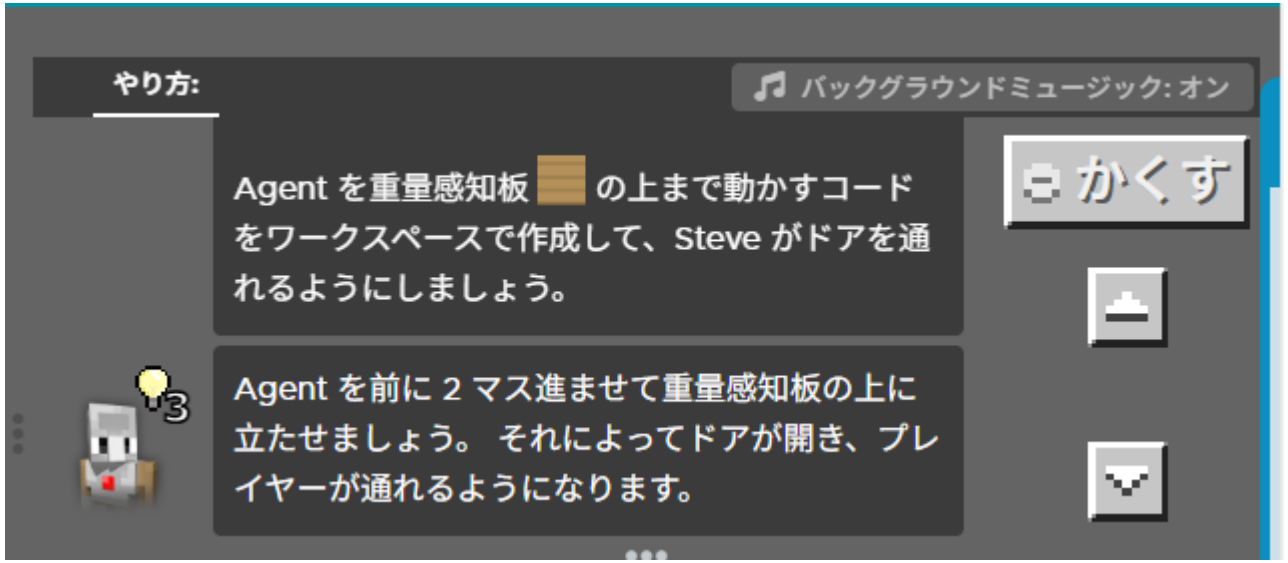
これでもOKみたい。

Minecraft Voyage Aquatic（マインクラフト大航海）を選んでみました。

レベル１では、箱の中からボートを取り出すのが目的です。主人公（のロボット？）を、箱の前または隣まで移動させましょう。

命令を並べたら「RUN / 実行」を押すと結果がわかります。失敗したらやり直し！

画像は、教材のウェブページ（http://hourofcode.jp/）からのスクリーンショットです。



Minecraft Hero’s Journey（マインクラフトヒーローの旅）を選んでみました。

Agent（アシストロボット）を操縦して、「重量感知板」の上にのせてください。

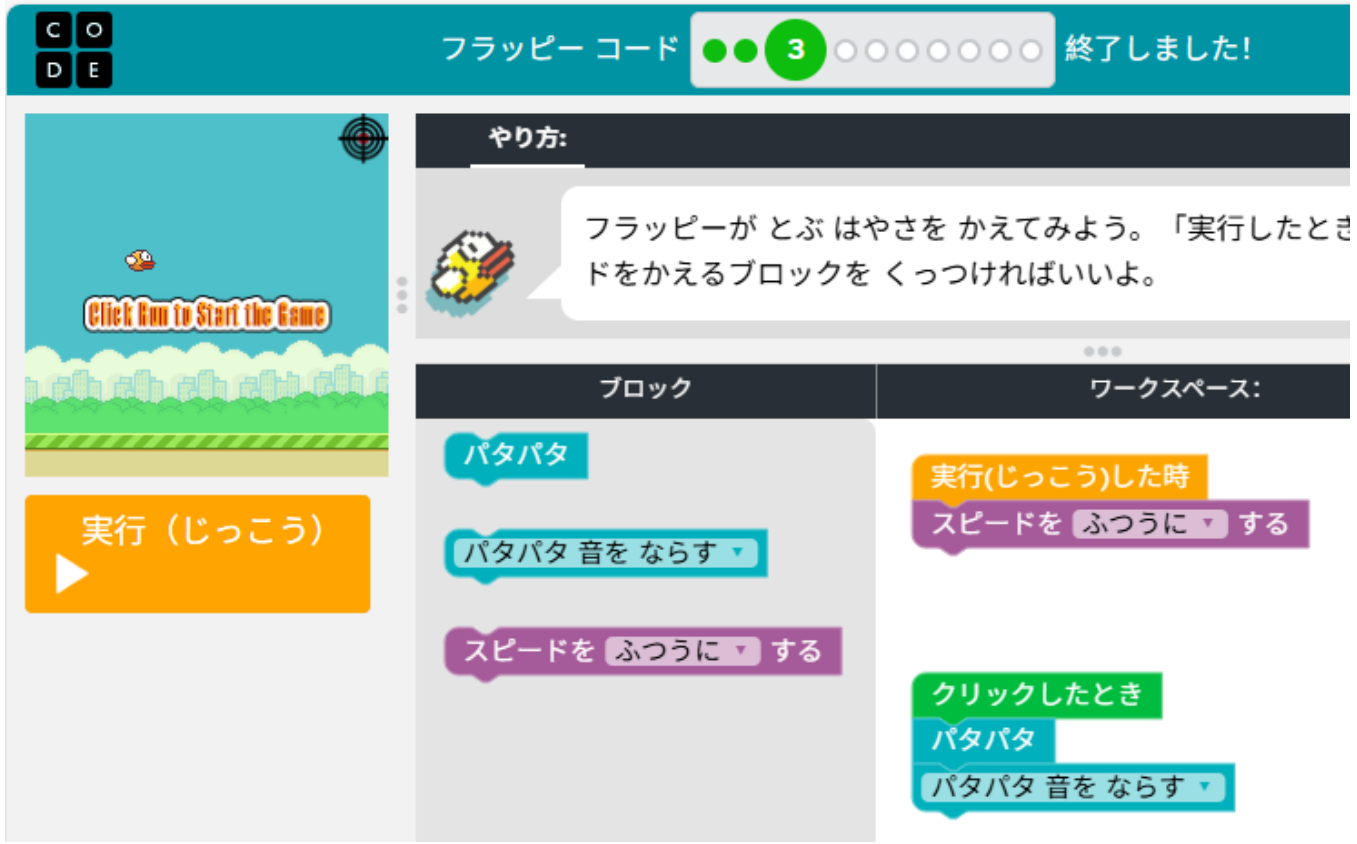
そのあと、主人公ロボットをカーソル（左図、またはキーボード）で移動させます。

画像は、教材のウェブページ（http://hourofcode.jp/）からのスクリーンショットです。

フラッピーゲームを造る

https://hourofcode.com/flap

ドラッグ&ドロップ・プログラミングで、自分だけのフラッピーバード・ゲームを造りましょう。



「実行」を押してから、左上のゲーム画面をクリックで操作します。

画像は、教材のウェブページ（http://hourofcode.jp/）からのスクリーンショットです。

アナとエルサとコードを書く
https://hourofcode.com/frzn

画面（氷）に、いろいろな図形を書いてみましょう。

簡単なプログラムで複雑な図形も描けるのを是非体験してみて。

画像は、教材のウェブページ（http://hourofcode.jp/）からのスクリーンショットです。