

Hack the Dazzling Raspbertree : Extension - Rainbow LEDs



This worksheet requires you to of completed the worksheet '*Hack the Dazzling Raspbertree*' before attempting this extension worksheet.

Now that we have mastered the basics of using the network library to create single coloured lights on our Christmas tree, we can now try to create a rainbow by sending multiple colours to our tree lights!

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

Then to start we need to import all our packages that we are going to use. We will be using `Network` , `sleep` and `colorsys` . We are going to use sleep so we can pause the code for a split second and we are going to use `colorsys` to generate the beautiful rainbow of colours.

```
import colorsys
from time import sleep
import network
```

Now, we need to tell the code what led we want to change and what the tree's IP is. Then we need to open a connection with the Christmas tree so that we can send it colours.

```
led = 1
tree_ip = "1.1.1.1"
network.call(tree_ip)
```

Now we need to create an infinite loop so that we can run the code forever if we wanted. We use this loop to keep the code running until we stop it.

```
while True:
```

Now we need to make a for loop. First, we create a variable called `x`. A for loop runs the code inside it as long as the variable is not bigger than the second number, every time all the code has ran, it adds 1 to the variable. The first number is what the variable is equal to, and the second number is what the variable can not be bigger than. In the code below, the variable `x` starts off at 0 and it runs the code until it is 100..

```
for x in range(0, 100):
```

Now we are going to use the `colorsys` package to give us the red, green and blue colour values we need to make the rainbow. `colorsys` works by changing the `HSV` value (*That is a colour numbering format that is used for different shades of colour*) into the separate red green and blue values.

```
r, g, b = colorsys.hsv_to_rgb(x / 100.0, 1.0, 1.0)
```

Now we will have a value of R, G, B these are our red green and blue values. To use then we are going to first times the value by 100 to be able to use them in with the led. We are using `int` to turn the value into an integer, also known as a number.

```
r = int(r * 100)
g = int(g * 100)
b = int(b * 100)
```

How we need to sent the light values and the led number back to the tree. So we'll first form a `list` with the led value then we will add into the last the values for red, green

and blue. This `list` will be sent to the Christmas tree so it can work out what led it needs to change and the colour it should change it to. Then we use `str` to turn it into a string that the network package can accept.

```
message = (led, r, g, b)
message = str(message)
network.say(message)
```

After we need to `sleep` the code for a split second so the colour has time to show before we move on to the next colour.

```
sleep(0.05)
```

What the code we have written does is changes the Red, Green, Blue values between 0 - 100 between a range where x is 0 - 100 to produce a rainbow of colour. We are using the `sleep` to allow the colour to show for a split second so we are able to see the colour before moving on to the next. Our `while True:` keeps the code looping forever or till we use **CTRL** and **C** to force stop the code.

This is what you should have:

```
import colorsys
from time import sleep
import network

led = 1
tree_ip = "1.1.1.1"
network.call(tree_ip)

while True:
    for x in range(0, 100):
        r, g, b = colorsys.hsv_to_rgb(x / 100.0, 1.0, 1.0)
        r = int(r * 100)
        g = int(g * 100)
        b = int(b * 100)
        message = (led, r, g, b)
        message = str(message)
```

```
network.say(message)  
sleep(0.05)
```

So now we can save it and run to code to see if it works. If not call a volunteer over.