

For this project you will need:

- A buddy OR a second microbit

To use the radio, you first need to **import the radio** functions. To do this type the following:

```
import radio
```

Now Python knows that you want to use the radio functions, but you still need to turn on the radio, and choose a group to belong to, so that you don't **interfere** with other micro:bits!

The default group is 0. With your buddy, decide on a different number, from **1-255**, then type:

```
radio.on()
radio.config(group=___) put your chosen group number here!
```

Now you are ready to send and receive messages!

To start with, you are going to tell the micro:bit to send a message when you press button A. Check the Inputs/Outputs worksheet if you can't remember how to do this.

To send a message, type the following code into your **if button A** statement:

```
radio.send("___") put your message here!
```

You now want to expand your program, so that as well as sending messages, it can receive them.

To expand your **if statement**, use an **elif**, which is short for else if. For example:

```
if button_a.was_pressed():
    do something if True; otherwise, try the next statement
elif button_b.was_pressed():
    do something if True; otherwise, try the next statement
else:
    do something else (if none of the above statements are True).
```

To receive a message, you need to type the following code into your **elif button B** statement:

```
message = radio.receive()
```

You have created a **variable** called message which will always be equal to the incoming message.

Underneath this, add the line `display.scroll(str(message))`, to display the new message.

Now **test your code** with your buddy – can you both send and receive messages?

You may be receiving the message **None**! This means that there is no message to receive – can you use an if statement to **only** display messages if the message is `not None`?

For this project you will need:

- A buddy OR a second microbit

To use the radio, you first need to **import the radio** functions. To do this, add this block from the radio menu: `import radio`

Now Python knows that you want to use the radio functions, but you still need to turn on the radio, and choose a group to belong to, so that you don't **interfere** with other micro:bits!

The default group is 0. With your buddy, decide on a different number, from **1-255**.

```
radio.on()
radio.config( group=___ )
```

put your chosen group number here!

Now you are ready to send and receive messages!

To start with, you are going to tell the micro:bit to send a message when you press button A. Check the Inputs/Outputs worksheet if you can't remember how to do this.

To send a message, connect the following block into your **if button A** statement:

```
radio.send(" ")
```

put your message here!

You now want to expand your program, so that as well as sending messages, it can receive them.

To expand your **if statement**, use an **elif**, which is short for else if. For example:

```
if button_a .was_pressed() :
    do something if True; otherwise, try the next statement
elif button_b .was_pressed() :
    do something if True; otherwise, try the next statement
else:
    do something else (if none of the above statements are True).
```

To receive a message, you need to add the following code into your **elif button B** statement:

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

You have created a **variable** which will always be equal to the incoming message.

Underneath this, add the line `display.scroll(incoming)` to display the new message.

Now **test your code** with your buddy – can you both send and receive messages?

You may be receiving the message **None**! This means that there is no message to receive – can you use an if statement to **only** display messages if the message `is not None`?