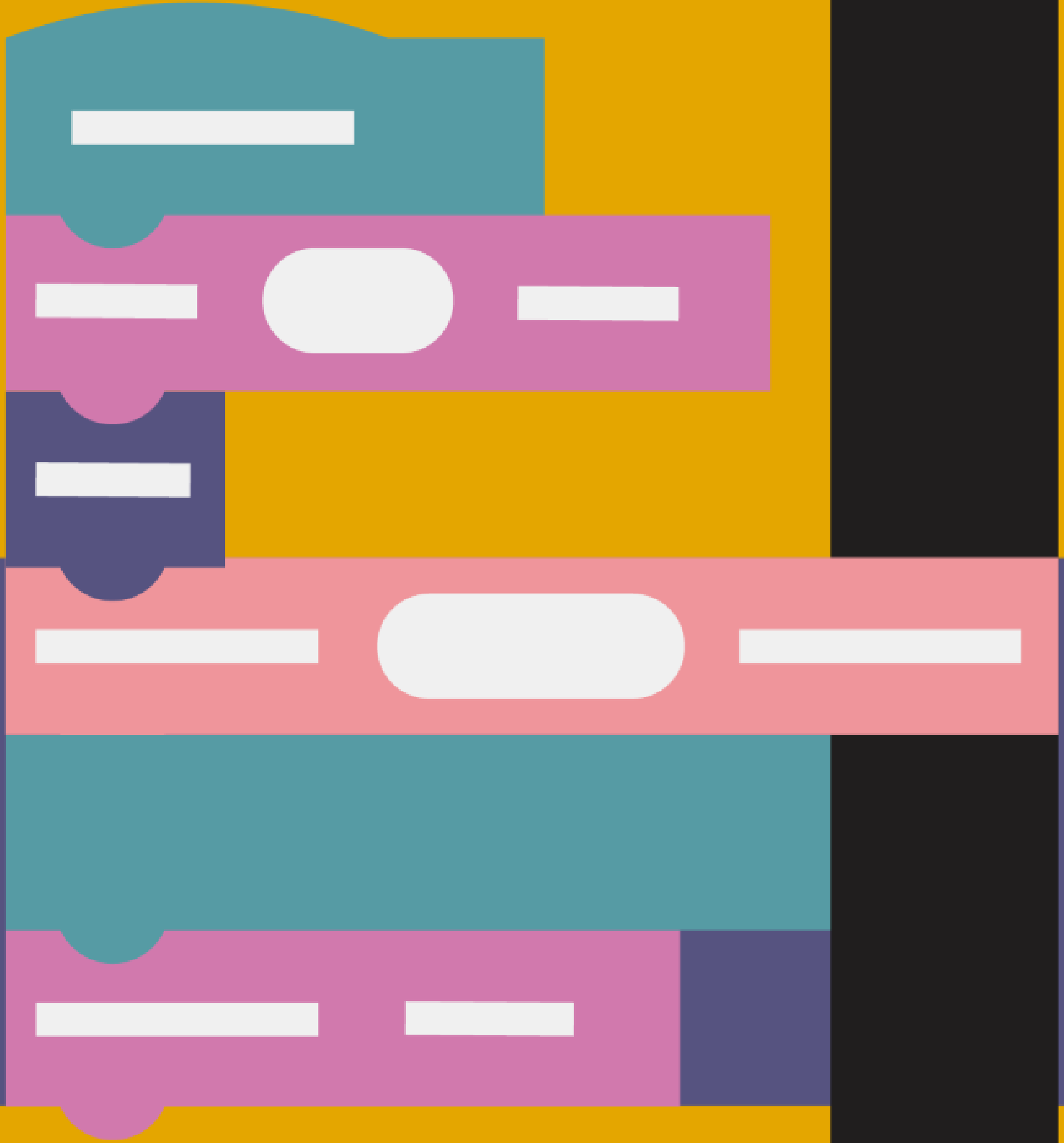


Globisens

xploris

CODING MATH

Flashing Stars

xploris

CODING MATH

Globisens

FLASHING STARS

1. Introduction
2. Theory
3. Activity setup
4. Coding
5. Activity summary

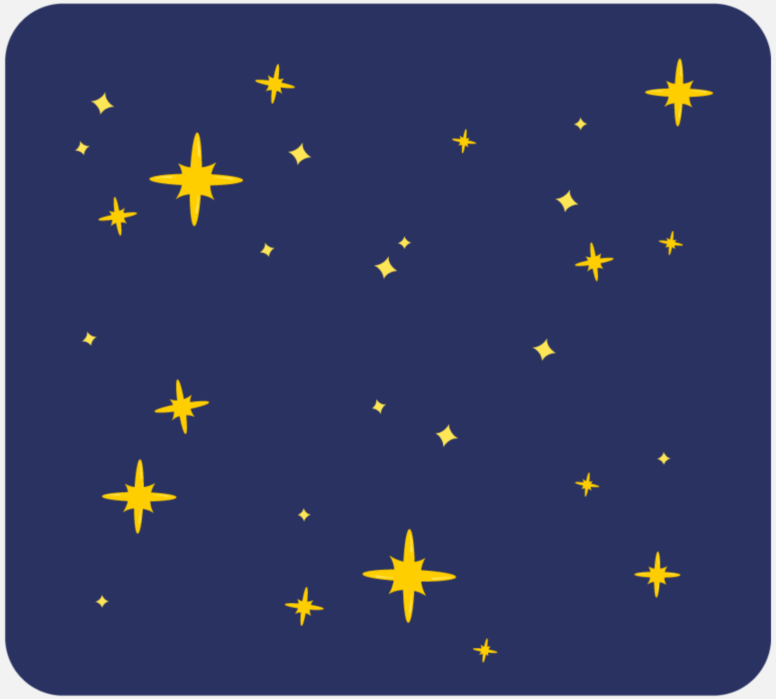
1 Introduction

Welcome to a new adventure in math and coding!

Today we are going to explore how technology uses random numbers to create interesting visual effects. Imagine that in the night sky, the stars twinkle in different random positions. Our goal will be to program a white star to twinkle in random positions on the Xploris screen.

Why is this important?

Random is a fundamental concept in mathematics and programming, used in simulations, graphics and video games. Learning to generate random positions helps us understand how numbers work in different contexts.



2 Theory

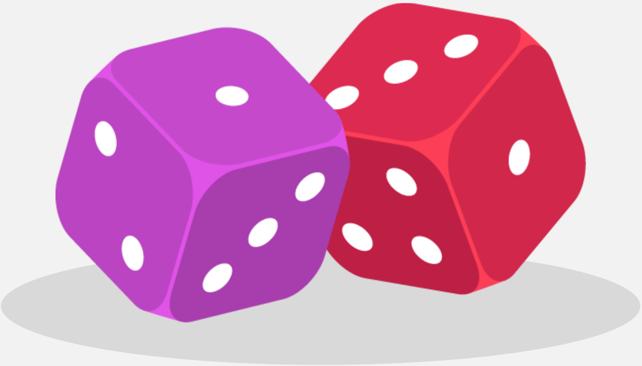
What is Randomness?

Random sequence occurs when something **happens without following a fixed order**. In other words, we cannot predict exactly what will happen.

Examples of randomness in real life:

- **Rolling a die:** We don't know what number will come up.
- **The weather:** It can be sunny, rainy or snowy without always following a pattern.
- **A raffle:** We cannot know which number will win before it is chosen.

Randomness is everywhere!



2 Theory

How Do We Use Randomness in Mathematics?

In math, we use randomness to **do simulations, play games and solve problems.**

Example: Roll a die

A die has **6 sides** (1, 2, 3, 4, 5 and 6).
Each time we roll it, any of these numbers can come up.

Probability
If there are 6 numbers on the die and only one can come up, the probability of getting a 3 is **1 out of 6**.

How Do We Use Randomness in Programming?

Computers cannot think for themselves, but they can **simulate randomness** with numbers.
In Python, we use a special function called randint() to **pick random numbers**.
Example in Python:

```
import random
x = random.randint(1, 128) # Chooses a number between 1 and 128
y = random.randint(1, 64) # Chooses a number between 1 and 64
```

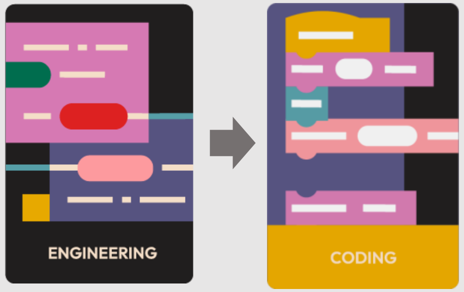
Each time we run this code, x and y will have different values.

3 Activity setup

Turn on your Xploris and connect it to your computer or tablet.

Open the XploriLab software on your computer or tablet.

Once inside XploriLab, select the icon to connect the device via USB cable or bluetooth as applicable.



Go to the ENGINEERING section and then to CODING.

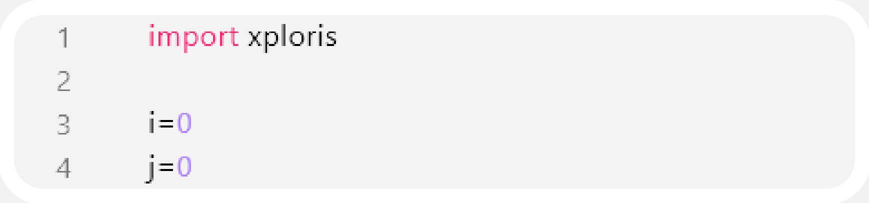
4 Coding

1

To begin, click on XploriLab to enable the Python programming.

2

Import the Xploris library and create the variables i and j, which will store random numbers.



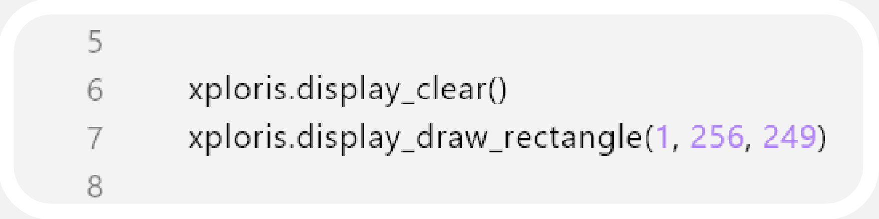
```
import xploris

i=0
j=0
```

What are i and j for?
These are numbers that will represent **the position of the star** on the screen.
“i” is the X coordinate (horizontal).
“j” is the Y-coordinate (vertical).

4 Coding

3

Before drawing the star, **clear the screen and create a background.**

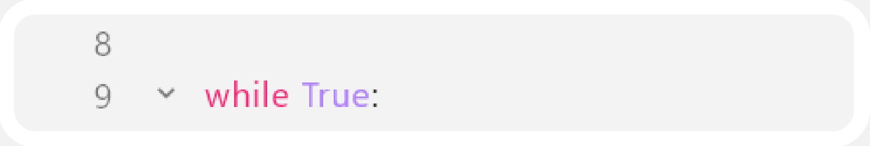


```
xploris.display_clear()
xploris.display_draw_rectangle(1, 256, 249)
```

Why are we doing this?
“xploris.display_clear()” clears the Xploris screen. “xploris.display_draw_rectangle(1, 256, 249)” creates a **background rectangle** for the animation. Rectangle starts on pixel **1** and ends on pixel **256** – setting all pixels to **249** = blue color.

4

The star **never stops blinking**. Use an **infinite loop** for this.



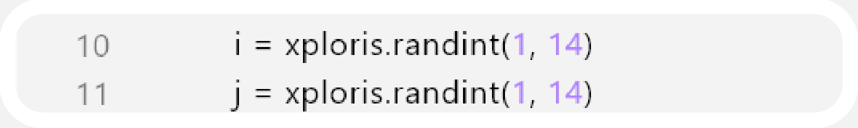
```
while True:
```

Note: The instructions that we want to be **repeated inside the loop** must be written **below it with a tabulation** so that the computer understands that they are part of the loop.

4 Coding

5

Each time the program is repeated, **new random positions** for the star must be **created**. For this, the following must be included:



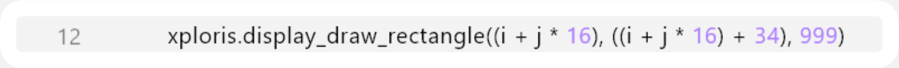
```
i = xploris.randint(1, 14)
j = xploris.randint(1, 14)
```

"xploris.randint(1, 14)" chooses a number between **1 and 14** for i and j.

This causes the star to **appear in a different place each time.**

6

To display the star on the screen, we use a command that draws a bright pixel at the chosen position:



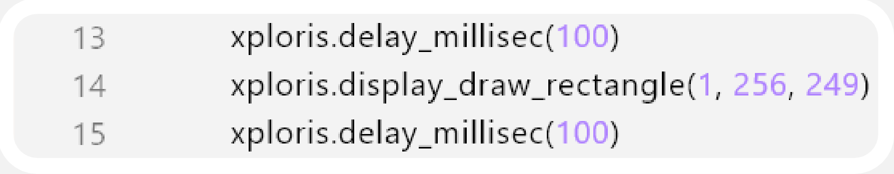
```
xploris.display_draw_rectangle((i + j * 16), ((i + j * 16) + 34), 999)
```

- "xploris.display_draw_rectangle()" draws a shape on the screen.
- "(i + j * 16)" defines the start pixel of the star.
- "(i + j * 16) + 34) defines the end pixel for the star.
- "999" is the color **white**.

4 Coding

7

To make the star **turn on and off**, you need to add a **short delay** before clearing the screen.



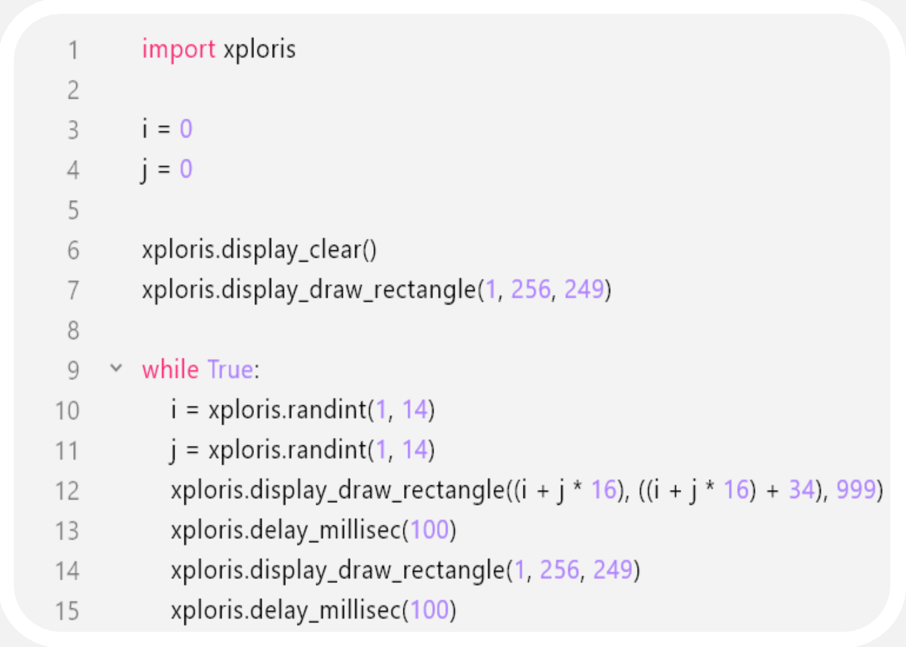
```
        xploris.delay_millisec(100)
        xploris.display_draw_rectangle(1, 256, 249)
        xploris.delay_millisec(100)
```

“xploris.delay_millisec(100)” makes the star **stay on the screen for 100 milliseconds**.

Then **the previous star is deleted** by displaying the same star in the back ground blue color**.** And now it can appear in a new place.

8

The code should look like this:



```
import xploris

i = 0
j = 0

xploris.display_clear()
xploris.display_draw_rectangle(1, 256, 249)

while True:
    i = xploris.randint(1, 14)
    j = xploris.randint(1, 14)
    xploris.display_draw_rectangle((i + j * 16), ((i + j * 16) + 34), 999)
    xploris.delay_millisec(100)
    xploris.display_draw_rectangle(1, 256, 249)
    xploris.delay_millisec(100)
```

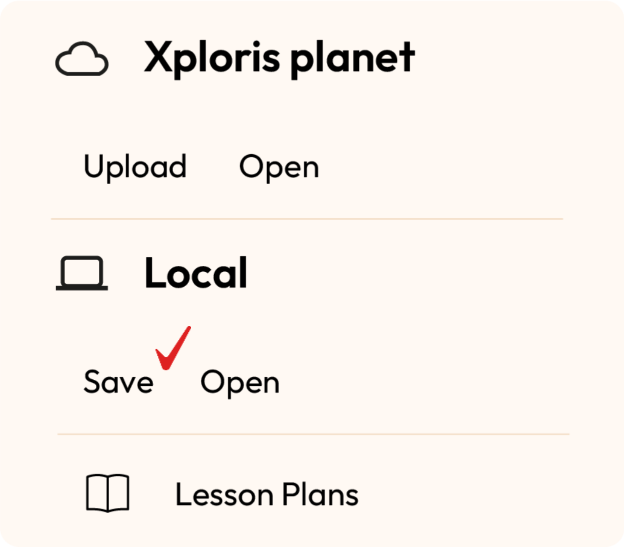
4 Coding

To make sure that the program works correctly, we will follow these final steps:

Press the three-bar icon at the top and select the "Save" option. Then, assign a name and save your program.

Press the "Upload" button in the Xplorilab interface. This will transfer the program to the Xploris device.

Once the program is loaded, press the "Play" button on Xplorilab software. Look the Xploris screen and you will see how the star changes position while it flashes.



5 Activity summary

- We learned what randomness is and how it is used in programming.
- We discovered how a computer chooses numbers at random.
- We programmed a flashing stars in Xploris.

Globisens

xploris

CODING MATH

Flashing Stars