

# Hack the Dazzling Raspbertree

---



## What will you be doing in this activity?

---

In this activity, you will be hacking the LED Christmas tree using Python through a computer network. A computer network is a big set of computers that are connected through a wire (or through Wi-Fi nowadays). Computer networks are everywhere. Schools and businesses would have all their computers wired together, and even your house is just one big personal computer network.

## Opening and setting up Python

---



To open the Python 3 development environment (IDLE3) we will first need to click the terminal icon on the top bar (highlighted by the box in the picture above). Once terminal has opened type in `sudo idle3 &` and press enter. Python IDLE should now open.

**TIP : Be sure to create a new script for your code by going to File > New File**

# Networking with Python

For us to network with python we will need to import the `network` package. We will import this package using the `import` function. Like so...

```
import network
```

Then to talk to the tree, we will need to use network's `call` function to open a connection to the LED tree. The `call` function uses an IP address. Each device that's in a network will have its own unique IP address, even the tree! In our script, we'll store the tree's IP as a variable called `tree_ip`. Replace the `1.1.1.1` IP with the tree's current IP address.

```
tree_ip = "1.1.1.1"  
network.call(tree_ip)
```

## Talking to the Christmas Tree

The Pi tree has a special way of controlling the LEDs. Each LED has three colour values, one for red, one for green and one for blue. On top of that, we will also need an ID to reference each individual LED. Each colour value goes from 0 to 255 (0 meaning off and 255 meaning full brightness). Each value is put together into a string. The following diagram also explains this.

`network.say("[ ID , R , G , B ]")`

LED ID

Red Green Blue

from 0 to 255

To store the code, we will create a message variable called (you guessed it) **"message"** ! This example is to change LED number 1 to be a bright white colour. But why not pick a different LED to turn on, not just number 1?

```
message = "[1,255,255,255]"
```

Then, to send it to the tree, all you have to use is the `say` function which gives the IP, that you called at the start of the script using the `call` function, the code to changes the LEDs.

```
network.say(message)
```

Now save your code, and run it using **F5**. You should see the LED you chose, light up. If it does then your code works and you can move on to the next task, if it doesn't work then check your code and if it still doesn't work then grab a volunteer.

*So far, this is what you should have...*

```
import network
tree_ip = "1.1.1.1"
network.call(tree_ip)
message = "[1,255,255,255]"
network.say(message)
```

Now try changing other LEDs to different colours and play about with them (why not try changing some of the colour's values to zero as well?)

## Making the LEDs randomly twinkle

---

Now that you are a master at controlling the tree's LEDs, let's take is a step further. In this section, we will learn how to make an LED flash at random moments. First of all, make a new script by going to **File > New file**.

We'll start off by importing the `network` package again but we will also need to import two other libraries, them being the `random` package and the `sleep` function from the `time` package. We'll import them all using,

```
import network
import random
from time import sleep
```

And just like last time, we'll call the tree using the network's call function. Replace the `1.1.1.1` ip with the tree's current IP address.

```
tree_ip = "1.1.1.1"
network.call(tree_ip)
```

Now let's set the default colour for it to be before and after the twinkle as well as the twinkle colour.

```
message = "[1,100,175,85]"
twinkle_message = "[1,255,255,255]"
```

For us to make it randomly twinkle more than once, we will need to put it into a loop, in this case, we will put it into a `while` loop and set the condition to true. Everything that is tabbed in after the loop must be tabbed. **Indentation is extremely important with Python!** It uses it to figure out which code should be inside the loop and which should be before/after it.

```
while True:
```

To get a random time for the LED to flash after, we will use the random package's `randint` function to randomly choose a number between two numbers, the first one being the smallest value it could be and the second value being the largest value it can be. Let's set the values to 1 and 10.

```
random_time = random.randint(1, 10)
```

After using `randint` to get a random time we will need to make the LED twinkle. To do this we will set the LED to the default colour and sleep for the random time and make the LED flash a bright white. Then after, reset back to the default colour. We will do this by using the following code...

```
network.say(twinkle_message)
sleep(0.25)
network.say(message)
sleep(random_time)
```

Now you should have:

```
import network
import random
from time import sleep

tree_ip = "1.1.1.1"
network.call(tree_ip)
message = "[1,100,175,85]"
twinkle_message = "[1,255,255,255]"

while True:
    random_time = random.randint(1, 10)
    network.say(twinkle_message)
    sleep(0.25)
    network.say(message)
    sleep(random_time)
```

Now **save** and press **F5** to run the script and see your LED flash!

**Now Try** changing the random time values and running the script again!

## Controlling a Range of LEDs

Now let's try to control a range of LEDs at the same time with a small snippet of code.

First, we are going to need to open up a new file, so go to **File > New File** to create a new file.

For us to network using python we will need to import the **network** package again. We will import this package using the **import** function. Like so...

```
import network
```

Then we will continue by stating our Christmas tree and making the initial call to it with the following. *Remember to change the IP address!*

```
tree_ip = "1.1.1.1"  
network.call(tree_ip)
```

Now we are going to use the `range` loop function of python to send the Christmas tree the led number but with incrementing our LED number each time for the range of LEDs we want to control.

When we set the message, we state the colour here, as an example, we have set each colour to 255 to make the led white. You can change each of these values to your own colour in the order that was shown before. Then in `message` we have `x` refer to the number that is set by the range.

we are using `str` to turn what is a list of number to string of numbers and symbol.

```
for x in range(1, 4):  
    message = (x, 255, 255, 255)  
    message = str(message)  
    network.say(message)
```

This code is saying to control LEDs 1-3 with the colour combination that we stated in `message`. This is done by having the `network.say` repeat each time we increase the X number.

Now you should be able to run the code and control the multiple LEDs!

## What next?

---

Can you...

- Control 10 LEDs at once using an loop?
- Make an LED come up as a rainbow?
- Try having the colour of the twinkle become random every time a new twinkle happens.