# ACTIVITY SHEET

***SESSION 2***

## Animations with Python

In this activity, we are going to understand how to make animations within the PictoBlox python environment. As an activity we will make the Tobi sprite walk to and fro on the PictoBlox stage.

*This activity sheet belongs to* _____

## MATERIALS REQUIRED

□ Computer/Laptop/Tab with PictoBlox installed

## STEP-BY-STEP

Follow the steps below:

1. Open a **New file** in PictoBlox:
   - **1.1.** Click on **PictoBlox** icon on your device and
   - **1.2.** Select the coding environment as **Python Coding**
2. The sprite object for Tobi will already be initiated.

```
sprite = Sprite('Tobi')
```

3. We need to change the sprite's position along the x-axis and y-axis, for that we need to include **gotoxy([1], [2])** function in a below-given manner:
   - **3.1.** Parameters of **gotoxy([1], [2])** function**:**
     - **3.1.1. [1] = value on x-axis**
     - **3.1.2. [2] = value on y-axis**

```
sprite.gotoxy(0, -100)
```

4. Next, we will set up the rotation style to "left-right".

```
sprite.setrotationstyle("left-right")
```

5. We will use the following functions for Tobi using the sprite instance:
   - **5.1. move([1]):** To make Tobi move on the stage. Here [1] = Number of steps.
   - **5.2. bounceonedge():** This function makes Tobi turn around when it reaches the edge of stage.
   - **5.3. nextcostume():** Changes the costume of Tobi, to make it look like Tobi is moving.

```
sprite.move(10)
sprite.bounceonedge()
sprite.nextcostume()
```

*STEP 5*

6. In order to make Tobi move to and fro forever, we will run the above 3 lines of code forever. We will do this using the **while loop**. The while loop can be described as follows:

   **6.1.** A condition loop which is evaluated before running the contents of the loop.

   **6.2.** The contents of the loop are run only if the condition evaluates to **True**.

   **6.3.** If the condition becomes **False**, the loop stops running and the code written after the loop is then executed.

   **6.4.** Since we want the loop to be executed forever (until we stop the code), we will set the condition to always return True: 1/True.

```
while 1:
```

7. We will add the code within the **while True** loop for running the code inside it (Tobi walking) forever.

```
while 1:
  sprite.move(10)
  sprite.bounceonedge()
  sprite.nextcostume()
```

8. When we run the code now, we can see Tobi is walking very fast and switching the costume quickly as well. To make the code run slower, we can add a delay in the loop. We do this using the **sleep([1])** function from the **time** library. This function pauses the code for the specified number of seconds added as parameter. Let us pause the code for **0.1** seconds on each loop.

9. The final code will be as follows:

```
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)
```

**10.** Press the **Run** button to test the code.

## SAVING THE PROGRAM

Save the project file as **Tobi Walking**. Check Activity Sheet CH1 if you missed how to save the project.