## Lesson 24 – Activity Sheet 1

## Getting Started – part two

This activity sheet will provide you with some example code and how to use functions in the program code. Functions are useful as they,

1. Enable you to keep the program code neat and tidy
2. Store several instruction and lines of code under one word. To run the function, you simply call it
3. Functions can be reused, which means you dint have to repeat lines of code
4. You can edit or update a function without having to edit the main section of the program

**Functions always go at the start of a program after the imports.** Let’s look at an example,

def cuddles(): #PET needs attention

display.show(Image.SAD)

#sleep(1000)

while True:

if pin0.is\_touched(): #touch 0 and GND

display.show(Image.HEART)

sleep(2000)

reset()

This function controls when the PET needs a cuddle. The function starts with the term **def** and then the name of the function, cuddles, (you can name the function any word you want to) then add the ( ): First it displays a sad face, to let you know it wants a cuddle, then a while True loop checks for the pin 0 being touched / stroked. (Remember to hold the GND pin at the same time). If you stroke the PET then it displays the hear image and then the program resets using the machine module.

The function can be changed simply by altering this code, for example

def cuddles(): #PET needs attention

display.show(Image.SAD)

#sleep(1000)

while True:

if pin0.is\_touched(): #touch 0 and GND

display.show(Image.HEART)

**PET says ‘thank you’**

**PET plays a tune**

sleep(2000)

reset()

## The Program

.The program basically asks the user to interact with the PET:

* A happy face is displayed, do nothing the PET is happy
* A sad face is displayed, the PET needs a cuddle, touch / stroke the 0 pin and GND pin
* A ‘asleep’ face is displayed, wake up your PET up by shaking it
* The house image is display, the PET wants to go for a walk, press button A and B together.

This program builds on the previous lesson code which uses the random module to select a random number between 1 and 50. If the number selected is ‘1’ then the micro:PET wants a cuddle and triggers the cuddles function.

If the random number is ‘2’, then the micro:PET is asleep and the wakeup function is run. Shake the micro:bit to wake the PET up.

If the random number is ‘3’, then the house image is displayed and the micro:PET wants to go for a walk, press button A and B together.

If none of theses numbers are selected then the program just displays a Happy face to show that the PET is happy.

The program code only selects the random numbers between 1 and 4 for testing purpose. Ensure that you increase the top number to say 100 or even 500, otherwise the PET will fall asleep every few seconds!

## The program code

# Add your Python code here. E.g.

from microbit import \*

import speech

import time

import random

def cuddles(): #PET needs attention

display.show(Image.SAD)

#sleep(1000)

while True:

if pin0.is\_touched(): #touch 0 and GND

display.show(Image.HEART)

sleep(2000)

reset()

def wakeup(): #Shake to wake up

display.show(Image.ASLEEP)

while True:

if accelerometer.was\_gesture("shake"):

display.scroll("Thanks")

reset()

def walk(): #press A and B

display.show(Image.HOUSE)

while True:

if button\_a.is\_pressed() and button\_b.is\_pressed():

display.show(Image.SMILE)

sleep(1000)

reset()

running = True

while True:

while running == True:

needs = random.randint(1, 4) **#change after testing**

sleep(1000)

if needs == 1:

cuddles()

running = False

sleep(1000)

elif needs == 2:

wakeup()

sleep(1000)

running = False

elif needs == 3:

walk()

sleep(1000)

running = False

else:

display.show(Image.HAPPY)

sleep(1000)

* Add additional features to the functions such as speech, animations or music. You could trigger the servo to control a movement on the PET.
* Add additional random interactions by inserting more ELIF statements. Create a function control what happens when this number is selected.
* Create a function for a complex set movements or interaction and then use it anywhere in the program by calling it.