# micro:bit Nol プログラムの記入と実行

\_\_\_\_\_組\_\_\_\_番 氏名\_\_\_

1. editor を検索

検索

https://python.microbit.org/v/2.0 にいったら Try the beta に飛ぶ。

2. 編集画面



3. 初期のプログラム解説

	コード	内容
1.	# Imports go at the top	①コメントアウト
2.	from microbit import *	②ライブラリを読込
3.		
4.		
5.	# Code in a 'while True:' loop repeats forever	
6.	while True:	③インデントがさがった部分をずっと行う:
7.	display.show(Image.HEART)	④LED に"HEART"というイメージを出す
8.	sleep(1000)	⑤1000 ミリ秒の間、動きを止める
9.	display.scroll('Hello')	⑥ディスプレイに「Hello」とスクロールする

①コメントアウト 実際のプログラムの動作に影響しない文字列。

②マイクロビットのプログラムのライブラリ(プログラムコード集)を呼び出して使えるようにする。
 ③while で始まる文は「~の間:以降の事をする」という意味。この場合はずっと:以降の動作を繰り返すということになる。

④ディスプレイにイメージを出す。この場合は、micro:bit のライブラリに収録されている「HEART」と いうイメージを出している。

⑤sleep()はかっこの中にある時間、プログラムの動作を止める。かっこ内の単位はミリ秒。1秒は1000 ミリ秒。

⑥ディスプレイにかっこの中の文字列をスクロールさせる(日本語は使用できない)。

課題 1-1★ LED にスクロールさせる文字列を変えてみよう。(プログラムの画面の写真提出)

課題 1-2★ HEART のイメージを 10 秒間表示させてみよう(プログラムの画面の写真提出)

課題 1-3★★ HEART 以外にも以下のイメージがライブラリに収録されている。気になるものを表示さ せてみよう。(micro:bit の LED 画面の写真提出)

Image.HEART Image.HEART_SMALL Image.HAPPY		Image.HAPPY	Image.SMILE	Image.SAD
Image.CONFUSED Image.ANGRY Image.ASLEEP		Image.ASLEEP	Image.SURPRISED	Image.SILLY
Image.FABULOUS	Image.MEH	Image.YES	Image.NO	Image.CLOCK*
Image.ARROW_** Image.TRIANGLE		Image.TRIANGLE_LEFT	Image.CHESSBOARD	Image.DIAMOND
Image.DIAMOND_SMALL	Image.SQUARE	Image.SQUARE_SMALL	Image.RABBIT	Image.COW
Image.MUSIC_CROTCHET	Image.MUSIC_QUAVER	Image.MUSIC_QUAVERS	Image.PITCHFORK	Image.XMAS
Image.PACMAN	Image.TARGET	Image.TSHIRT	Image.ROLLERSKATE	Image.DUCK
Image.HOUSE	Image.TORTOISE	Image.BUTTERFLY	Image.STICKFIGURE	Image.GHOST
Image.SWORD	Image.GIRAFFE	Image.SKULL	Image.UMBRELLA	Image.SNAKE

\*1から12までの数字を指定できる。

\*\* N, NE, E, SE, S, SW, W, NW が指定できる。

課題 1-4★★★ これまでの知識を使い、3 回連続同じイメージが点滅する方法を考えてみよう。点滅 したあと, Hello を表示させよう。(プログラムの画面の写真提出)

ヒント:♥ 何もない文字 ♥ 何もない文字 ♥ 何もない文字 Hello

## micro:bit No2 エラーの見方と直し方(間違い探し)

\_\_\_\_\_組\_\_\_\_番 氏名\_\_\_\_\_

1. プログラムの記述を開始する場所は決まっている。また、正確に記述しなければならない。

		解説
1.	# Imports go at the top	
2.	from microbit import *	
3.		
4.		
5.	# Code in a 'while True:' loop repeats forever	
6.	while True:	
7.	display.show(Image.HEART)	インデント(タブ)が一つ。半角スペース4つ分
8.	sleep(1000)	インデント(タブ)が一つ。半角スペース4つ分
9.	display.scroll('Hello')	インデント(タブ)が一つ。半角スペース4つ分

一文字でも間違えたり、記入開始場所を間違えるとエラーになる。

2. 間違い探し

課題 2-1★ 配布されたファイルをテキストエディタに張り付けて、どこが違うかを探し、正しい答えを 書いてみよう。(このプリントの写真提出)

#### AttributeError (属性エラー)…ドット(.)の後の単語が間違っているとこうなる可能性大

\_\_\_\_\_行目の\_\_\_\_\_が間違っていて\_\_\_\_\_が正しい

#### SyntaxError(構文エラー)…特殊記号が間違っているとこうなる可能性大

\_\_\_\_\_行目の\_\_\_\_\_が正しいで\_\_\_\_\_が正しい

#### NameError (名前エラー)…単語やドット(.)の前の単語が間違っているとこうなる可能性大

\_\_\_\_\_行目の\_\_\_\_\_が間違っていて\_\_\_\_\_が正しい

行目の が間違っていて が正しい

※tab 一つ分は⇔と書いてもらえれば結構です。

また、タブは半角スペース4つ分を入力しても同じ解釈になります。

#### ImportError (インポートエラー)…ライブラリやモジュールが間違っているとこうなる可能性大

行目の が間違っていて が正しい

そのほかにも色々とエラーがあります。この簡単なプログラムでさえ、様々なエラーがあることを知っ ておこう。

コラム

プログラムで一番厄介なのは、このエラーです。何度もエラーを出すとへこたれます。実は私も教材 研究の段階でエラーを出したりして、いくつか授業案を作っていると、ダウンロードしたファイルの 数がすぐ 100 を超えてしまいました。エラーはつきものだという考えのもと、勉強してもらえれば幸 いです。ただ、やはりエラーは嫌なので、エラーが出たときにすぐ対処できるような想像力や知恵を身 に着けていきましょう。

上記以外のエラー(論理エラー)

論理エラーとは、プログラムの実行結果が意図した通りにならないエラーのことである。論理エラー は、転送が正常に行われ、プログラムの実行途中で異常終了することなく動作するものの、アルゴリズム が論理的に正しくない場合のことを意味する。例えば以下のコードは論理エラーである。

```
1 # Imports go at the top
2 from microbit import *
3
                                                Ĉ
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
       display.show(Image.HEART)
7
8
      sleep(1000)
9 display.scroll('Hello')
10
```

これを書くと、ずっとハートが流れ続け、Hello が表示されません。

3. 間違い探し問題作り

課題 2-2★★ 最初のプログラムの一部を自分で削除してみたり、何かを追加してみたり、あるいは大文字 にしてみたり、小文字にしてみたりして、どのようなエラーが起こるかどうか確認してみよう。5 種類のエ ラーを確認できたらポイント UP! (このプリントの写真提出)

行目の	を	に変えると、	エラーになる。
行目の	を	に変えると、	エラーになる。
行目の	を	に変えると、	エラーになる。
行目の	を	に変えると、	エラーになる。
行目の	を	に変えると、	エラーになる。
行目の	を	に変えると、	エラーになる。

4. エラー名把握

課題 2-3★★★ 上に出した以外にもエラーがある。どのようなエラーがあるか調べてみよう。「python エラー 一覧」などと調べてみましょう。内容がわからなくても、エラー名はわかったりするので、エラ ー名とその説明を書いてください。

エラー名	説明
Error	
Error	

Error	
Error	

プログラミング言語の分類

分類方法	種類	説明	
表現方法に	( )	文字で表現する。例:Python	
よる分類	( )	ブロックなど図形で表現する。例:スクラッチ	
変換方式に	( )言語	高級言語を一括して機械語に変換し実行する	
よる分類	( )言語 (スクリプト)	高級言語を1行ずつ機械語に変換し実行する	
	( )言語	処理手順を実行する順番に記述する	
方法論によ	関数型言語	関数を呼び出して処理を行う	
る分類	論理型言語	記号論理学の体系をもとにした言語	
	( )言語	オブジェクト指向を取り入れた言語	

代表的なエラーの種類

①.(ドット)のあとの単語が間違っている可能性がある。	
②特殊記号が間違っている可能性がある。	
③変数や. (ドット) の前の単語が間違っている可能性がある。	
<ul><li>④字下げ(インデント)が適切に行われていない可能性がある。</li></ul>	
⑤モジュールが適切に読み込まれていない可能性がある。	
⑥文法上も問題なく実行も途中で止まることもないが, 意図 した結果が得られない場合のエラー	

### micro:bit No3 変数代入とコメントアウトとキー操作

\_\_\_\_\_組\_\_\_\_番 氏名\_\_\_\_\_

プログラムを書いていると、どこまではうまくいっていて、どこからうまくいかないのかを切り分ける ために、コメントアウトをよく使います。その操作の練習をしよう。

# コメントアウト,アンコメント Ctrl + /

# 元に戻りたい場合 Ctrl + z

※複数行の場合、カーソルの位置から Shift を押しながら↓を押して複数行を選択して Ctlr + /

	コード	解説
1.	# Add your Python code here. E.g.	
2.	from microbit import *	
3.		
4.		53 という整数値を a という変数に代入する。
5.	a = 53	※プログラムは右の値を左に代入するという
6.	b = 59	表現をよく使います。
7.	c = 67	
8.	d = 4661	13 と"13"の違いは数値の 13 か文字列の 13 か
9.	e = "13"	f という変数に 40 という文字列を入れる。
10.	f = "40"	
11.	g = "0"	
12.		
13.	ans = a + b	
14.	ans = a - c	
15.	ans = b * c	
16.	ans = d / b	数値と文字列を足してみたらどうなるか?
17.	ans = a + e	文字列と文字列を足したらどうなるか?
18.	ans = e + f	文字列から文字列を引いたらどうなるか?
19.	ans = f - g	
20.		
21.	while True:	
22.	display.scroll(ans)	

課題 3-1★ 次の①~⑨の課題を行い、マイクロビットに書かれている表示を記入してみよう。エラーが 出る場合、何エラーなのかも書いてください。(このプリントの写真提出)

1	まずは 13 行目から 19 行目までコメントアウトしてください。	
2	13 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
3	14 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
4	15 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
5	16 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
6	17 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
7	18 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
8	19 行目だけアンコメントし実行した際に、どんな表示になるか。	表示
9	13~19 行目をアンコメントし実行した際に、どんな表示になるか。 (すべての行を実行)	表示

課題 3-2 わかったこと★★ 少なくとも三つは記述 (このプリントの写真提出)

・割り算の答えのこと。
・文字列と数値の足し算のこと
・文字列と文字列の足し算のこと。
・文字列と文字列の引き算のこと。
・複数行エラーのコードがあった際に、表示されるメッセージのこと。

課題 3-3★★★ 自分で上のプログラムを以下のような行為を自分でプログラミングし、どのような表示 がでたかを書いてみよう。(このプリントの写真提出)

- ① 0で割ってみる。
- ② 10/3をしてみた場合、小数点第何位まで表示されるか。

③ a = a + b をし, 21 行目の display.scroll(ans)を display.scroll(a)に変えてみたらどうなるか。

左側には計算部分の1行分のコードを記入,右側には結果を表示。エラーの場合も内容を表示

①	表示結果	_	
2	表示結果	_第(	)位まで表示
3	表示結果	_	

課題 3-4★★★ 変数 x に 7,変数 y に 3 を代入し,その変数同士の足し算の答えを z という変数に入れ,表示で「No10」となるようなプログラムを書いてください。(プログラムの写真提出)

ヒント

計算結果を整数値ではなく、文字列の型にしたい場合、z = str(z)とすれば、文字列に変換されます。
 それを 3-1 の 18 行目のような処理をすれば表示できます。

1. コメントアウト, アンコメント

プログラムの行の先頭に#ハッシュマークをつけてプログラムの実行を無効化することを (
)という。選択は2通りあり、①マウスで選択をする方法、②(
) を押しながら、(
)を押すことで選択することができる。選択したあとで、(
) を押すことで、コメントアウトができ、再び押すと#をとる(
)ができる。

コメントアウトやアンコメントを利用することで、効率的にプログラムを書くことができる。もし、 プログラムのコードを消してしまうと、再度必要になったときに再び記述しなければならないので、効 率が悪い。コメントアウト、アンコメントを使う利点は今までに書いた(

) 点や, (

)ことができる点がある。

	コード	実行される行
1.	コード①	
2.	コード②	
3.	# コード③	$\rightarrow$ $\rightarrow$
4.	# コード④	
5.	コード(5)	

	コード	実行される行
1.	# コード①	
2.	コード②	
3.	# コード③	$\rightarrow$ $\rightarrow$
4.	コード④	
5.	コード(5)	

2. 変数

変数は ( ) や ( ) を保管する箱のようなもので,代入演算子 ( )を使って最 初に変数に値を代入した時点で変数の宣言が行われる。

3. 演算子一覧

次は Python でよく使われる演算子の例である。

	算術・結合	演算子		比較演算子			ŕ	弋入演算子
+	数値の足	し算		==	=	( )	=	( )
	(		)	!=	=	等しくない		倫理演算子
	引き算			<		未満(より小さい)	and	かつ
*	(	)		>		超過(より大きい)	or	または
/	(	)		(	)	以下	not	ではない
%	(		)	(	)	以上	۲	ット演算子
								$\sim \mathcal{O}$

### micro:bit No4 LED イメージの自作

\_組\_\_\_\_番 氏名\_

LED のディスプレイに表示されるイメージはプリセットとして予めある程度は登録されていますが、 自分でイメージを自作したい場合があります。そんな場合にどうすればよいか勉強していきましょう。

microbit の LED ディスプレイは縦5 個、横5 個の赤色の LED からできています。LED を光らせる明るさは0(オフ)から9(最大の明るさ)の10段階で LED ごとに指定できます。

	コード	解説
1.	# Add your Python code here. E.g.	
2.	from microbit import *	
3.		
4.	jan = Image("00000:"	行ごとに区切って各 LED の明るさを指定
5.	"00000:"	コピー&ペーストをしながら、編集すると楽です
6.	"09999:"	(Ctrl+C,Ctrl+V)。ここでは最大の明るさの9を指定
7.	"99999:"	
8.	"99999")	一番最後はコロン(:)なし。グーの表示
9.		という内容を jan という <b>変数名</b> にしておく
10.	jan = Image("09090:"	上の内容をコピーし、適宜書き換えていく。
11.	"09090:"	
12.	"09090:"	
13.	"09999:"	
14.	"99999")	チョキの表示
15.		
16.	jan = Image("90909:"	
17.	"90909:"	
18.	"90909:"	
19.	"99999:"	
20.	"99999")	パーの表示
21.	display.show(jan)	ディスプレイに「jan」を表示させる。
22.		

課題 4-1★ 上のコードを書き、パーを表示させてみよう。次にパーの変数を宣言している部分をコメン トアウトし、チョキを表示させよう。(プログラムの画面の写真提出。パーとチョキの2枚分提出) ※上のプログラムのままだと jan という変数がグーからチョキに、チョキからパーに上書きされていき、 16 行目の変数 jan の内容が表示されるので、16 行目から 20 行目までをコメントアウトし、チョキの表 示を出してみたりしよう。

課題 4-2★★ 自分が作った LED イメージを点滅させたいこともあると思います。そんな場合は、 display.show と display.clear()を交互に繰り返し、間に sleep 時間を設ければ点滅ができます。(プログラ ムの画面の写真提出)

	コード	解説
1.	# Add your Python code here. E.g.	
2.	from microbit import *	
3.		
4.	jan = Image("00000:"	
5.	"00000:"	
6.	"09999:"	
7.	"99999:"	
8.	"99999")	
9.		
10.		
11.	吹	行数に注意してください。 20 行目まで
	Ш	とびます。
20.	"99999")	
21.		
22.	while True:	
23.	display.show(jan)	jan という変数を表示させる。
24.	sleep(200)	200 ミリ秒とめる。
25.	display.clear()	消す。
26.	sleep(200)	200 ミリ秒とめる。
27.		

### 課題 4-3★★ 点滅させるプログラムはどんな時に使うと効果的か簡単に意見を書いてください。

課題 4-4★★ 自作イメージをたくさんつくり、アニメーションにすることもできます。(micro:bit の LED 画面の動画提出)

	コード	解説
1.	# Add your Python code here. E.g.	
2.	from microbit import *	
3.		
4.	rock = Image("00000:"	
5.	"00000:"	
6.	"09999:"	
7.	"99999:"	
8.	"99999")	
9.		
10.	scissors = Image("09090:"	上のものをコピーし、適宜書き換える
11.	"09090:"	
12.	"09090:"	
13.	"09999:"	
14.	"99999")	
15.		
16.	paper = Image("90909:"	
17.	"90909:"	
18.	"90909:"	
19.	"99999:"	
20.	"99999")	
21.	jan= [rock, scissors, paper]	jan という名前で <b>リスト集</b> を作る。
		※プログラム的には配列という。
		[ や ]はエンターキーの近くにあります。
22.	while True:	
23.	display.show(jan, delay=200)	jan というリスト集を再生し、遅延時間 200 ミ
		リ秒を設定する。
24.		

課題 4-5★★★イメージの枚数や秒数も調整して自分なりのアニメーションを作成してみよう。 (micro:bit の LED 画面の動画提出)。どんなアニメーションでもいいです。

1. 効率よく作るためのコツ

コピー…()+()とペースト…()+()を使う。
 コメントアウトを効率的に使う。

2. 配列(リスト)

データが多い場合に、異なる名前の変数を複数用意することなく、複数の値を一つの名前(配列名)に よって管理する仕組みを()という。Python では())と呼んでいる。今回はグー、 チョキ、パーの変数を jan という配列名で入れている。

配列の中の各変数を()といい,要素の番号を()()という。添字を指定することで,要素への値の代入や参照ができる。配列の要素を1つの添字で指定する配列を
 しといい,2つの添字は,()という。

一次元配列	二次元配列
a=[1, 2, 3]	b=[[1,2,3],
	[4,5,6],
	[7,8,9]]
display.scoll(a[0]) は「1」が表示される。	display.scroll(b[0][2])は「3」が表示される
※添字は0から始まることに注意!	display.scroll(b[2][2])は「9」が表示される。

### micro:bit No5 疑似乱数と IF 文

ゲームなどでランダムな数字を発生させ、その値に応じてガチャのアイテムを引かせたりする場合が あります。どのように実装しているのでしょうか?

	ゴード	解說
1.	# Imports go at the top	
2.	from microbit import *	
3.	import random	今回はここが追加されることに注意!
4.		
5.	while True:	インデント下がった部分をずっと行う。
6.	value = random.randint(1,6)	randint はランダムの整数値を返し、変数 value に代
		入。この場合 1~6 まで。int は INTEGER の略。
7.	display.scroll(value)	変数 value に入った値を表示させる。
8.		

課題 5-1★ 上記のプログラムでランダムが生成されるかどうか調べる。(プログラムの画面の写真提出)

#### 疑似乱数 Tips

乱数とはいっても本当にランダムで発生しているわけではありません。実は一定の計算をして、表示させ、それを繰り返していくと、「全体的にランダムに実行されたように見える」という仕組みになっています。計算を読み解くと、次にでてくる値が予測できます。昔は線形合同法が有名でしたが、ランダム性がないと指摘されたので、現在メルセンヌツイスターという疑似乱数のアルゴリズムが使われることが多いです。microbitもメルセンヌツイスターを利用しています。

著作権配慮のため記載省略

そろそろ知っておきたいプログラムの意味。

オブジェクトとメソッド ○○○.△△△というプログラムをよく見かけます。オブジェクトとは、○ ○○にあたる部分で**モノ**と抽象化され、メソッドとは△△△にあたる部分でオブジェクトをどう扱うかを 定めた**方法**です。display.scroll はディスプレイ(モノ)をスクロールさせる(方法)という意味です。

	コード	解説
1.	# Imports go at the top	
2.	from microbit import *	
3.	import random	
4.		
5.	# Code in a 'while True:' loop repeats forever	
6.	while True:	
7.	value = random.randint(1,6)	
8.	display.scroll(value)	表示がうまくいったらコメントアウト
9.	if value <= 2:	今回 1,2 を GU にするので,<=を使用。
10.	display.scroll('GU')	
11.	elif value <= 4:	今回 3,4 を CHO にするので, <=を使用。
12.	display.scroll('CHO')	
13.	else:	それ以外は PA(事実上 5,6 の時)
14.	display.scroll('PA')	

課題 5-2★ IF 文を使い値によって別々に表示させるようにしよう。(プログラムの画面の写真提出)

IF 文の基本

2 通りの場合	3通り以上の場合
if 条件式 1:	if 条件式 1:
処理 A	処理 A
else:	elif 条件式 2:
処理 B	処理B
	else:
	処理C

条件式の比較演算子について

A == B	A は B 以上	$A \ge B$
A != B	A は B より小さい	A < B
A > B	AはB以下	A <= B
-	A == B $A != B$ $A > B$	A == B     A は B 以上       A != B     A は B より小さい       A > B     A は B 以下

※AやBは変数や数値が入る。=一つであると代入を意味するので、イコールは==と記入。

ヒント 数値が多くなった場合の IF の使い方	if x <= 30:	
疑似乱数が 1~100 の整数の中で 100 通り発生し,	処理 A	考え方:30以下の場合は処理 A
それぞれ処理 A の確率が 30%,処理 B の確率が	elif x <= 40:	(31~)40以下の場合は処理 B
10%, 処理 C の確率が 60%をさせたい場合, 右の	処理 B	それ以外は処理 C
ように記述する。	else:	というように記述する。
	処理 C	

課題 5-3★★ 上のじゃんけん表示において、No4 で行った LED でじゃんけんマークを表示させるよう にしよう。(microbit の LED の動画提出) ヒント:5 行の変数代入と display.show(○○)、sleep(○○)を 使います。

```
ヒント:プログラムのヒントです。
cho = Image("90909:"
"90909:"
"999999:"
"999999")
display.show(cho)
sleep(2000)
```

課題 5-4★★ グー:チョキ:パーの出る割合が 1:2:3 になるようなプログラムを作ろう。ヒント:ラン ダムを 6 通り発生させ、条件式の比較演算子をうまく使おう。じゃんけん表示は LED を自作して, display.show (gu) を使ってもよいし, display.scroll("gu")を使ってもよい。表面の右下のヒント参照 (プ ログラムの画面の写真全行提出)

1. 疑似乱数

乱数列のように見えるが、実際には確定的な計算によって求めている擬似乱数列による乱数
 ・( )…昔の疑似乱数生成アルゴリズム。ランダム性がないと指摘されている。
 ・( )…現在主流の疑似乱数生成アルゴリズム。日本人が考案。
 micro:bit の場合,疑似乱数を扱いたい場合,2行目に( )と書く。

2. IF 文のまとめ

何か値を判別したり、「押す・押さない」の判別をしたりするなど、条件に応じた挙動を実行させるためには、IF 文を使う。

分類	2通りの場合	3 通り(以上)の場合
コード	display.scroll('Akaten') display.scroll('Gokaku')	display.scroll('Akaten') display.scroll('Normal') display.scroll('Excellent')
考え方	○○未満 (以下, 以上, より大きい, と等しい)	40 点未満でまずは判別。 次に 75 点未満で判
	か,それ以外か。※条件を指定し,判別する。	別し、あとはそれ以外。

例:score という変数を判別したい場合。

#### 比較演算子一覧

		比較演算子			比較演算子
(	)	等しい	>		超過(より大きい)
(	)	等しくない	(	)	以下
	<	未満(より小さい)	(	)	以上

## micro:bit No6 ボタンと IF 文と疑似乱数

\_\_\_\_\_組\_\_\_\_\_番 氏名\_\_\_\_\_\_

IF 文はよく使うので繰り返し学習していきます。ボタンの使い方 IF 文です。

**じゃんけんプログラム**…a ボタンを押したら、「じゃんけん・・・(手)」と表示し、b ボタンがおされた ら、「あいこで・・・(手)」と表示させる。

	コード	解記
1.	# Imports go at the top	
2.	from microbit import *	
3.	import random	ランダムオブジェクトを使うためランダムを import。
4.		
5.		 じゃんけん変数を設定しておく。
6.	gu = Image("00000:"	
7.	"00000:"	今まで"gu"と書いていたが、gu という変数を扱い
8.	"09999:"	たい場合はそのまま gu と書く。
9.	"99999:"	
10.	"99999")	
11.		
12.	cho = Image("09090:"	
13.	"09090:"	日本は安仁として並八
14.	"09090:"	
15.	"09999:"	ここに今後使う変数を書いていく。
16.	"99999")	
17.		
18.	pa = Image("90909:"	
19.	"90909:"	
20.	"90909:"	
21.	"99999:"	
22.	"99999")	
23.		
24.	while True:	乱数を発生させる
25.	x = random.randint(1,3)	ボタンが押されたという if 文。ここは was で設定
26.	if button_a.was_pressed():	
27.	#display.scroll(x)	
28.	display.scroll("janken")	if 文の中にさらに if 文を書くことも可能。

29.	if x <= 1:	gu という変数名を扱いたい場合は gu
30.	display.show(gu)	gu と表示させたい場合は"gu"
31.	elif x <= 2:	
32.	display.show(cho)	
33.	else:	
34.	display.show(pa)	ここまで書いたら下は書かずに一回実行してみる。
35.	if button_b.is_pressed():	ここは is で設定してみる。上をコピペし, a.was→b.is
36.	display.scroll("aikode")	にかえる
37.	if x <= 1:	何回も繰り返される部分。
38.	display.show(gu)	インデント(タブ)あるいはスペース4
39.	elif $x \le 2$ :	つを使いながら繰り返したい処理を書
40.	display.show(cho)	いていく。
41.	else:	
42.	display.show(pa)	

is と was の違い~挙動をそれぞれ確認してみよう~ is は前処理が完全に終わらないと反応しない。was は前処理が実行中でも押しておけば反応する。

課題 6-1★ 上記のプログラムを書く。a ボタン b ボタンはディスプレイの横にある。(LED の動画提出) 課題 6-2★★ 以下はランダムに数字を発生させ、その結果によってじゃんけんの文字をスクロールさせ るプログラムである。以下のプログラムで間違いの部分はどこか答えなさい。(プリントの写真提出)

	コード	解説				
1.	# Imports go at the top	このプログラムでは,5行目で乱数 x の値が確定して				
2.	from microbit import *	しまい,その後,乱数 x の値がずっと使われる。				
3.	import random					
4.		(A. )行目のコードを				
5.	x = random.randint(1,3)					
6.	while True:	(B.	) 行目と(C.	)行目の間に		
7.	if button_a.was_pressed():					
8.	display.scroll("janken")	もっていな	»なければいけない。			
9.	if x <= 1:					
10.	display.scroll("gu")					
11.	elif x <= 2:					
12.	display.scroll("cho")					
13.	else:					
14.	display.scroll("pa")					

課題 6-3★★ a ボタンが押されたら、さいころ 1 個をふった目(・など)を表示させよう。変数名は先 頭に関してアルファベットしか使えません。one,two,などの変数名を設定しよう。display.show と sleep も使います。(LED の動画提出)

課題 6-4★★★ a ボタンが押されたら、さいころ1個をふった目(・など)を表示させ、b ボタンが押 されたらさいころ2個ふった目の数(1~12)を表示させよう。また、さいころ2個振った場合は同 様に確からしい状態で表示させること。(プログラムの写真提出。すべての行を撮って提出すること。 途中まで書いても部分点をあげます。)

ヒント1: さいころ1 個の値を変数 x、y にそれぞれ入れて、足した値をさらに変数 z などに代入し、変数 z をつくること。

ヒント2:1~9までは IF を使って, display.show でいけるが、10,11,12 は別の方法で表示させる必要 がある。LED の表示を 10,11,12 は自作し、IF 文などを使って表示させていこう。

if z <=9:

display.show(z)

ヒント2の例

10の場合のLEDの例 ten=Image("90999:" "90909:" "90909:" "90909:" "90999")

IF 文の使い方は様々です。以下に色々な例を載せておきます。 ①IF 文の不等号の使い方

IF 文は数学的な考え方を使えば、書き方は様々です。

age = 〇〇 #〇〇には年齢が入る



不等号を変えて=を調整すれば同じになります。

②IF 文のネスト

IF 文の中にさらに IF 文を使うことも可能です。



③IF 文のネストを使わず、3パターンあるという意味で考える。

②のように書かず、以下のように書いても同じ意味です。



## micro:bit No7 音楽利用とループと初期化

\_\_\_\_組\_\_\_\_\_番 氏名\_\_\_\_\_\_

ここからは、ループに入った時のプログラムとループに入る前のプログラムでコードをどこに記述していけばよいのか考えてみよう。

/ i i i i i i i i i i i i i i i i i i i	タ	イ	マー	1	0
---	---	---	----	---	---

	コード	
1.	# Imports go at the top	
2.	from microbit import *	
3.	import music	今回はここが入ることに注意。
4.		
5.	count = 0	ループに入る前に count 変数を 0 にしておく
6.	while True:	(これを初期化という。)
7.	if button_a.was_pressed():	a ボタンが押された時
8.	count = count + 10	10 ずつ増やしていく
9.	display.scroll(count)	
10.	if button_b.was_pressed():	b ボタンが押された時
11.	while True:	ずっと繰り返す。
12.	if $count == 0$ :	0なら終了させる。
13.	music.play(music.RINGTONE)	音楽を鳴らす。
14.	break	ループを抜ける。
15.	display.scroll(count)	値を表示させ、
16.	count = count - 1	1を引く。count -= 1 という書き方でもよい。
17.		

count 変数がループに入る前にあることに注意。最初に0にしておく。初期化という。while の中にあったりすると、ループする度に0になってしまい、プログラムとして成り立たなくなる。

課題 7-1★ 上記のプログラムを書く。(プログラムの画面の写真提出)

課題 7-2★★ 次のコードは上の 12~14 行目と 15~16 行目を入れ替えただけですが、プログラム的に バグなプログラムです。

	コード	
1.	# Imports go at the top	
2.	from microbit import *	
3.	import music	

4.		
5.	count = 0	
6.	while True:	
7.	if button_a.was_pressed():	
8.	count = count + 10	
9.	display.scroll(count)	
10.	if button_b.was_pressed():	
11.	while True:	
12.	display.scroll(count)	
13.	count = count - 1	count -= 1 という書き方でもよい。
14.	if $count == 0$ :	
15.	music.play(music.RINGTONE)	
16.	break	11 行目のループを終わらせるための break
17.		

なぜなのか理由を書いてみよう。(このプリントの写真提出)

検証方法:プログラムを転送したあと、aボタンを押さずにbボタンから押してみる。

課題 7-3★★ (人数チェッカーの制作)a ボタンを押したら、押した回数をカウントし表示させ、b ボ タンが押されたら、その値を表示させる。a ボタンとb ボタンが同時に押されたら、リセットさせる。 ヒント:a と b の同時押しは、if button\_a.is\_pressed() and button\_b.is\_pressed(): です。同時押しは、a ボタンを押しながら、b ボタンを押すと実現できます。リセットは本当にリセットされたかわからないの で, 直前で音楽を鳴らしたり, 直前で display.scroll("reset")としたりして, わかりやすくしよう。(プロ グラムの写真全行提出)

課題 7-4★★★ 以下のような表示になるためにはどのようにすればよいか、プログラムをインターネットで調べながらループ (while あるいは for)を使って書きなさい。(プログラムの画面の写真提出)

表	示				ヒント
1	12	123	1234	12345	・python の for 文の書き方を検索して調べる。
					・文字列を入れる変数を文字列型で初期化しておく。
					・数字を for 文で1ずつあげていく。
					<ul> <li>・あげた数字を文字列にして、文字列の変数に結合させる。(文字列)</li> </ul>
					の結語は文字列同士の+でいけます。)

1. ループ (反復処理)

今までなぜかずっと素通りしてきた while True:。この部分を本日は掘り下げていく。反復処理(ルー プ)は while 文の他に for 文というものもあるが、これは次回行う。

while True: … 「以下のことをずっと行う」という意味である。( )ともいう。 ( )をかけばループから抜けることができる。プログラムは入力を受け続けていることが 多いので,基本的に while True のような無限ループをしていることが多い。

2. 初期化

コンピュータプログラムでは、プログラム内で利用するデータ、変数などを利用可能な初期状態にする ことを( )という。宣言した変数に初期値を書き込む処理などが該当し、プログラムの冒頭や 関数、メソッドの冒頭などで実行されることが多い。変数の初期化が必要な理由は、もしも初期化をしな い状態でその変数を用いて演算処理を行おうとすると、( ( ))が発生し、 プログラムがとまってしまう。

例	:	最初	に	変数	1C1	代入	、さ	n	る	時に	使	う	0
---	---	----	---	----	-----	----	----	---	---	----	---	---	---

例:数值型	例:文字列型
a = 1	s = 'aaaaa'
count = 0	s1 = "
など,最初に数値を代入しておく。	何か文字列を代入したり、シングルクオーテーシ
	ョン(あるいはダブルクオーテーション)でくく
	り、文字列型として定義しておく。

3. ループと初期化

初期化とループを使いプログラムを書いていくことはよく起こる。

例:aを3回表示させる。プログラム

ゴード	解説
while display.scroll('a')	初期化する。 i が 3 未満の時繰り返す。 a を表示する。 i を 1 プラスさせて,ループの最初に戻る。

これをするとiが0の時aが1回,1の時aが1回,2の時aが1回表示されるので,合計aが3回表示 される。

### micro:bit No8 リスト(配列)とループ②

\_\_\_\_組\_\_\_\_番 氏名\_\_\_\_\_

前回まで while 文を扱ったので今回は for 文を扱います。

	コード	解説		
1.	# Imports go at the top			
2.	from microbit import *			
3.				
4.				
5.	while True:			
6.	if button_a.was_pressed():	a ボタンが押されたら,		
7.	for i in range(1,11):	iを1から始めて10回数値をあげていく。		
8.	display.scroll(i)	i を表示。		
9.	if i % 2 == 1:	もし2で割って1余れば(奇数なら)		
10.	display.show(Image.SMILE)	スマイルを		
11.	sleep(3000)	3 秒表示させ,		
12.	else:	そうでなければ,		
13.	display.show(Image.ANGRY)	アングリーを		
14.	sleep(3000)	3 秒表示させる。		

課題 8-1★ a ボタンがおされたら、1SMILE 2ANGRY 3SMILE 4ANGRY・・・と 10 まで表示さ る。(プログラムの画面の写真提出)

課題 8-2★★ b ボタンがおされたら、10 回、1~10 の間で乱数を発生させその数字を表示し、もし、乱 数発生時に 10 がでれば音楽を鳴らして、ループを止める。(プログラムの画面の写真提出)

ヒント1: import random と import music を忘れずに書くこと。

ヒント2:乱数 random.randint(1,10)を発生させ,適当な変数名に代入し,それを表示する。 ヒント3:もし,その変数が 10 であったら,音楽 music.play(music.RINGTONE)を鳴らして break する。

課題 8-3★★ num というリストに 13,19,20,23,25 を入れ,それを順番に表示させる。(プログラムの画 面の写真提出)

	コード	解記
1.	# Imports go at the top	
2.	from microbit import *	
3.		
4.		
5.	num = [13, 19, 20, 23, 25]	ここで num[0]から num[4]まで値を代入する
6.	for i in $range(0,5)$ :	0から5回繰り返す。
7.	display.scroll(i)	要素番号を表示させる。挙動がわかったらコメントアウト
8.	display.scroll(num[i])	num[0]~num[4]までの 5 つ分を表示させる。
9.		

課題 8-4★★★ 8-3 の配列の要素に対して,それぞれ素数(prime number)か,合成数(composite number) かを判定するプログラムを作る。(プログラムの画面の写真全行提出。途中でも部分点あげます。) アウトプット例:  $[13 = p \quad 19 = p \quad 20 = c \quad 23 = p \quad 25 = c$ 」と出力

ヒント1:判別したい数字を解析する前に prime 変数を初期化しておく。 prime = True など ※prime = True や prime = False のように, 真偽値を入れる型のことを bool 型という。 ヒント2:変数 jを用意し, 2 から順番にその数字-1 の数まで割っていき,一つでも割れるものがあれ ば, prime = False にし,ループを抜ける (解析)。割ったあまりが0 になるのは, num[i]%j == 0 など とすると表現できる。 ヒント3: prime が True であれば「数=p」と表示し,そうでなければ「数=c」と表示する。 ※bool 型の場合, if prime == True: という書き方でもよいし, if prime: という書き方でもよい。 ヒント4:数値と文字列の結合は, display.scroll(str(数の情報) + '=p')でいけます。

### 今回はもう一つの反復処理である for 文を学びます。

	書き方	具体的なコード	解説
	for 変数 in range(回数):	for i in range():	a を 10 回表示
Û	処理 A	display.scroll('a')	
	for 変数 in range(值 A,值 B,[增減值]):	for a in range(,):	第三引数を省略すると,+1 増
2	処理 A	display.scroll(a)	える。1~10[(値 B)-1]まで表
			示
	for 変数 in range(值 A,值 B,[增減值]):	b = [ 10,13,15,17]	b のリストを表示させる。
3	処理 A	for i in range(,):	この場合, b[0]~b[3]の値を4
		display.scroll(b[i])	つ表示させる。

反復処理を使う場合,繰り返す回数として使いたいのか,変数の値として使いたいのか,配列の添字の番 号として使いたいのかどれで使いたいかを考えながら使っていくと理解しやすくなります。

練習問題

1	2
for a in range(10):	for a in range(0,10):
display.scroll('b')	display.scroll(a)
3	4
for a in range(10):	list = [1,2,3,4,5]
display.scroll('a')	for i in range(0,5):
	display.scroll(list[i])
5	
list = [1,2,3,4,5]	
for i in range(1,5):	
display.scroll(list[i])	

語群

ア. aを 10 回表示		イ. bを1	イ.bを10回表示		ウ.1~10 まで表示。	
エ. 1~5	まで表示	オ. 0~10	オ. 0~10 まで表示		カ. 0~9 まで表示	
キ.0~4 まで表示		ク. 2~5	ク. 2~5 まで表示			
答え						
① (	) ② (	) ③ (	) ④ (	) ⑤ (	)	

## micro:bit No9 様々なセンサー

microbit の中には傾き, 温度, 明るさ, サウンドなど様々なセンサーが入っています。それらの使い方に ついて考えていきましょう。accelerometer オブジェクトを使い、様々なメソッドを使うことで実現でき ます。左右 x 軸では、get\_x()メソッドを利用すれば、x 軸の傾きを取得することができ、同じく get\_y(),get\_z()も使用することができます。さらに、microbit へのジェスチャーがプリセットとして登録 されており、current\_gesture()メソッドを使うことで登録された行為を文字列で読み取ることができます。

傾きの読み取り

	コード	
1.	# Imports go at the top	
2.	from microbit import *	
3.		
4.		
5.	while True:	
6.	reading = accelerometer.get_x()	そのほかに get_y() , get_z()がある。
7.	if reading > 20:	数値が低いととても敏感に反応してしまうため、閾値
		として 20 ぐらいを設定しておく。
8.	display.show("R")	プラスなら右方向
9.	elif reading < -20:	
10.	display.show("L")	マイナスなら左方向
11.	else:	
12.	display.show("-")	

課題 9-1★ 上記のプログラムを書いてみよう。(プログラムの画面の写真提出)

ジェスチャーの読み取り

	コード	
1.	# Imports go at the top	
2.	from microbit import *	
3.	import music	
4.		
5.	while True:	
6.	display.show(Image.HAPPY)	
7.	gesture = accelerometer.current_gesture()	micro:bit を振るとセンサーが決まった動きを
8.	if gesture == "shake":	感知し gesture に"shake"という値を入れる

9.	display.show(Image.CONFUSED)	
10.	music.play(music.DADADADUM)	

課題 9-2★ 上記のプログラムを書いてみよう。(プログラムの画面の写真提出)

光センサーとサウンドセンサー

	コード	
1.	# Imports go at the top	
2.	from microbit import *	
3.		
4.	while True:	
5.	<pre>sl = microphone.sound_level()</pre>	soundlevel を略して, sl とした。
6.	ll = display.read_light_level()	lightlevel を略して, ll とした。
7.	if button_a.was_pressed():	もしaボタンがおされたら
8.	display.scroll("soundlevel is")	サウンドレベルは
9.	display.scroll(sl)	00
10.	display.scroll("lightlevel is")	ライトレベルは
11.	display.scroll(ll)	00

課題 9-3★★ a ボタンが押されたら,温度と騒音の数値を表示させる。(プログラムの画面の写真提出)

課題 9-4★★★ (カップラーメンタイマー制作)カップラーメンの上において, b ボタンを押したら, 無限ループを開始させ, 40 度以上になったら,数値を 83 (約 3 分)からカウントダウンさせ,0 になっ たら音を鳴らすプログラムを作る。ただ,40 度以上の環境や 83 (約 3 分)からのカウントダウンは待っ ていられないので,30 度以上になったら数値を 10 からカウントダウンし,0 になったら音がなるプログ ラムを作成してください。音がなったあとは break をしてください。(プログラムの画面の全行の写真提 出。途中でも部分点をあげます。)

ヒント1:temp = temperature() で温度を temp という変数に入れることができます。

※裏側の左の 7mm 程度の正方形のチップ (prosessor) が温度センサー。指で温めると数値あがっていき

### ます。

ヒント2: display.scroll(temp)とすれば、温度の数値を把握することができます。

ヒント3: import music を忘れずに書いておくこと。

#### 試験対策まとめ

今回, 試験対策まとめはありません。

# micro:bit NoIO ロボットの制作と試走

\_\_\_\_\_組\_\_\_\_\_番 氏名\_\_\_\_\_\_

1. モジュールのロード

左側の[Project]→[Open]から配布された haizen.py を選択。

$\odot$	micro:bit	(a)   <del>K</del>	2	Untitled project	Q Q	÷
	Project name Untitled project	L	1 2 3	# Imports go at the top from <b>microbit</b> import *	5	
	main.py	1	4	# Code in a lubile Truct lean repeats forewar	C	
IIN Reference			5 6 7 8 9	<pre>while True: display.show(Image.HEART) sleep(1000) display.scroll('Hello')</pre>		0 1 2 3V GHD
(+) Ideas			10			Show serial vertex :
ne api		:				shake v b v
	1)					δ ·
Project	Onen a hev or Puthon	(2) ther	files			Ø         ~           §         ~
ø	O Reset project					🔺 (A) (B) 🗸 🗸
0	Beta release	Feedback	¥	Send to micro:bit :	🖻 Open	

- 2. 動作チェックの簡単なプログラム
- a ボタンを押したら、まっすぐ走り、b ボタンを押したら、p0~p2 のセンサのアナログ値を表示させる。

	コード	解記
1.	# Imports go at the top	
2.	from microbit import *	
3.	from haizen import *	動きに必要なファイルを読み込ませます。
4.		
5.	motor = WheelMotor()	インスタンス化という。説明は後日。
6.		
7.	while True:	
8.	if button_a.was_pressed():	
9.	while True:	
10.	motor.set_speed(30,30)	第一引数が左車輪,第二引数が右車輪
11.	if button_b.was_pressed():	
12.	while True:	
13.	$sensor0 = pin0.read_analog()$	pin0 のアナログ値を読み取り sensor0 変数に代入

14.	<pre>sensor1 = pin1.read_analog()</pre>	
15.	<pre>sensor2 = pin2.read_analog()</pre>	
16.	display.scroll("p0=")	p0 と表示し,
17.	display.scroll(sensor0)	センサの値を表示させる。
18.	display.scroll("p1=")	
19.	display.scroll(sensor1)	
20.	display.scroll("p2=")	
21.	display.scroll(sensor2)	

動きを停めたり、リセットさせたい場合は、後ろのリセットボタンを押してください。

3. 組み立て

説明書をみて組み立ててください。組み立てたら micro:bit を指して、試走しましょう。チェックする ポイントは以下の通りです。

①まっすぐ走るかどうか。まっすぐ走らないのであれば,set\_speed(30,30)を調整しましょう。
 ②センサがうまく認識されているかどうか。黒にいる時と白にいるときを調べる。p1 が白で p2 が黒の時なども調べる。白にいるときと黒にいるときどちらが数値は高くなるか調べてください。

センサーチェック用黒線



#### 調べて書く

p0 黒	p1 黒	p2 黒

p0 白	p1 白	p2 白

p0 白	p1 白	p2 黒