

A virtual micro:pet

In the late 1990s
the “must have” toy
was a Tamagotchi

These were
electronic “pets”
that you had to
keep alive by
feeding and playing
with

They were huge



A virtual micro:pet

We can make a virtual pet using the micro:bit

This uses programming skills and ideas you've already learned, although it adds in one new idea

Decide what animal will be your pet

Display a welcome message at the start of your program

A virtual micro:pet

Use buttons A and B to **interact** with your pet

Use **button A** to feed your pet. When you press button A the pet should be happy – use a smiling image

(Image.HAPPY)

Use **button B** to play with your pet. When you press button B the pet should be silly – use a silly image (Image.SILLY)

But if you try and do **both** things at the same time that confuses your pet. When you press button A **and** button B the pet should be confused – use a confused image

(Image.CONFUSED)

At all other times the pet's image should show

A virtual micro:pet

Here's a basic framework for your program:

```
1  #Virtual pet program
2  from microbit import *
3
4  display.scroll("Hello")
5
6  while True:
7      #show your pet here
8
9      if button_a.is_pressed():
10         #what to do is A is pressed
11
12     elif button_b.is_pressed():
13         #what to do if B is pressed
14
15     elif button_a.is_pressed() and button_b.is_pressed():
16         #what to do is A and B are both pressed
17
```

A virtual micro:pet

Reminders:

- the “normal” thing for your pet to do is to show the image of the animal
- don't forget that you can use `sleep()` to slow things down
- a **while loop** will keep things running forever
- using **if** and **elif** will help deal with the things your pet needs to do
- be really careful about **indents** in your code

A virtual micro:pet

You can add in a section to do interact with your pet by shaking the micro:bit by using the accelerometer:

```
14
15     elif button_a.is_pressed() and button_b.is_pressed():
16         #what to do is A and B are both pressed
17
18     elif accelerometer.was_gesture("shake"):
19         #what to do if the microbit is shaken
20
```

You might also use other features of the accelerometer