

# Animations with Python Session 2





### **Topics covered**

- 1. Backdrops and Costumes
- 2. Functions to Control Sprite
- 3. Activity: Walking Tobi Animation







## **Backdrops and costumes**







### What is a Backdrop?

- A backdrop is one of the many frames, or backgrounds, that a Stage can have.
- The Stage can change its look to any of its backdrops.







### **Choosing a Backdrop**

You can choose a backdrop from

- the backdrop library
- uploading a file from the computer
- creating one using the paint editor

Click **Choose the Backdrop** and select any backdrop you want. We're choosing Blue Sky.







### What are Costumes?

- A costume is one out of possibly many "frames" or alternate appearances of a sprite.
- Sprites can change their look to any of its costumes.
- Every sprite has at least one costume.
- The costumes available for the sprite are shown in the **Costumes** tab, next to the **Python** tab.



![](_page_5_Picture_6.jpeg)

![](_page_5_Picture_8.jpeg)

### How to Create Costumes?

There are four ways of getting a costume for a sprite or stage.

- From the costume library
- Drawing one yourself using the inbuilt paint editor
- Getting an image or multiple images from your desktop
- Taking an image using a webcam

![](_page_6_Picture_6.jpeg)

![](_page_6_Picture_7.jpeg)

![](_page_6_Picture_9.jpeg)

![](_page_7_Picture_0.jpeg)

![](_page_7_Picture_2.jpeg)

![](_page_7_Picture_4.jpeg)

![](_page_7_Picture_5.jpeg)

### 1. move()

This function is used to move the sprite a certain number of steps forward. It takes only one input:

#### Number of Steps – Integer

#### Example:

sprite = Sprite('Tobi')
import time
sprite.move(10)
time.sleep(1)
sprite.move(50)

![](_page_8_Figure_6.jpeg)

![](_page_8_Picture_7.jpeg)

|      |      | t Upload Firmware        |             | 55 |
|------|------|--------------------------|-------------|----|
|      | Y    | (X:0,Y:180)              |             |    |
|      |      |                          |             |    |
|      |      |                          |             |    |
|      | 2.0% |                          |             |    |
|      | ~    |                          |             |    |
|      |      | ( <mark>X:0,Y:0</mark> ) | (X:240,Y:0) |    |
| ,200 |      | 200                      | 200         | X  |
|      |      |                          |             |    |
|      |      |                          |             |    |
|      | .00  |                          |             |    |
|      | 235  |                          |             |    |
|      |      |                          |             |    |
|      |      | (Y.0 Y. 100)             |             |    |

![](_page_8_Picture_10.jpeg)

### 2. bounceonedge()

- The function checks to see if its sprite is touching the edge of the screen with the move function and if it is, the sprite will point in a direction that mirrors the direction from which it was coming.
- It uses a line perpendicular to the edge to determine the reflection angle. ulletExample

```
sprite = Sprite('Tobi')
sprite gotoxy(0,0)
sprite.setrotationstyle('left-right')
while 1:
 sprite.move(3)
 sprite.bounceonedge()
```

![](_page_9_Picture_5.jpeg)

![](_page_9_Figure_8.jpeg)

### 3. setrotationstyle()

The function changes the rotation style of the sprite in-project. There are three options for this function:

- **all around:** All around means the sprite can face any of the 360 degrees. It is the default.
- left-right: Left-right means sprite can only face left or right, and any other  $\bullet$ directions are rounded. The sprite will also be horizontally flipped when facing left in the left-right style.
- **don't rotate:** Don't rotate means that the sprite always faces as in 90°.  $\bullet$

![](_page_10_Picture_6.jpeg)

![](_page_10_Picture_12.jpeg)

#### 4. switchcostume()

This function is used to switch the sprite's costume to a specific costume. It takes only one input:

#### **Costume Name** – String

#### Example

sprite = Sprite('Tobi')
sprite.switchcostume("Tobi walking 1")
sprite.say("Tobi Walking 1", 2)
sprite.switchcostume("Tobi walking 2")
sprite.say("Tobi Walking 2", 2)

![](_page_11_Picture_6.jpeg)

![](_page_11_Picture_7.jpeg)

![](_page_11_Picture_9.jpeg)

### 5. nextcostume()

The function changes its sprite's costume to the next one in the costume pane, but if the current costume is the last in the list, the function will loop to the first.

```
sprite = Sprite('Tobi')
sprite.switchcostume("Tobi walking 1")
sprite.say(sprite.costume("name"), 2)
sprite.nextcostume()
sprite.say(sprite.costume("name"), 2)
```

![](_page_12_Picture_4.jpeg)

![](_page_12_Picture_6.jpeg)

![](_page_12_Picture_8.jpeg)

### 6. gotoxy()

This function is used to change the sprite's specified x and y coordinates on the stage. It takes two inputs:

X Position – Integer from -240 to 240

**Y Position** – Integer from -180 to 180

### Example

```
sprite = Sprite('Tobi')
sprite.gotoxy(0, 0)
sprite.say('This is Center', 2)
sprite.gotoxy(100, 100)
sprite.say('This is Top Right', 2)
sprite.gotoxy(-100, -100)
sprite.say('This is Bottom Left', 2)
```

![](_page_13_Picture_7.jpeg)

![](_page_13_Figure_8.jpeg)

![](_page_13_Picture_10.jpeg)

![](_page_14_Picture_0.jpeg)

### **Tobi Walking**

In this activity, we will create a project that will make Tobi (a character) walk on the stage region in PictoBlox.

![](_page_14_Picture_3.jpeg)

![](_page_14_Picture_4.jpeg)

![](_page_14_Picture_6.jpeg)

![](_page_15_Picture_0.jpeg)

First, select the Tobi.py file from the Project Files section and by default, the 1. syntax will be written in sprite as an object.

sprite = Sprite('Tobi')

- 2. We need to change the sprite's position along the x-axis and y-axis, for that we need to include gotoxy() function in a below-given manner: sprite.gotoxy(0, -100)
- Next, we will set up the rotation style to "left-right". 3. sprite.setrotationstyle("left-right")

![](_page_15_Picture_5.jpeg)

![](_page_15_Picture_9.jpeg)

## **TOBI WALKING**

- 4. Now we need to use the while loop.
- The basic working of a while loop is, that a condition is evaluated before processing the body of the loop.
- If the condition is true, then the body of a loop is executed.
- Once the condition becomes false, the control goes out of the loop.
- After exiting the loop, the control goes to the statements which are immediately after the loop.
- Since we want the loop to be executed infinitely, we will set the condition to True or 1.

while 1:

![](_page_16_Picture_8.jpeg)

![](_page_16_Picture_10.jpeg)

![](_page_17_Picture_0.jpeg)

5. Next we will add the sprite functions: move(), bounceonedge() **nextcostume()** in the while loop:

```
sprite = Sprite('Tobi')
import time
sprite.gotoxy(0, -100)
sprite.setrotationstyle("left-right")
while 1:
 sprite.move(10)
 sprite.bounceonedge()
```

![](_page_17_Picture_3.jpeg)

#### and

![](_page_17_Picture_6.jpeg)

## **TOBI WALKING**

- An animation is created when the costume of the sprite is switched rapidly.
- But we should have the right costumes as well.
- Delete all costumes except Tobi Walking 1 and Tobi Walking 2.

![](_page_18_Picture_4.jpeg)

![](_page_18_Picture_5.jpeg)

![](_page_18_Picture_7.jpeg)

## **TOBI WALKING**

• The final code is:

```
sprite = Sprite('Tobi')
import time
sprite.gotoxy(0, -100)
sprite.setrotationstyle("left-right")
while 1:
 sprite.move(10)
 sprite.bounceonedge()
 sprite.nextcostume()
 time.sleep(0.1)
```

![](_page_19_Picture_3.jpeg)

• Save the file as **Tobi Walking.sb3** on your system.

![](_page_19_Picture_5.jpeg)

![](_page_19_Picture_7.jpeg)

![](_page_20_Picture_0.jpeg)

![](_page_20_Picture_1.jpeg)

![](_page_20_Picture_2.jpeg)

![](_page_20_Picture_3.jpeg)