

# toio を用いて 基本技法からアルゴリズム までを学ぼう

## ステージ1【基本操作要素技法レベル】

2023/12/8

### 0. 本教材の使い方

#### この教材で学ぶことと、プログラム(コード)作成のヒント

- この教材では、プログラムを作り基本的な考え方を学んでいきます。プログラミングを体系的に学ぶためには、逐次処理、繰り返しや条件分岐などの技法レベルだけではなく、データ処理技法、基本アルゴリズムとの相互の関係性を学ます
- 本教材には3つの特徴があります。一つがスマートステップで段階的に学んでいくこと、2つめが繰り返し同じような内容を行う、3つめは課題からプログラムを考える他に間違いを探すおよび作成されたプログラムを読んで理解する特徴があります。これによりプログラミングに対する理解を定着させる効果が期待できます
- 最後に、プログラムの正答例は掲載していません。その理由は、プログラムが正しいかどうかは、自分が考えた結果通りになっているかどうかを自分で考えてもらうようになっているためです

#### 操作方法について

- 本教材に設置されているリンクを開く場合は次のようにいてください
  - [Windowsの場合] CTRキーを押しながらリンクをクリックする
  - [Macの場合] Commandキーを押しながらリンクをクリックする
- アプリ版の場合は、本教材の各ページにある2次元バーコードをタブレットで読みでアプリを起動してください
- マウスに左右のボタンがない場合に、右クリックで(複製、貼り付けなどを行う)サブメニューを開く場合
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック

- 本教材に設置されているリンクを開いた後は必ず次のようにいてください。
  - 【縁のはた】あるいは【マップの作成】をクリック
  - コード作成エリアの右下にある、 ボタンをクリック

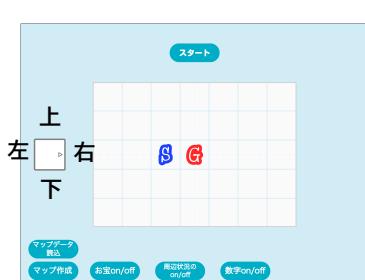
# 1. 逐次, 繰り返し, 条件分岐の基本

ステージ1

【基本操作要素技法レベル】

## 1. 逐次, 繰り返し, 条件分岐の基本

i 一つ前に進もう



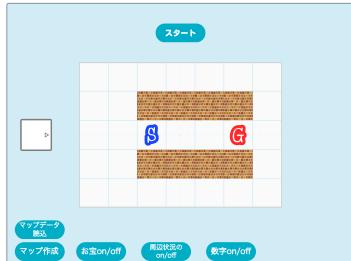
この課題で使えるとされている主な命令ブロック	
スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して, 指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します

### この課題で学ぶこと, プログラム(コード)作成のヒント

- この課題では、教材の使い方とプログラムの基本を学びます
- 命令ブロックを組み合わせてスタート「S」からゴール「G」までの動きを作ります
- その動きのことをアルゴリズム(日本語では手順)と呼びます
- アルゴリズム(手順)を考えて命令ブロックを組み合わせていきます
- 【スタート位置に移動して( )に向ける】と【前に一マス進む】命令ブロックの動きを確認しましょう

# 1. 逐次、繰り返し、条件分岐の基本

## ii 3つ前に進もう



この課題で使えるとされている主な命令ブロック

スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します

## この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、教材の使い方とプログラムの基本を学びます
- 命令ブロックを組み合わせて動きを作ります
- スタートの「S」からゴールの「G」までの経路を考えます
- SからGに行くには、何マス進めばよいでしょうか
- その数だけ【前に一マス進む】命令ブロックをつなぎましょう

# 1. 逐次、繰り返し、条件分岐の基本

## iii L字の経路に沿って進もう



この課題で使えるとされている主な命令ブロック

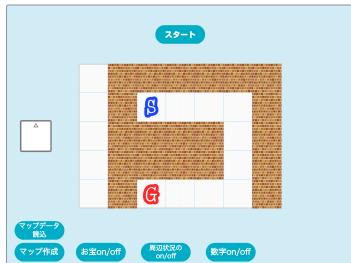
スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが進んでいる方向から右に90度回転します
左に90度回転	toioが進んでいる方向から左に90度回転します

## この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、教材の使い方とプログラムの基本を学びます
- 命令ブロックを組み合わせて動きを作ります
- スタートの「S」からゴールの「G」までの経路を考えます
- SからGに行くには、何マス進めばよいでしょうか
- 途中でL字に曲がっています。どちらに向きを変えるとよいでしょうか
- toioの向きは、進行方向からどちら側になるかで決まります

# 1. 逐次、繰り返し、条件分岐の基本

## iv コの字の経路に沿って進もう



### この課題で使えるとされている主な命令ブロック

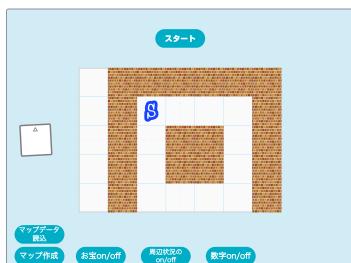
スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが進んでいる方向から右に90度回転します
左に90度回転	toioが進んでいる方向から左に90度回転します

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、教材の使い方とプログラムの基本を学びます
- 命令ブロックを組み合わせて動きを作ります
- スタートの「S」からゴールの「G」までの経路を考えます
- SからGに行くには、何マス進めばよいでしょうか
- 途中でL字に曲がっています。どちらに向きを変えるとよいでしょうか
- 命令を順番に実行するプログラムの構造を「逐次処理」と言います

# 1. 逐次、繰り返し、条件分岐の基本

## v 四角形に沿って進もう



### この課題で使えるとされている主な命令ブロック

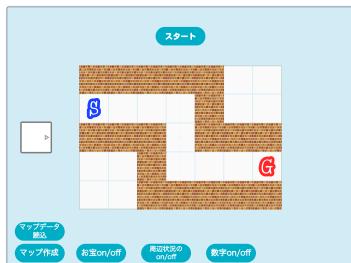
スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが進んでいる方向から右に90度回転します
左に90度回転	toioが進んでいる方向から左に90度回転します

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、教材の使い方とプログラムの基本を学びます
- 命令ブロックを組み合わせて動きを作ります
- スタートの「S」から始まり、時計回りに動き、もう一度「S」まで戻ってくる経路を考えます
- 途中で口の字に曲がっています。どちらに向きを変えるとよいでしょうか
- 命令を順番に実行するプログラムの構造を「逐次処理」と言います

# 1. 逐次、繰り返し、条件分岐の基本

## vi 間違い探し: 正しく経路に沿って進むよう修正しよう



### この課題で使えるとされている主な命令ブロック

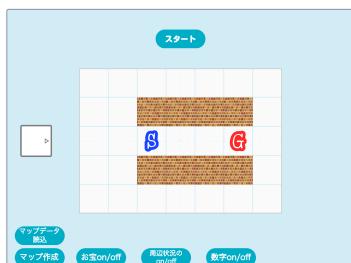
スタート位置に移動して 右 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが進んでいる方向から右に90度回転します
左に90度回転	toioが進んでいる方向から左に90度回転します

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、すでに作られているプログラムの間違いを探して修正します
- プログラムの間違いを修正することを「デバッグ」と言います
- プログラムを読んで、どのようにtoioが動作するか考えます。それが正しいかをマップ上で考えた経路と同じかを考えましょう
- 違つたら、その部分がプログラムのどの部分かを考えましょう
- 難しい場合は、まずは動かして、間違つた部分を確認しましょう
- プログラムを読む力が付くとプログラムを作成する能力が飛躍的に高まります

# 1. 逐次、繰り返し、条件分岐の基本

## vii 同じ動作をまとめて見よう ~まっすぐ進もう



### この課題で使えるとされている主な命令ブロック

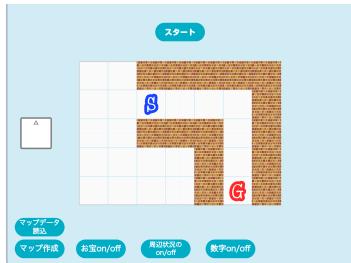
前に一マス進む	toioが向いている方向に一マス移動します
3 回繰り返す	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、繰り返し処理の考え方を学びます
- 同じマップを使った「ii 3つ前に進もう」で作成したプログラムを思い出そう
- 【前に一マス進む】命令ブロックが連続して何回使われていますか
- その回数を、繰り返しをする命令ブロックに指定して、繰り返される命令をワニの口の中に入れてみよう
- 回数を指定して繰り返す命令は、同じ命令ブロックが連続して使われる場所を探して、それが何回使われるかを考えると良いです

# 1. 逐次、繰り返し、条件分岐の基本

## viii 同じ動作をまとめて見よう ~L字に進もう



この課題で使えるとされている主な命令ブロック

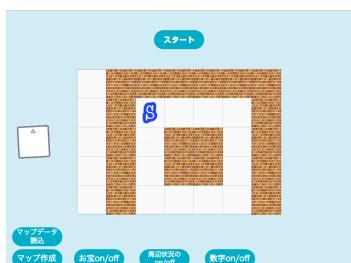
前に一マス進む	toioが向いている方向に一マス移動します
3回繰り返す	かいすうしていくりかえすせいやょめいれいわにくちのぱしょいれめいれいえると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、繰り返し処理の考え方を学びます
- 同じマップを使った「iii L字の経路に沿って進もう」で作成したプログラムを思い出そう
- 【前に一マス進む】命令ブロックが連続して何回使われていますか
- その回数を、繰り返しをする命令ブロックに指定して、繰り返される命令をワニの口の中に入れてみよう
- 回数を指定して繰り返す命令は、同じ命令ブロックが連続して使われる場所を探して、それが何回使われるかを考えると良いです

# 1. 逐次、繰り返し、条件分岐の基本

## ix 同じ動作をまとめて見よう ~四角形にそって進もう



この課題で使えるとされている主な命令ブロック

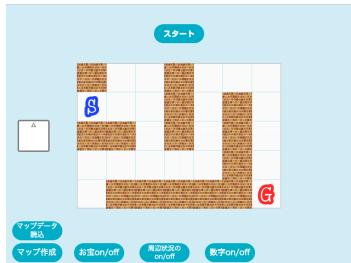
前に一マス進む	toioが向いている方向に一マス移動します
3回繰り返す	かいすうしていくりかえすせいやょめいれいわにくちのぱしょいれめいれいえると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、繰り返し処理の考え方を学びます
- 同じマップを使った「v 四角形に沿って進もう」で作成したプログラムを思い出そう
- 命令ブロックの組合せ同じ箇所が何回使われていますか
- その回数を、繰り返しをする命令ブロックに指定して、繰り返される命令をワニの口の中に入れてみよう
- 回数を指定して繰り返す命令は、同じパターンになる命令ブロックが連続して使われる場所を探して、それが何回使われるかを考えると良いです

# 1. 逐次、繰り返し、条件分岐の基本

× 間違ひ探し: 正しく経路に沿って進むよう修正しよう その2



この課題で使えるとされている主な命令ブロック

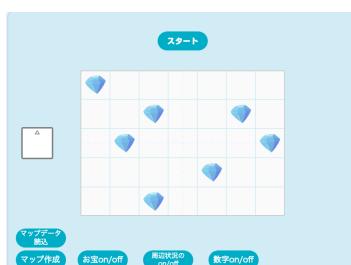
前に一マス進む	toioが向いている方向に一マス移動します
3 回繰り返す	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

## この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、すでに作られているプログラムの間違ひを探して修正(デバッグ)を行います。間違ひの箇所は1カ所です
- プログラムを読んで、どのようにtoioが動作するか考えます。それが正しいかをマップ上で考えた経路と同じかを考えましょう
- 違つたら、その部分がプログラムのどの部分かを考えましょう
- 難しい場合は、まずは動かして、間違つた部分を確認しましょう

# 1. 逐次、繰り返し、条件分岐の基本

xi お宝があつたら LED を光らせよう



この課題で使えるとされている主な命令ブロック

もし もし、〇〇なら、■■を行 い、でなければ△△を行う	条件分岐の命令(制御)ブロックです。「もし、〇〇なら、■■を行い、でなければ△△を行う」ことを意味しています。 〇〇は「もし」の後の□に入れます。■■は上の□の中、△△は「でなければ」の後の□の中に入れます
= お宝	条件です、=(等しい)の右と左の〇の中身が同じなら「真」、そうでなければ「偽」です
真下のアイテム	toioを置いてあるマップのマスに書かれているデータを取得します
ランプの色 を タイマー秒つける	toioのLEDを指定した秒数だけ光らせます

## この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、条件分岐の考え方を学びます
- マットの上にtoioを置きます。置いた場所にお宝があれば以下のように動作するプログラムを作成します。置く場所は自由です
- toioを置いた場所にお宝の上にあればtoioのLEDが青く光、そうでなければ赤く光ります
- toioを置いたマスのデータは、命令(変数)ブロックである【真下のアイテム】で取得できます

# 1. 逐次、繰り返し、条件分岐の基本

## xii 経路上にお宝があったら LED を光らせよう



### この課題で使えるとされている主な命令ブロック

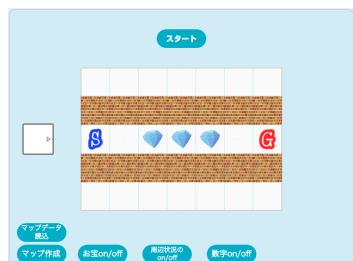
	条件分岐の命令(制御)ブロックです。「もし, ○○なら, ■■を行う」ことを意味しています。 ○○は「もし」の後の枠に入れます。■■は上の□の中に入れます
	条件です, =(等しい)の右と左の○の中身が同じなら「真」, そうでなければ「偽」です
	toioを置いてあるマップのマスに書かれているデータを取得します
	toioのLEDを指定した秒数だけ光らせます

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、条件分岐の考え方を学びます
- toioをマットの上の「S」に置きます。経路にそってゴール「G」まで移動させます
- toioがお宝の上にあればtoioのLEDを赤く光らせます。そうでなければ何も光りません
- toioを置いたマスのデータは、命令(変数)ブロックである【真下のアイテム】で取得できます

# 1. 逐次、繰り返し、条件分岐の基本

## xiii お宝が何個あったか数えよう



### この課題で使えるとされている主な命令ブロック

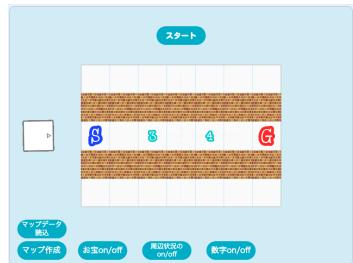
	条件分岐の命令(制御)ブロックです。「もし, ○○なら, ■■を行う」ことを意味します。 ○○は「もし」の後の枠に入れます。■■は上の□の中に入れます
	変数「お宝の数」の値を0にする命令です。
	変数「お宝の数」の値に1を加えて「お宝の数」の値とする
	変数「お宝の数」の値と「個」と「秒」をつなげた文字列を2秒言います

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、条件分岐の考え方を学びます
- toioをマットの上の「S」に置きます。経路にそってゴール「G」まで移動させます
- toioがお宝の上にあれば変数「お宝の数」に1を加えます。最初はお宝の数は0から始めましょう
- toioを置いたマスのデータは、命令(変数)ブロックである【真下のアイテム】で取得できます

# 1. 逐次、繰り返し、条件分岐の基本

## xiv 経路上に隠された数字はいくつ？



この課題で使えるとされている主な命令ブロック

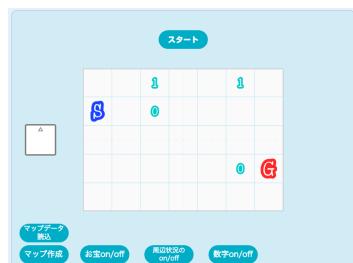
	もし [ ] なら [ ] でなければ [ ]	条件分岐の命令(制御)ブロックです。「もし, ○○なら, ■■を行い, そうでなければ△△を行う」ことを意味しています。 ○○は「もし」の後の枠に入れます。■■は上の口の中, △△は「でなければ」の後の口の中に入れます
	S_数のリスト ▾ に 真下のアイテム が含まれる	変数「真下のアイテム」の値が数字なら条件を満たす(真)、そうでなければ条件を満たさない(偽)と判断する

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、条件分岐の考え方を学びます
- toioをマットの上の「S」からスタートして、経路にそってゴール「G」まで移動させます
- toioが数字の上にあれば、その数字をしゃべります

# 1. 逐次、繰り返し、条件分岐の基本

## xv 真下の数字が1だったら右へ、0だったら左へ



この課題で使えるとされている主な命令ブロック

新しく使う命令はありません

### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、条件分岐の考え方を学びます
- toioをマットの上の「S」からスタートして、経路にそってゴール「G」まで移動させます
- toioが数字「0」の上にいたら【左に90度回転】して前に進めます。数字が「1」の場合は【右に90度回転】して前に進めます。
- 「0」か「1」以外の場合は、まっすぐ前に進みます

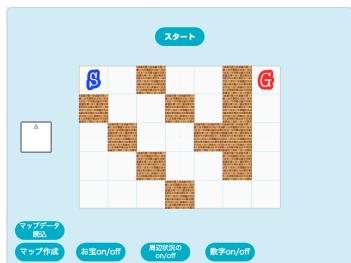
# 1. 逐次、繰り返し、条件分岐の基本

## xvi 間違い探し: 正しく経路に沿って進むよう修正しよう その3



この課題で使えるとされている主な命令ブロック

新しく使う命令はありません



### この課題で学ぶ事と、プログラム(コード)作成のヒント

- この課題では、プログラムの間違いを探して修正します
- プログラムが示しているtoioの動きと、実際のマップでどのように動かせば良いかとの差が間違いになります
- まずは、このプログラムがどのように動くか考えてみましょう
- 次に、このマップをどのように進めばよいか考えてみましょう
- 違いはありましたか？ 違っている場所を修正しましょう

# 1. 逐次、繰り返し、条件分岐の基本

## xvii ブロックを避けてゴールを目指せ



この課題で使えるとされている主な命令ブロック

新しく使う命令はありません



### この課題で学ぶ事と、プログラム(コード)作成のヒント

- この課題では、ブロックを避けてゴールまでたどり着く経路を考えてプログラムを作ってください
- 使えるブロックは、コード作成エリアにあるブロックだけです

# 1. 逐次、繰り返し、条件分岐の基本

## xviii 数字を順番にたどろう！



この課題で使えるとされている主な命令ブロック

新しく使う命令はありません



### この課題で学ぶこと、プログラム(コード)作成のヒント

- この課題では、数字を順番に通りをゴールまでたどり着く経路を考えてプログラムを作ってください
- 使えるブロックは、コード作成エリアにあるブロックだけです

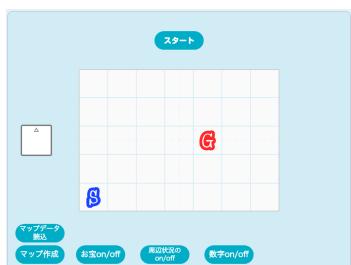
# 1. 逐次、繰り返し、条件分岐の基本

## xix これはどんな動きをするかな？



この課題で使えるとされている主な命令ブロック

新しく使う命令はありません



### この課題で学ぶこと、プログラム(コード)作成のヒント

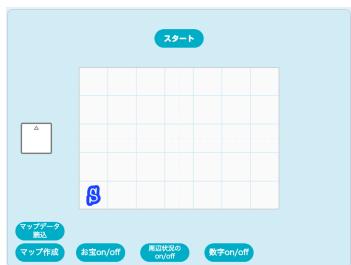
- これは、コード作成エリアにあるプログラムがどのようにtoioを動かすかを考える課題です
- プログラムを読んでtoioがどのように動くか経路を想像しましょう
- その後、実際に動かして思った通りの動きをしたか確認しましょう
- もし違った動きをしたら、もう一度プログラムを読んで確認しましょう

## 1. 逐次、繰り返し、条件分岐の基本 xx 自由に動きを作つてみよう！



この課題で使えるとされている主な命令ブロック

新しく使う命令はありません



### この課題で学ぶこと、プログラム(コード)作成のヒント

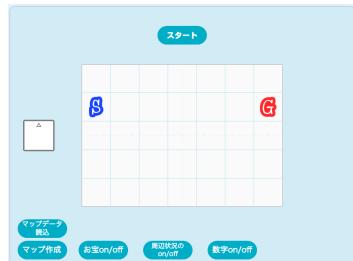
- これは、コード作成エリアにあるプログラムがどのようにtoioを動かすかを自由に考える課題です
- スタートからtoioを自由に動かしてみよう
- 使う命令ブロックは、コード作成エリアにあるブロックを複製(コピー)して使ってください。

## 2. 逐次処理を極めよ！

ステージ1

【基本操作要素技法レベル】

## 2. 逐次処理を極めよ！ i ひたすらまっすぐ進め



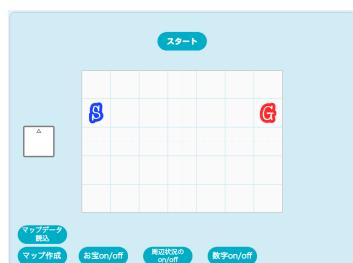
この課題で使えるとされている主な命令ブロック

スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路を考えます
- SからGに行くには、何マス進めばよいでしょうか。その数だけ【前に一マス進む】命令ブロックをつなぎましょう
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

## 2. 逐次処理を極めよ！ ii 間違探し:ひたすらまっすぐ進めるようにしよう



この課題で使えるとされている主な命令ブロック

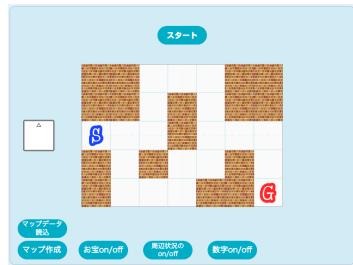
スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路をたどるプログラムがすでに作られています
- このプログラムは正しく動くか考えてみましょう。考えたら実際に動かしてみましょう。考えた通りに動きましたか
- 正しく動かない場合は、正しくゴールに行くように直しましょう
- この直す作業を「デバッグ」と呼びます

## 2. 逐次処理を極めよ！

### iii 迷路をゴールしよう



#### この課題で使えるとされている主な命令ブロック

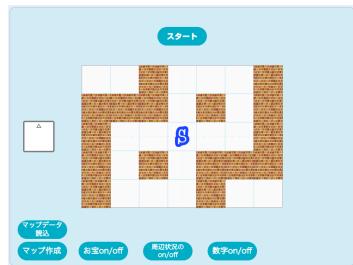
スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路を考えます
- 経路は複数あります。それぞれの経路でプログラムを作成してみましょう
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

## 2. 逐次処理を極めよ！

### iv 8の字を描くように進もう



#### この課題で使えるとされている主な命令ブロック

スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」から右方向に動くようにします。その後、8の字を描くように経路を巡り、最後にスタート位置まで戻ります。
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

## 2. 逐次処理を極めよ！

v 間違い探し:8の字を描くように進むように修正しよう



この課題で使えるとされている主な命令ブロック	
スタート位置に移動して 右 向ける	スタート位置「S」まで移動して、指定された方向(上下左右) にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

かだい まなぶこと さくせい  
この課題で学ぶことと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
  - スタートの「S」から始まり、右方向に進み8の字を描きながらスタートまで戻る経路をたどるプログラムがすでに作られています
  - このプログラムは正しく動かしてみましょう。考えたら実際に動かしてみましょう。考えた通りに動きましたか
  - 正しく動かない場合は、正しくゴールに行くように直しましょう

？ ちくじしょり きわめよ  
逐次処理を極めよ！

たからうえすべてとおつ いこう  
vi お宝の上を全て通つてゴールに行こう その1



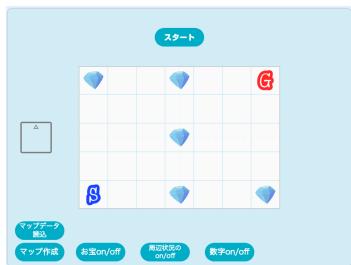
この課題で使えるとされている主な命令ブロック	
スタート位置に移動して 右に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

かだい まなぶこと さくせい  
この課題で学ぶこと プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
  - スタートの「S」からゴールの「G」までの経路を考えます。途中、すべてのお宝の上を通って進みましょう
  - 経路は複数あります。それぞれの経路でプログラムを作成してみましょう
  - 必要な命令は複数をして使います
    - [Macの場合] Controlキーを押しながらマウスボタンをクリック
    - [Windowsの場合] マススの右ボタンをクリック

## 2. 逐次処理を極めよ！

### vii お宝の上を全て通つてゴールに行こう その2



#### この課題で使えるとされている主な命令ブロック

スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路を考えます。途中、すべてのお宝の上を通つて進みましょう
- 経路は複数あります。それぞれの経路でプログラムを作成してみましょう
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

## 2. 逐次処理を極めよ！

### viii ブロックを避けてお宝を通りながらゴールに行こう



#### この課題で使えるとされている主な命令ブロック

スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路を考えます。途中、ブロック(壁)を避けて、すべてのお宝の上を通つて進みましょう
- 経路は複数あります。それぞれの経路でプログラムを作成してみましょう
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

## 2. 逐次処理を極めよ！

ix 間違い探し: ブロックを避けてゴールしよう



この課題で使えるとされている主な命令ブロック

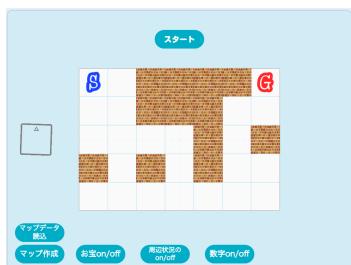
スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

この課題で学ぶこと、プログラム(コード)作成のヒント

- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」から始まり、ゴールの「G」まで行く経路をたどるプログラムがすでに作られています
- このプログラムは正しく動くか考えてみましょう。考えたら実際に動かしてみましょう。考えた通りに動きましたか
- 正しく動かない場合は、正しくゴールに行くように直しましょう

## 2. 逐次処理を極めよ！

x プログラムを読み解く:これはどんな動きをするかな？



この課題で使えるとされている主な命令ブロック

スタート位置に移動して [右] 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
右に90度回転	toioが向いている方向から右に90度回転します
左に90度回転	toioが向いている方向から左に90度回転します

この課題で学ぶこと、プログラム(コード)作成のヒント

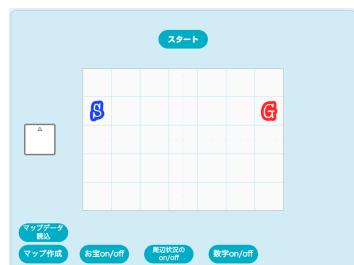
- 逐次処理の考え方を学びましょう。逐次処理は、命令ブロックを組み合わせた順番に命令を実行することを言います
- スタートの「S」から始まり、ゴールの「G」まで行く経路をたどるプログラムがすでに作られています
- このプログラムを読んでみて、toioはどの経路を通って動くか考えてみましょう
- 次に、プログラムを動かして、予想通り動いたか確認してみましょう
- 予想と違った場合は、もう一度プログラムを読んでみましょう。このプログラムには間違いは無いように作られています

# くりかえし きわめ 3. 繰り返しを極めよう！

ステージ1

【きほんそうさようそぎほう  
基本操作要素技法レベル】

## 3. 繰り返しを極めよう！ すすめ ひたすらまっすぐ進め



この課題で使えるとされている主な命令ブロック

スタート位置に移動して 右 向ける

前に一マス進む

3 回繰り返す

スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます

toioが向いている方向に一マス移動します

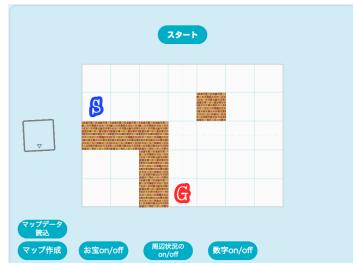
回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路を考えます
- SからGに行くには、同じ動作(命令ブロック)が何個必要かを考えてみましょう。必要な回数だけ繰り返しを行います
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

### 3. 繰り返しを極めよう！

まちがいさがし 間違ひ探し: ゴールに向かって進もう



この課題で使えるとされている主な命令ブロック

スタート位置に移動して 右 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
3 回繰り返す	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することです
- スタートの「S」からゴールの「G」までの経路をたどるプログラムを繰り返し処理で作られています
- このプログラムは正しく動くか考えてみましょう。考えたら実際に動かしてみましょう。考えた通りに動きましたか
- 正しく動かない場合は、正しくゴールに行くように直しましょう

### 3. 繰り返しを極めよう！

めいろ 迷路をゴールしよう



この課題で使えるとされている主な命令ブロック

スタート位置に移動して 右 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
3 回繰り返す	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

#### この課題で学ぶこと、プログラム(コード)作成のヒント

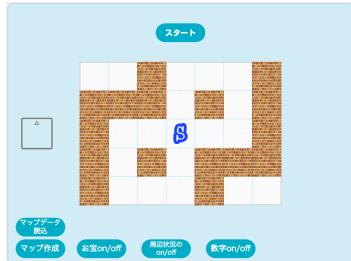
- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路を考えます
- SからGに行くには、同じ動作(命令ブロック)が何個必要かを考えてみましょう。必要な回数だけ繰り返しを行います
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

### 3. 繰り返しを極めよう！

#### iv 8の字を描くように進もう



この課題で使えるとされている主な命令ブロック



スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
3 回繰り返す	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することを言います
- スタートの「S」からスタートに戻るの8の字の経路を考えます
- SからGに行くには、同じ動作のまとまりが何組必要かを考えてみましょう。ブロックのまとまりを必要な回数だけ繰り返しさせます
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

### 3. 繰り返しを極めよう！

#### v プログラムを読み解く:これはどんな動きをするかな？



この課題で使えるとされている主な命令ブロック



スタート位置に移動して <b>右</b> に向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
3 回繰り返す	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行します
- スタートの「S」から始まり、スタートまで8の字を描きすすむ経路をたどるプログラムがすでに作られています
- このプログラムを読んでみて、toioはどの経路を通って動くか考えてみましょう
- 次に、プログラムを動かして、予想通り動いたか確認してみましょう。このプログラムには間違いは無いように作られています
- 繰り返し処理の中に繰り返し処理があります。外側の繰り返し処理は、中にある繰り返し処理を繰り返します。このような構造を「入れ子」と言います。

### 3. 繰り返しを極めよう！

vi お宝の上を全て通つてゴールに行こう その1



#### この課題で使えるとされている主な命令ブロック

スタート位置に移動して <b>右</b> 向ける	スタート位置「S」まで移動して、指定された方向(上下左右)にtoioを向けます
前に一マス進む	toioが向いている方向に一マス移動します
<b>3 回繰り返す</b>	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路上にあるお宝を通つて進むプログラムを考えます
- SからGに行くには、同じ動作(命令ブロック)が何個必要かを考えてみましょう。必要な回数だけ繰り返しを行います
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

### 3. 繰り返しを極めよう！

vii お宝の上を全て通つてゴールに行こう その2



#### この課題で使えるとされている主な命令ブロック

<b>右に90度回転</b> <b>左に90度回転</b>	toioが向いている方向から左か右に90度回転します
前に一マス進む	toioが向いている方向に一マス移動します
<b>3 回繰り返す</b>	回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

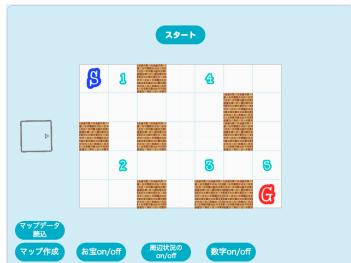
#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路上にあるお宝を通つて進むプログラムを考えます
- SからGに行くには、同じ動作(命令ブロック)が何個必要かを考えてみましょう。必要な回数だけ繰り返しを行います
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック



### 3. 繰り返しを極めよう！

#### viii ブロックを避けて数字を順番に通りながらゴールに行こう



この課題で使えるとされている主な命令ブロック

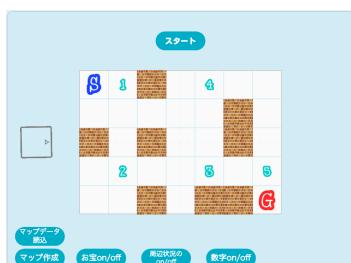
右に90度回転	左に90度回転	toioが向いている方向から左か右に90度回転します
前に一マス進む		toioが向いている方向に一マス移動します
3 回繰り返す		回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行することを言います
- スタートの「S」からゴールの「G」までの経路上にある数字の順番に通って進むプログラムを考えます
- 繰り返し処理を活用してSからGに行きましょう
- 必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マススの右ボタンをクリック

### 3. 繰り返しを極めよう！

#### ix プログラムを読み解く:これはどんな動きをするかな？



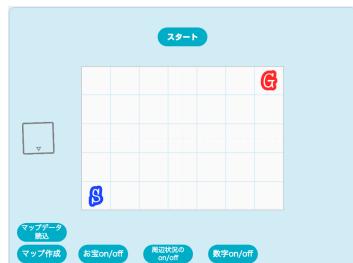
この課題で使えるとされている主な命令ブロック

右に90度回転	左に90度回転	toioが向いている方向から左か右に90度回転します
前に一マス進む		toioが向いている方向に一マス移動します
3 回繰り返す		回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 繰り返し処理の考え方を学びましょう。繰り返し処理は、命令ブロックを指定した回数や条件で繰り返し命令を実行します
- スタートの「S」からゴールの「G」までの経路上にある数字の順番に通って進むプログラムがすでに作られています
- このプログラムを読んでみて、toioはどの経路を通って動くか考えてみましょう
- 次に、プログラムを動かして、予想通り動いたか確認してみましょう。このプログラムには間違いは無いように作られています

### 3. 繰り返しを極めよう！ ×自由に動きを作つてみよう！



この課題で使えるとされている主な命令ブロック

右に90度回転	左に90度回転	toioが向いている方向から左か右に90度回転します
前に一マス進む		toioが向いている方向に一マス移動します
3 回繰り返す		かいすう してい くりかえす せいやよ めいれい すうじ 回数を指定して繰り返す(制御)命令ブロックです。数字を変えると、ワニの口のような場所に入れた命令ブロックを、その数だけ繰り返しを行います

#### この課題で学ぶこと、プログラム(コード)作成のヒント

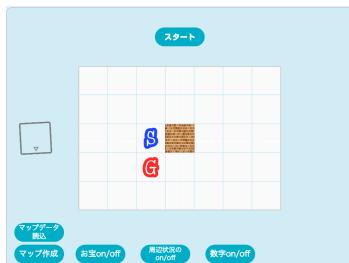
- スタートから初めて、ゴールに進むプログラムを作ろう。
- 数字やお宝のアイテムはマップにありません。
- 動きの命令(一つ前に進む、右に90度回転、左に90度回転)と繰り返し命令だけを使ってうごかしましょう。
- 使うことができるブロックは表示されているものだけです。必要な命令は複製をして使います
  - [Macの場合] Controlキーを押しながらマウスボタンをクリック
  - [Windowsの場合] マウスの右ボタンをクリック

## 4. 条件分岐を極めよう！

### ステージ1

【基本操作要素技法レベル】

## じょうけん ぶんき きわめ 4. 条件分岐を極めよう！ めまえかべ i 目の前の壁をよけよう



この課題で使えるとされている主な命令ブロック

正面のアイテム	toioの正面にあるアイテムを格納している変数です。ブロック、お宝、外などの文字が格納されます
= ブロック	条件を判断するブロックです。=の左右に指定された値が同じであるかを判断します
もし なら	条件分岐の命令ブロックです。「もし」の後に条件をいれます。その条件が正しいとワニの口の中にある命令ブロックを実行します

この課題で学ぶこと、プログラム(コード)作成のヒント

- 条件分岐の考え方を学びましょう。条件分岐は、条件によって動かす命令ブロックを変更する場合に使います
- スタートの「S」位置でtoioは右に向いています。正面にブロック(壁)があったら下に向きを変えて進みます。ブロックがなければ、向きを変えずにそのまま進みます。ここでは、条件によって向きをかえるかどうかを決定するために条件分岐を使います。
- 条件分岐の基本的な書き方を覚えましょう
- 必要な命令は複製をして使います

## じょうけん ぶんき きわめ 4. 条件分岐を極めよう！ めまえかべさけ ii 目の前の壁を避けてゴールへ



この課題で使えるとされている主な命令ブロック

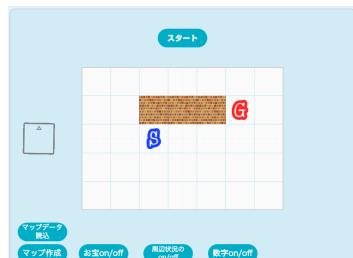
正面のアイテム	toioの正面にあるアイテムを格納している変数です。ブロック、お宝、外などの文字が格納されます
= ブロック	条件を判断するブロックです。=の左右に指定された値が同じであるかを判断します
もし なら	条件分岐の命令ブロックです。「もし」の後に条件をいれます。その条件が正しいとワニの口の中にある命令ブロックを実行します

この課題で学ぶこと、プログラム(コード)作成のヒント

- 条件分岐の考え方を学びましょう。条件分岐は、条件によって動かす命令ブロックを変更する場合に使います
- スタートの「S」位置でtoioは右に向いています。条件分岐処理と繰り返し処理の命令を使って、目の前に壁(ブロック)があった場合、下に一マス進むようにプログラミングしよう
- 必要な命令は複製をして使います

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！

### iii コードを読み解く：どのような動作をするかな



#### この課題で使えるとされている主な命令ブロック

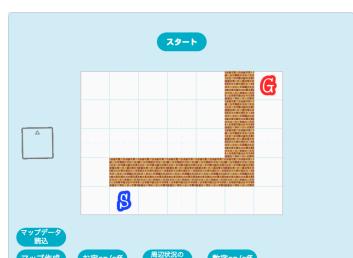
周辺状況 (前後左右) の 4 番目	Toioの周辺にあるアイテムが格納されている変数(リスト) です。ブロック、お宝、外などの文字が格納されます 2番目が前、4番目が左、6番目が右、8番目が後ろです
= ブロック	条件を判断するブロックです。=の左右に指定された値が 同じであるかを判断します
もし なら でなければ	条件分岐の命令(制御)ブロックです。「もし、○○なら、■■ を行い、そうでなければ△△を行う」ことを意味しています。 ○○は「もし」の後の枠に入れます。■■は上の口の中、△△ は「でなければ」の後の口の中に入れます

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 条件分岐の考え方を学びましょう。条件分岐は、条件によって動かす命令ブロックを変更する場合に使います
- まずはコードを読んで動きを想像してみよう。どのように動くか、想像通りに動いたかな？
- 周辺 状況 (前後左右)の4番目 というブロックは、toioの進行方向左側に何があるか(ブロック、お宝、数など)がわかる命令です

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！

### iv 壁(ブロック)を左側にみながらゴールへ



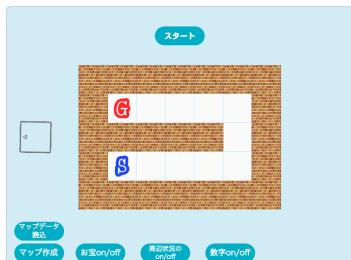
#### この課題で使えるとされている主な命令ブロック

周辺状況 (前後左右) の 4 番目	Toioの周辺にあるアイテムが格納されている変数(リスト) です。ブロック、お宝、外などの文字が格納されます 2番目が前、4番目が左、6番目が右、8番目が後ろです
= ブロック	条件を判断するブロックです。=の左右に指定された値が 同じであるかを判断します
もし なら でなければ	条件分岐の命令(制御)ブロックです。「もし、○○なら、■■ を行い、そうでなければ△△を行う」ことを意味しています。 ○○は「もし」の後の枠に入れます。■■は上の口の中、△△ は「でなければ」の後の口の中に入れます

#### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいかを考えよう
- たとえば、以下のように場合をいくつか分けて考えます
  - 左側に壁がないとき、左に90度回転して、一マス進めます
  - 左側に壁があるとき、そのまま一マス進めます
- 周辺 状況 (前後左右)の4番目 というブロックは、toioの進行方向左側に何があるか(ブロック、お宝、数など)がわかる命令です

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！ ▼ コの字の経路に沿って進もう



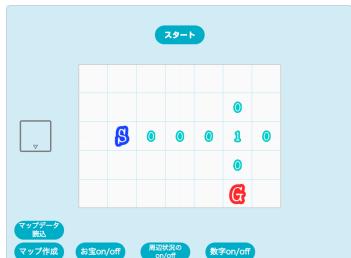
### この課題で使えるとされている主な命令ブロック

	<p>Toioの周辺にあるアイテムが格納されている変数(リスト)です。ブロック、お宝、外などの文字が格納されます</p> <p><b>= ブロック</b></p> <p>条件を判断するブロックです。<math>=</math>の左右に指定された値が同じであるかを判断します</p>
	<p>条件分岐の命令(制御)ブロックです。「もし、○○なら、■■」を行い、そうでなければ△△を行うことを意味しています。○○は「もし」の後の枠に入れます。■■は上の口の中、△△は「でなければ」の後の口の中に入れます</p> <p><b>もし...なら</b></p> <p><b>でなければ</b></p>

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいかを考えよう
- たとえば、以下のように場合をいくつか分けて考えます
  - 正面に壁がないとき、そのまま一マス進めます
  - 正面に壁があって、左側に壁がないとき、左に90度回転して、一マス進めます
- 周辺状況(前後左右)の4番目というブロックは、toioの進行方向左側に何があるか(ブロック、お宝、数など)がわかる命令です

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！ ▼ 真下の数字によって行動を決めよう



### この課題で使えるとされている主な命令ブロック

	<p>Toioの周辺にあるアイテムが格納されている変数(リスト)です。ブロック、お宝、外などの文字が格納されます</p> <p><b>= ブロック</b></p> <p>条件を判断するブロックです。<math>=</math>の左右に指定された値が同じであるかを判断します</p>
	<p>条件分岐の命令(制御)ブロックです。「もし、○○なら、■■」を行い、そうでなければ△△を行うことを意味しています。○○は「もし」の後の枠に入れます。■■は上の口の中、△△は「でなければ」の後の口の中に入れます</p> <p><b>もし...なら</b></p> <p><b>でなければ</b></p>

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいか、条件分岐を使った方法を考えよう。
- たどりつくじどうまとどうさしうりょう ゴールにたどり着くと自動的に動作が終了します。
- たとえば、以下のように場合をいくつか分けて考えます。どのような条件でゴールにたどり着くか考えてみましょう
  - 真下の数字が「1」のとき、右に90度回転して、一マス進めます
  - そうでないとき、そのまま一マス進めます

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！

まちがいさがし むかつすすもう  
vii 間違い探し: ゴールに向かって進もう



### この課題で使えるとされている主な命令ブロック

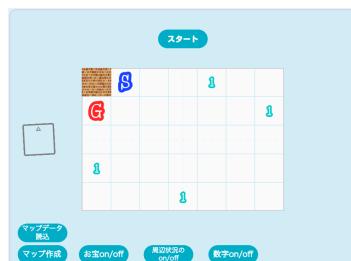
真下のアイテム	正面のアイテム	toioの真下あるいは正面にあるアイテムを格納する変数です。プロック、お宝などの文字と数字が格納されます
= ブロック		条件を判断するブロックです。=の左右に指定された値が同じであるかを判断します
もし	なら	条件分岐の命令(制御)ブロックです。「もし、○○なら、■■を行い、そうでなければ△△を行う」ことを意味しています。○○は「もし」の後の枠に入れます。■■は上の口の中、△△は「でなければ」の後の口の中に入れます

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいか、条件分岐を使った方法を考えよう。ゴールにたどり着くと自動的に動作が終了します。
- スタートの「S」からゴールの「G」までの経路をたどるプログラムを繰り返し処理と条件分岐処理で作られています
- このプログラムは正しく動くか考えてみましょう。考えたら実際に動かしてみましょう。考えた通りに動きましたか
- 正しく動かない場合は、正しくゴールに行くように直しましょう

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！

きのう ついか ました すうじ じ おん  
viii 【機能を追加】真下の数字が1の時にビープ音をならそう



### この課題で使えるとされている主な命令ブロック

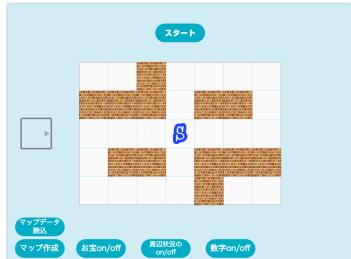
音 67 を 1 秒鳴らす	Toioがビープ音を1秒鳴らします。音を変更する場合は「音」の後の数値、長さは「秒」の前の数値で変えることができます	
もし	なら	条件分岐の命令(制御)ブロックです。「もし、○○なら、■■を行い、そうでなければ△△を行う」ことを意味しています。○○は「もし」の後の枠に入れます。■■は上の口の中、△△は「でなければ」の後の口の中に入れます

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいか、条件分岐を使った方法を考えよう。ゴールにたどり着くと自動的に動作が終了します。
- すでにtoioがマットの外枠に沿って移動するプログラムが作られています。ここに真下の数字が1の場合にビープ音を鳴らす命令を組み込んでみましょう

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！

### ix【完成させよう】8の字を永遠に回る



かだい つか この課題で使えるとされている主な命令ブロック

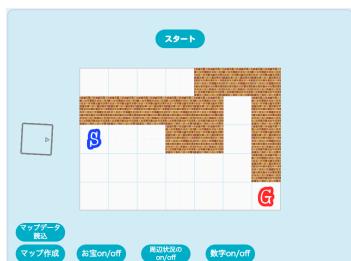
周辺状況（前後左右）▼ の 4 番目 = 空	ひだりがわ へんすう しゅうへんじょうきょう ぜんごさゆう ばんめ Toioの左側(リスト変数 周辺 状況(前後左右)の4番目) のマスが「空」(何もない)ならば条件が成立する
周辺状況（前後左右）▼ の 6 番目 = 空	みぎがわ へんすう しゅうへんじょうきょう ぜんごさゆう ばんめ toioの右側(リスト変数 周辺 状況(前後左右)の6番目) のマスが「空」(何もない)ならば条件が成立する

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいか、条件分岐を使った方法を考えよう。  
たどりつく じどうまと どうさ しゅうりょう ゴールにたどり着くと自動的に動作が終了します。
- スタートの「S」の位置から右側にす進み8の字の経路を永遠とたどるプログラムを繰り返し処理と条件分岐処理で作ります。
- すでに正面のマスが「ブロック」「外」でなければ一マス進むプログラムが作られています。これを改良してプログラムを完成させよう。

## じょうけんぶんき きわめ 4. 条件分岐を極めよう！

### x【コードを読み解く】どのような動作をするかな



かだい つか この課題で使えるとされている主な命令ブロック

周辺状況（前後左右）▼ の 4 番目 = 空	ひだりがわ へんすう しゅうへんじょうきょう ぜんごさゆう ばんめ Toioの左側(リスト変数 周辺 状況(前後左右)の4番目) のマスが「空」(何もない)ならば条件が成立する
周辺状況（前後左右）▼ の 6 番目 = 空	みぎがわ へんすう しゅうへんじょうきょう ぜんごさゆう ばんめ toioの右側(リスト変数 周辺 状況(前後左右)の6番目) のマスが「空」(何もない)ならば条件が成立する

### この課題で学ぶこと、プログラム(コード)作成のヒント

- 回数を指定した繰り返しを使わないで、どのような場面でtoioをどのように動かせばよいか、条件分岐を使った方法を考えよう。  
たどりつく じどうまと どうさ しゅうりょう ゴールにたどり着くと自動的に動作が終了します。
- スタートの「S」の位置からゴール「G」にたどり着くプログラムを繰り返し処理と条件分岐処理で作成しています。
- どのようなアルゴリズム(手順)でゴールまでたどり着くか考えましょう。

# きほんそうさようそぎほう 5. 基本操作要素技法レベル そうごうかだい 総合課題

ステージ1

【基本操作要素技法レベル】

きほんそうさようそぎほう 4. 基本操作要素技法レベル 総合課題 そうごうかだい  
この課題で使えるとされている主な命令ブロック  
新しく使う命令はありません



この課題で学ぶこと、プログラム(コード)作成のヒント

- 基本操作要素技法レベルで学んだことを生かして迷路を解くプログラムを作成してみよう
- スタートの「S」の位置からゴール「G」にたどり着くプログラムを作ろう。考え方のヒントは以下の通りです
  - 左側にブロックや外となっていて、正面が空の場合は、一マス進む
  - 左側にブロックや外となっていて、正面もブロックや外となっている場合は、その場で右に90度回転
  - 左側が空の場合は、左に90度回転して、一マス進む

この課題で学ぶこと、プログラム(コード)作成のヒント

- 基本操作要素技法レベルで学んだことを生かして迷路を解くプログラムを作成してみよう
- スタートの「S」の位置からゴール「G」にたどり着くプログラムを作ろう。考え方のヒントは以下の通りです
  - 左側にブロックや外となっていて、正面が空の場合は、一マス進む
  - 左側にブロックや外となっていて、正面もブロックや外となっている場合は、その場で右に90度回転
  - 左側が空の場合は、左に90度回転して、一マス進む