

Mission1 プログラムをつくってみよう！

【1】プログラミングを始めよう



- **プログラム**とは、コンピュータがうごくための手順をかきあらわしたものです。
- プログラムをつくるためには、目的に合う **プログラミング言語**を使います。
- プログラムでは、コンピュータの動作 (**命令**といいます) を、1つずつ**入力**していきます。
- プログラムをかいたら、**実行**します。
- 正しく動作しなかったら、プログラムの誤りを見つけて**デバッグ**します。

手順1 コンピュータを起動し、ブラウザで次のサイトを開こう。

<https://manabu-tech.net/nakopad/>

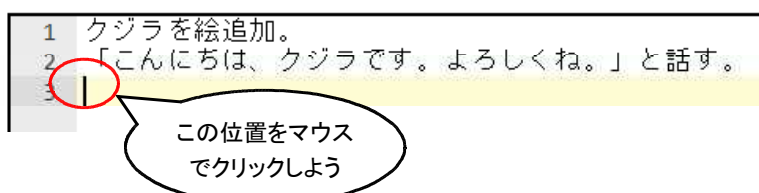
手順2 まず、プログラムを実行してみよう！



★プログラミングの流れを知ろう！ ①プログラムを入力 → ②実行 → ③実行結果の表示

手順3 プログラムを消してみよう。

(1) **カーソル**を、プログラムの末尾に移動させよう。



カーソルキー



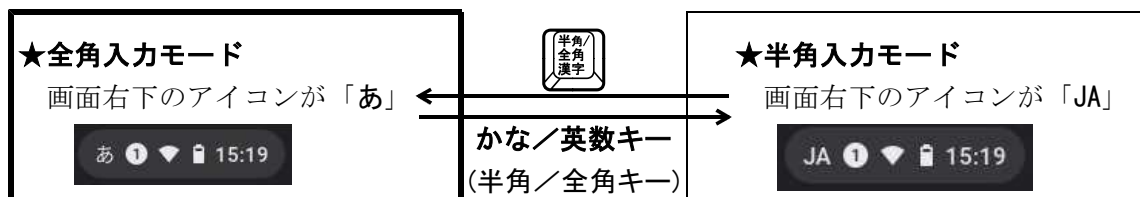
を使って移動させることもできるよ！

(2) **バックスペースキー** (エンターキーの上にある  または )を押して、文字を消してみよう。

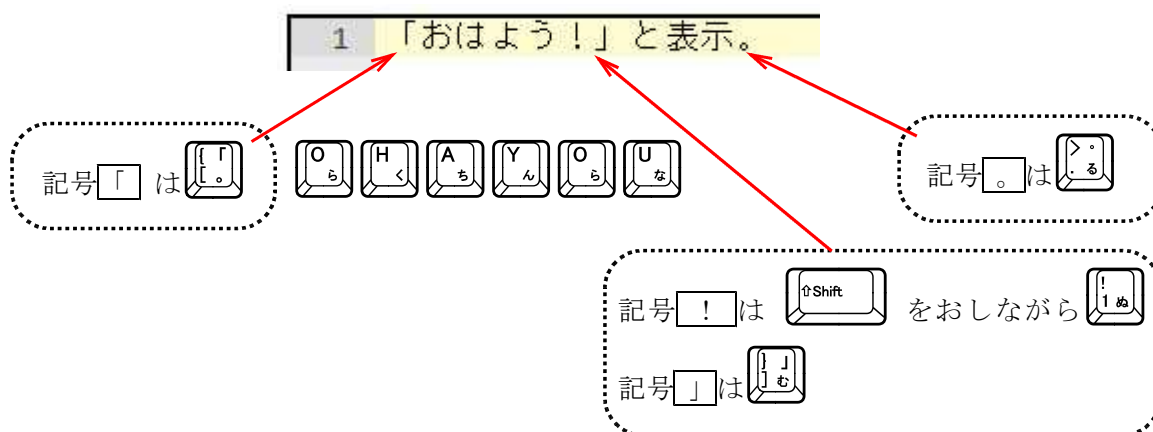


手順4 自分プログラムを入力してみよう！

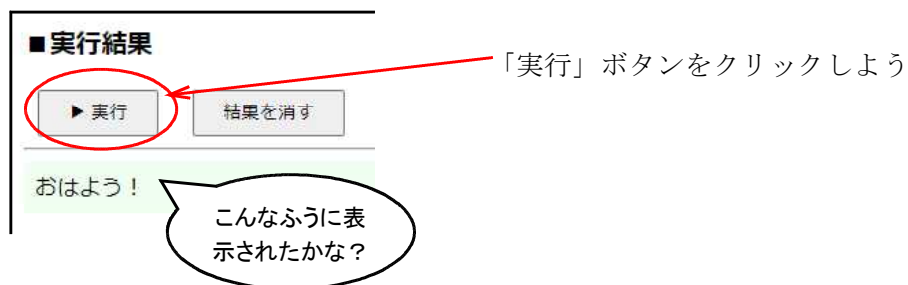
(1) 全角入力モードに切り換えよう。



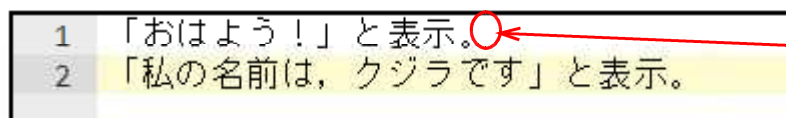
(2) プログラムを入力してみよう。



(3) 入力できた人は実行してみよう！



手順5 うまくできた人は、プログラムを増やして、実行してみよう。



ここでエンターキー
押して改行しよう。

【2】様々な命令を使ってみよう！



- 「…」と表示 「…」の中の文章を画面に表示する命令です。
- 「…」と言う 「…」の中の文章を、ダイアログで表示する命令です。
- (計算式)と表示 「…」を使わずに計算式を書くと、計算した結果(答え)を表示します。
- {…} 中カッコの中に計算式を書くと、計算した結果(答え)を表示します。

手順1 画面の **結果を消す** ボタンをクリックしよう。

手順2 次のようにプログラムを**修正**して、実行しよう。どんな結果になるかな？

```
1 「おはよう！」と言う。
2
```

手順3 次のようにプログラムを**入力・修正**して、実行しよう。どんな結果になるかな？

①

```
1 1 + 2 を表示。
2
```

②

```
1 1 + 2 - 3 * 4 / 5 を表示。
2
```

記号 **[]** は +

記号 **[+]** は +

記号 **[-]** は

記号 **[*]** は +

記号 **[/]** は → で変換して
(全)スラッシュを選ぶ

③

```
1 「答えは { 1 + 2 - 3 * 4 / 5 } 」と表示。
2
```

{ } (中カッコ)

- カギカッコ「…」の中の文字は、そのまま表示される。
- 中カッコ {…} の中の計算式は、その結果が表示される。

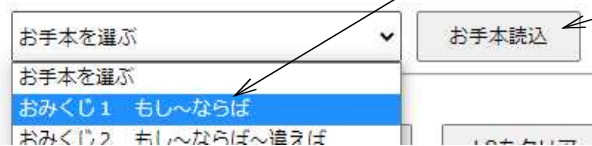
④ 自分でプログラムをかきかえて、様々な計算をさせてみよう！

【3】おみくじをつくってみよう！



- 人が入力した情報や、コンピュータが計算した結果などを、**変数**に入れて記憶させます。
- プログラムが、書かれた順に実行される処理を**順次**といいます。
- 条件に応じて、処理を変える処理を**分岐**といいます。

手順1 画面下の「お手本を選ぶ」から「おみくじ1」を選んで、**お手本読込**ボタンをクリックしよう。



手順2 読み込んだプログラムを実行してみよう。

(プログラム)

```

1 おみくじ = (6の乱数) + 1。
2 おみくじを表示。
3
4 もし、おみくじ = 1ならば
5   「大吉」と表示。
6   ここまで。
7
8 もし、おみくじ = 2ならば
9   「凶」と表示。
10  ここまで。

```

(実行結果)

1 ← 変数 **おみくじ** の値
 大吉 ← **もし、～ならば** の結果

★プログラムを読み解いて、次のことを理解しよう！

- **変数** には、名前をつけて、値や文字を代入することができる。
- (6の乱数+1) とすると、1～6までのどれか1つの整数を選んでくれる。
- もし、(条件式) ならば

空白2文字 条件に当てはまるときの処理
 ここまで。

}

「分岐」の書き方を覚えよう。

手順3 プログラムを改良して、**おみくじ** の値が3, 4, 5, 6の時の処理を加えてみよう。



プログラムを入力したら、**実行**して動作を確かめてみよう。

手順4 <チャレンジ！>

プログラムを改良して、**おみくじ** の値が7以上の処理を加えてみよう。

【3】＜深めよう＞ 変数の使い方を理解しよう



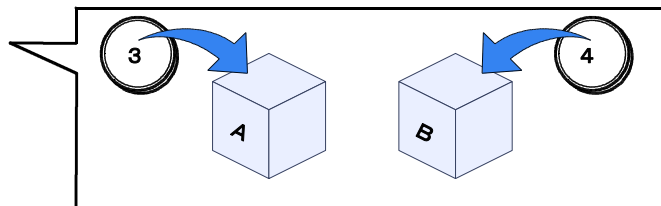
- **変数**を使って、値を計算することができます。
- これまでに学習した**命令**や**変数**を上手に組み合わせると、目的に合うプログラムを作ることができます。

手順1 まず、次のプログラムを実行してみよう。

```
3 + 4 を表示。
```

手順2 これを、次のように修正して、実行しよう。

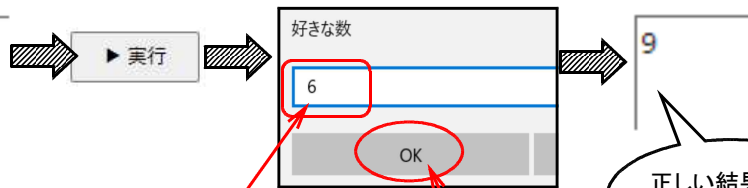
```
A = 3。
B = 4。
A + B を表示。
```



Aという名前の変数（値を入れるはこ）には、3を入れる。
Bという名前の変数（値を入れるはこ）には、4を入れる。

手順3 さらに、次のように修正して、実行しよう。

```
A = 3。
B = 「好きな数」と尋ねる。
A + B を表示。
```



好きな値を入力して、OKボタンを押そう。

正しい結果が表示されたかな？

手順4 考えよう！ ①プログラムの2行目 **「好きな数」と尋ねる** は、どんな動作をしたかな？
② **B = 「好きな数」と尋ねる** では、変数Bに何を入れたかな？

答え：①入力ダイアログを表示して、値を入力させる。 ②変数Bには、自分が入力した値を入れる。

手順5 このプログラムを修正して、好きな2つの数を入力して、その積（かけ算の結果）を表示するプログラムを考えてつくろう！

（プログラムの例）

```
1 A = 「1つめの数」と尋ねる。
2 B = 「2つめの数」と尋ねる。
3 「A * B = {A * B} 」と表示。
4
```

（実行した結果の例）

A * B = 15

Mission2 数当てゲームをつくってみよう！

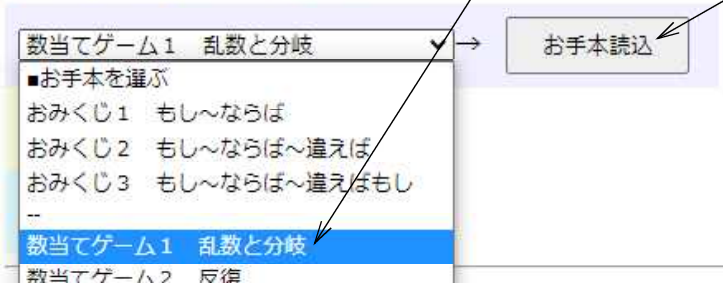


- **変数**を使って、値を計算することができます。
- これまでに学習した**命令**や**変数**を組み合わせて、目的に合うプログラムを作りましょう。
- プログラムの順に実行される部分を、**順次処理**といいます。
- 条件によって動作が変わる部分を、**分岐処理**といいます。

【1】分岐 もし～ならば

手順1 画面下の「お手本を選ぶ」から「数当てゲーム1」を選んで、**お手本読込**ボタンをクリックしよう。

■ プログラムを読み込むときは…



手順2 読み込んだプログラムを実行してみよう。

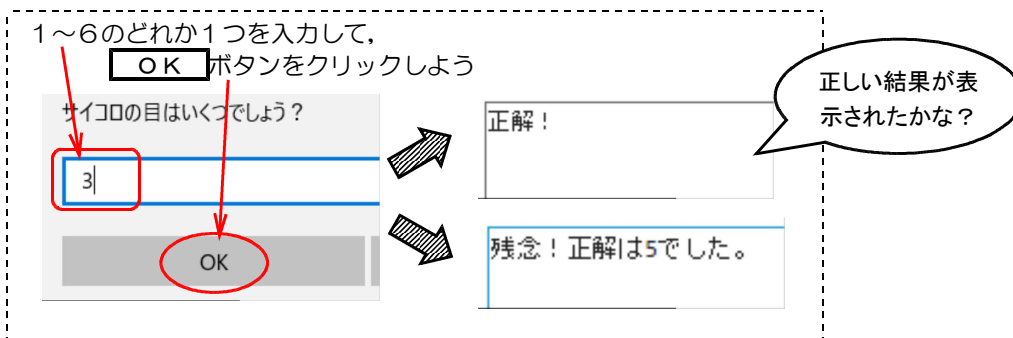
(プログラム)

```

1 サイコロ = (6の乱数) + 1。
2 答え = 「サイコロの数を当ててみよう」と尋ねる。
3
4 もし、答え = サイコロならば
5   「正解！」と表示。
6 違えば
7   「残念。正解は{サイコロ}でした。」と表示。
8 ここまで。

```

(実行結果の例)



★ プログラムを読み解いて、次の書き方を理解しよう！

- **もし、(条件式) ならば**
(条件式にあてはまるときの処理)
- 違えば**
(あてはまらないときの処理)
- ここまで。**

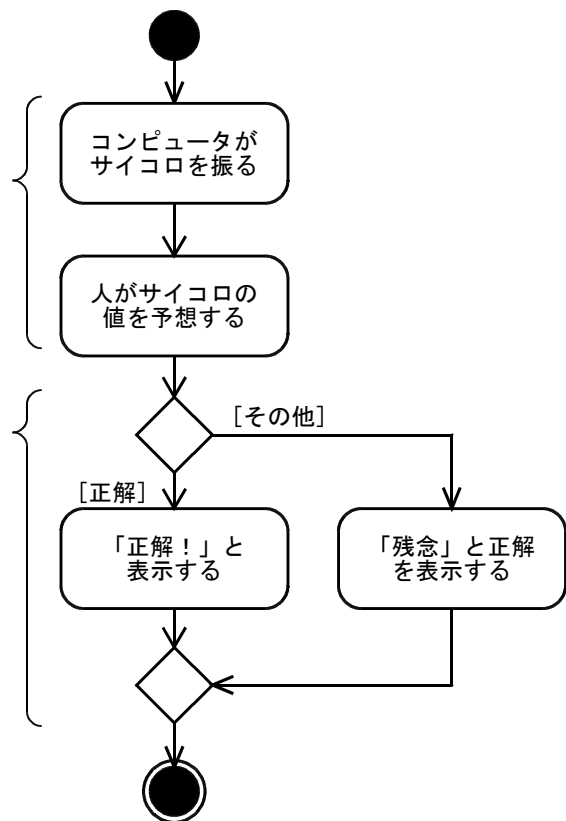
手順3 プログラムの動作を、説明してみよう。

1 行目	サイコロ = (6 の乱数) + 1。 ① 6 の乱数は、0 ~ 5 のどれか 1 つの値をコンピュータが決める。	サイコロ という変数に ① ~ のどれか 1 つの 値を入れる。
2 行目	答え = 「サイコロの数を当ててみよう」と尋ねる。 ③ ②	② 尋ねる という命令で 入力ダイアログが表示される。
3 行目	(改行だけの行は、何もしない)	あなたの予想を入力すると、 その値が ③ という 変数に入る。
4 行目	もし、答え = サイコロ ならば ④	④ 答え という変数の値と ④ という変数の値が
5 行目	「正解！」と表示。 ⑤	同じだったら ⑤ と 表示する。
6 行目	違えば	違う値だったら
7 行目	「残念。正解は「サイコロ」でした。」と表示。	⑥
8 行目	ここまで。 ⑥	と表示する。

答え：①1～6 ②入力ダイアログ ③答え ④サイコロ ⑤正解！ ⑥残念。正解は*でした。

★覚えよう！

- このプログラムの処理と処理の順番を、
アクティビティ図(右のような図)で
かき表すことができます。
- 1～2行目のように、プログラムの順番に
実行される処理を、**順次処理**といいます。
- 4～8行目のように、条件によって結果が
異なる処理を、**分岐処理**といいます。



手順4 このプログラムを修正して、コンピュータが決めた**2つの数の合計を予想して当てるゲーム**のプログラムを考えてつくろう！

【2】反復 ○回～ここまで

手順1 画面下の「お手本を選ぶ」から「数当てゲーム2」を選んで、**お手本読込**ボタンをクリックしよう。

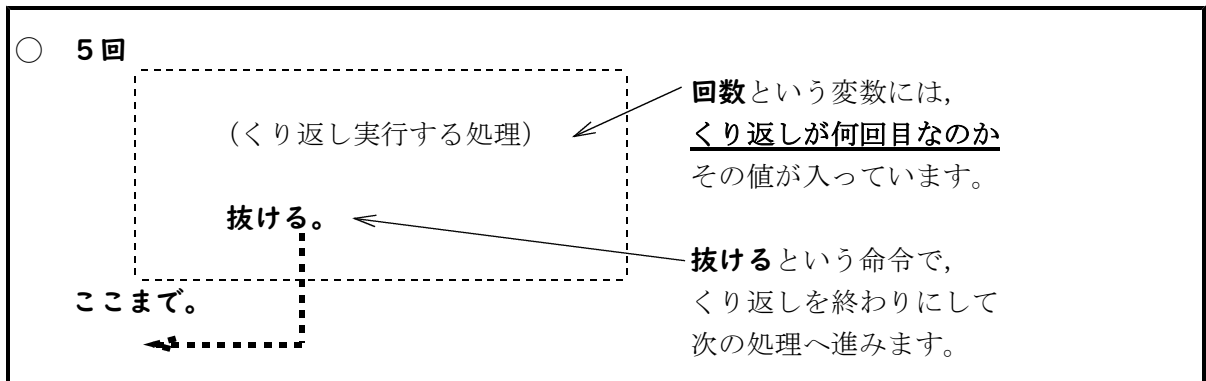
手順2 読み込んだプログラムを実行してみよう。

```

1 サイコロ = (6の乱数) + 1。
2
3 5回
4     答え = 「{回数} 回目 サイコロの目はいくつ?」と尋ねる。
5
6     もし、答え = サイコロならば
7         「正解!」と表示。
8         抜ける。
9     違えば
10        「残念。」と表示。
11        ここまで。
12    ここまで。
13    「正解は {サイコロ} でした。」と表示。
14

```

★プログラムを読み解いて、次の書き方を理解しよう！



★なぜ、8行目に **抜ける** という命令を入れてあるのか、考えてみよう。

★Aくんは、13行目にある「**正解は {サイコロ} でした。**」と表示。を、7行目と8行目の間にに入れてもよいと考えました。そうすると、動作はどのように変化するかな。考えてみよう。

手順3 このプログラムを、次のように改良してみよう。

Bさん「このプログラムのままだと、正解しなかったときに『残念。』しか出てこないから、少しヒントがほしいよね」
 Cさん「そうだね。じゃあ**もし～ならば～違えば～ここまで**を使って、ヒントを加えられな
 いかなあ」
 Bさん「うん。正解しなかったときに『もっと大きいよ』『もっと小さいよ』って表示され
と、正解しやすくなるよね!」

★読み込んだプログラムを改良して、正解しなかったときに『もっと大きいよ』『もっと小さいよ』
 と表示する処理を加えてみましょう。

【補足】プログラムを保存しよう

先生へ提出したいときは…	まだ提出しないけど保存したいときは…
<p>■プログラムを保存するときは…</p> <p>ファイル名 <input type="text"/> .txt → <input type="button" value="▲このファイル名で保存する"/></p>	<p><input type="button" value="△LSに一時保存する"/> <input type="button" value="◇LSのプログラムを消去"/></p>
(1) ファイルとして、フォルダに保存する	(2) 自分のコンピュータの内部に、一時保存する
<p>手順1 ファイル名を入力する</p> <p><input type="text" value="ファイル名"/> .txt</p> <p>手順2 <input type="button" value="▲このファイル名で保存する"/> をクリック</p> <p>手順3 Windows や iOS の場合は、あなたのコンピュータの「ダウンロード」フォルダにファイルが保存されています。 ↓ そのファイルを、自分が保存したいフォルダへ移動させましょう。</p> <p>手順4 Chromebook の場合は、保存するフォルダを選んで保存します。</p>	<p>手順1 このボタンをクリックする</p> <p>手順2 『保存していいですか?』と聞かれたら <input type="button" value="OK"/> を押す。</p> <p>※こちらの方法では、プログラムを1つだけ保存することができます。</p> <p>手順2 で <input type="button" value="OK"/> を押すと、以前にこの方法で保存したプログラムは消えてしまいます。</p> <p>※保存したプログラムを消去したいときは <input type="button" value="◇LSのプログラムを消去"/> ボタンをクリックします。</p>

次の授業で、保存したファイルを開くときは…	次の授業で、保存したプログラムを開くときは…
<p><input type="button" value="▼保存したファイルを読み込む"/></p>	<p><input type="button" value="▽LSから読み込む"/></p>
<p>手順1 このボタンをクリックする</p> <p>手順2 保存したファイルを選んで、<input type="button" value="開く"/> を押す。</p> <p>手順3 『読み込んでいいですか?』と聞かれたら <input type="button" value="OK"/> を押す。</p>	<p>手順1 このボタンをクリックする</p> <p>手順2 『読み込んでいいですか?』と聞かれたら <input type="button" value="OK"/> を押す。</p>

【3】メディア 音を鳴らしてみよう！

手順1 次のプログラムを入力して、実行してみよう

```
1 正解音=ピンポンを音追加。
2 正解音を音再生。
3
```

音が鳴った
かな？

学習用なでしこパッド（このサイト）では、あらかじめピンポンとブブーという音のファイルが用意されています。

```
1 残念音=ブブーを音追加。
2 残念音を音再生。
3
```

★プログラムを読み解いて、次の書き方を理解しよう！

- (変数) = (音のファイル) を音追加。 …音のファイルを読み込み、鳴らす準備をします。
- (変数) を音再生。 …読み込んだ音のファイルを再生します。

手順2 <課題> 正解ならピンポンと、不正解ならブブーと、音を鳴らしてみよう！

(1) 保存したプログラムを読み込もう。

※保存していない人は、画面下の「お手本を選ぶ」から「数当てゲーム3」を選んで、**お手本読込**ボタンをクリックしよう。

(プログラムの例)

```
1 サイコロ = (6の乱数) + 1。
2
3 5回
4   答え = 「{回数} 回目 サイコロの目はいくつ？」と尋ねる。
5
6   もし、答え = サイコロならば
7       「正解！」と表示。
8       抜ける。
9   違えば
10      もし、答え > サイコロならば
11          「もっと小さいよ」と言う。
12      違えば
13          「もっと大きいよ」と言う。
14      ここまで。
15  ここまで。
16  ここまで。
17  「正解は {サイコロ} でした。」と表示。
18
```

(2) **手順1**を参考にして、音を鳴らすプログラムを追加してみよう！

手順3 <チャレンジ！> さらに改良してみよう

★ヒントや正解を、画面に表示するとともに、音声で話すようにできるかな？



「～」を声出す。 という命令を使うよ！