micro:bit MicroPython cheatsheet

Imports

Import every class, function and variable
```
from microbit import *
```

Import only the display class
```
from microbit import display
```

Import the microbit library
```
import microbit
```

Buttons

Was a button pressed?
```
button_a.was_pressed()
```

Is a button currently pressed?
```
buttonb_.is_pressed()
```

Music

Play happy birthday
```
music.play(music.BIRTHDAY)
```

Create an array called **tune** of "NOTE OCTAVE:DURATION" then play it.
```
tune = ["C4:4", "E4:4", "G4:4",
        "C5:4", "G4:4","E4:4"]
music.play(tune)
```

Play a pitch [Frequency(Hz), Duration(ticks)]
```
music.pitch(440, 6)
```

Set the tempo
```
music.set_tempo(ticks=4, bpm=120)
```

Input/Output Pins

Is a pin currently being touched?
```
pin0.is_touched()
```

Return the current value on a pin
```
pin1.read_analog()
```

Write a value to a pin
```
pin2.write-digital(1)
```

LED Display

Scroll a string across the display
```
display.scroll('hello world')
```

Show an image on the display
```
display.show(Image.DUCK)
```

Return the light level from the display
```
display.read_light_level()
```

Sleep

Pause the program for a number of milliseconds (ms)
```
sleep(500)
```

Accelerometer

Gestures: **up, down, left, right, face up, face down, freefall, 3g, 6g, 8g, shake**

Was the micro:bit shaken?
```
accelerometer.was_gesture("shake")
```

Is the micro:bit currently falling?
```
accelerometer.is_gesture("freefall")
```

What is the value of the accelerometer x axis?
```
accelerometer.get_x()
```

Compass

Run the compass calibration routine
```
compass.calibrate()
```

What is the compass heading from 0 - 360 degrees?
```
compass.heading()
```

What is the field strength on the y axis in nano teslas?
```
compass.get_y()
```

Radio

Import the radio module
```
import radio
```

Turn the radio on or off
```
radio.on()
```

Send a string via radio
```
radio.send('duck')
```

Return whatever radio message was received
```
radio.receive()
```

Temperature

What is the current temperature?
```
temperature()
```

micro:bit MicroPython cheatsheet

Images

```
HEART, HEART_SMALL, HAPPY, SMILE, SAD,CONFUSED, ANGRY,ASLEEP,
SURPRISED, SILLY, FABULOUS, MEH, YES, NO, CLOCK12, CLOCK11, CLOCK10,
CLOCK9, CLOCK8, CLOCK7, CLOCK6, CLOCK5, CLOCK4, CLOCK3, CLOCK2, CLOCK1,
ARROW_N, ARROW_NE, ARROW_E, ARROW_SE,ARROW_S, ARROW_SW, ARROW_W,
ARROW_NW, TRIANGLE, TRIANGLE_LEFT, CHESSBOARD, DIAMOND, DIAMOND_SMALL,
SQUARE, SQUARE_SMALL, RABBIT, COW, MUSIC_CROTCHET, MUSIC_QUAVER,
MUSIC_QUAVERS, PITCHFORK, XMAS, PACMAN, TARGET, TSHIRT, ROLLERSKATE,
DUCK, HOUSE, TORTOISE, BUTTERFLY, STICKFIGURE, GHOST, SWORD, GIRAFFE,
SKULL, UMBRELLA, SNAKE
```

Music

```
DADADADUM, ENTERTAINER, PRELUDE, ODE, NYAN, RINGTONE, FUNK, BLUES,
BIRTHDAY, WEDDING, FUNERAL, PUNCHLINE, PYTHON, BADDY, CHASE, BA_DING,
WAWAWAWAA,JUMP_UP, JUMP_DOWN, POWER_UP, POWER_DOWN
```

Neopixels

Import the Neopixel module

```
import neopixel
```

Initialise a strip of Neopixels (pin, number of Neopixels)

```
neopixel.Neopixel(pin0, 10)
```

Send the current colour data to the Neopixels

```
neopixel.Neopixel.show()
```

Conditions

Play happy birthday

```
if accelerometer.was_gesture("left"):
     display.scroll('Left')
elif accelerometer.was_gesture("right"):
     display.scroll('Right')
else:
     display(clear)
```

Loops

Show a beating heart forever:

```
while True:
     display.show(Image.HEART)
     sleep(10)
     display.show(Image.HEART_SMALL)
     sleep(10)
```

Variables

Set the compass heading to a variable

```
direction = compass.heading()
```

Set the received radio message to a variable

```
incoming = radio.receive()
```