

# **Drawing Art with Coding and Mathematics**

## **Coding & STEM 4 Schools**

**An Introduction to Coding and Computational Thinking**

**Presented by Mr Daniel Hickmott**

**October 11th 2019**

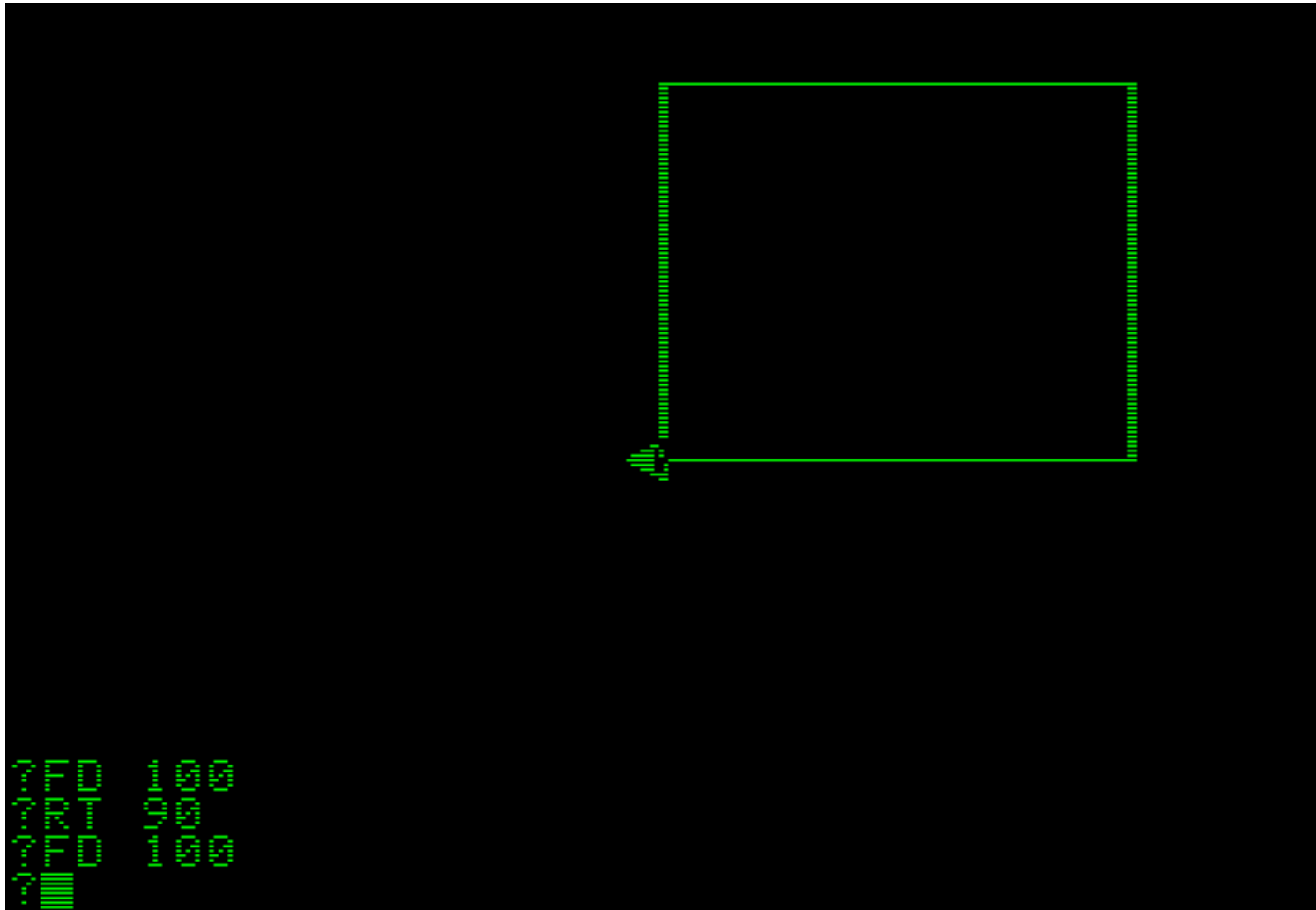
# Activity

- In this activity you will:
  - code in Pencil Code ([A Hybrid Coding Environment](#))
  - use Coding and Mathematics to draw art
  - apply concepts that you learned about in previous sessions, such as [Sequences](#) and [Loops](#)
  - learn about another **Computational Concept** called [Operators](#)

# Coding and Mathematics

- (In Coding) "...the purpose of math is not to get a good score on a test. The purpose of the math is to get your program to work. It is a self-teaching lesson."  
David Bau (Creator of Pencil Code)
- Coding and the teaching of Mathematics have been intertwined since the first educational Coding languages (for example, LOGO)
- What are some ways that you could (or do) teach Coding alongside Mathematics?





**my world**

**MENU**

- square 40
- Hexagon
- Triangle
- Rhombus
- Trapezoid
- Pentagon
- N-GON 7 25
- Star 50
- 5Star
- 6Star 60
- n-Star 7 99
- Tile
- squares
- r-arc
- r-petal
- l-arc
- l-petal
- flower

**STARS**

**n-Star**

Input N SIZE

Repeat N

fd SIZE

bk SIZE

rt

360 / N

The image shows a Scratch code editor with a script to draw an octagon. The script is as follows:

```
1 speed 500
2 movexy -150, 0
3 pen purple, 10
4 pd()
5 for [1..8]
6   dot purple, 50
7   fd 100
8   rt 45
9
```

The script is executed, resulting in a purple octagon with a turtle icon at the bottom-left vertex. The octagon is drawn with a pen color of purple and a pen thickness of 10. The turtle icon is positioned at the bottom-left vertex of the octagon.

# Pencil Code

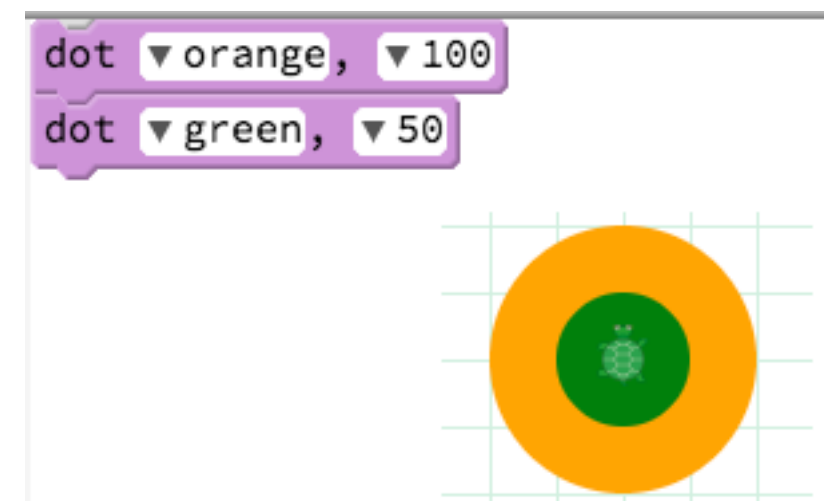
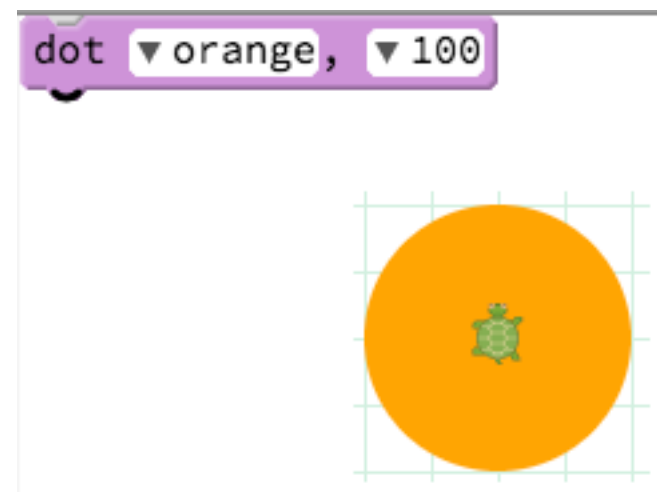
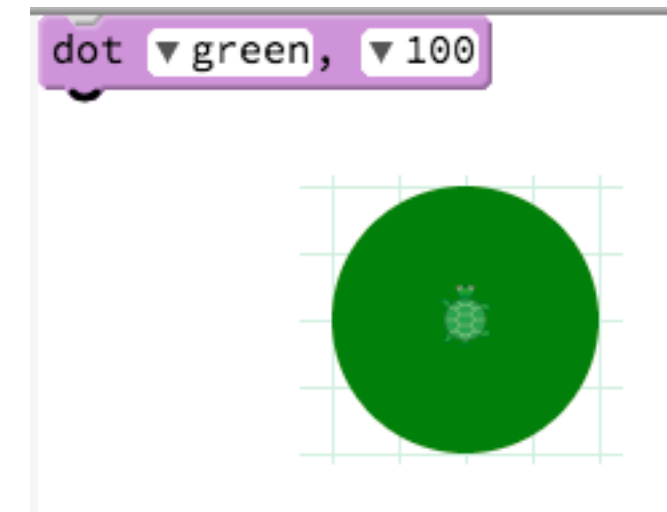
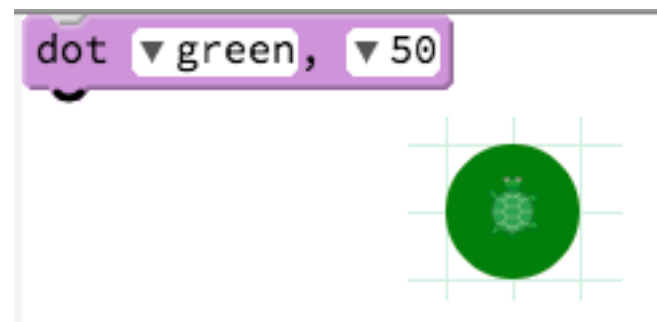
- A free, online [Hybrid Coding Environment](#)
- Uses a [General-Purpose Programming Language](#) called CoffeeScript, which can be used for web development
- You can also change the language to JavaScript, which is more commonly used by professional coders
- You can switch between [Blocks](#) and [Text](#) views
- 3 themed interactive tutorials: [Draw](#), [Jam](#) and [Imagine](#)

# Drawing a Sky Full of Stars

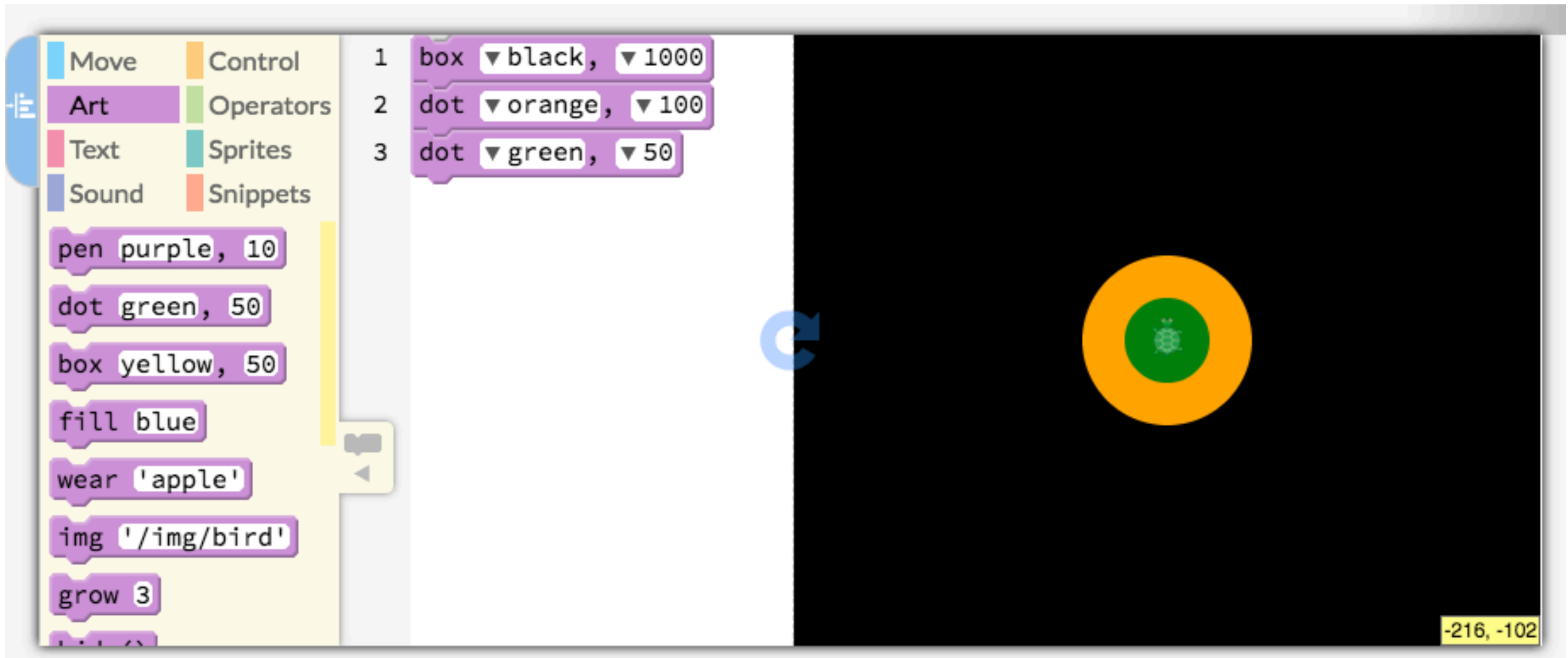




# Drawing a Dot



# Change the Background Colour



The image shows a Scratch code editor with a script to change the background color. The script consists of three blocks:

- 1 box ▼ black, ▼ 1000
- 2 dot ▼ orange, ▼ 100
- 3 dot ▼ green, ▼ 50

The stage shows a black background with a green circle and a yellow ring. The coordinates for the green circle are -216, -102.

# Pencil Code Colour Reference

white	gainsboro	silver	darkgray	gray	dimgray	black
whitesmoke	lightgray	violet	orchid	magenta	darkviolet	darkmagenta
ghostwhite	thistle	plum	mediumorchid	darkorchid	blueviolet	purple
aliceblue	lavender	lightsteelblue	mediumpurple	mediumslateblue	rebeccapurple	indigo
lightcyan	powderblue	lightskyblue	cornflowerblue	slategray	darkslateblue	blue
azure	paleturquoise	skyblue	deepskyblue	lightslategray	slateblue	mediumblue
mintcream	aquamarine	lightblue	darkturquoise	dodgerblue	royalblue	darkblue
honeydew	palegreen	lightgreen	cyan	cadetblue	steelblue	navy
lightgoldenrodyellow	yellow	lawngreen	turquoise	mediumturquoise	darkcyan	midnightblue
lightyellow	palegoldenrod	chartreuse	mediumaquamarine	lime	teal	darkslategray
beige	wheat	greenyellow	mediumspringgreen	limegreen	lightseagreen	green
ivory	moccasin	khaki	springgreen	darkseagreen	mediumseagreen	darkgreen
lemonchiffon	papayawhip	gold	yellowgreen	goldenrod	seagreen	forestgreen
cornsilk	blanchedalmond	tan	darkkhaki	darkorange	olivedrab	darkolivegreen
floralwhite	navajowhite	burlywood	orange	peru	darkgoldenrod	olive
oldlace	antiquewhite	sandybrown	coral	chocolate	saddlebrown	red
linen	bisque	lightsalmon	salmon	orangered	sienna	darkred
seashell	peachpuff	darksalmon	lightcoral	tomato	crimson	maroon
snow	mistyrose	lightpink	rosybrown	indianred	deeppink	firebrick
transparent	lavenderblush	pink	hotpink	palevioletred	mediumvioletred	brown

# Different Background Colours

The image shows a Scratch code editor interface. On the left, a palette of blocks is visible, including Move, Art, Text, Sound, Control, Operators, Sprites, and Snippets. The script area contains the following code:

```
1 box ▼ skyblue, ▼ 1000  
2 dot ▼ gold, ▼ 100  
3 dot ▼ skyblue, ▼ 50
```

Below the script area, there are several other code blocks in the palette:

- pen purple, 10
- dot green, 50
- box yellow, 50
- fill blue
- wear 'apple'
- img '/img/bird'
- grow 3

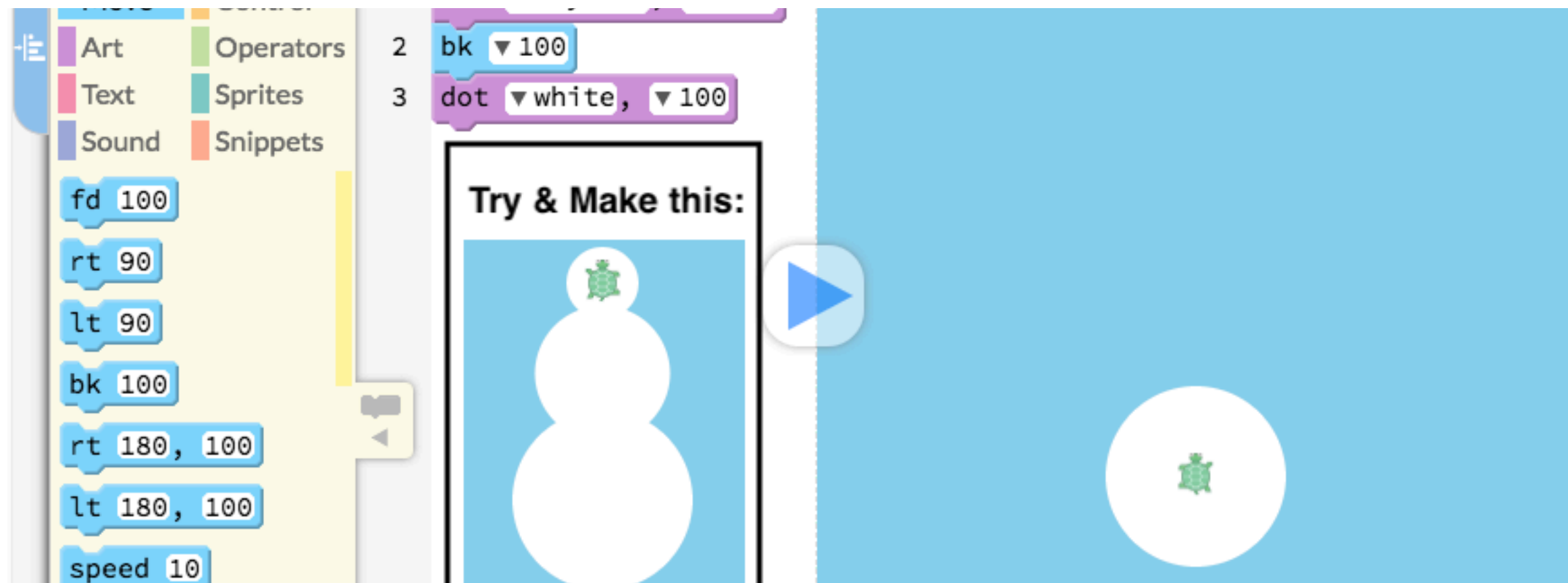
The stage on the right shows a blue background with a yellow circle in the center. A small green turtle icon is visible inside the circle. A blue circular arrow icon is positioned between the script area and the stage. In the bottom right corner of the stage, there is a small yellow box containing the text "-147, 13".

# Moving the Turtle

- We can also move the Turtle on the Canvas:
  - `fd`: it moves forward a number of steps
  - `bk`: it moves backwards a number of steps
  - `lt`: it turns by a number of degrees to the left
  - `rt`: it turns by number of degrees to the right
- We will start by moving the Turtle forward

# Drawing a Snow Kid in a Sequence

- Use the blocks below a starter and add `fd` and `dot` blocks to draw a Snow Kid



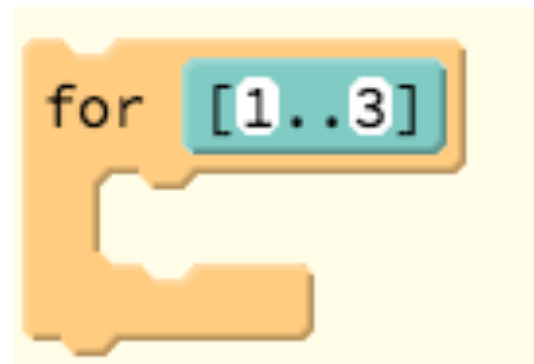
The image shows a Scratch code editor with a sequence of blocks. The blocks are:

- `fd 100`
- `rt 90`
- `lt 90`
- `bk 100`
- `rt 180, 100`
- `lt 180, 100`
- `speed 10`

Below these blocks, there is a section titled "Try & Make this:" which shows a snowman made of three white circles on a blue background. A play button is next to it. To the right, a larger blue area shows a single white circle with a green turtle icon in the center, representing the starting point of the drawing process.

# Loops in Pencil Code

- Do you remember what **Loops** are?
- We use **Loops** to repeat steps multiple times
- For example, the **repeat** block in Scratch
- Pencil Code has a command called **for**, in the **Control** section, which is similar to repeat



# Drawing a Line of Dots

The image shows a Scratch code editor with a script area on the left and a stage on the right. The script area contains the following code:

```
1 bk 100
2 for [1..3]
3   dot green, 50
4   fd 100
```

The stage on the right shows a green grid with three green dots arranged vertically. A blue circular arrow icon is positioned to the left of the dots. In the bottom right corner of the stage, there is a yellow box containing the coordinates `-149, -51`.



# Speeding the Turtle Up

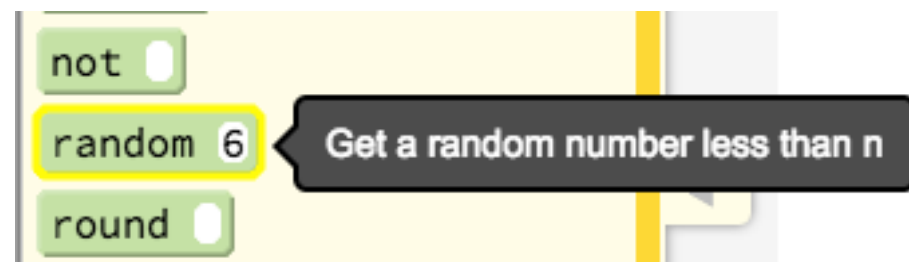
The image shows a Scratch code editor with a script on the left and a stage on the right. The script consists of the following blocks:

```
1 speed 50
2 bk 100
3 for [1..10]
4   dot green, 25
5   fd 25
```

The stage shows a green turtle at the top center of a grid. Below the turtle is a vertical line of 10 green dots. A blue circular arrow icon is positioned to the left of the stage. In the bottom right corner of the stage, there is a yellow box containing the text "-13, 178".

# Operators

- All of the blocks in the **Operators** section
- Variety of different functions and purposes
- E.g. addition, subtraction, multiplication & division
- We will focus on the **random** block in today's session



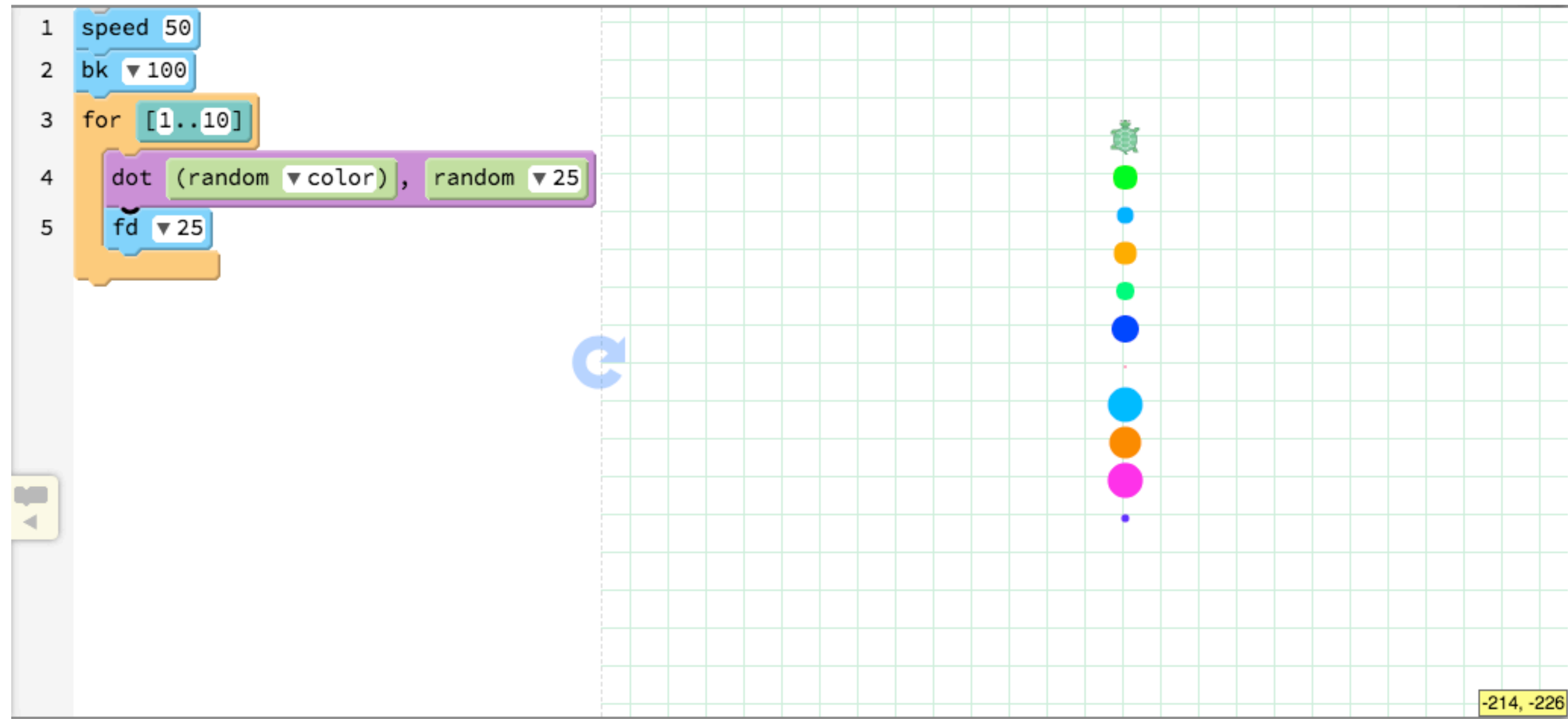
# Random Sized Dots

The image shows a Scratch code editor with a script on the left and a stage on the right. The script consists of the following blocks:

```
1 speed 50  
2 bk 100  
3 for [1..10]  
4   dot green, random 25  
5   fd 25
```

The stage on the right shows a vertical line of 10 green dots of varying sizes, drawn by the script. A blue circular arrow icon is visible on the stage grid.

# Random Coloured Dots

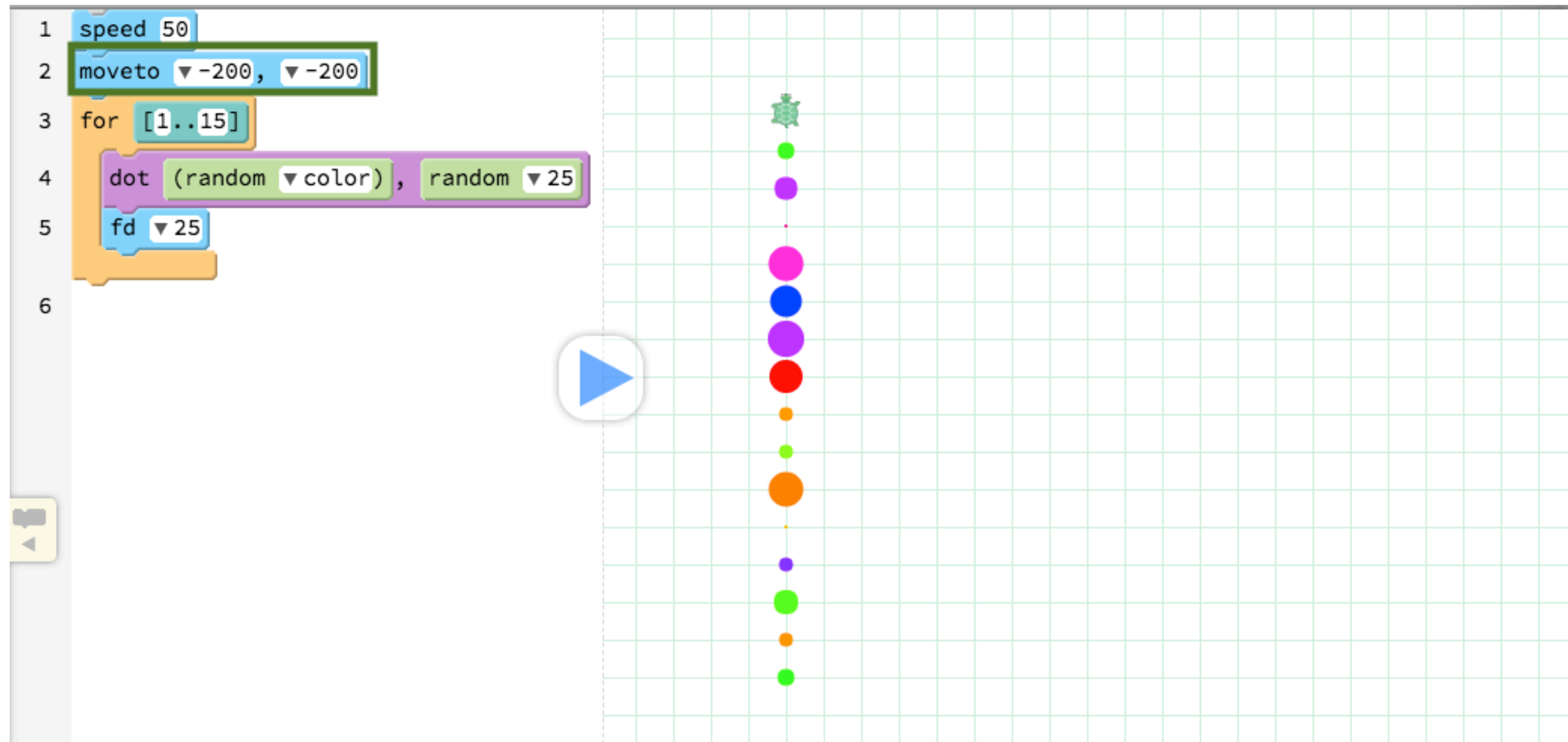


The image shows a Scratch workspace with a script area on the left and a stage on the right. The script area contains the following code:

```
1 speed 50
2 bk 100
3 for [1..10]
4   dot (random color), random 25
5   fd 25
```

The stage shows a vertical line of 10 colored dots. The dots are arranged in a vertical line, with the top dot being a small green turtle icon. The dots below it are colored in a sequence: green, blue, orange, green, blue, cyan, orange, magenta, and a very small purple dot at the bottom. A blue circular arrow icon is visible on the left side of the stage, indicating a refresh or reset action. The coordinates `-214, -226` are displayed in the bottom right corner of the stage.

# Moving the Turtle Around

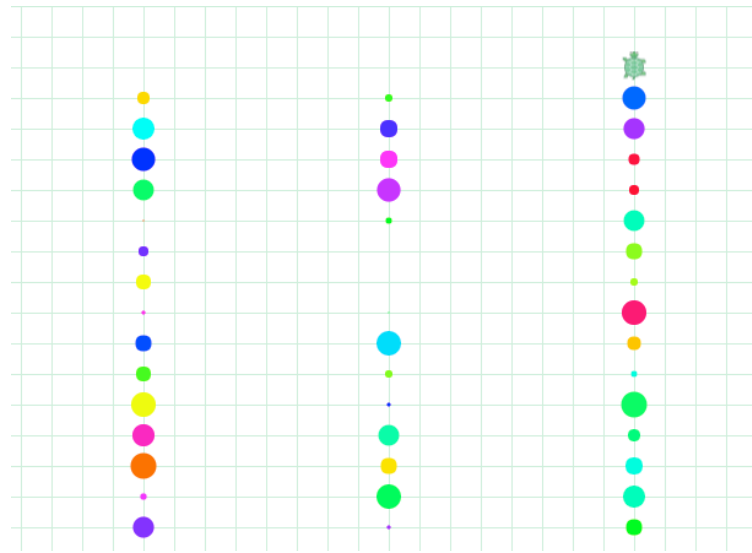


The image shows a Scratch code editor with a script on the left and a canvas on the right. The script consists of the following blocks:

- 1. `speed 50`
- 2. `moveto -200, -200` (highlighted with a green box)
- 3. `for [1..15]` loop
- 4. `dot (random color), random 25` block (highlighted with a purple box)
- 5. `fd 25` block (highlighted with a blue box)
- 6. Empty line

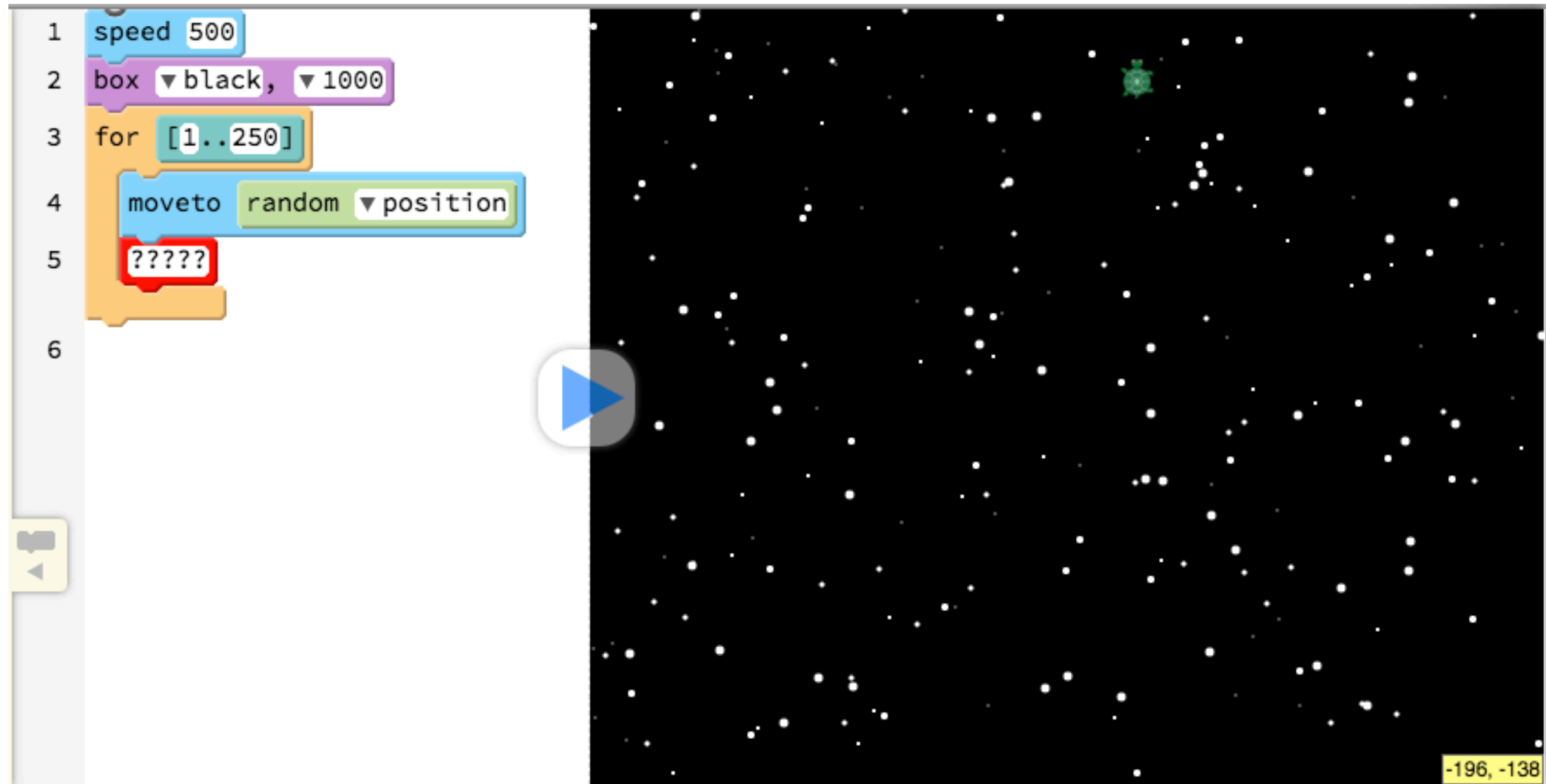
The canvas on the right shows a green grid with a vertical line of 15 colored dots. A small green turtle icon is positioned at the top of the line. A blue play button is visible on the left side of the canvas.

# Drawing Three Lines



- Use the `moveto` command with the `for` loop to draw a picture similar to the one above
- Hint: by switching to the `Text` mode, you can copy and paste commands

# Drawing a Starry Night

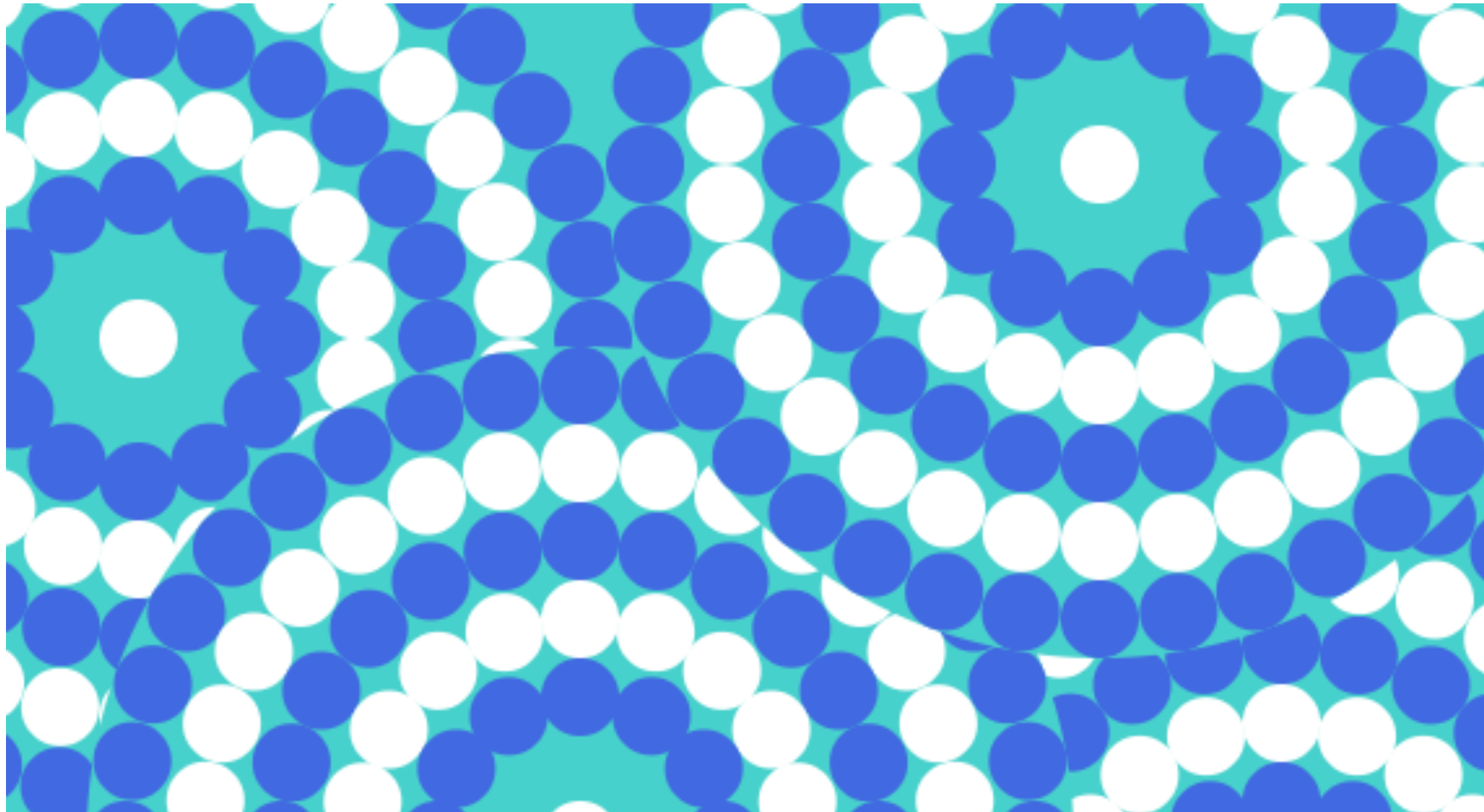


The image shows a Scratch script editor on the left and a preview window on the right. The script consists of the following blocks:

- 1. `speed 500`
- 2. `box ▼ black, ▼ 1000`
- 3. `for [1..250]`
- 4. `moveto random ▼ position`
- 5. `?????` (highlighted in red)
- 6. (empty line)

The preview window displays a black background with numerous white dots of varying sizes representing stars. A single green star is visible in the upper left quadrant. A play button is centered over the preview window. In the bottom right corner of the preview window, the coordinates `-196, -138` are displayed.

# Creating Dot Paintings

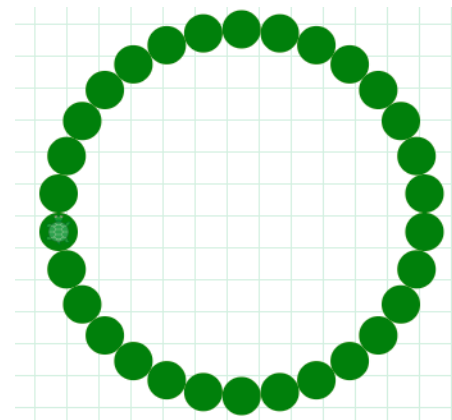




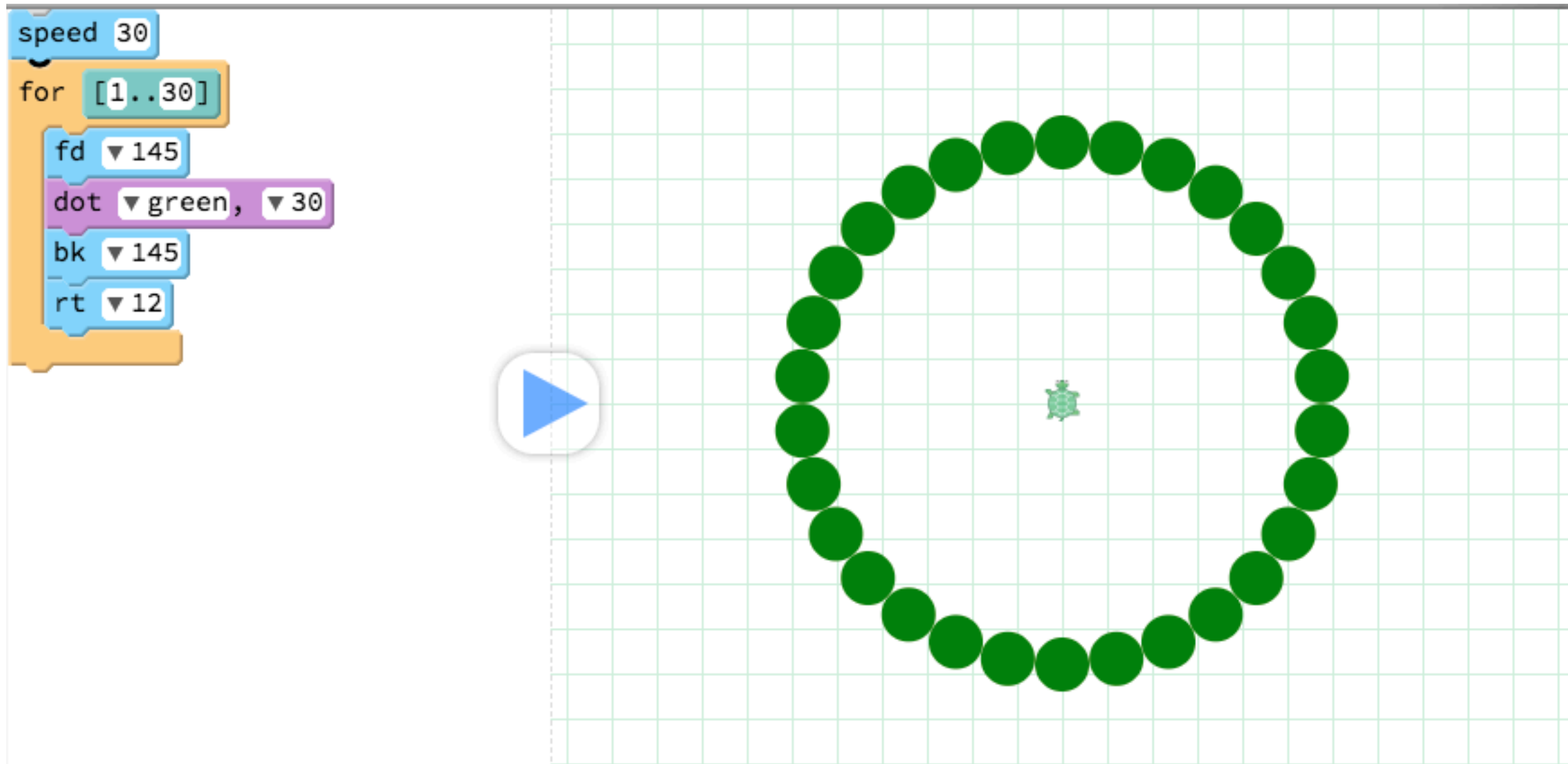
# Drawing a Circle of Dots

```
1 speed 50
2 for [1..3]
3   dot ▼ green, ▼ 30
4   fd ▼ 30
5   rt ▼ 12
```

— What should we change to make it draw this pattern?



# Forward and Backwards

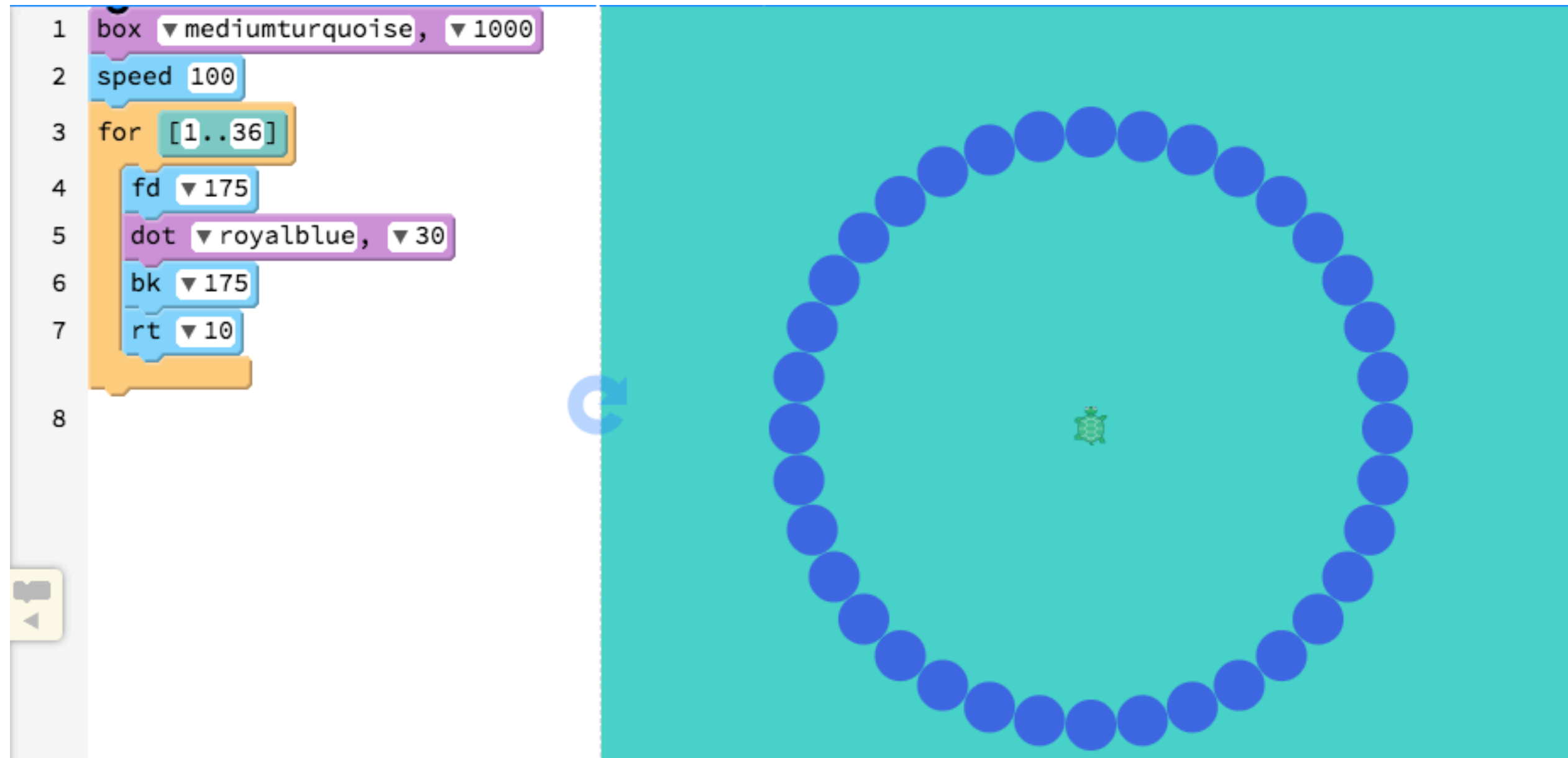


The image shows a Scratch workspace with a code editor on the left and a stage on the right. The code editor contains the following blocks:

- speed 30
- for [1..30]
- fd 145
- dot green, 30
- bk 145
- rt 12

A blue play button is visible on the left side of the stage. The stage shows a small green turtle icon at the center, surrounded by a circle of 30 green dots. The dots are arranged in a circle with a radius of approximately 145 pixels, and the turtle is positioned at the center of this circle.

# Background and Colour Change



The image displays a Scratch script on the left and its visual output on the right. The script consists of the following blocks:

- 1. `box` block with `mediumturquoise` and `1000`.
- 2. `speed` block with `100`.
- 3. `for` loop block with `[1..36]`.
- 4. `fd` block with `175`.
- 5. `dot` block with `royalblue` and `30`.
- 6. `bk` block with `175`.
- 7. `rt` block with `10`.
- 8. An empty line.

The visual output on the right shows a teal background with a circle of 36 blue dots. A small green turtle icon is positioned at the center of the circle. A blue circular arrow icon is located between the code and the output.

# Drawing the Other Circles

<b>Pattern</b>	<b>Degrees in Each Turn</b>	<b>Times Repeated</b>	<b>Steps Forwards &amp; Backwards</b>
1	10	36	175
2	12	30	145
3	15	24	115
4	20	18	85
5	30	12	55

## An Extra 2 Dots

- In the next steps we will draw patterns that overlap with each other, so we will also add a **big dot**
- Put the command at the top of your script (before all the **for** blocks): **dot 380, mediumturquoise**
- We also want a **dot in the middle** at the end
- After you have drawn all of the other patterns, add this command (to draw the last dot in the middle): **dot 30, white**

# Creating a Function

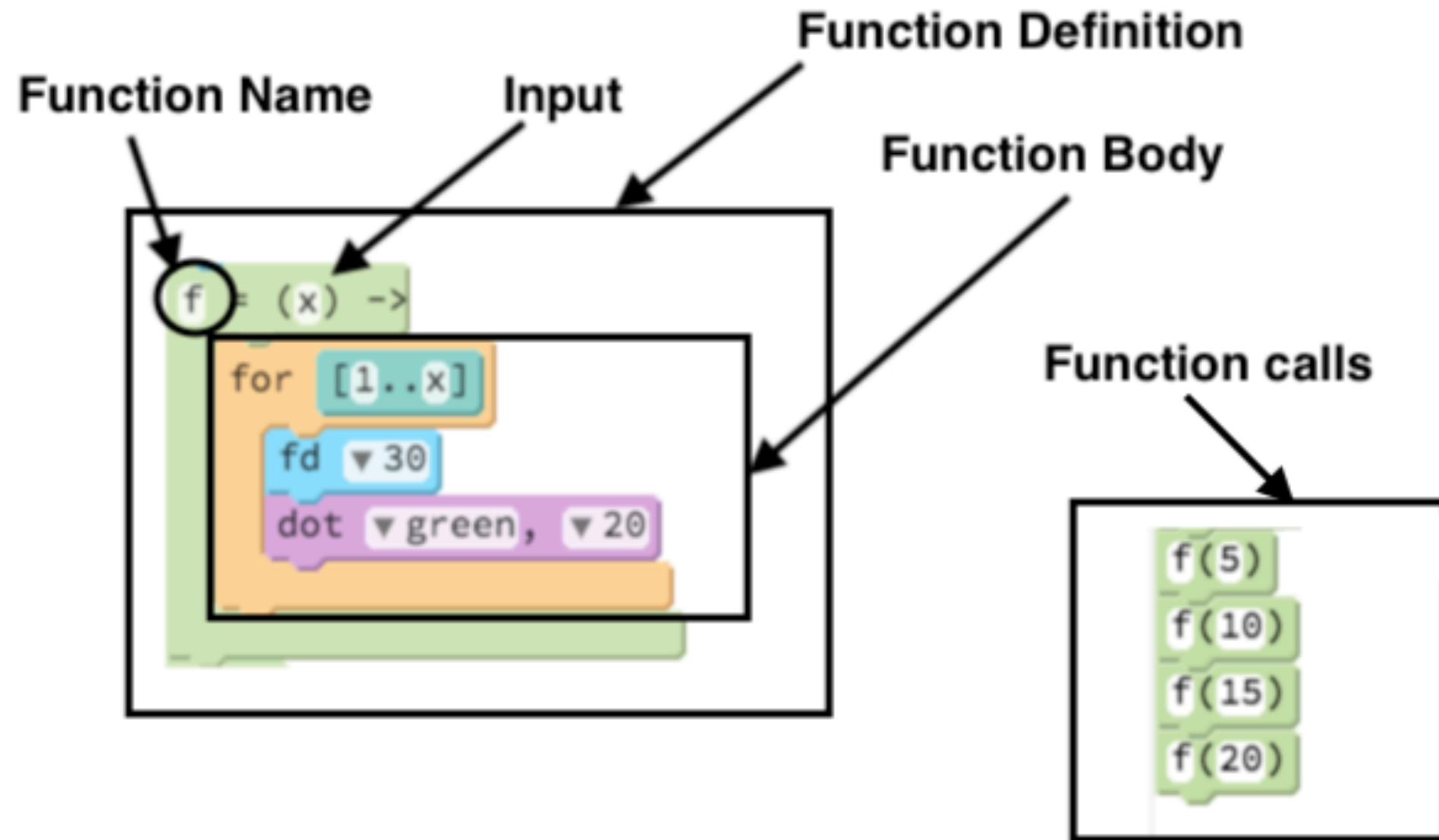
- A **Function** is a way of "encapsulating" common instructions
- We can take a set of instructions (such as drawing the dot patterns) and put them into one instruction
- If you have used **Custom Blocks** (through the **Make a block** button) in **Scratch**, you have created a **Function**
- **fd**, **speed** and other blocks are examples of in-built **Functions** in **Pencil Code**, but you can also create your own **Functions** as well

# Creating a Function to Draw the Pattern

- Say that we wanted to draw our pattern of circles on other spots on the Canvas more than once
- How could we do that?
- We could copy/paste the instructions a few times **but** there is a way that is neater

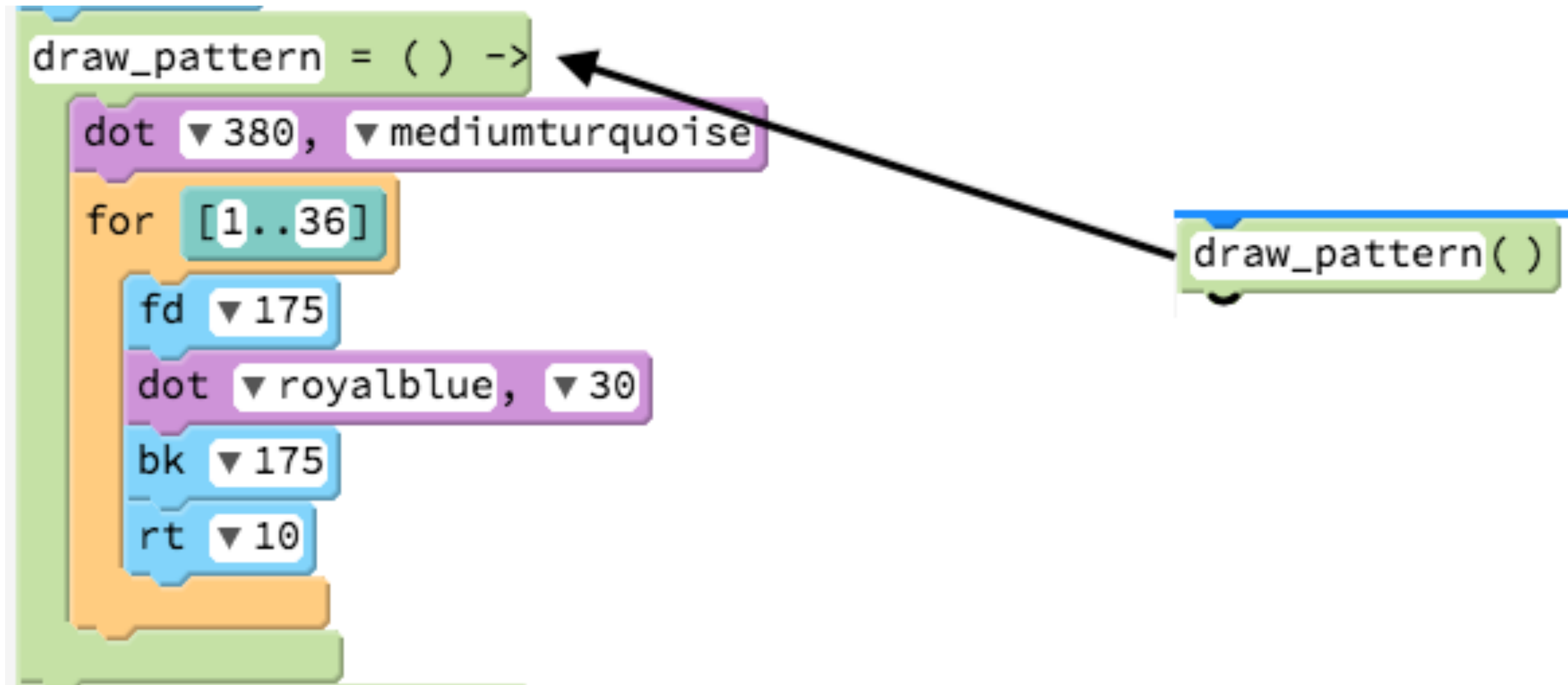
```
1 speed 500
2 draw_pattern( )
3
```

# Functions in Pencil Code



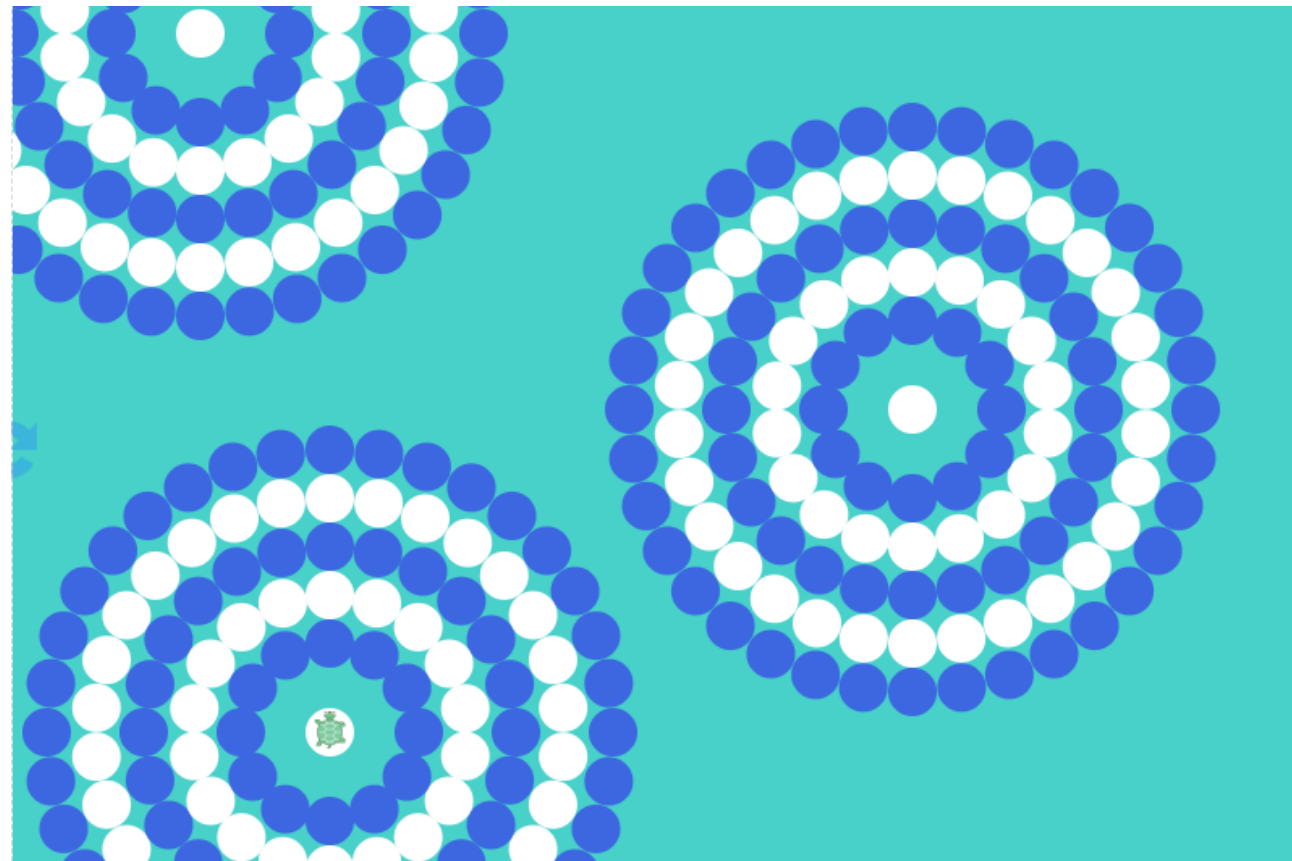


# Function Example



# Drawing the Pattern in Different Spots

- How can you combine the `draw_pattern` and `moveto` blocks to make a picture like below?



# Extensions

- If you finish all of the previous steps, you could:
  - change the colours of the different patterns
  - add more patterns on the Stage
  - put all of the instructions that draw a pattern into one **Function** called **draw\_circle** with inputs: **repeat**, **steps** and **degrees**
  - make it so that the colour of the dots are **random**