

Jasmine Tea 基礎-プログラミングをもっと知ろう（生徒用）

1. 作品を共有してみよう

Jasmine Tea のサイトでは、自分で作った作品を共有することができます。

ただ作ること自体が好きで作る人もいるとおもいますが、今回は自分で作成したゲームをみんなが遊べる状態にしてフィードバック（意見）をもらいながら改善をしていきましょう。

情報を共有するときのルール

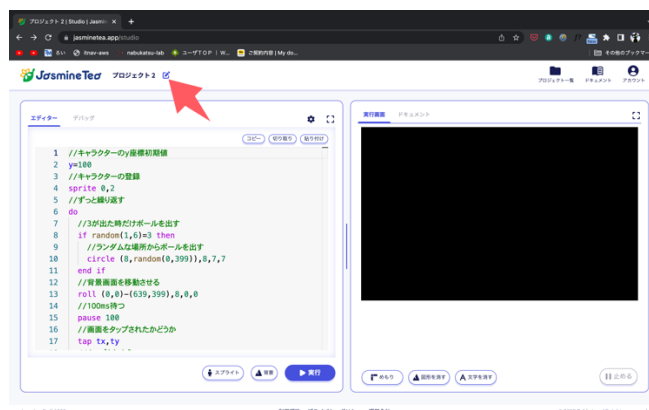
作品を共有するときにいくつかルールがあるのでまずは下記の情報をよく確認してください。

1. 既に存在しているゲームや、アニメ・ゲームなどのキャラクター、著作権のある音楽を模倣した素材が含まれる作品を共有しないようにしてください。
2. 名前、住所、電話番号などの個人情報は載せないようにしましょう。
 - 特にプロジェクト名などは学校によって学籍番号や氏名にする場合もあるので注意してください。
3. 人を傷つけるような表現などを使っていないか、よく確認してください。
 - 差別を助長するようなものも厳禁です。

共有用 URL の設定・共有

共有ルールを確認でき、すべて問題なければ実際にプロジェクトを共有していきましょう。

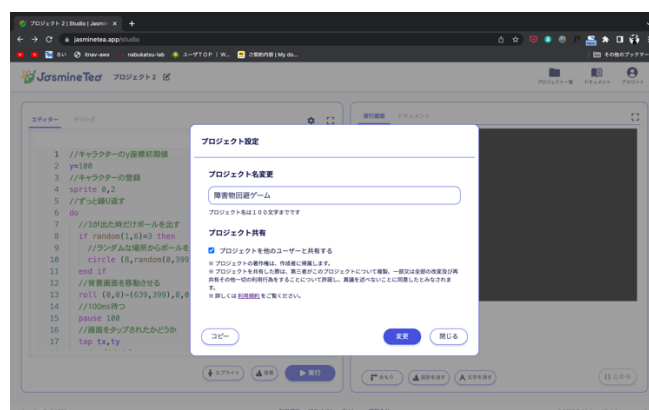
作品を共有するには Jasmine Tea エディタの左上の「プロジェクト設定」を開きます。



プロジェクト設定画面を開いたら、プロジェクト名を任意の名前に変更してください。

その下に「プロジェクト共有」の項目があるので、「プロジェクトを他のユーザーと共有する」にチェックをつけます。

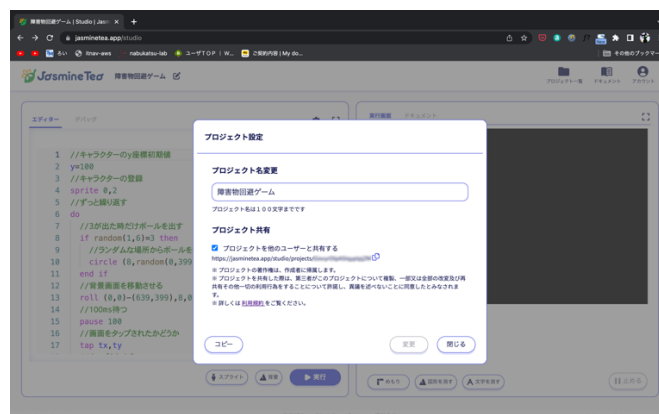
プロジェクト共有に関する注意事項を確認し、「変更」をクリックしましょう。



そうすると、下の画像のようにプロジェクトを共有するための URL が表示されます。

こちらの URL を共有すると、共有された人のパソコンやタブレット上であなたが作ったゲームを実際に遊ぶことができるようになります。

ぜひこの URL をコピーして、遊んで欲しい人に共有しましょう。



他の人の作品を遊んでみよう

同じグループ、またはクラスメイトが作成したゲームを実際に遊んでみましょう！

ただ遊ぶだけでなく、下の項目のような点に注目しながら自分と違う点などを確認していきましょう。

- 今回の Jasmine Tea 教材から変化（カスタマイズ）している部分はどこか
- 特に工夫されていて、面白いなと感じるポイントはどこか
- ゲームの難易度はちょうどいいか。
- こういう機能があったらもっと面白いなと感じた部分はあるか
- 自分のゲームに取り入れたら面白そうな部分はあるか

上記は一例で、他にもいろんな視点で他の人の作品を見てください。

フィードバック（グループワーク）

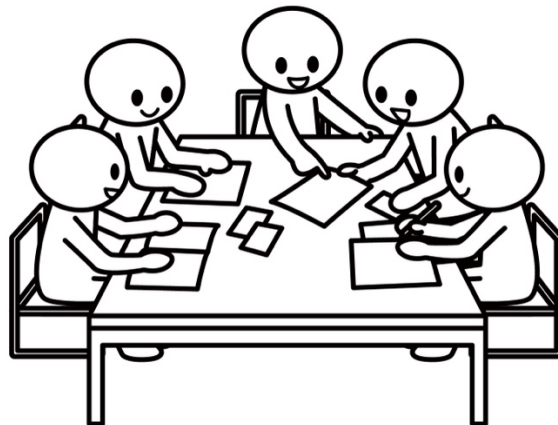
情報を共有しよう

いろんな生徒の作品を見た後は、実際に自分の感想を相手に伝えましょう。

「良かったところ」「もっとこうすればより面白くなるかも」といったアドバイスを送り合ってください。

注意点としては

単純な悪口や誹謗中傷などを行わないことです。



自分の作品に対するアドバイスはしっかりと聞きましょう。

また、その中から良いアイデアや意見はしっかりとメモを取っておいてください。

おそらく自分が想定していた難易度と、他のユーザーがプレイして感じた難易度には大きな開きがあるかと思います。

それは単純に開発する人は何度もそのゲームをプレイしながら難易度調整しているので、プレイヤースキルがついた状態なのに対し、初めて遊ぶ人は一切プレイヤースキルがない状態からスタートするからです。

ですので、実際プレイしたことのない人、もしくはあまり詳しくない人などに遊んでもらって感想を聞くことが重要になってきます。

こちらのフィードバックをもとに自身のゲームをカスタマイズしていきますので、きちんとメモを取っておいてください。

2. ユーザーを意識してみよう

レベルデザインとは

レベルデザインとは、ゲームの難易度に関わる要素を設計していくことです。

- どこに障害物を置いたらプレイヤーは手応えを感じてくれるのか？

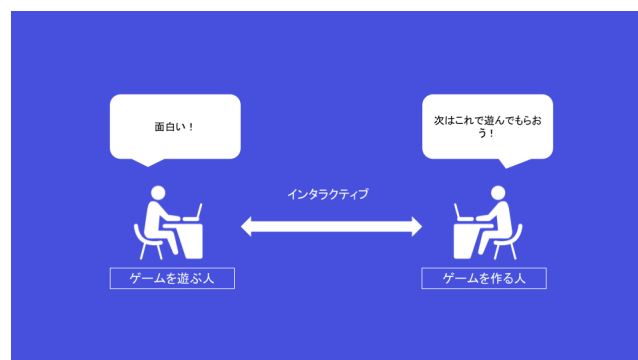
- ここでジャンプして欲しいから前方に落とし穴を作ろうか？
- 一見不規則な動きだけど実は法則性があって、それに気づけたプレイヤーが楽しめるようにしよう！

など、プレイヤーを驚かせたり、感動させたりする要素を調整し、何度もプレイしたくなるような設計をしていくことをレベルデザインと呼びます。

ゲームは誰かに遊んでもらうもの

現在はゲームというと、コンピュータやスマートフォンに入っているアプリなどを想像する人が多いと思います。

FPS やマルチ対戦以外のゲームは 1 人で遊んでいるように思えますが、プレイヤーはゲームを通してその向こうの「つくった人」と一緒に遊んでいると言えます。そう考えるとゲームは全て**インタラクティブ**（双方向の対話型）な遊びと言えますね。



自分の行動に呼応してくれる「誰か」がいることでプレイヤーは「予想」を繰り返し、「結果」を得ることができます。それは同時にプレイヤーに行動する理由を与えます。

理由があり予想し、実践し結果を得る体験を繰り返しプレイヤーに提供することが面白いゲーム作りの要となります。

フィードバックをもとに改善

実際に自分が作ったゲームを誰かに遊んでもらうことで、さまざまな視点からのフィードバックを得られたと思います。

- 障害物が多すぎて難しい／逆に少なくて簡単すぎる

- ボールを上下に動かした方が面白いかも
- たまにスピードが違うボールが出たら飽きずに続けられるかも

などなど、いろんな意見を参考に、もう一度自分の作品をカスタマイズしてみましょう！

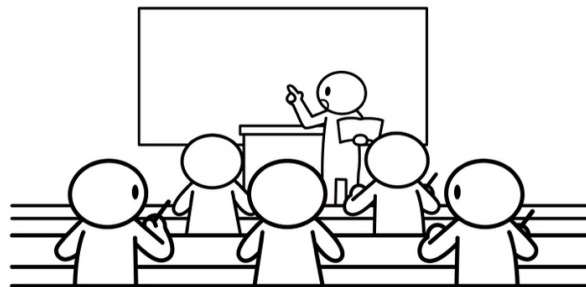
自分だけが楽しむゲームではなく、ゲームで遊ぶプレイヤーを楽しませるためのゲーム作りを意識してください。

発表会

カスタマイズが終わったら、グループごと・クラスごとに発表してみましょう。

同じゲームを作っているけど、カスタマイズの方向性でさまざまな可能性があると思います。

他の人はどんなことに気を付けて開発をしたのかを聞いてみましょう。



3. Jasmine Tea の応用 1

さまざまな機能（命令/関数）について

今回の教材で紹介している機能（画面を移動させる・キャラクターを動かす・衝突判定など）を利用してカスタマイズを行なってもらいました。

しかし、Jasmine Tea には今回のゲーム作成で紹介できなかった機能がたくさんあります。

その中から、よくゲーム開発で利用される機能を一部紹介しますので、ぜひ活用してみてください。

time()

プログラム実行からの経過時間を取得します。

`time` では、プログラムが実行されてからこの関数が実行されたときまでの経過時間が、この関数の結果の値となります。経過時間は、ミリ秒（1000 分の 1 秒）単位です。

もし経過時間が最大数値（2147483647）を超えたときは、経過時間は 0 に戻ります。

サンプル

```
s=time()  
  
do  
  t=time()  
  n=(t-s)/1000  
  locate 0,0  
  print n;"秒"  
loop
```

crash(n1,n2)

2 つのSpriteが衝突しているかどうかを確認します。

- `n1, n2` - 衝突しているかどうかを確認したいSprite番号の id 式(0~31)。

2 つのSpriteが衝突していたときは、-1 がこの関数の結果の値になります。もし衝突していなかったときは、0 がこの関数の結果の値となります。

衝突しているかどうかの判定は、Spriteを構成する四角形（16x16 ドット、もしくは 32x32 ドット）が重なっているかどうか、でチェックします。

いずれかのSprite、または両方のSpriteが非表示だったときは、この関数の結果の値は常に 0 となります。

サンプル

```
if crash(0,1) then
  stay 0
  stay 1
  locate 15,10
  print "GAME OVER"
end
end if
```

beep

ブザー音を鳴らします。

```
beep w, f, l
```

- w - 波の種類 (0: 正弦波, 1: 方形波, 2: のこぎり波, 3: 三角波)
- f - 周波数 (1 ~ 22050)
- l - 長さ (1 ~ 10000)

w, f, l を省略すると w: 1, f: 2000, l: 200 の数値で音が鳴ります。

サンプル

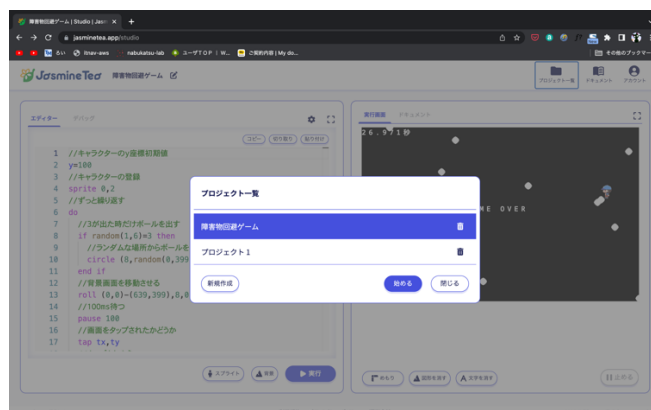
```
beep 0, 2000, 800
```

実習

では実際に作成した回避ゲームにタイムとブザー音を追加したいと思います。

まずは前回作成したゲームを開きましょう。

別のプロジェクトを開いていた場合は右上のプロジェクト一覧から対象のプロジェクトを選択することができます。



ボールにぶつかったらブザー音が鳴るようにしよう

ブザー音は 1 行コードを追加するだけで鳴らすことができます。

下のコードをボールとキャラクターがぶつかった時という条件の中に追記してください。

```
//ブザーを鳴らす  
beep 1,2000,800
```

キャラクターとボールがぶつかったらという条件は `touch` 関数を利用していましたね。

ぶつかったタイミングでブザーが鳴っていれば成功です。

カスタマイズをしよう

3 つの値を色々変更してみてください。様々な音が鳴ります。

また、下のように記述すると 2 つの音を順番に鳴らすこともできるのでいろいろ試してみましょう。

```
beep 1,2000,200  
pause 200  
beep 1,1000,200
```

💡 音の波の種類や周波数に関しては実際に色々数値を変化させてどのように変化するかを確認してみましょう。同じ周波数でも波形が変わるとどのような変化が起こるのか確認してみてください。自分で検索して調べてもいいと思います。

タイマーをつけよう

次にタイマーをつけて何秒ボールに当たらずにプレイできたかを画面に表示したいと思います。

まず `do~loop` の上の行に下のコードを追記してください。

```
s=time()
```

次に `do~loop` の中に以下のコードを追記してください。

```
t=time()  
n=(t-s)/1000  
locate 0,0  
print n;"秒"
```

これで下の画像のように画面に経過時間が表示され、ゲームオーバーになると止まるようになったかと思います。

経過時間が動かないときは、コードに誤字がないか、記述する場所が合っているかを確認しましょう。



カスタマイズをしよう

現在は画面の左上に表示してあるだけですが、例えばゲーム実行中は表示せず、ゲームオーバーになったタイミングで画面の中央に表示するようにしたり、カウントダウンにしてー

定時間生き残ったらゲームクリアにするようにしたりと、タイマーを使えばいろんなゲーム性を持たせることができるようになりますと思います。

ぜひ色々試してみましょう。

他の機能も使ってみよう

今回は数ある機能のうちの一部を紹介しました。

Jasmine Tea には他にもいろいろな機能が備わっているのでぜひいろんな機能を触ってみましょう。

命令/関数については、Jasmine Tea 公式の「命令/関数辞書」というページを参照してください。

<https://jasminetea.app/docs/references/index>

4. Jasmine Tea の応用 2

実習

今回は少し複雑な機能として、声で操作できるようにカスタマイズしてみたいと思います。

VUI について

Amazon Echo や Google Home などのスマートスピーカーの登場で、音声によるユーザーインターフェース（利用者が操作する部分・接点）を利用する機会が増えてきました。

長い年月をかけて操作性の工夫が積み重ねられてきた GUI（グラフィカルユーザーインターフェース：マウスやキーボードで操作するもの）と違い、VUI（ボイスユーザーインターフェース：声で操作するもの）は、一般的に利用されるようになってからの月日がまだまだ浅いため、いざ VUI を使ったサービスを作ろうとすると、どのように使えばいいのか困ってしまいます。

まだまだ利用者も開発者も試行錯誤の段階なので、これからどんどん新しい VUI の形ができてくると思いますので、ぜひ皆さんも VUI を使ったゲーム開発にチャレンジして色々なアイデアを生み出してみましょう！

もしかしたら 10 年後は声で端末を操作するのが当たり前になっているかもしれません。

<https://techplay.jp/column/224>

音声入力を取得する

まず `listen` 命令を使って音声入力を取得したいと思います。

💡 `listen` 命令とは

ウェブブラウザが持つ音声認識機能を使って、マイクを ON にして発話内容を聞きはじめます。認識された発話内容は、`hear$` 関数によって取得することができます。

この `listen` 命令が実行されたときは、マイクが ON になって発話内容の聞き取りが開始された後に、すぐに次の行が実行されます。聞き取り中かどうかは、`listening` 関数によって判断することができます。

`do~loop` の上の行に下記のコードを追記してください。

```
//マイクから音を入力  
listen
```

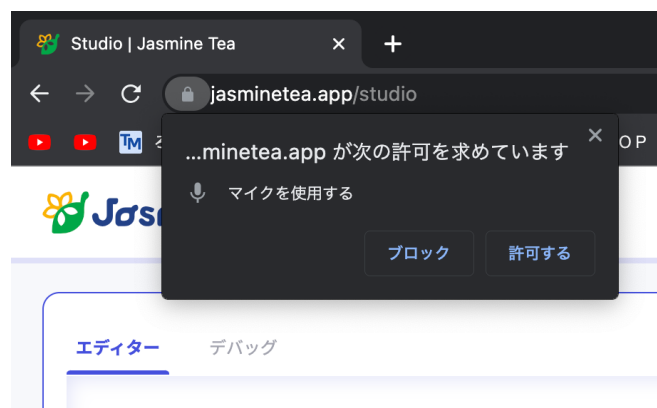
次に、`do~loop` 内に以下のコードを追記しましょう。

```
//変数 s$にマイクから取得した音データを文字として代入する  
s$=hear$()  
//マイクで認識中でなければ認識させる  
if not listening() then  
  listen  
end if  
print s$  
//もし変数 s$の値に「上」が入っていれば  
if include(s$,"上") or include(s$,"うえ") or include(s$,"梅") then  
  y=y-16  
end if  
//もし変数 s$の値に「下」が入っていれば
```

```
if include(s$,"下") or include(s$,"した") or include(s$,"舌") then  
  y=y+16  
end if
```

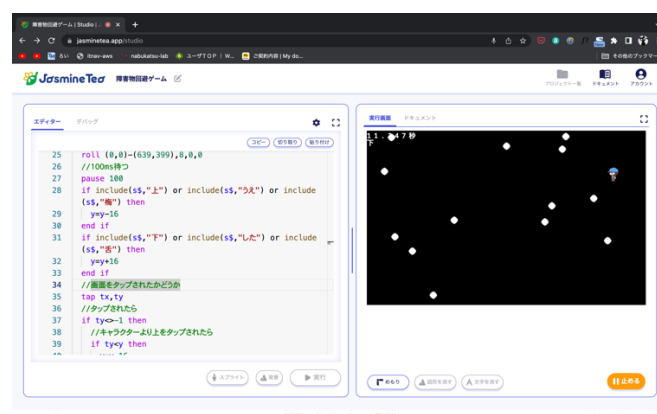
書き終わったら実行してみましょう。

ブラウザ上からマイクの機能を利用するときには下の画像のようなウィンドウが表示されるかと思うので、「許可する」をクリックしてください。



実行したらマイクに話しかけてみましょう。

「上」や「下」という言葉に反応してキャラクターが上下に動いていれば成功です。



音声認識がうまくいかないときは

2文字だけだとあまりマイクが認識しない場合があります。

そのようなときは「上に」とか「下へ」などと少し喋る言葉を増やしてみてください。

2 文字よりも認識しやすくなると思います。

また、`if include(s$, "上") or include(s$, "うえ") or include(s$, "梅") then` でどんな言葉を「上」として認識するかを決めています。

実行画面を確認し、自分が「上」として発音した内容が別の文字として認識されていた場合はその言葉もこの条件文の中に追加しておきましょう。

タップされたら動かすコードを消しておく

現在は音声認識とタップの 2 つの入力でキャラクターを動かしているのでタップされたら動かすコードは削除かコメントアウトしておきましょう。

コメントアウトとは、プログラムを編集するときに特定の箇所をコメント化して一時的に除外することです。これによってコメント化したコードは実行されなくなります。

コメントアウトは対象のコードの先頭（左側）に「`//`」を記述するだけです。

```
//画面をタップされたかどうか
//tap tx,ty
//タップされたら
//if ty<>-1 then
//  キャラクターより上をタップされたら
//  if ty<y then
//    y=y-16
//  キャラクターより下をタップされたら
//  else
//    y=y+16
//  end if
//end if
```

VUI を導入したことにより、またゲームの難易度が変わりましたね。

タップなら簡単に避けられたボールが声で操作になったとたん一気に難しくなったと思うので再度ゲーム難易度を変更してください。

💡 チャレンジ問題

- 「上にジャンプ」「下にジャンプ」と言ったら通常よりも多く移動できるようにしよう
- 「ストップ」と言ったら 1 秒間画面を停止させるようにしよう

ぜひいろんな方法を試してください。

今後のプログラミング学習について

Jasmine Tea を使ってプログラミングの基本構文や考え方などを学んできました。

これらの内容はどんなプログラミング言語にも共通して役立つものです。

まだまだ紹介しきれていない機能やプログラムの基本的な考えなども多くあるので、Jasmine Tea のウェブサイトにあるドキュメントを見ながらどんどんいろんなゲーム開発に挑戦するのもいいと思います。

<https://jasminetea.app/docs>

また、プログラミングはモノづくりの手段なので、例えば「3D のゲームを作りたい」「自分の Web サイトを作りたい」「スマートフォンのアプリを作りたい」など、目的に合わせてこれから学ぶプログラミング言語を選択していきましょう。

Jasmine Tea でプログラミングを学んだ後、次のステップとしてどんなものを学べばいいのか、いくつかの候補をご紹介します。

Unity

2D、3D ゲームを作りたい/VR や ARなどを体験してみたい人向け



```
using System;
```

```
class Test
{
    static void Main(string[] args)
    {
        string message = "Hello World!";
        Console.WriteLine(message);
    }
}
```

Unity とは、ユニティ・テクノロジーズ社が提供する、ゲーム開発プラットフォームです。

最近ではゲーム以外にも医療・建築・アート作品や、VR・AR コンテンツの作成

など幅広い場所で利用され始めているのも特徴です。

Unity はマルチプラットフォームといって、Unity1 つでパソコン、スマートフォン、Web アプリ、家庭用ゲーム機など様々な機器で実行できるアプリを作ることができます。

Unity で使用するプログラミング言語は「C#」という言語で、少し難しい部分もありますが、コードをあまり書かなくても簡単なゲームを開発することができるので初心者にもおすすめです。

最初のチュートリアルとして、Unity 公式の「玉転がし」から始めてみましょう。

<https://learn.unity.com/project/bolt-roll-a-ball-tutorial>

Web フロントエンド

Web サイト・Web アプリケーションを作りたい人向け




```
let message = "Hello World!";  
console.log(message);
```

ブラウザで検索すると、多くの Web サイトを見ることができると思います。

また、Google カレンダーやみなさんが学習した Jasmine Tea など、Web ブラウザ上で動く Web アプリケーションなどもどんどん増えてきています。

そういった Web 上で動くものを作りたい場合は Web フロントをまずは学習してみましょう。

多くの Web サイトは「HTML, CSS, JavaScript」という言語を利用し作成されています。

JavaScript は現在世界で 1 番使われている言語で、Web フロント以外にもサーバーサイド開発などにも利用されています。JavaScript を習得することで、フロントエンドとバックエンドの両方で利用できるため、幅広いスキルを身につけることができます。

Web フロントのチュートリアルは「progate」がおすすめです。

アカウントの作成をする必要がありますが、基礎部分は無料で学習でき、ブラウザ上で完結するので環境設定などをする必要がありません。

<https://prog-8.com/>

Python

人工知能や機械学習に挑戦してみたい人向け



```
message = "Hello World!"  
print(message)
```

Python は、構文がシンプルで読みやすいプログラミング言語です。初心者にも扱いやすく、多くのプロジェクトで利用されています。

AI 技術以外にも YouTube や Instagram などの Web アプリケーション開発などにも使われており、入門書やサンプルコードなどもたくさんあるので、独学でプログラミングを学びたい人にもやさしい言語だと思います。

Python のチュートリアルも「progate」がおすすめですが、他にもたくさんの入門サイトがあるのでぜひ色々調べてみてください。

下記のサイトもおすすめです。

<https://www.python.jp/train/index.html>

今回紹介したもの以外にもたくさんのプログラミング言語が存在します。

自分がプログラミングを学ぶ目的にあった言語を選択してください。

プログラミングの可能性は今後ますます広がっていきます。さまざまなものがプログラミングで開発され、いろんな場面で活用され始めてきました。

この先もさまざまな分野において IT 化が加速し、今後もプログラミングを扱う人が増えて生活に欠かすことのできない存在であり続けると思います。

将来プログラマーになるかどうかに関わらず、身の回りで使われているプログラミングの基礎知識やプログラミングの考え方などを理解できていれば必ずこれからの人生に役立つと思いますので、ぜひしっかりと学び続けていって欲しいと思います。