

# micro:bit 「サンプル プログラミング集」 (第5.0版)

【Pythonプログラミング + microPython  
分冊版】

大分県立芸術文化短期大学  
情報コミュニケーション学科 非常勤講師  
望月陽一郎  
[mochizuki@oita-pjc.ac.jp](mailto:mochizuki@oita-pjc.ac.jp)

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)

# 目次

## 【Pythonプログラミング編】

・関数など

001 温度を表示する

002 明るさを表示する

003 ボタンで温度を表示する

004 ボタンで明るさを表示する

・条件 (if文)

006 暗くなるとLEDが点く

・音

007 Happy Birthdayを鳴らす

・複数の条件

013 コンパス (Compass)

・無線

014 2台で送受信

・傾き

015 水準器

016 傾きメーター

・乱数と繰り返し

044 ランダム数値表示

045 じゃんけん

## 【microPython編】

001 「HelloWorld」と「ハートマーク」を表示する

002 LED上に「O」を表示する

003 傾きを表示する

004 0から9までを表示する

# micro:bit 「サンプル プログラミング集」 Pythonプログラミング編

大分県立芸術文化短期大学  
情報コミュニケーション学科 非常勤講師  
望月陽一郎  
[mochizuki@oita-pjc.ac.jp](mailto:mochizuki@oita-pjc.ac.jp)

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)

# 目次

## 関数など

- 001 温度を表示する
- 002 明るさを表示する
- 003 ボタンで温度を表示する
- 004 ボタンで明るさを表示する

## 条件 (if文)

- 006 暗くなるとLEDが点く

## 音

- 007 Happy Birthdayを鳴らす

## 複数の条件

- 017 コンパス (Compass)

## 無線

- 018 2台で送受信

## 傾き

- 019 水準器
- 020 傾きメーター

## 乱数と繰り返し

- 052 ランダム数値表示
- 053 じゃんけん

# micro:bit プログラミングサイト

<https://makecode.microbit.org/#editor>

ブロック編集/Python画面  
の切り替え

ホーム画面へ

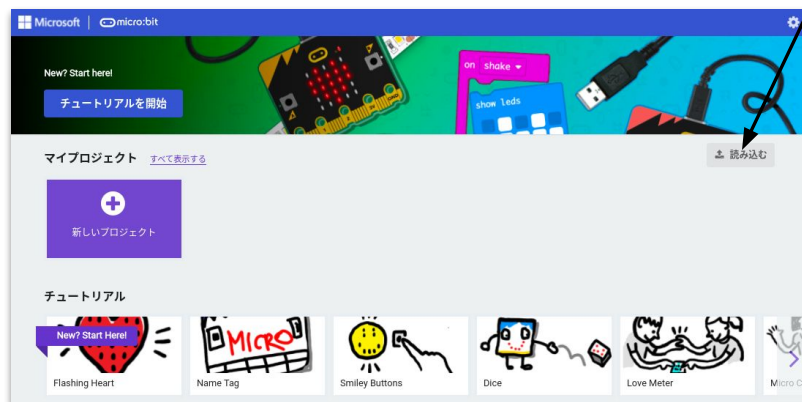


確認用シミュレータ

Python編集画面

作成したプログラムを  
USB接続した  
micro:bit にコピー

ローカル保存しておいた  
ファイル読込



## 【説明】

micro:bit プログラミングサイト

<https://makecode.microbit.org/#editor>

にアクセスしてプログラミングします。

- ・ Python画面でプログラミング。
- ・ 確認用シミュレータで確認する。
- ・ USB接続したmicro:bit にプログラムをコピー。
- ・ micro:bit 本体で動作させる。

( ・ micro:bitを接続せずにプログラムをダウンロードすることもできます。 )

※ダウンロード保存しておいたプログラムを読み込んで修正することもできます。

## 001 温度を表示する

### 【コード】

```
def on_forever():  
    basic.show_number(input.temperature())  
basic.forever(on_forever)
```

### 【解説】

- ・ ・ ・ def on\_forever(): ・ ・ ・ 関数「ずっと実行」  
 実行内容
- ・ ・ ・ basic.show\_number ・ ・ ・ 数値を表示
- ・ ・ ・ input.temperature() ・ ・ ・ 温度の値
  - ・ 明るさの値 input.light\_level()
  - ・ 磁力の値 input.magnetic\_force(Dimension.X)
- ・ ・ ・ basic.forever(on\_forever) ・ ・ ・ 関数on\_forever を呼び出し

## 002 明るさを表示する

### 【コード】

```
def on_forever():  
    led.plot_bar_graph(input.light_level(), 255)  
    basic.pause(1000)  
basic.forever(on_forever)
```

### 【解説】

- ・ ・ ・ led.plot\_bar\_graph ・ ・ ・ LEDにグラフを表示 255は最大値（変更可）
- ・ ・ ・ basic.pause() ・ ・ ・ 一時停止 1000は停止時間（ミリ秒）

### 003 ボタンで温度を表示する

#### 【コード】

```
def on_button_pressed_b():  
    basic.show_number(input.temperature())  
input.on_button_pressed(Button.B, on_button_pressed_b)
```

#### 【解説】

- ・ ・ ・ def on\_button\_pressed\_b():   ・ ・ ・ 関数「Bボタンを押すと実行」
- ・ ・ ・ input.on\_button\_pressed(Button.B, on\_button\_pressed\_b)
  
- ・ ・ ・ 呼び出し
  - ・ ・ ・ A+Bボタンは、Button.AB on\_button\_pressed\_ab

### 004 ボタンで明るさを表示する

#### 【コード】

```
def on_button_pressed_a():  
    led.plot_bar_graph(input.light_level(), 255)  
    basic.pause(1000)  
    basic.clear_screen()  
input.on_button_pressed(Button.A, on_button_pressed_a)
```

#### 【解説】

- ・ ・ ・ basic.clear\_screen()   ・ ・ ・ 画面を消去



## 006 暗くなるとLEDが点く

### 【コード】

```
def on_forever():
    if input.light_level() < 15:
        basic.show_leds("""
            #####
            #####
            #####
            #####
            #####
            """)
    elif input.light_level() < 30:
        basic.show_leds("""
            .....
            .###.
            .###.
            .###.
            .....
            """)
    else:
        basic.show_leds("""
            .....
            .....
            .....
            .....
            .....
            """)
basic.forever(on_forever)
```

### 【解説】

```
... if input.light_level() < 15:
    ... if 文 もし~ならば、
        elif input.light_level() < 30:
            ... 追加条件
        else:
            ... さもなくば
... basic.show_leds("""
    .....
    .###.
    .###.
    .###.
    .....
    """)
    ... LEDのオンオフ
    ... .消灯 #点灯
```

## 007 Happy Birthdayを鳴らす

### 【コード】

```
def on_button_pressed_a():
    music.play_tone(392,750)
    music.play_tone(392,music.beat(BeatFraction.QUARTER))
    music.play_tone(440,music.beat(BeatFraction.WHOLE))
    music.play_tone(392,music.beat(BeatFraction.WHOLE))
    music.play_tone(523,music.beat(BeatFraction.WHOLE))
    music.play_tone(494,music.beat(BeatFraction.DOUBLE))
    music.play_tone(392,750)
    music.play_tone(392,music.beat(BeatFraction.QUARTER))
    music.play_tone(440,music.beat(BeatFraction.WHOLE))
    music.play_tone(392,music.beat(BeatFraction.WHOLE))
    music.play_tone(587,music.beat(BeatFraction.WHOLE))
    music.play_tone(523,music.beat(BeatFraction.DOUBLE))
input.on_button_pressed(Button.A, on_button_pressed_a)
```

### 【解説】

- ・ ・ ・ music.play\_tone(440,music.beat(BeatFraction.WHOLE))
  - ・ ・ ・ 440 音の高さ (ラ)  
392 ソ 440 ラ 494 シ 523 上のド  
587 上のレ など
  - ・ ・ ・ WHOLE 音の長さ一拍 (4分音符)  
DOUBLE 二拍 (2分音符)  
QUARTER 1/4拍 (16分音符)  
HALF 1/2拍 (8分音符)  
750 数値 (付点4分音符)

## 013 コンパス (Compass)

### 【コード】

```
def on_forever():
    if 315 < input.compass_heading() or input.compass_heading() < 45:
        basic.show_string("N")
    elif 45 < input.compass_heading() and input.compass_heading() < 135:
        basic.show_string("E")
    elif 135 < input.compass_heading() and input.compass_heading() < 225:
        basic.show_string("S")
    elif 225 < input.compass_heading() and input.compass_heading() < 315:
        basic.show_string("W")
    else:
        basic.clear_screen()
basic.forever(on_forever)
```

### 【解説】

- ・ ・ ・ or ・ ・ ・ または
- ・ ・ ・ and ・ ・ ・ かつ
- ・ ・ ・ basic.show\_string() ・ ・ ・ 文字列の表示

## 014 2台で送受信

### 【コード°】

```
def on_received_number_deprecated(receivedNumber):
    if receivedNumber == 0:
        basic.show_string("Receive!")
        basic.pause(1000)
        basic.clear_screen()
radio.on_received_number_deprecated(on_received_number_deprecated)

def on_button_pressed_a():
    radio.send_number(0)
input.on_button_pressed(Button.A, on_button_pressed_a)

radio.set_group(1)
```

### 【解説】

- ・ ・ ・ radio.set\_group() ・ ・ ・ 無線グループの設定
- ・ ・ ・ radio.send\_number() ・ ・ ・ 数値を送信
- ・ ・ ・ radio.on\_received\_number\_deprecated  
(on\_received\_number\_deprecated) ・ ・ ・ 数値を受信したら
- ・ ・ ・ on\_received\_number\_deprecated(receivedNumber):  
　　・ ・ ・ 受信した数値の処理関数

## 015 水準器

### 【コード】

```
x = 2
y = 2
def on_forever():
    global x, y
    led.unplot(x, y)
    x = (-1 - input.acceleration(Dimension.X) + 1000) / 400
    y = (-1 - input.acceleration(Dimension.Y) + 1000) / 400
    led.plot(x, y)
basic.forever(on_forever)
```

### 【解説】

- ・・・ global グローバル変数の利用
- ・・・ led.unplot(x, y)
  - ・・・ LED 位置x,yの消灯 x 0~4 y 0~4
- ・・・ led.plot(x, y)
  - ・・・ LED 位置x,yの点灯 x 0~4 y 0~4
- ・・・ x = (-1 - input.acceleration(Dimension.X) + 1000) / 400
  - ・・・ X方向の位置変化 → x の座標 (0~4) に

## 016 傾きメーター

### 【コード】

```
y = 0
x = 2
def on_forever():
    global y, x
    y = 0
    led.unplot(x, y)
    x = (-1 - input.acceleration(Dimension.X) + 1000) / 400
    for index in range(5):
        led.plot(x, y)
        y += 1
    basic.pause(100)
    basic.clear_screen()
basic.forever(on_forever)
```

### 【解説】

- ・・・ for index in range(5):
  - ・・・ ループ (繰り返し)
  - ・・・ range(5) ・・・ 0~4 (5回)
- ・・・ y += 1 ・・・ yに1を足す (1増える)

#### 044 ランダム数値表示

【コード】

```
def on_button_pressed_a():
    basic.show_number(randint(0, 9))
input.on_button_pressed(Button.A, on_button_pressed_a)
```

#### 045 じゃんけん

【コード】

乱数 = 0

```
def on_button_pressed_a():
```

```
    global 乱数
```

```
    乱数 = randint(0, 2)
```

```
    basic.show_number(乱数)
```

```
    if 乱数 == 0:
```

```
        basic.show_leds("""
```

```
            # # # # .
```

```
            # # # # .
```

```
            # # # # #
```

```
            # # # # .
```

```
            . # # # .
```

```
            """)
```

```
    elif 乱数 == 1:
```

```
        basic.show_leds("""
```

```
            . # . # .
```

```
            . # # . .
```

```
            # # # # .
```

```
            # # # # .
```

```
            . # # # .
```

```
            """)
```

```
    else:
```

```
        basic.show_leds("""
```

```
            . . . . .
```

```
            # # # # .
```

```
            # # # # #
```

```
            # # # # .
```

```
            . # # # .
```

```
            """)
```

```
input.on_button_pressed(Button.A, on_button_pressed_a)
```

【解説】

・・・randint(0, 9) ・・・0~9の間で乱数（ランダムに）

【解説】

・・・乱数 = 0 ・・・変数に日本語が使える。

・・・if 乱数 == 0: ・・・== 等しい（= は右から左への代入）

## コードを打つ時に気をつけること

- ・ 縦の列をそろえます。

```
def OO():  
    basic. . . .  
    ↑
```

このタブは、「半角スペース4つ」。Tabキーは使いません。

- ・ 「,」の次は、見やすくするため半角スペースを入れます。
- ・ 「=」 「+」などの両側には、半角スペースを入れます。
- ・ 問題があるときは、画面の下に  
「Problems 数字」のように表示されます。

# micro:bit 「サンプル プログラミング集」 microPython編

大分県立芸術文化短期大学  
情報コミュニケーション学科 非常勤講師  
望月陽一郎  
[mochizuki@oita-pjc.ac.jp](mailto:mochizuki@oita-pjc.ac.jp)

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)



# 目次

001 「HelloWorld」と「ハートマーク」を表示する

002 LED上に「○」を表示する

003 傾きを表示する

004 0から9まで表示する

# micro:bit microPythonプログラミングサイト

<https://microbit.org/ja/code/>

micro:bit プログラムしよう アイデア はじめよう 教える 購入 日本語

## プログラムで想像力を発揮する

BBC micro:bit は、ブロック、JavaScript、Pythonを使ってプログラムできるって知ってましたか？  
もしBBC micro:bitを使ったことがなければ、試してみてくださいクイックスタートガイド（簡単にはじめる方法）。

### MakeCodeエディター

Microsoft提供のMakeCodeエディターを使え

エディターへのアクセスに問題があつたり、エディターをオフラインで使いたい場合は、FAQをご確認ください。

プログラムしよう  
リファレンス  
Lessons

### Pythonエディター

Pythonエディターは自分のプログラミングのスキルをさらに向上させたい人に最適です。選ばれたプログラム例とあらかじめ用意された様々な画像や音楽は、プログラミングの手助けになります。グローバルなPythonコミュニティの成果物によって動いています。Find out more about the latest updates in version 2 of the Python Editor.

プログラムしよう  
リファレンス

## オンラインエディター

ダウンロード on the App Store

micro:bitアプリでは、Bluetoothを使用してワイヤレスであなたのmicro:bitにプログラムを送り込みます。ケーブルは必要ありません！使用方法についての詳細については、このページを見てください。アプリケーション（ソフトウェア）。

## 【説明】

micro:bit プログラミングサイト

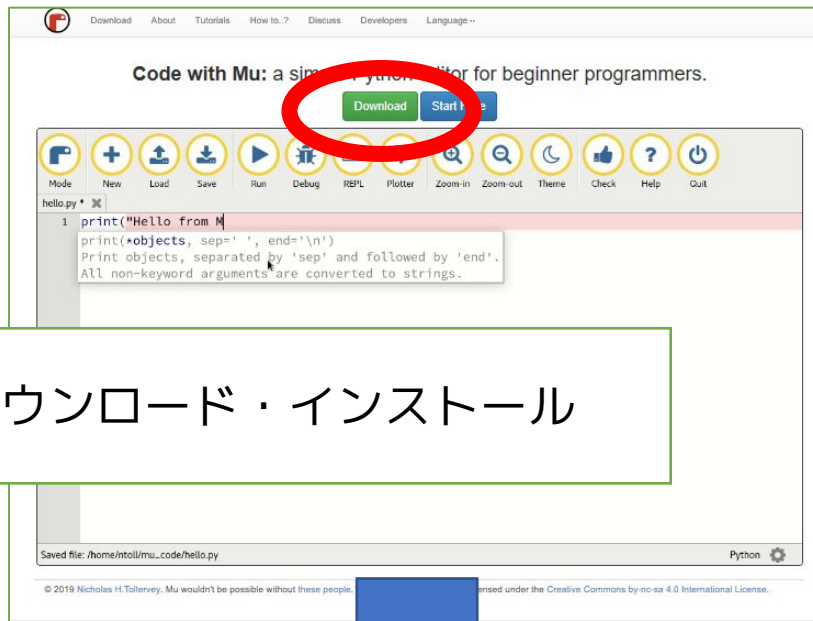
<https://microbit.org/ja/code/>

のPythonエディタで、オンラインプログラミングします。

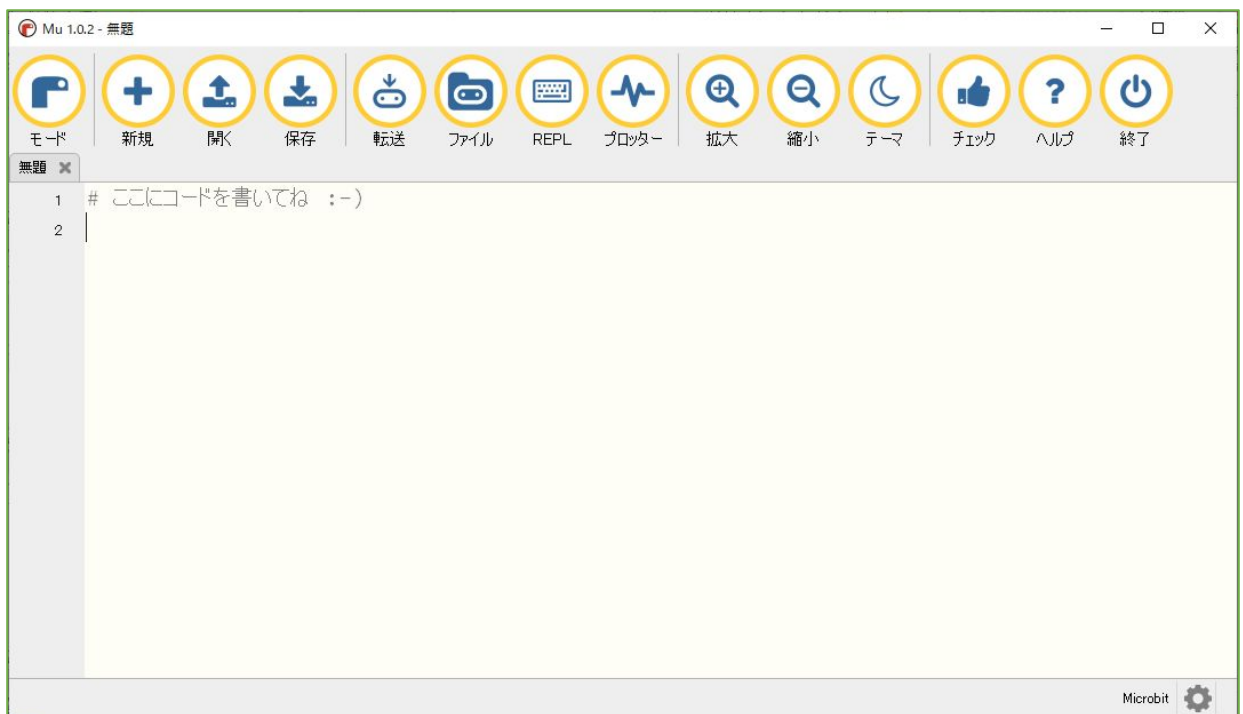
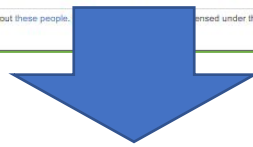
- ・ Download→プログラムをダウンロードしてmicro:bitへ送る
- ・ Load/Save→\*.py (Mu編集用ファイル)やプログラムを読み込み・保存できる

# micro:bit microPythonプログラミングツール Mu

<https://codewith.mu/>



ダウンロード・インストール



## 【説明】

micro:bit プログラミングツール Mu サイト

<https://codewith.mu/>

からダウンロード・インストールします。

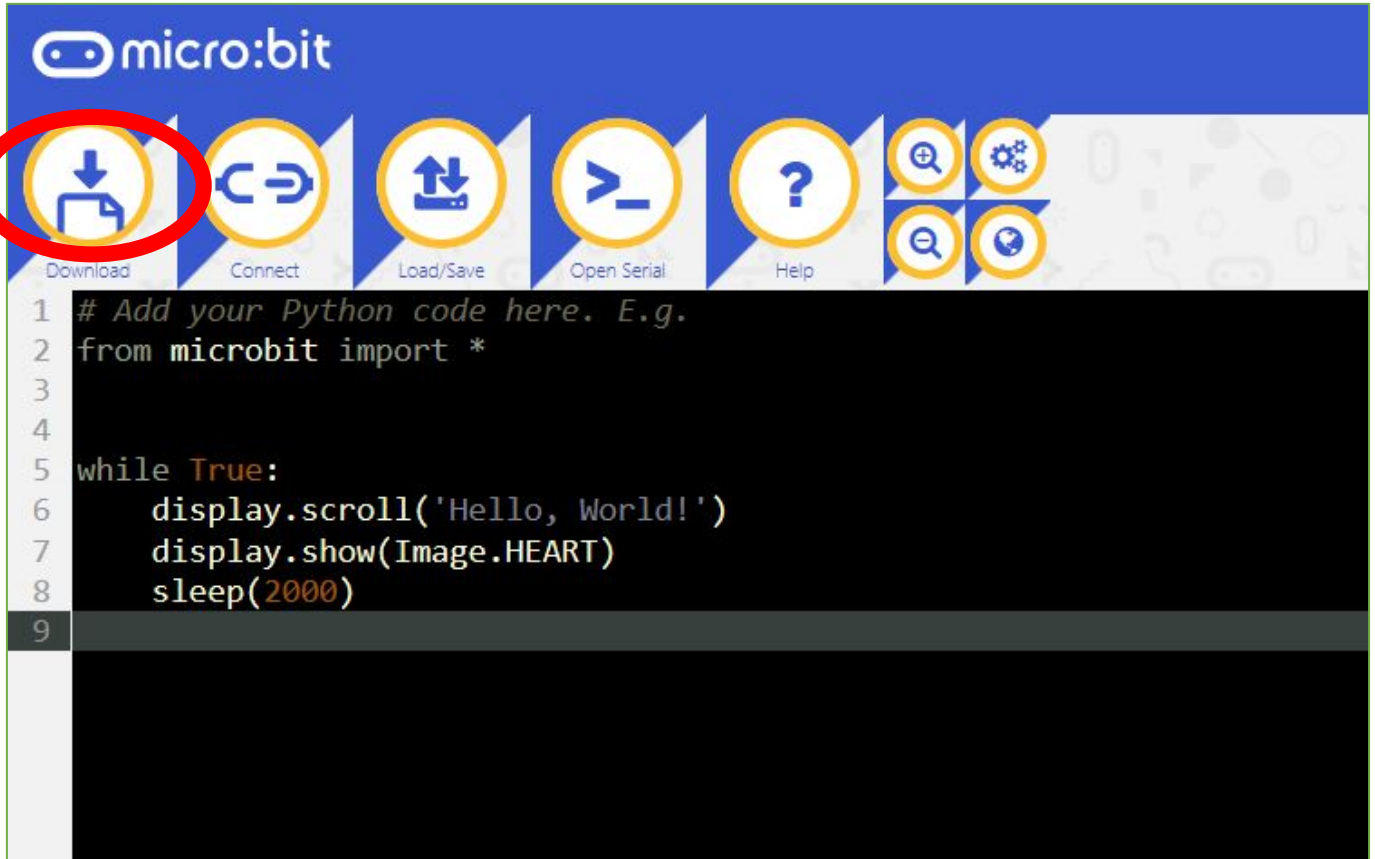
※オンラインエディターとの違い

- ・ 転送 → プログラムを接続されているmicro:bitへ
- ・ 保存 → \*.py (Mu編集用ファイル)

# 001 「HelloWorld」と「ハートマーク」を表示する

<https://python.microbit.org/v/2.0>

ボ	セ	L	出
通	入	オ	v2



# . . . プログラムに関係しないメモ  
from microbit import \* . . . micro:bit用モジュールの読み込み  
while True . . . ずっと繰り返す  
display.scroll(' ') . . . LEDに文字列を表示  
display.show(Image.HEART) . . . LEDに「ハート」を表示  
sleep(2000) . . . 2000ミリ秒=2秒停止

ダウンロード → micro:bit に転送

## 【説明】

# IMAGEの種類

ボ	セ	L	出
通	入	オ	v2

Image.HEART ハート  
Image.HEART\_SMALL 小ハート  
Image.HAPPY 笑顔  
Image.SMILE 笑った口  
Image.SAD 悲しい顔  
Image.CONFUSED 困り顔  
Image.ANGRY 怒った顔  
Image.ASLEEP 眠っている顔  
Image.SURPRISED びっくり顔  
Image.SILLY 変顔  
Image.FABULOUS 素晴らしい顔  
Image.YES はい (チェック)  
Image.NO いいえ (×)  
Image.CLOCK12, Image.CLOCK11,  
Image.CLOCK10, Image.CLOCK9,  
Image.CLOCK8, Image.CLOCK7,  
Image.CLOCK6, Image.CLOCK5,  
Image.CLOCK4, Image.CLOCK3,  
Image.CLOCK2, Image.CLOCK1  
○時 (短針を表示)  
Image.ARROW\_N, Image.ARROW\_NE,  
Image.ARROW\_E, Image.ARROW\_SE,  
Image.ARROW\_S, Image.ARROW\_SW,  
Image.ARROW\_W, Image.ARROW\_NW  
矢印表示 (上を北として)  
Image.TRIANGLE 三角  
Image.TRIANGLE\_LEFT 左三角  
(RIGHTはない)

Image.CHESSBOARD チェス盤  
Image.DIAMOND ダイヤ  
Image.DIAMOND\_SMALL 小ダイヤ  
Image.SQUARE 四角  
Image.SQUARE\_SMALL 小四角  
Image.RABBIT うさぎ  
Image.COW 牛  
Image.MUSIC\_CROTCHET 四分音符  
Image.MUSIC\_QUAVER 八分音符  
Image.MUSIC\_QUAVERS つないだ八分音符  
Image.PITCHFORK フォーク  
Image.XMAS クリスマス  
Image.PACMAN パックマン  
Image.TARGET ターゲット  
Image.TSHIRT Tシャツ  
Image.ROLLERSKATE ローラースケート  
Image.DUCK アヒル  
Image.HOUSE 家  
Image.TORTOISE カメ  
Image.BUTTERFLY チョウ  
Image.GHOST おばけ  
Image.SWORD 剣  
Image.GIRAFFE キリン  
Image.SKULL どくろ  
Image.UMBRELLA 傘  
Image.SNAKE ヘビ

文部科学省  
高等学校情報科「情報I」教員研修用教材より  
※オプション例は、2019/9 別途調べました。

## 002 LED上に「〇」を表示する

ボ	セ	L	出
通	入	才	v2

<https://python.microbit.org/v/2.0>



The screenshot shows the Microbit Python IDE interface. At the top, there is a blue header with the "micro:bit" logo and several icons for "Download", "Connect", "Load/Save", "Open Serial", "Help", and a 2x2 grid of icons for search, settings, and other functions. Below the header is a code editor with the following Python code:

```
1 from microbit import *
2
3 maru = Image("09990:"
4             "90009:"
5             "90009:"
6             "90009:"
7             "09990")
8 display.show(maru)
9 sleep(1000)
10 display.clear()
11
```

〇〇=Image("09990:" . . . 0がLED消灯、9が最も明るい  
. . .  
"09990:") . . . 変数として設定  
display.show(〇〇) . . . 変数を表示  
display.clear() . . . LED表示を消す



## 【説明】

002 LED上に「○」を表示する

・・・順次の例として、LEDで○を1秒間点灯したあと消灯するプログラムです。

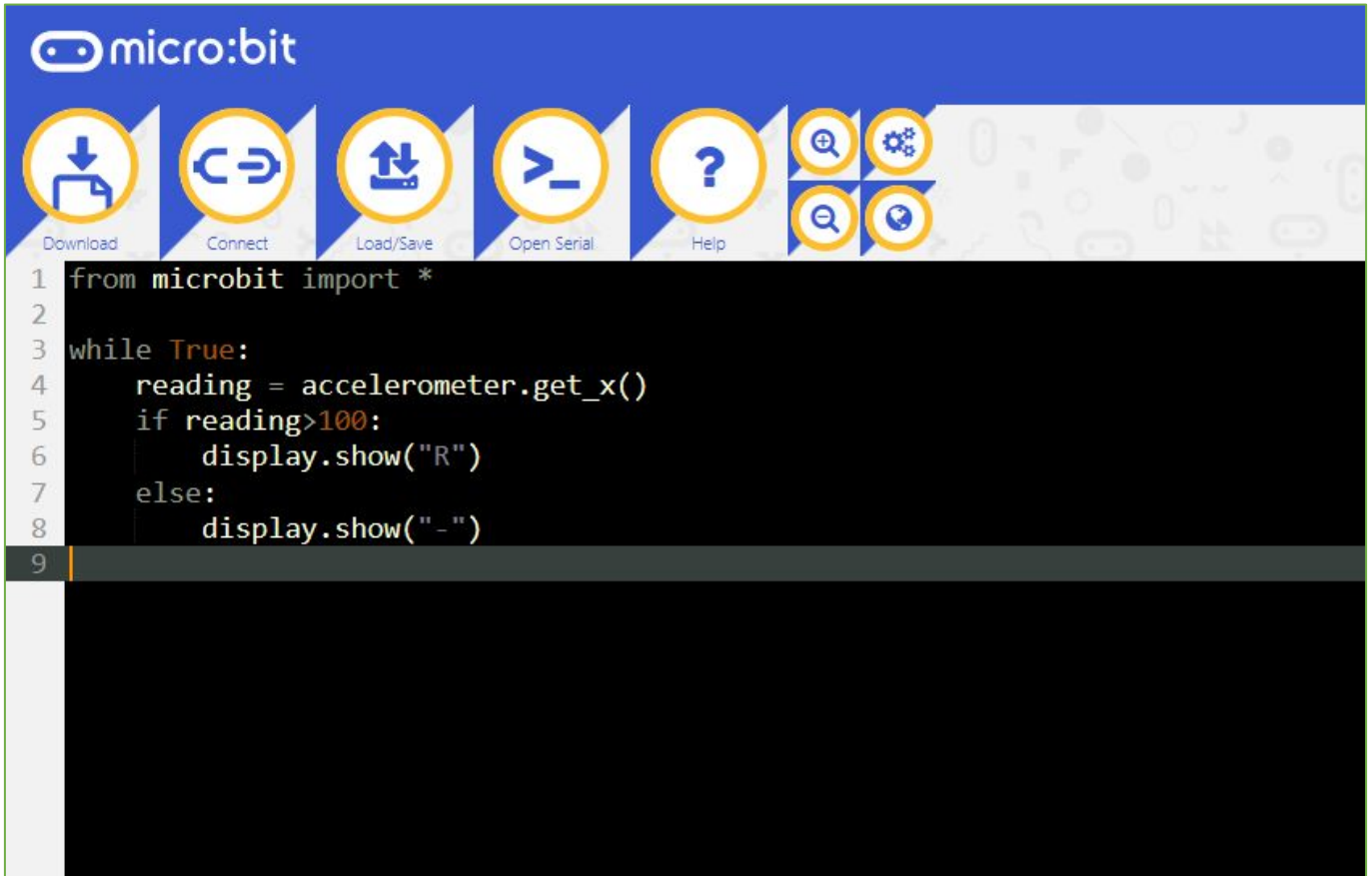
ポ	セ	L	出
通	入	オ	v2

文部科学省  
高等学校情報科「情報I」教員研修用教材より

## 003 傾きを表示する

<https://python.microbit.org/v/2.0>

ボ	セ	L	出
通	入	才	v2



```
1 from microbit import *
2
3 while True:
4     reading = accelerometer.get_x()
5     if reading > 100:
6         display.show("R")
7     else:
8         display.show("-")
9
```

reading=accelerometer.get\_x()  
... 変数に加速度センサーの値を入力  
... >100 右に傾き  
( <100 左に傾き =100 左右水平)

※accelerometer.get\_y()  
... >100 手前に傾き <100 向こうに傾き

## 【説明】

### 003 傾きを表示する

・・・ 分岐の例として、micro:bitの加速度センサが右に傾いているデータを検出すれば、LEDに「R」が点灯し、そうでなければLEDに「-」が点灯するプログラムです。

### ※温度センサーの例

```
1 from microbit import *
2
3 while True:
4     reading = temperature()
5     display.scroll(str(reading)+'C/')
6     sleep(1000)
7
```

ボ	セ	L	出
通	入	オ	v2

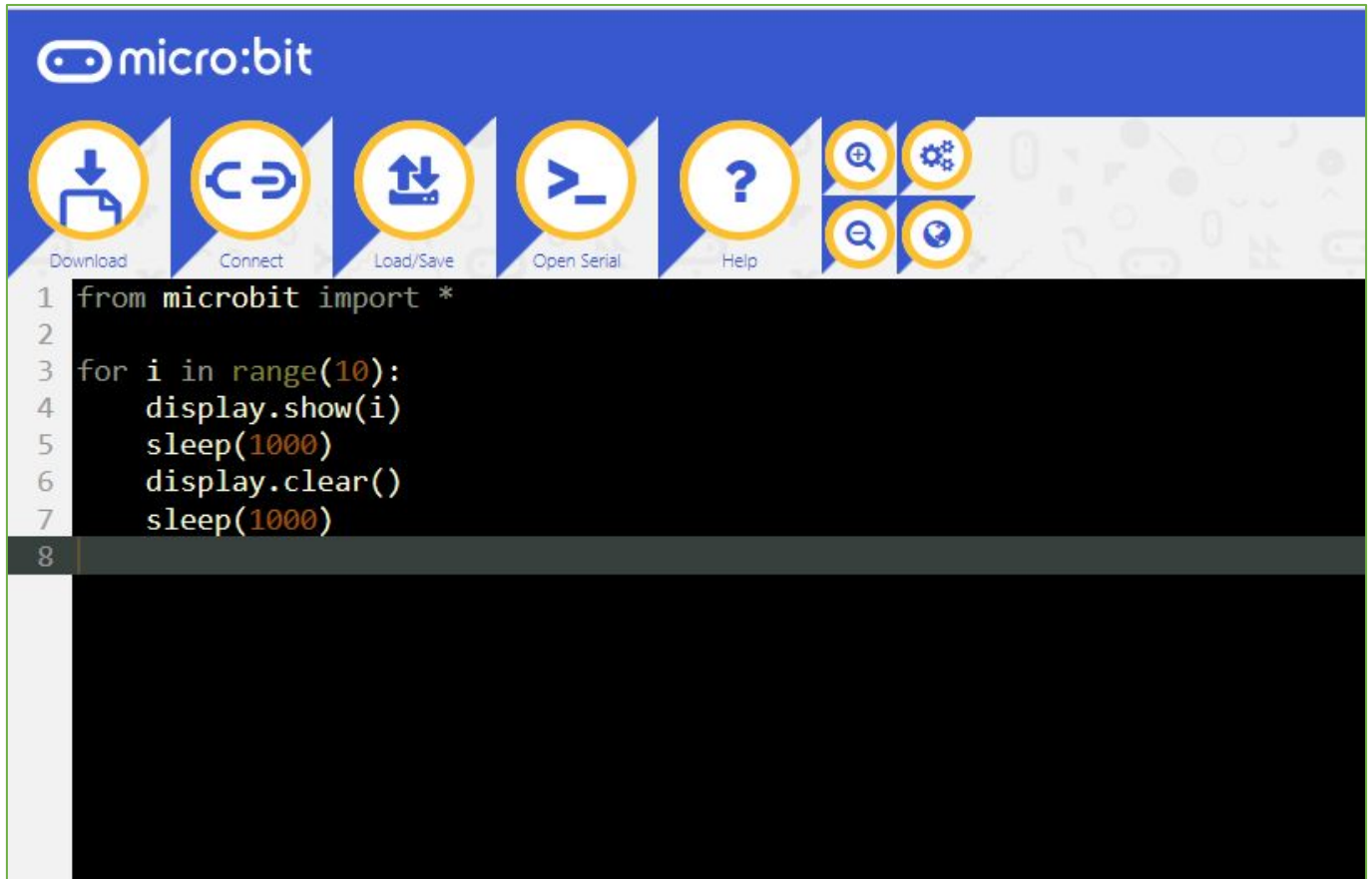
2020/1/13作成

文部科学省  
高等学校情報科「情報I」教員研修用教材より

## 004 0から9までを表示する

ボ	セ	L	出
通	入	才	v2

<https://python.microbit.org/v/2.0>



for i in range(10) . . . i を0~9までカウントしながら繰り返し  
display.show(i) . . . i に入っている値を表示

## 【説明】

004 0から9までを表示する

・・・反復の例として、0から9までの数字を順に表示するプログラムです。

ボ	セ	L	出
通	入	才	v2

文部科学省  
高等学校情報科「情報I」教員研修用教材より

# micro:bit 「サンプル プログラミング集」 (第5.0版)

【Pythonプログラミング + microPython  
分冊版】

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)

2018/10/19 第1.0版  
2018/10/24 修正  
2018/11/02 第1.1版  
2018/11/03 修正  
2019/03/12 第1.98版  
2019/05/05 第2.0版  
2019/11/12 第2.1版  
2019/12/30 第2.11版  
2020/01/13 第3.0版  
2020/08/25 第3.5版  
2020/08/30 修正  
2021/01/04 第4.0版  
2021/03/04 修正  
2022/01/01 修正  
2022/02/01 第5.0版

