Saturday Kids

Level 2 Python Applications

Lesson 5 - Intro to NeoPixel

By the end of the lesson, students should be able to:

Intro to NeoPixel

- describe what is a neopixel
- use neopixel functions to light up and change the colours of the LEDs on a 16 LED NeoPixel ring
- recall and import libraries
- recall and use for loops to repeat a code for a specific number of times
- use tuples to store values
- recall and use variables to store data and change the value of variables
- recall and use while loops to create a forever loop
- use the sleep() function to slow down the program
- recall and use conditionals to determine if a variable is larger than a certain value
- practice looking up RGB colours to light up the NeoPixel in specific colours
- connect the NeoPixel to the microbit via the breakout board and female-male jumper cables and ensuring that the NeoPixel's GVS ports are attached correctly and securely to the GVS pins on the breakout board
- recall and use randint to return a random integer

Version

Date: January 2020

Format: 8 lessons x 2 hours

Important! View speaker notes for details

<8) =) ;D :)/>
CODE IN THE COMMUNITY

Things to note

- 🔌 **Unplugged** = Activities not involving technology (Videos, Kinaesthetic activities etc.)
- 💬 **Discussion** = Get the students to think and respond about a question
- 🎎 **Guided** = Demonstration → Instructor does the activity while the student mimics)
- 👐 **Unguided** = Instructor will give the students the task and show what the final result should look like and give the students a certain amount of time to do it by themselves before moving on to “Check for Understanding”
- 🎯 **Check for Understanding** = Instructor will go through the solution with them or get a student to share the solution
- 🚀 **Sandbox** = Free-Play (Students recap what they learnt from the entire day by creating a project)
- 🎁 **Bonus** = This is given to students who are fast-paced

Materials Needed

Per student:

- 1x microbit set
  - 1x microbit
  - 1x usb
  - 1x battery pack
  - 2x AAA batteries
- 1x breakout board set
  - 1x breakout board
  - 1x 16 LED NeoPixel ring
  - 3x female-male jumper cable
- 1x Chromebook/Laptop

Frequently Asked Questions

“What if I can't finish the activities for that particular day? ”

- In the event that you can't finish all of the activities in the given time, DO NOT rush to finish the concepts and just continue where you left off the next week.
- The bonus activities are for the faster students that have completed the general task that was given to the whole class. You do not need to cover this with everyone.

“How do you know you’ve been teaching the right way?”

- When students are able to create their sandbox with minimal to no help from you.

Frequently Asked Questions

“What is the purpose of this course?”

- For students to practice applying Python knowledge learnt previously on a micro-controller and build structures with Strawbees to present their creations to tasks given.
- Students also learn more basic coding concepts through Python and use them via computational thinking.

“What is computational thinking?”

- Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions. We can then present these solutions in a way that a computer, a human, or both, can understand.

Frequently Asked Questions

“Why must I follow the speaker notes and teach in a certain way? I prefer to freestyle.”

- For follow-up purposes as there will be cases where you might be unable to teach your class on a particular day and another instructor will need to cover you.

Frequently Asked Questions

“ Can students bring home the Strawbees structure?”

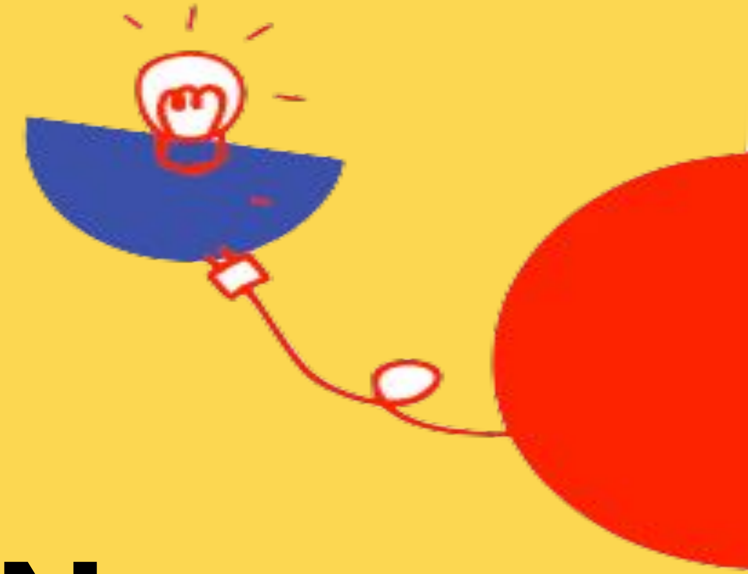
- No, but they can take pictures of their structures before dismantling them.

“ Can students bring home the microbit set?”

- No, but they can take videos of their projects/ pictures of their structures before dismantling them.

“Can students buy the Strawbees or microbit set?”

- No.



PYTHON

LEVEL 2

<8) =) ;D :)/>
CODE IN THE COMMUNITY

Attendance Taking

Please ensure that your attendance has been taken at the start of every lesson.

You will need to attain 80% attendance in order to graduate from this course.

Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	...
✔️	✔️	✔️	✔️	✔️	...

15 min

Recap & Agenda

Today's Lesson

- Introduction to NeoPixel
- Lighting up the NeoPixel Ring
- Light Show
- Rainbow Snake

45 min

NeoPixel

Unplugged

10 min

Introduction to NeoPixel

What are NeoPixels?

Unplugged

NeoPixels are multi-coloured programmable LEDs. They are made of RGB (Red, Green, Blue) color pixels and are available in a wide variety of shapes and sizes.

Unplugged

Observe your NeoPixel

How many LEDs are there on the NeoPixel?
What’s its shape?

16 LEDs

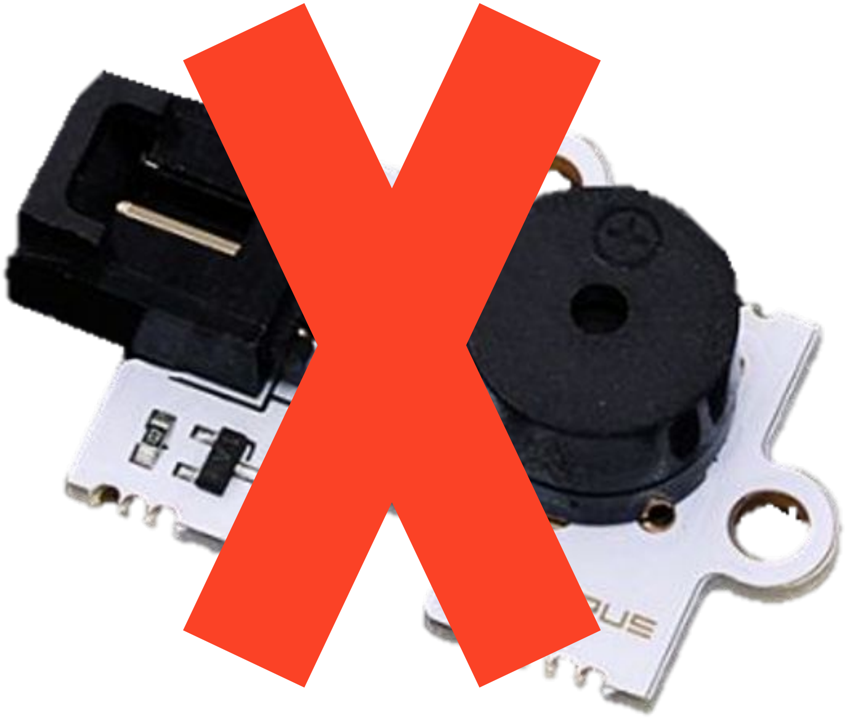
Circle (Ring)

Are there any constraints regarding the use of Neopixels?

Power!

Longer NeoPixels (more pixels) uses more power!

Unplugged

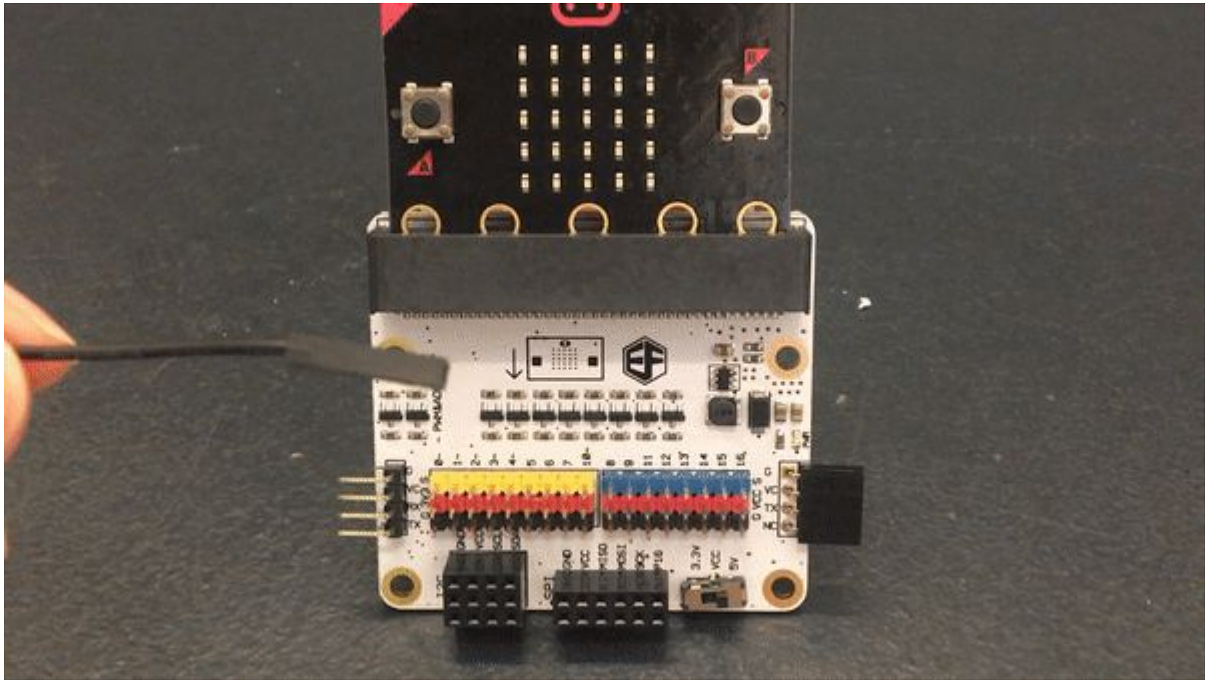


The buzzer cannot be played together with the NeoPixel at the same time, UNLESS powering up with an external power source!

🎎 Guided

Setting up the NeoPixel Ring

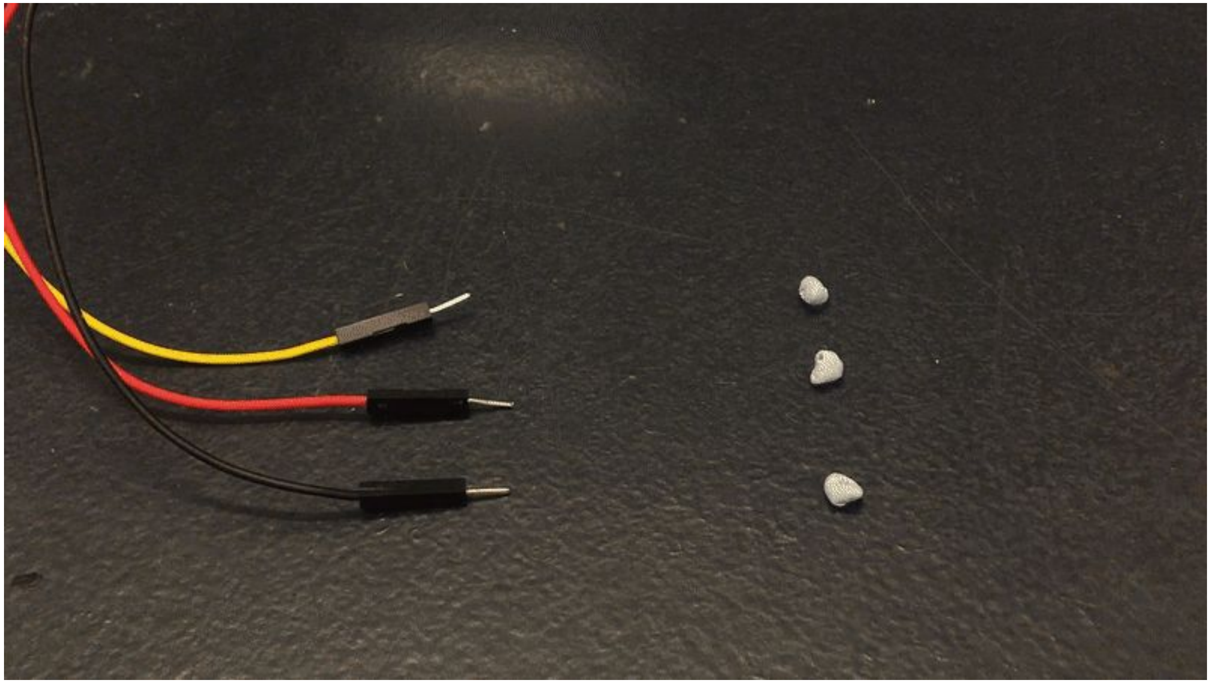
Connect the female end of the jumper cable to pin 0 of the breakout board.



🎎 Guided

Setting up the NeoPixel Ring

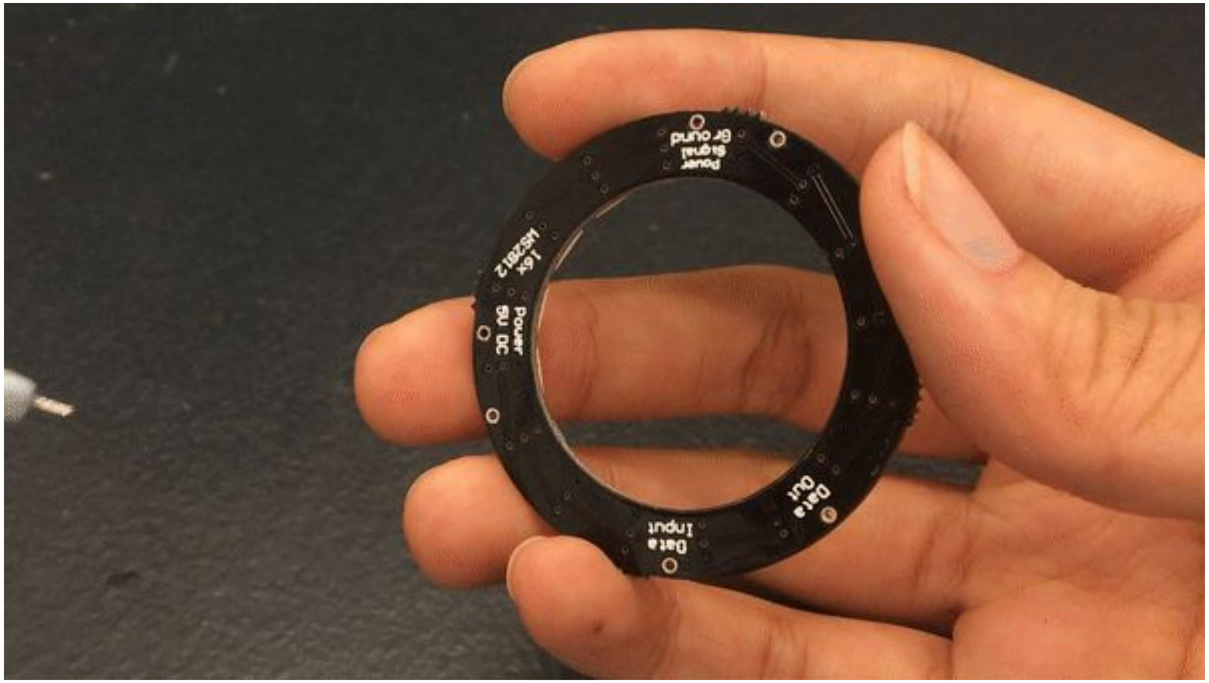
Secure the male end of the jumper cable by using blue tack.



Guided

Setting up the NeoPixel Ring

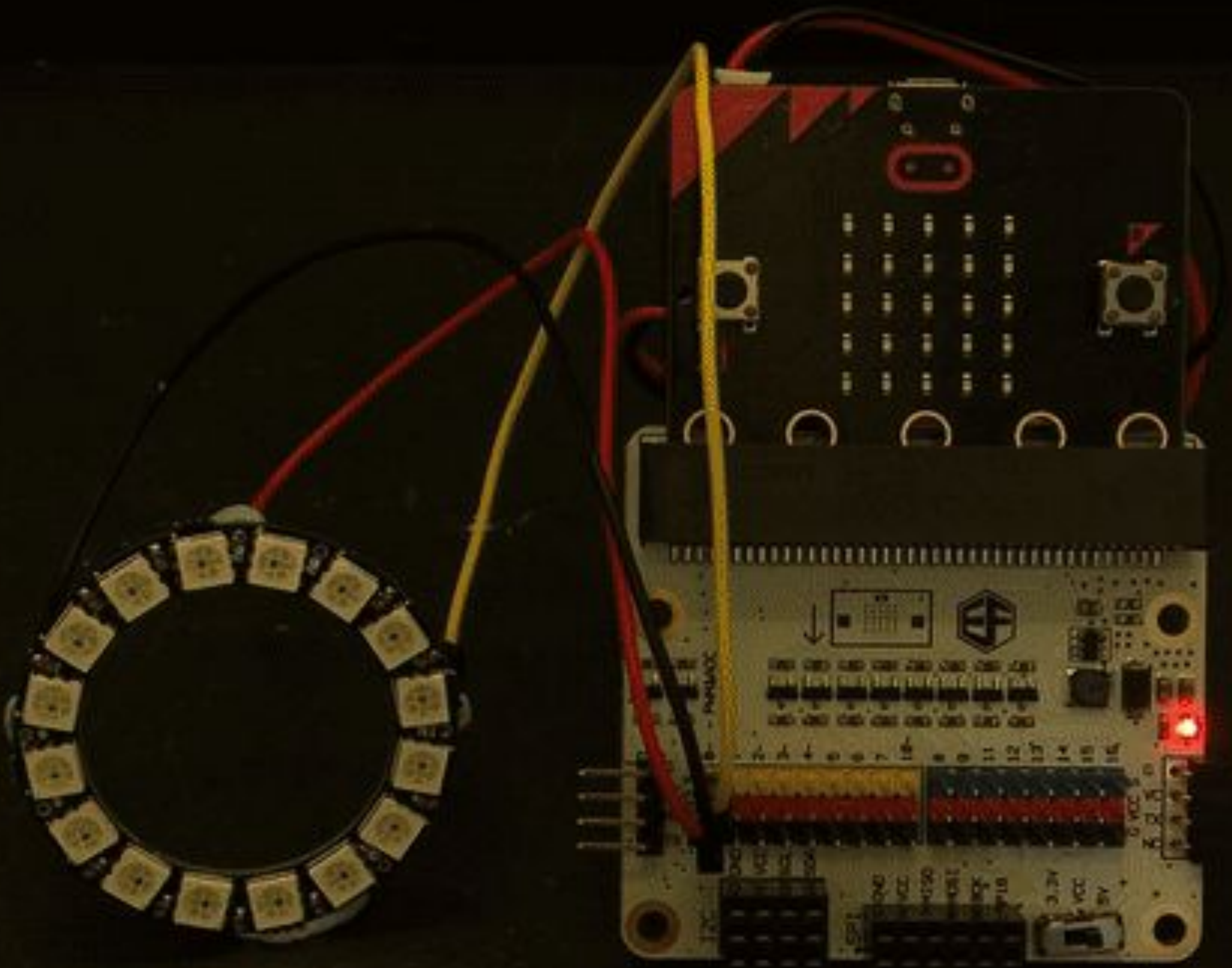
Connect the male end of the jumper cable to the NeoPixel.



🎎 Guided

15 min

Lighting up the NeoPixel Ring



🎎 Guided

Programme the micro:bit

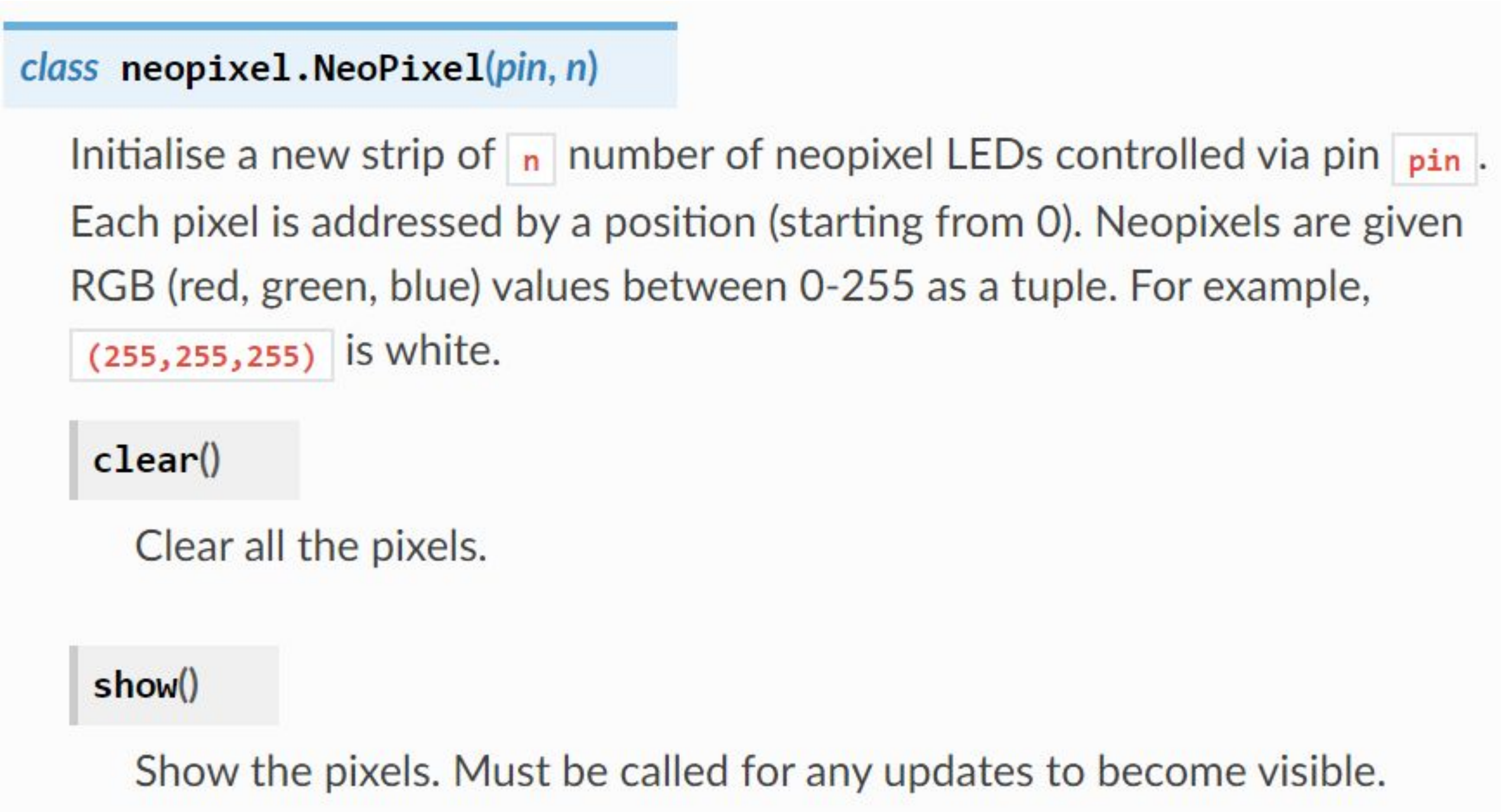
Turn on the NeoPixel’s LEDs one at a time

🎎 Guided

🔍 Find the Function to initialise the NeoPixel

Guided

Initialise the NeoPixel



class **neopixel.NeoPixel**(*pin*, *n*)

Initialise a new strip of `n` number of neopixel LEDs controlled via pin `pin`. Each pixel is addressed by a position (starting from 0). Neopixels are given RGB (red, green, blue) values between 0-255 as a tuple. For example, `(255,255,255)` is white.

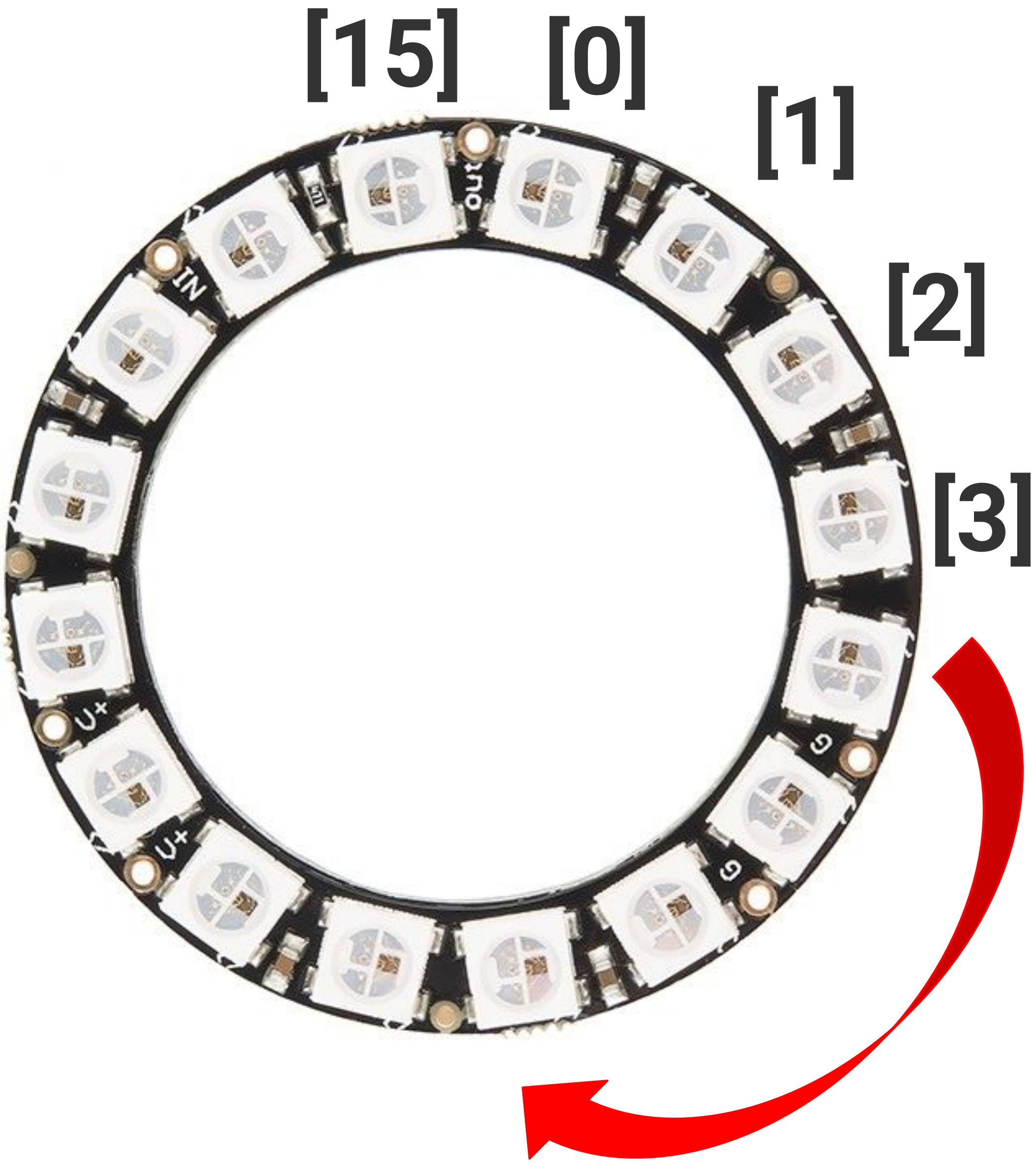
clear()

Clear all the pixels.

show()

Show the pixels. Must be called for any updates to become visible.

Unplugged



Each pixel is assigned a position starting from number 0

🎎 **Guided**

Tuple

A tuple containing 3 values (red,green,blue) is used to light up each pixel

(red, green, blue)

Each values range from 0 to 255

📥 Download the Project onto the micro:bit

Nothing Happens

Why?

🎎 Guided

🔍 Find the Function that allows us to show the colours on the NeoPixel

Guided

How to Correct the Code

```
np.show()
```

💾 Give your project a name and save it!

🙌 **Unguided**

20 min

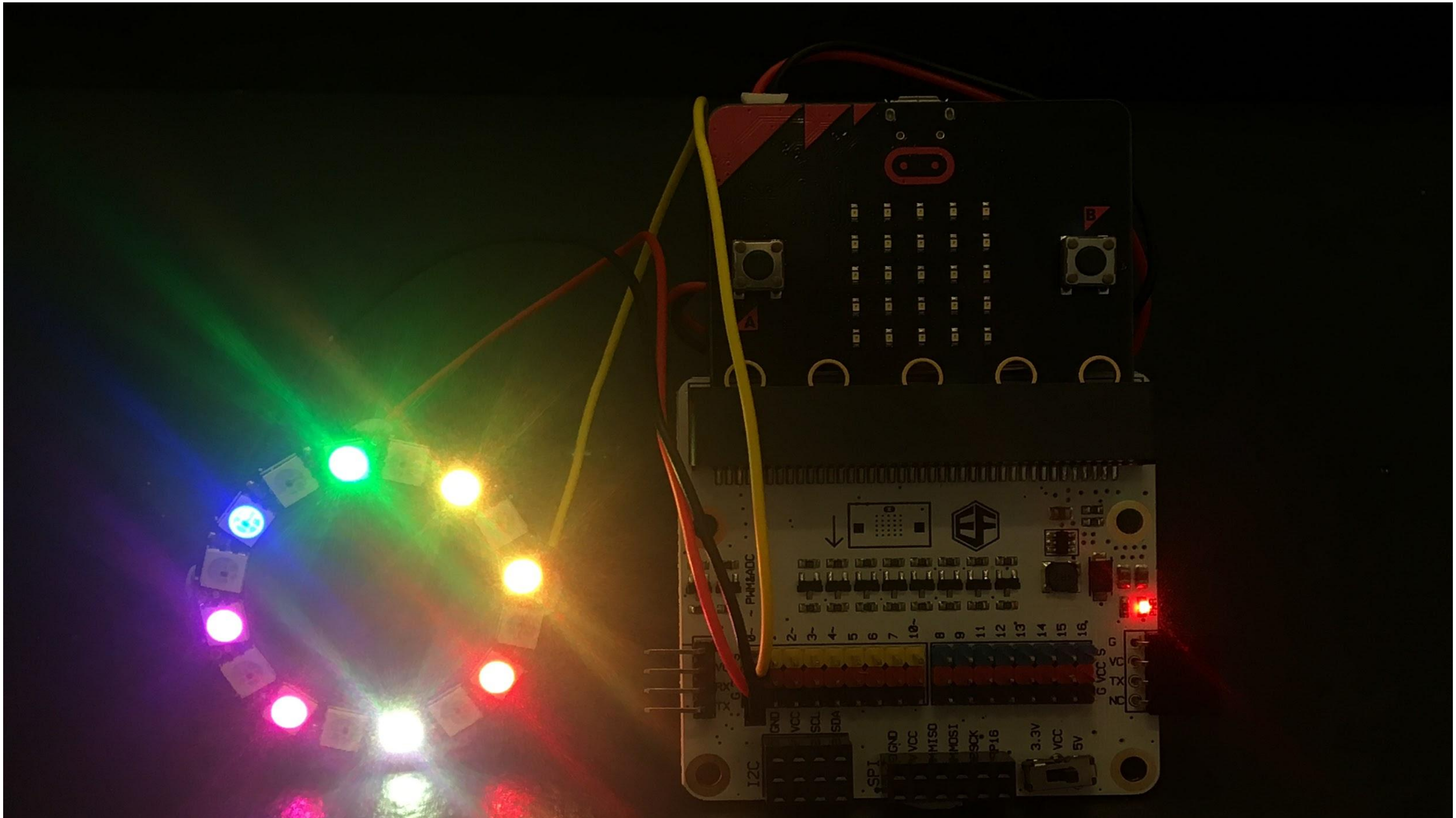
Light Show

👐 **Unguided**

Random + Specific Lights

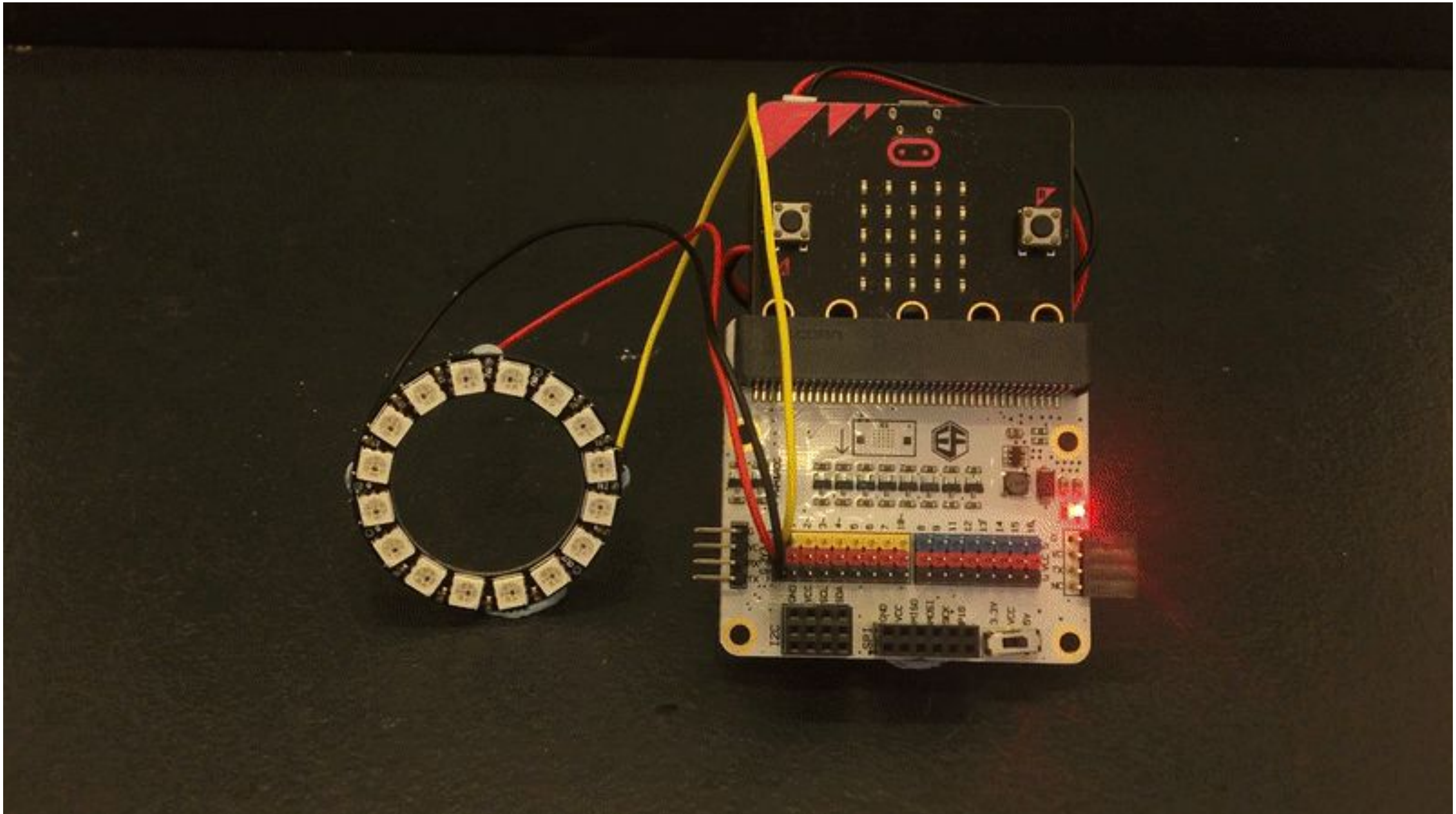
🙌 **Unguided**

Specific Lights



🙌 **Unguided**

Random Lights



🙌 **Unguided**

Programme the micro:bit

Light up each LED in EITHER (1) a random colour, (2) a specific colour OR (3) both random and specific colours

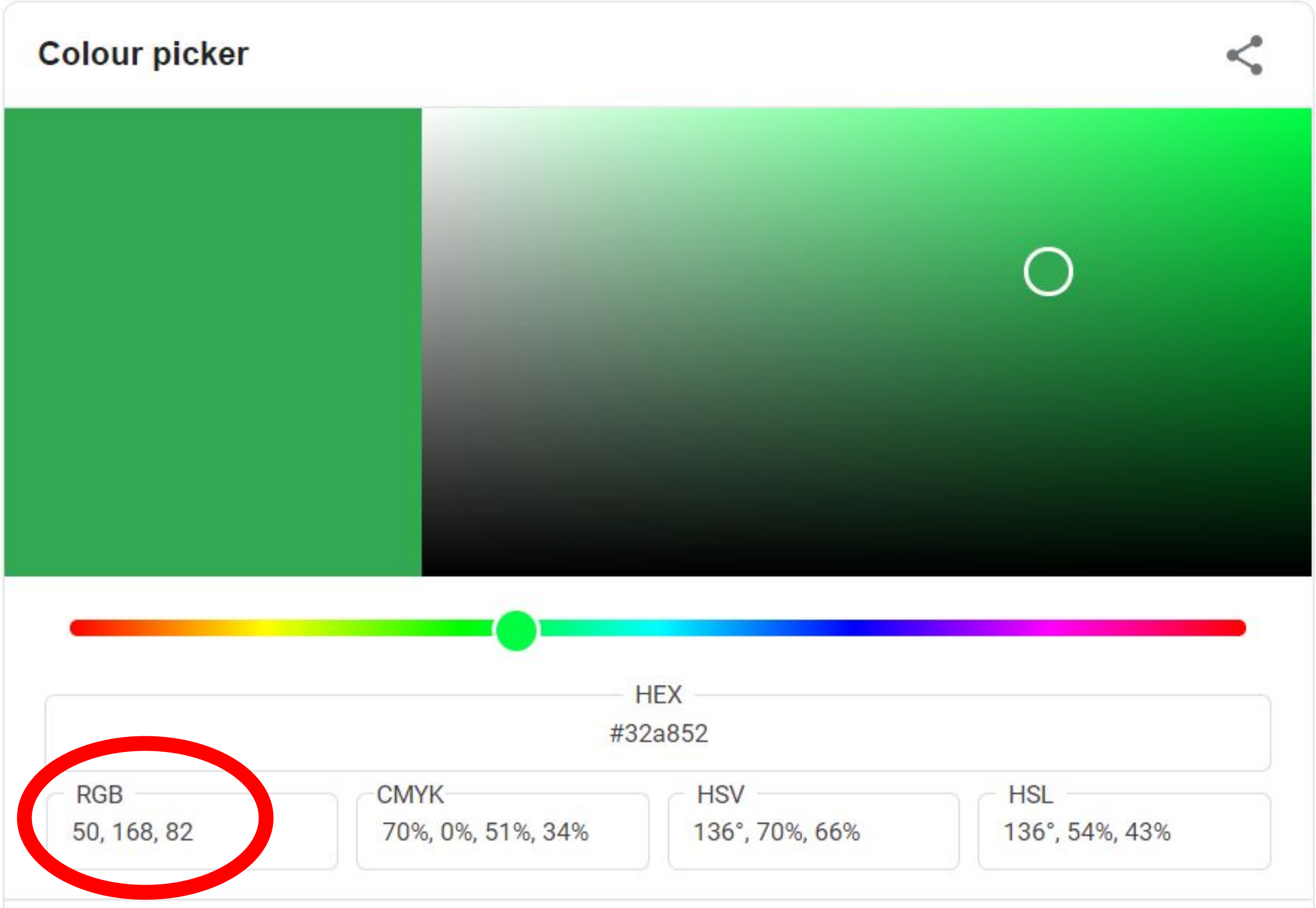
👐 Unguided

~~👀~~ Hints

Google “colour picker” for the RGB values of the different colours

Guided

RGB values of colours



Colour picker

HEX
#32a852

RGB
50, 168, 82

CMYK
70%, 0%, 51%, 34%

HSV
136°, 70%, 66%

HSL
136°, 54%, 43%

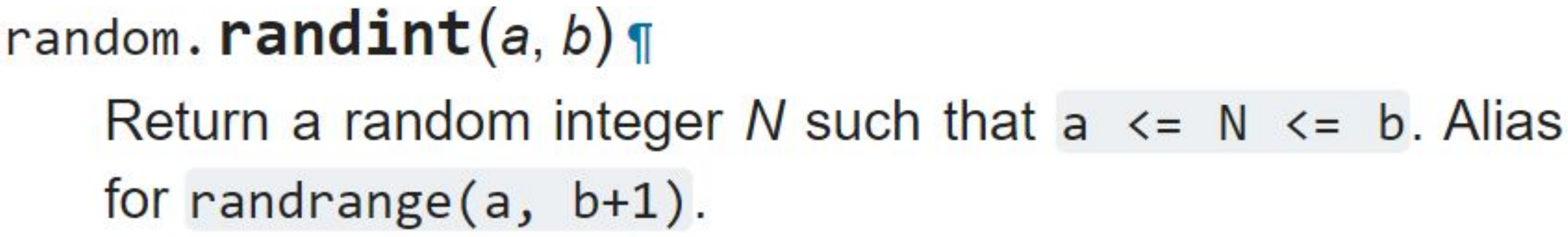
👐 **Unguided**

~~👀~~ Hints

Find the Function that generates a random integer

Guided

Generate a Random Integer



random.**randint**(*a*, *b*) ¶

Return a random integer *N* such that `a <= N <= b`. Alias for `randrange(a, b+1)`.

💾 Give your project a name and save it!

10 min

Break

Guided

30 min

Rainbow Snake

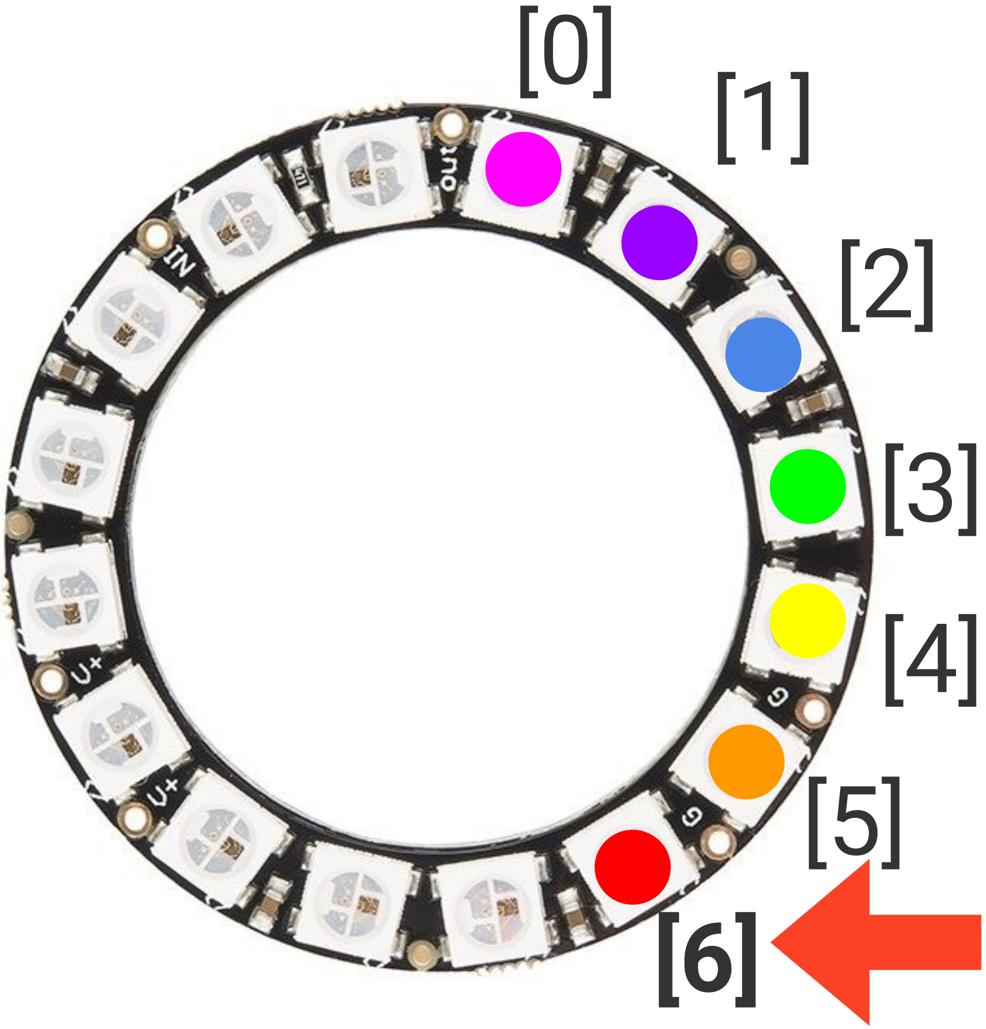
🎎 **Guided**

Programme the micro:bit

Light a rainbow snake running around the NeoPixel ring

Guided

Head and colours of the snake



The snake is made up of 7 colours.

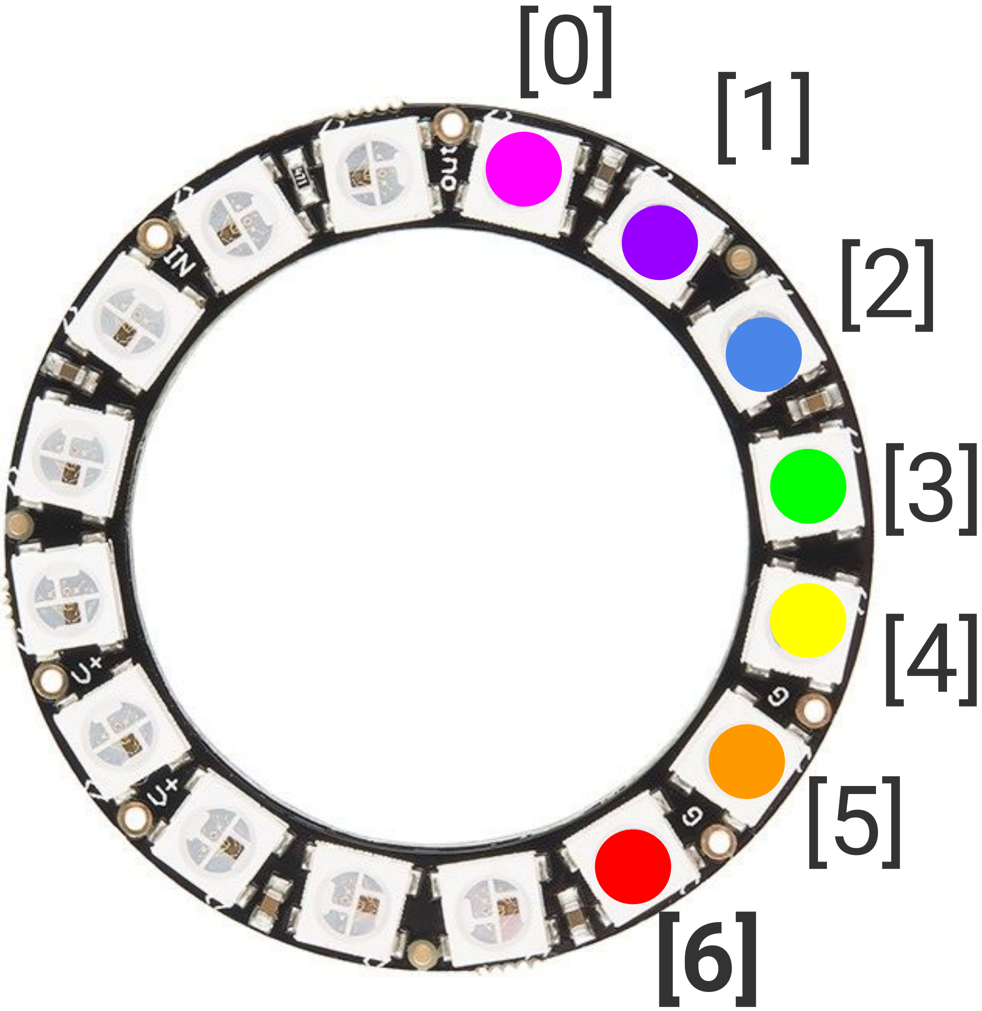
The head of the snake will be assigned to position 6

Guided

RGB values of the snake

- ● red (255,0,0)
- ● orange (255,137,0)
- ● yellow (255,150,0)
- ● green (0,255,0)
- ● blue (0,0,255)
- ● indigo (76,0,153)
- ● violet (153,0,76)

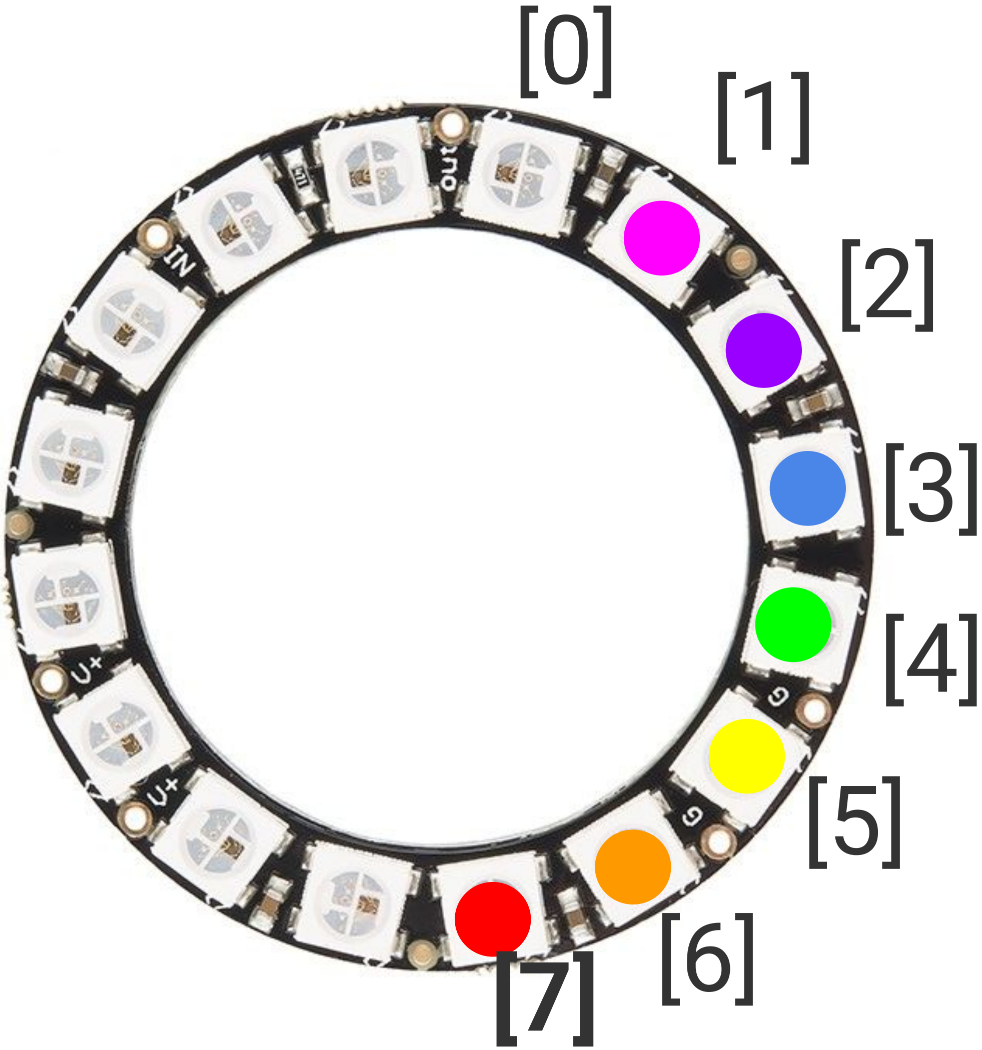
Colours of the snake



As the snake moves, the position of the LEDs will change

Guided

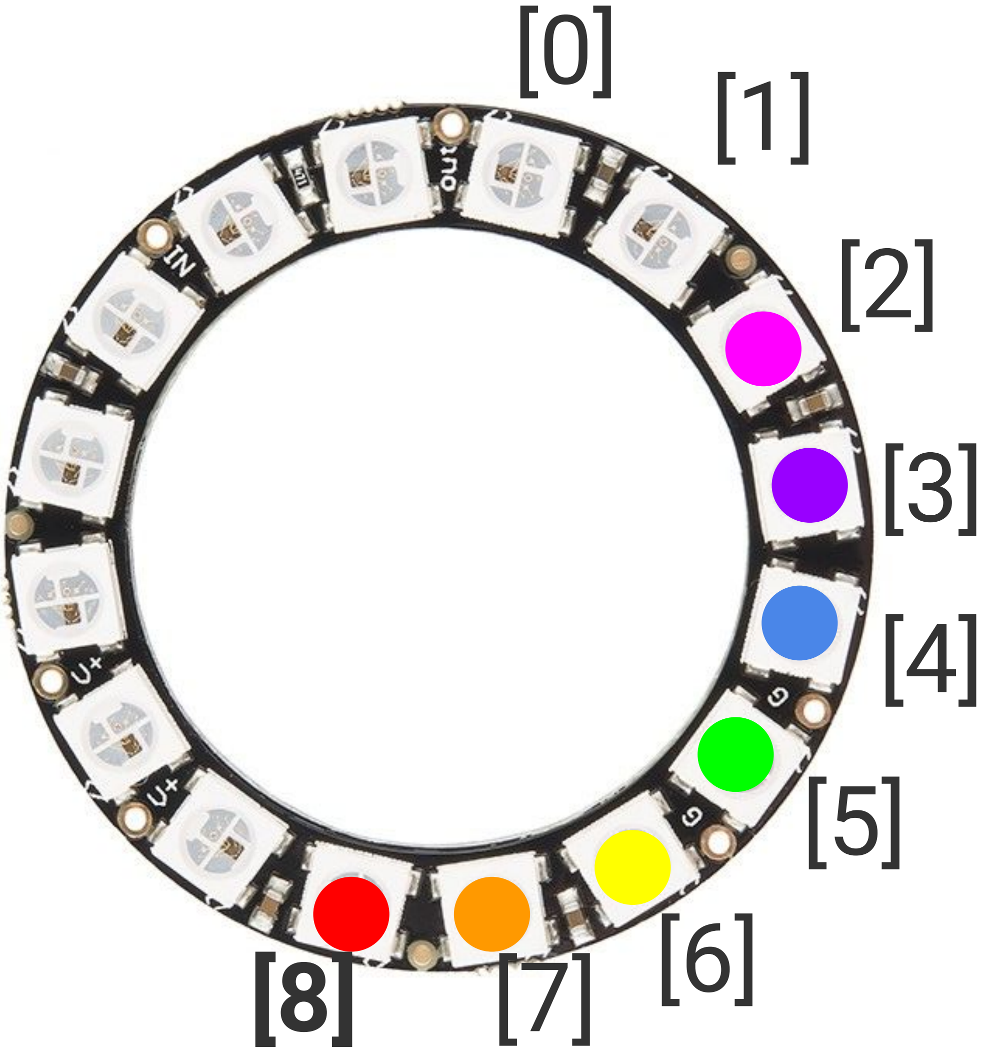
Colours of the snake



As the snake moves, the position of the LEDs will change

🎎 Guided

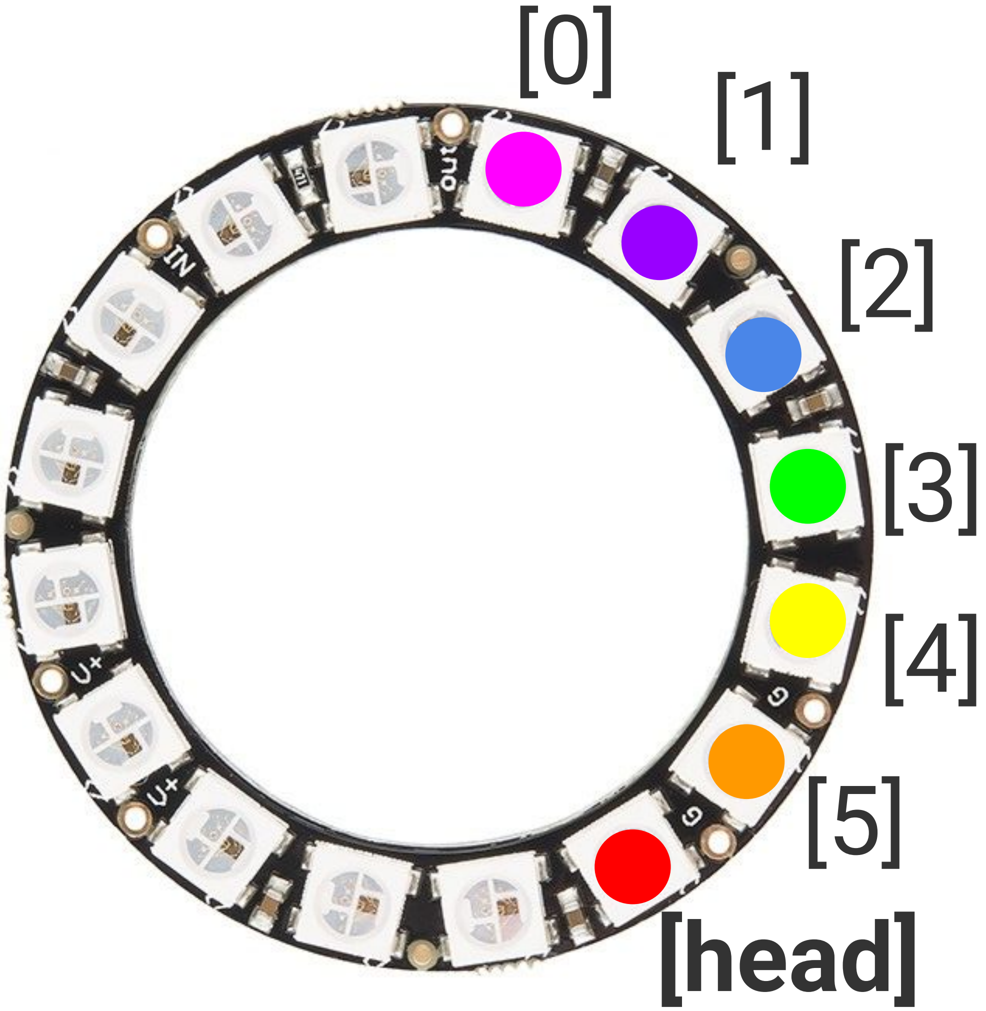
Colours of the snake



As the snake moves, the position of the LEDs will change

🎎 Guided

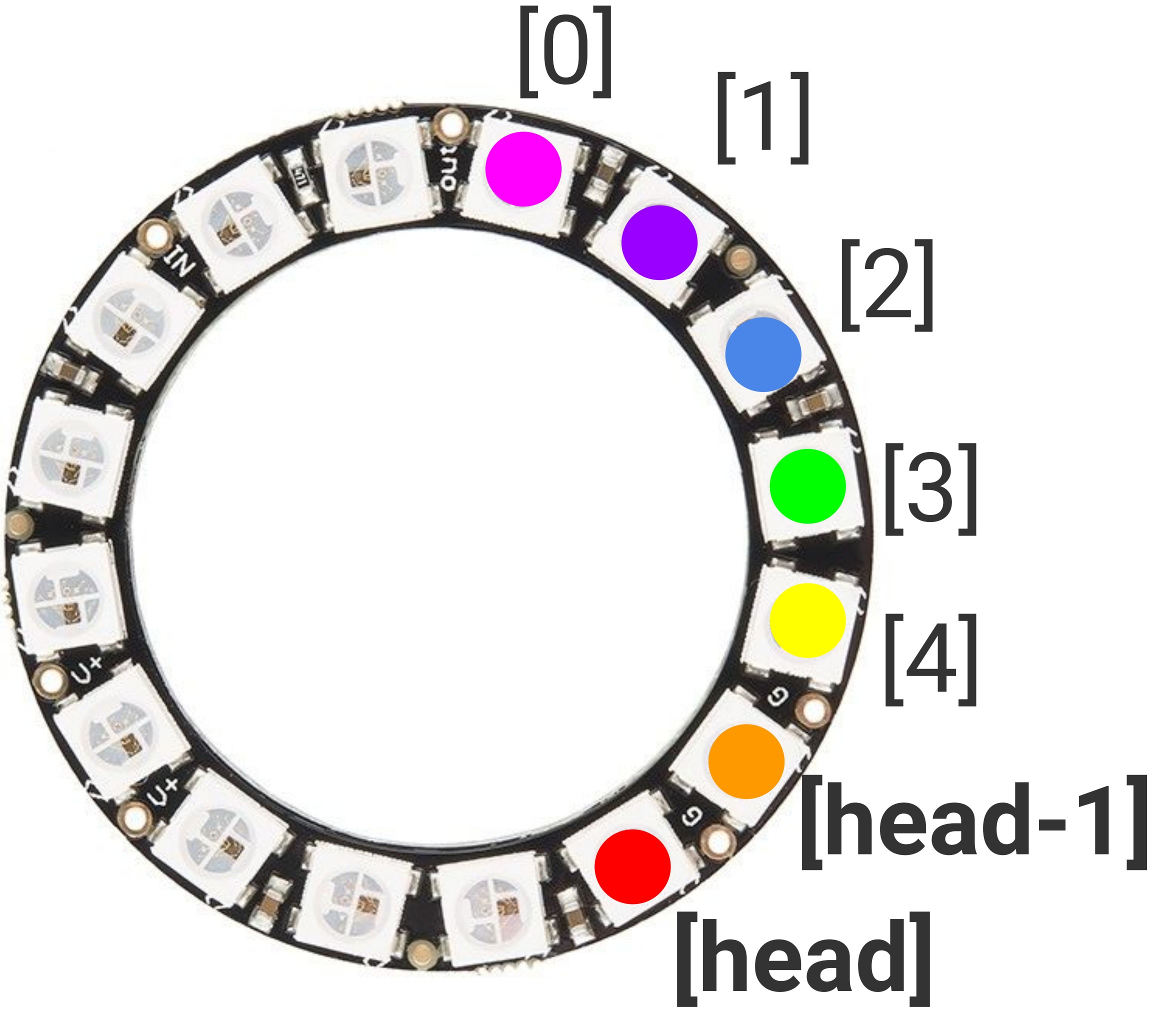
Colours of the snake



Hence, assign a variable to indicate the various LED parts of the snake!

🎎 Guided

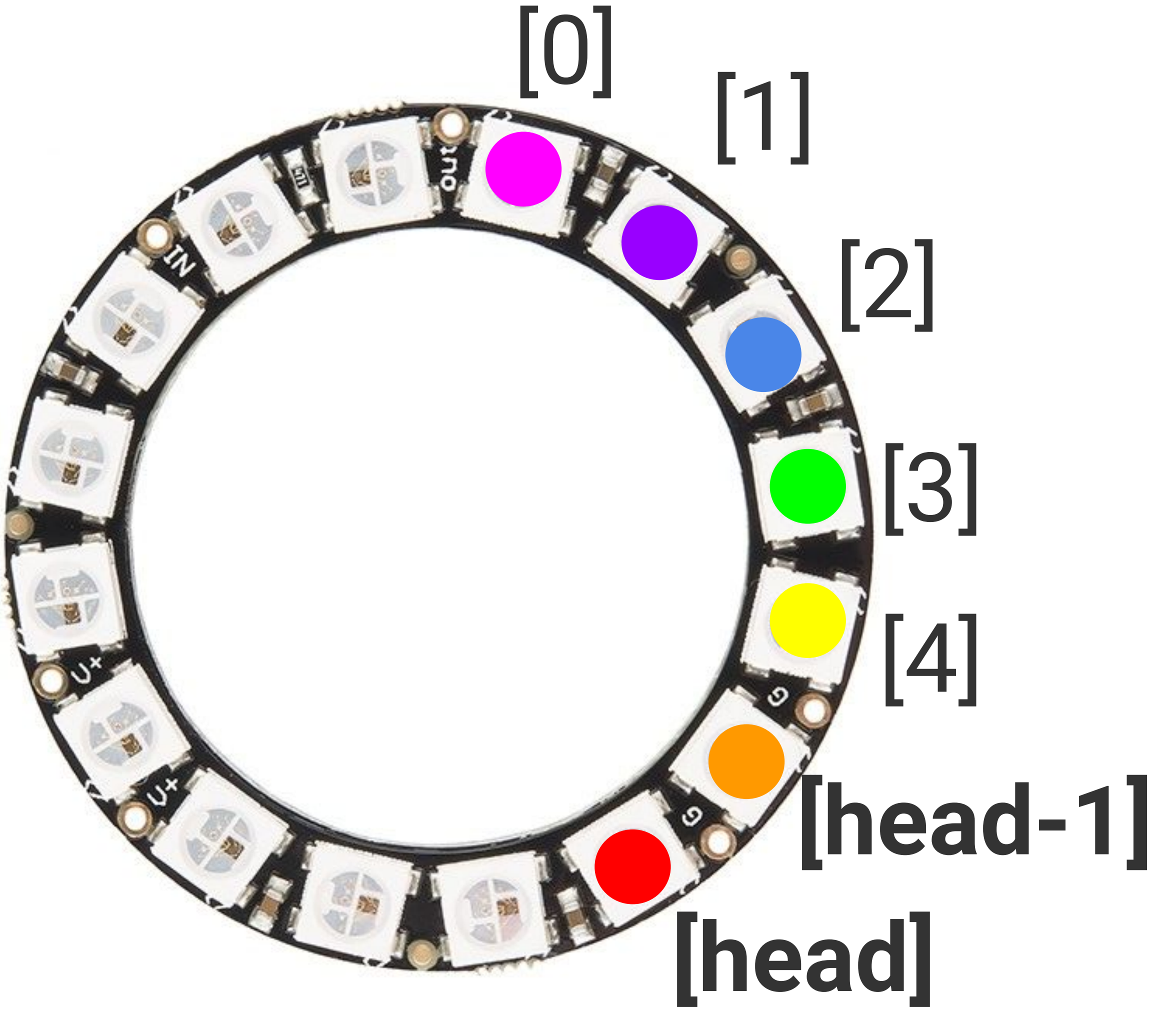
Colours of the snake



The neck of the snake is just one LED away from the head of the snake

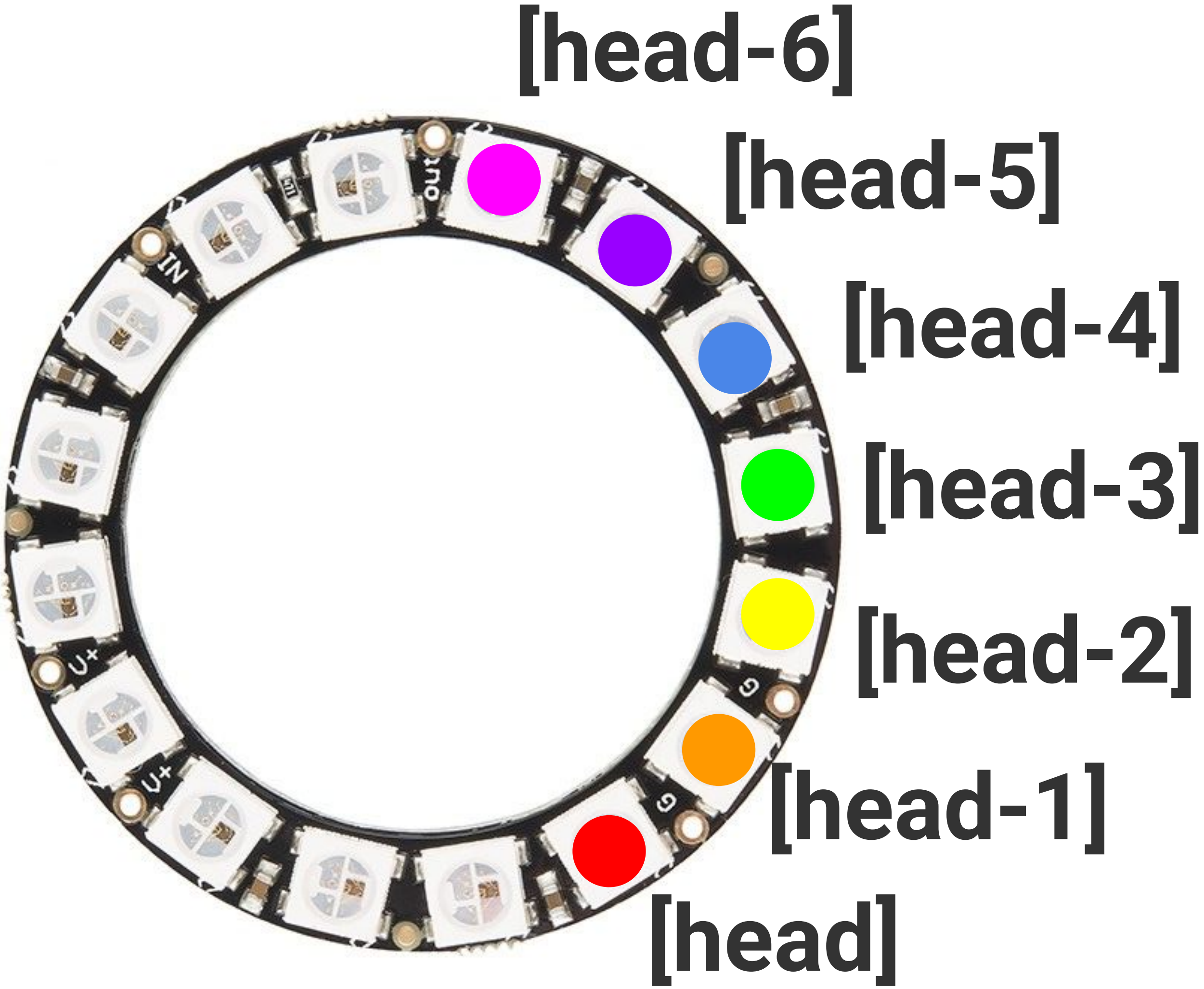
🎎 Guided

Colours of the snake



What are the indexes to indicate the rest of the snake?

Colours of the snake



[head-6]

[head-5]

[head-4]

[head-3]

[head-2]

[head-1]

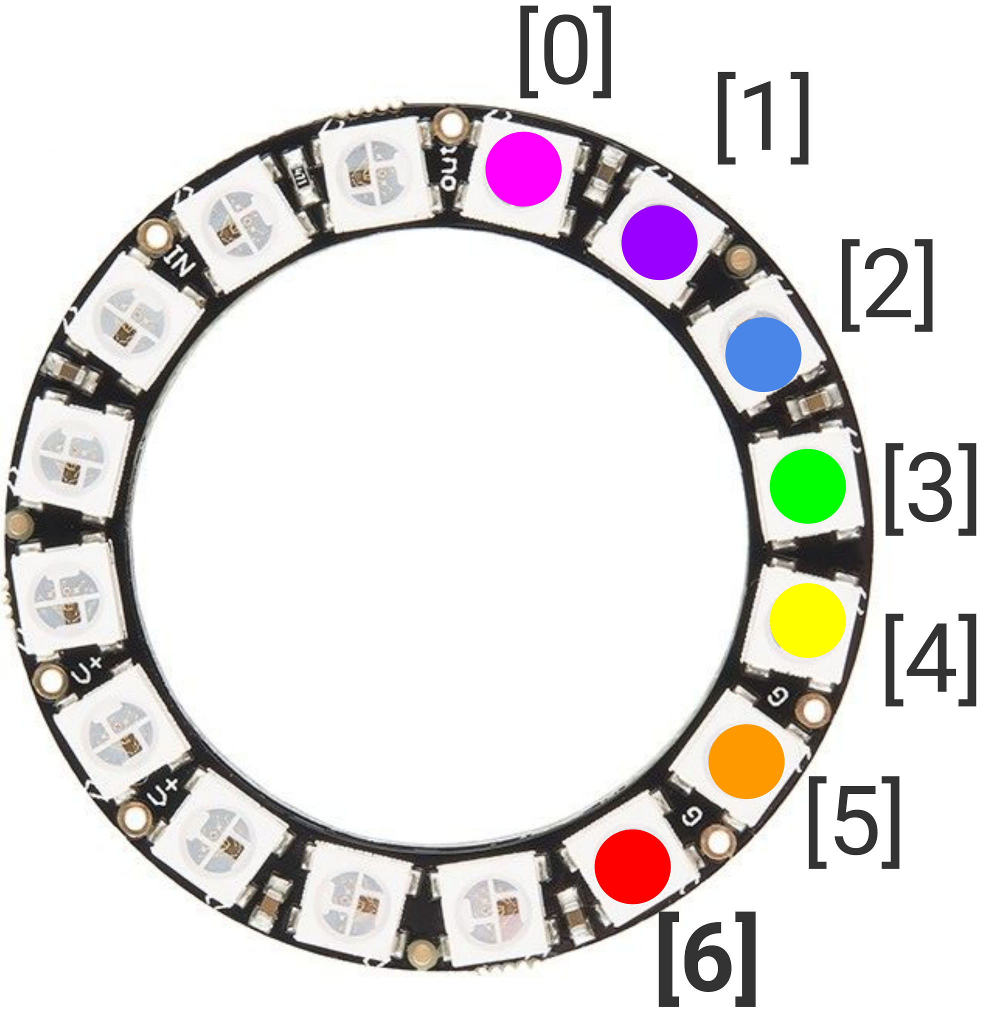
[head]

Did you get it right?

Assign the rest of the snake to the rest of the colours.

Guided

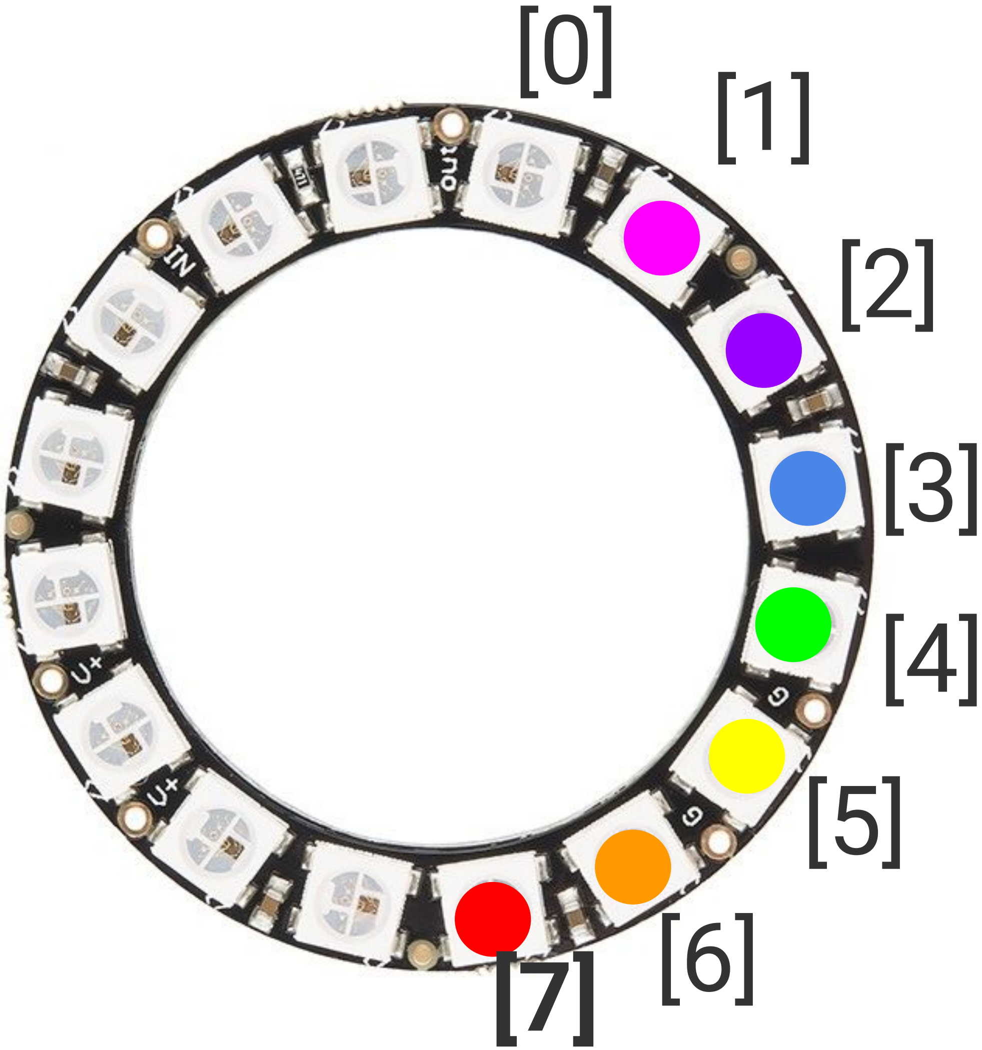
Moving snake



Remember that as the snake moves, the head of the snake moves to 1 position higher

Guided

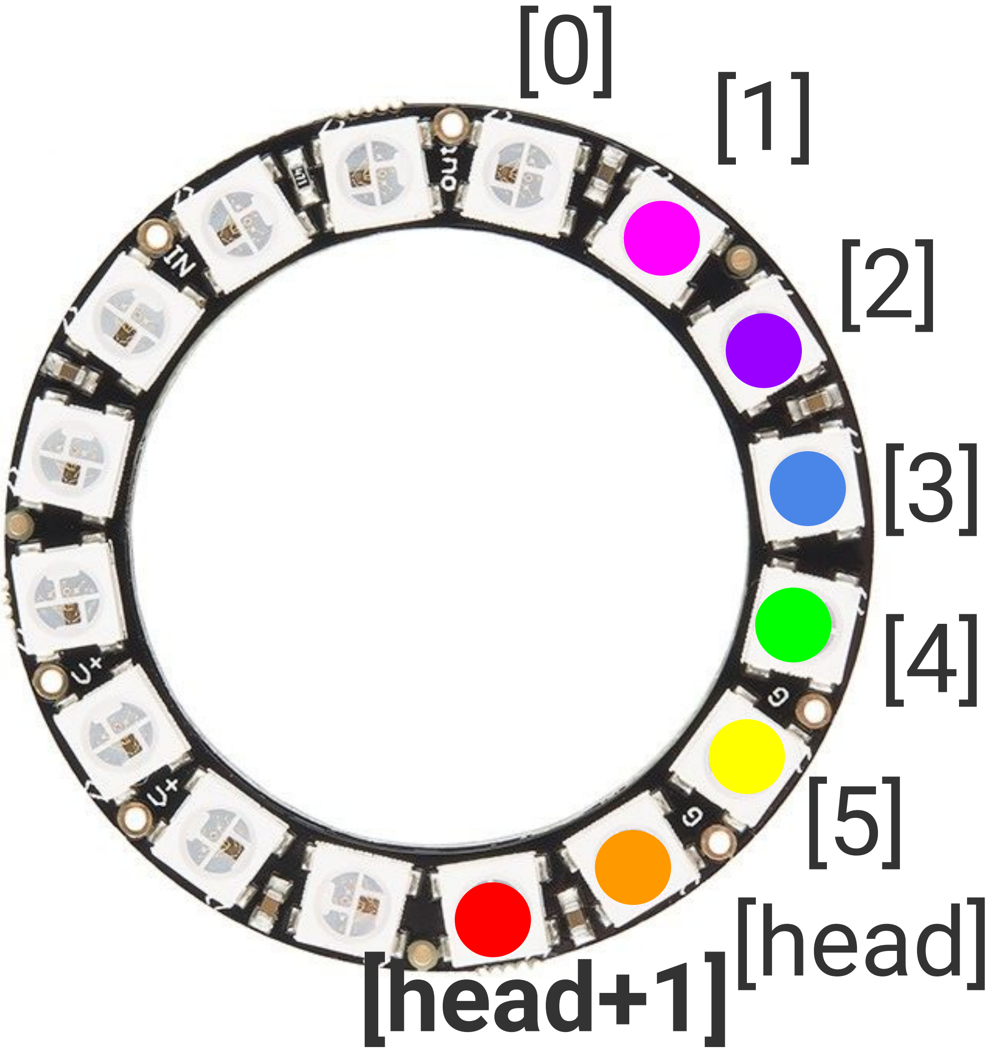
Moving snake



Remember that as the snake moves, the head of the snake moves to 1 position higher

🎎 Guided

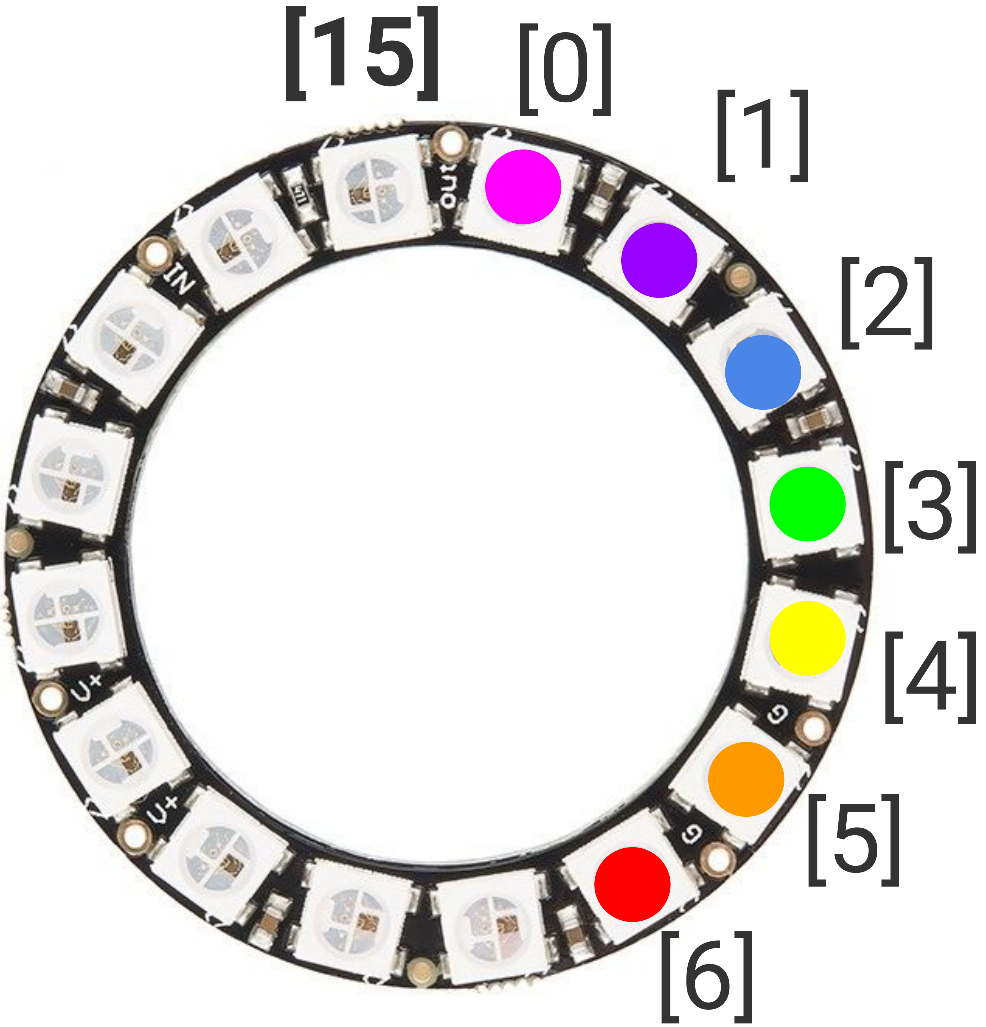
Moving snake



Hence, the new head of the snake is at **[head+1]**

Guided

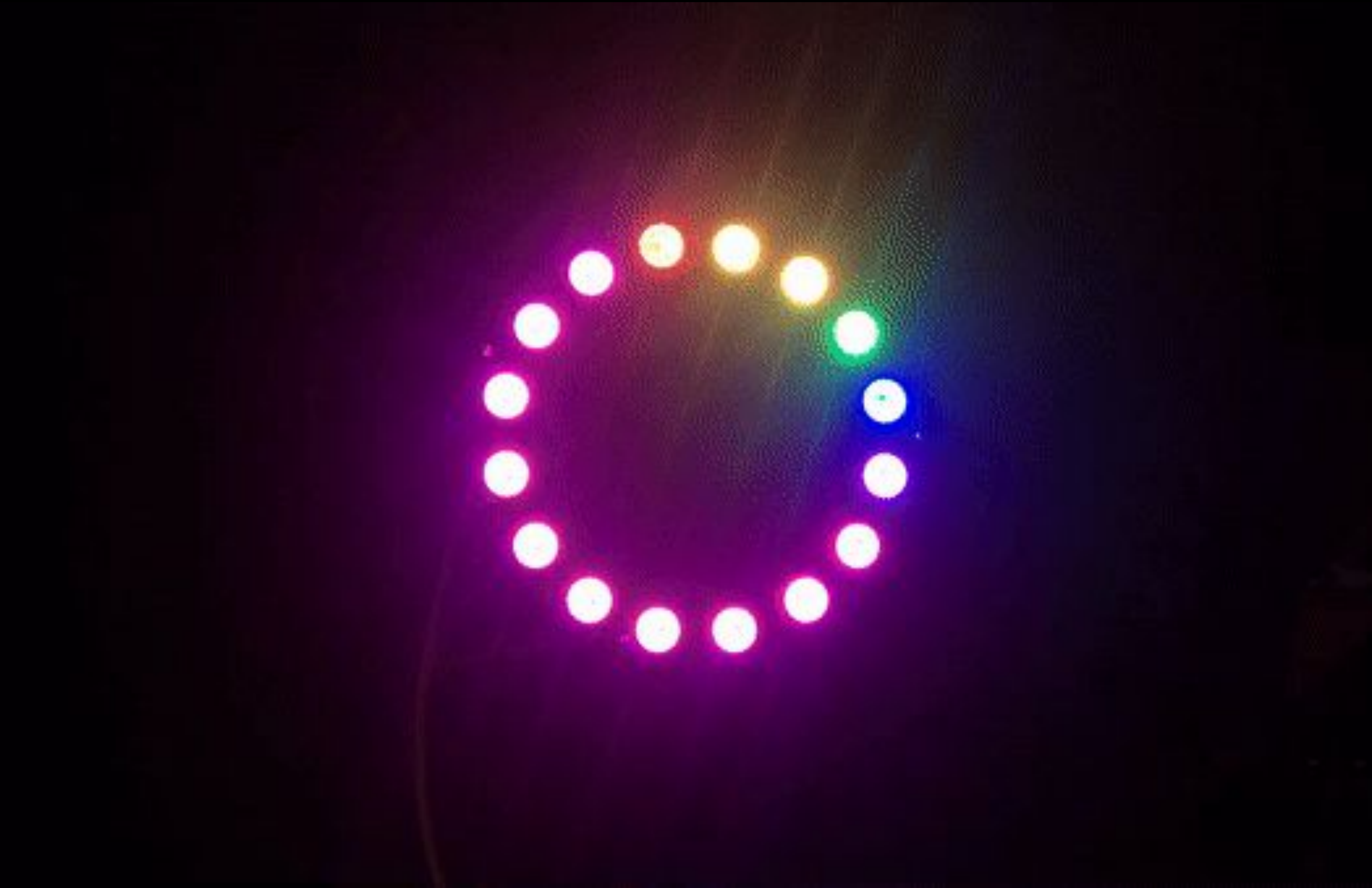
Restarting the snake



Once the snake reaches the 16th LED, we need to restart the head of the snake from [0].

Remember that the final LED position is [15]!

📥 Download the Project onto the micro:bit



🎎 Guided

The snake leaves a trail of violet LED lights

Why?

Guided

How to Correct the Code

```
white=(255,255,255)

np[head-7]=white
```

💾 Give your project a name and save it!