

# Python & Java {4} Teachers



# Initials

Level 1 - Python



Funded by the  
Erasmus+ Programme  
of the European Union

{4}

Python & Java 4 Teachers

# Introduction

You probably think that art and programming couldn't be more unlike, but actually they might be more similar than you think!



There are many modern artists whose main medium is code. For example: Mark Dorf, Josh Davis and Kyle McDonald.

Here are a few websites that merge together art and programming:

- [Silk – Interactive Generative Art \(weavesilk.com\)](http://weavesilk.com)
- [Dream by WOMBO](#)



# Signing Your Work



For centuries artists have signed their art. The artist's signature has become a key part of the art-creating process – it signifies that the art is finished.

No matter what your signature looks like, what form it takes or where you put it, no art work is complete without one!  
This is no different when we mix together  
Technology and Art!



# Task

In this project we will be using turtle graphics in Python to write out our initials.

Turtle is a python library that draws graphics.

After we import Turtle we can give it commands like forward, backward, right, left etc.

## Extension:

- Change the colour of the letters.
- Change the speed of the turtle.



# Process

This program should:

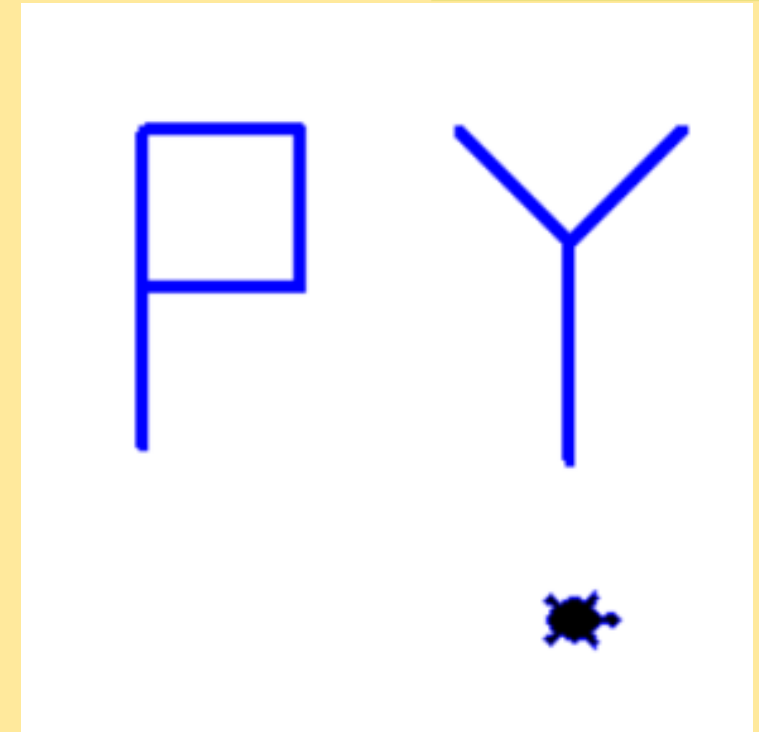
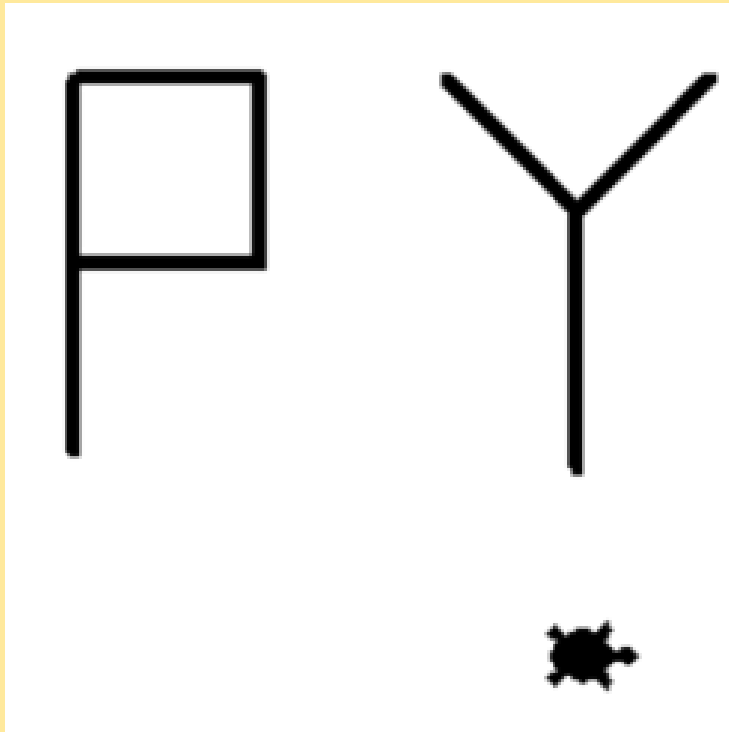
- ✓ Import the python Turtle graphics Library
- ✓ Control the speed fill colour and line color of the turtle,
- ✓ Apply previous knowledge of angles to help move the turtle along.



# What it will look like...

{4}

Python & Java 4 Teachers



# Python Libraries

Python Libraries are a set of useful functions that eliminate the need for writing codes from scratch.

```
1 from random import *
2
3 import Tkinter
4
5 import statistics
6
```

They can be brought into the program using the 'import' keyword and can save valuable time when writing complex programs.

One common example of a python library is the 'random' module, used often for generating pseudo-random numbers

# Step 1

Import turtle library.

Here we have imported the Turtle library which mean we can now create graphics with the code we will write.

```
1 #Writing PY in turtle graphics as level 1a
2 #Importing Turtle library
3
4 from turtle import *
5
6 #Creating the turtle shape cursor
7 shape("turtle")
8
9 #pensize changes the thickness of the line
10 pensize("4")
11
```

By changing the shape we can change the shape of the cursor to look like a turtle for example. The pen size determines the thickness of the lines the turtle draws.

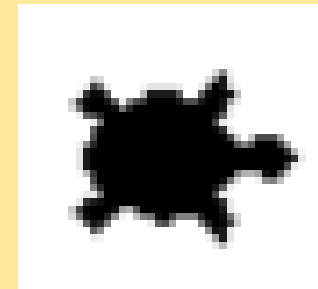


# What it will look like...

{4}

Python & Java 4 Teachers

```
1 #Writing PY in turtle graphics as level 1a
2 #Importing Turtle library
3
4 from turtle import *
5
6 #Creating the turtle shape cursor
7 shape("turtle")
8
9 #pensize changes the thickness of the line
10 pensize("4")
11
```



# Simple Turtle Commands

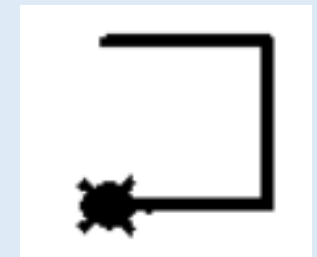
The turtle understands simple commands such as left turns, right turns, move forwards and move backwards.

## Simple Turtle commands:

- `fd(10)` – moves the turtle forward 10 pixels,
- `backward(5)` – moves the turtle backward by 5 pixels,
- `right(35)` – moves the turtle clockwise by an angle of 35 degrees,
- `left(55)` – moves the turtle counter-clockwise by an angle of 55 degrees;
- `goto(x, y)` – moves the turtle to the position x, y.

Another command turtle uses is `pu()` and `pd()`, which stands for pen up and pen down. This means that the turtle can move without leaving a line behind it.

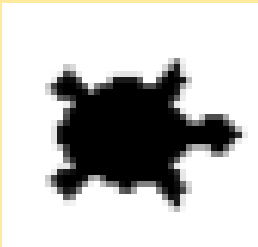
```
54 fd(50)
55 rt(90)
56 fd(50)
57 lt(90)
58 backward(50)
```



# Step 2

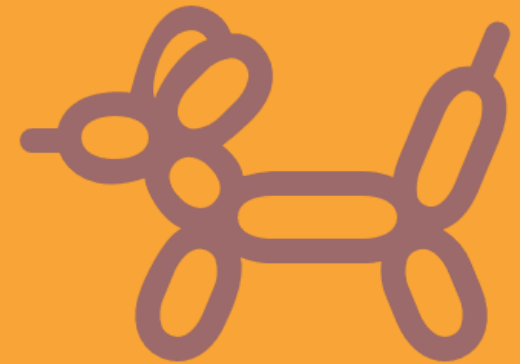
Drawing the letter 'P'.

By default the turtle faces the right whenever a program starts.



```
19 #Main code
20 #Capital P instructions
21 lt(90)
22 fd(100)
23 rt(90)
24 fd(50)
25 rt(90)
26 fd(50)
27 rt(90)
28 fd(50)
29 #Finished Letter p
```

Now turtle has been imported we can give it simple commands in order to successfully draw the letters. Since it starts facing the right, we have to turn it counter-clockwise 90 degrees so we can start drawing the 'P'. The turtle then moves forward 100 pixels and follows the rest of the commands to complete the 'P'.

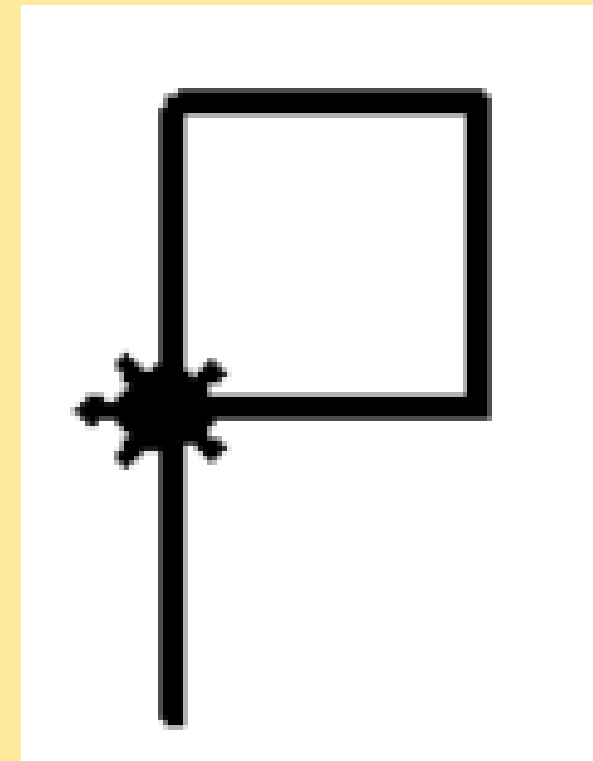


# What it will look like...

{4}

Python & Java 4 Teachers

```
19 #Main code
20 #Capital P instructions
21 lt(90)
22 fd(100)
23 rt(90)
24 fd(50)
25 rt(90)
26 fd(50)
27 rt(90)
28 fd(50)
29 #Finished Letter p
```



# Step 3

Moving the turtle over.

```
27 #Finished Letter p
28 #start lifting up pen to move for the Y
29 pu()
30 rt(180)
31 fd(100)
32 lt(90)
33 fd(50)
34 pd()
35 #Finished moving now to write the capital Y
```

Now have finished the first letter we will use the pen up function so that whilst the turtle is moving no extra lines will be created. Once the turtle has finished moving we use pen down so that we can now start drawing again.



# What it will look like...

{4}

Python & Java 4 Teachers

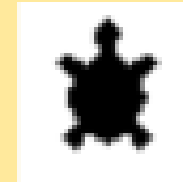
```
27 #Finished Letter p
28 #start lifting up pen to move for the Y
29 pu()
30 rt(180)
31 fd(100)
32 lt(90)
33 fd(50)
34 pd()
35 #Finished moving now to write the capital Y
```



# Step 4

Drawing the letter 'Y'.

```
37 #Finished moving now to write the capital Y
38 #A 'Y' moves in a diagonal direction so we turn the turtle 135^.
39 rt(135)
40 fd(50)
41 lt(90)
42 fd(50)
43 #need to turn back onself to complete the letter
44 lt(180)
45 fd(50)
46 lt(45)
47 fd(70)
48 #Finished the letter 'Y'
```



The turtle finished moving facing upwards, so before moving again the turtle has to turn clockwise 135 degrees. The turtle then creates a 'V' shape for the top of the 'Y'. Line 43 explains that the turtle needs to turn back on itself so that it can draw the line in the middle.

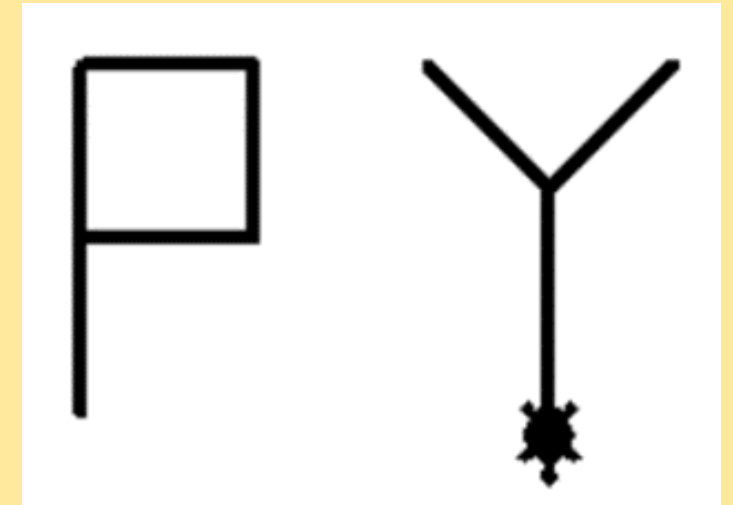


# What it will look like...

```
37 #Finished moving now to write the capital Y
38 #A 'Y' moves in a diagonal direction so we turn the turtle 135^.
39 rt(135)
40 fd(50)
41 lt(90)
42 fd(50)
43 #need to turn back onself to complete the letter
44 lt(180)
45 fd(50)
46 lt(45)
47 fd(70)
48 #Finished the letter 'Y'
```

{4}

Python & Java 4 Teachers



## CHALLENGE:

➤ Try and draw your own initials in Turtle Graphics.

# Extension

```
1 #Writing PY in turtle graphics as level 1a
2 #Importing Turtle library
3
4 from turtle import *
5
6 #Creating the turtle shape cursor
7 shape("turtle")
8
9 #pensize changes the thickness of the line
10 pensize("4")
11
12 #EXTENSION
13 pencolor("blue")
14
15 #delay changes the speed of the turtle
16 delay(0)
17
```

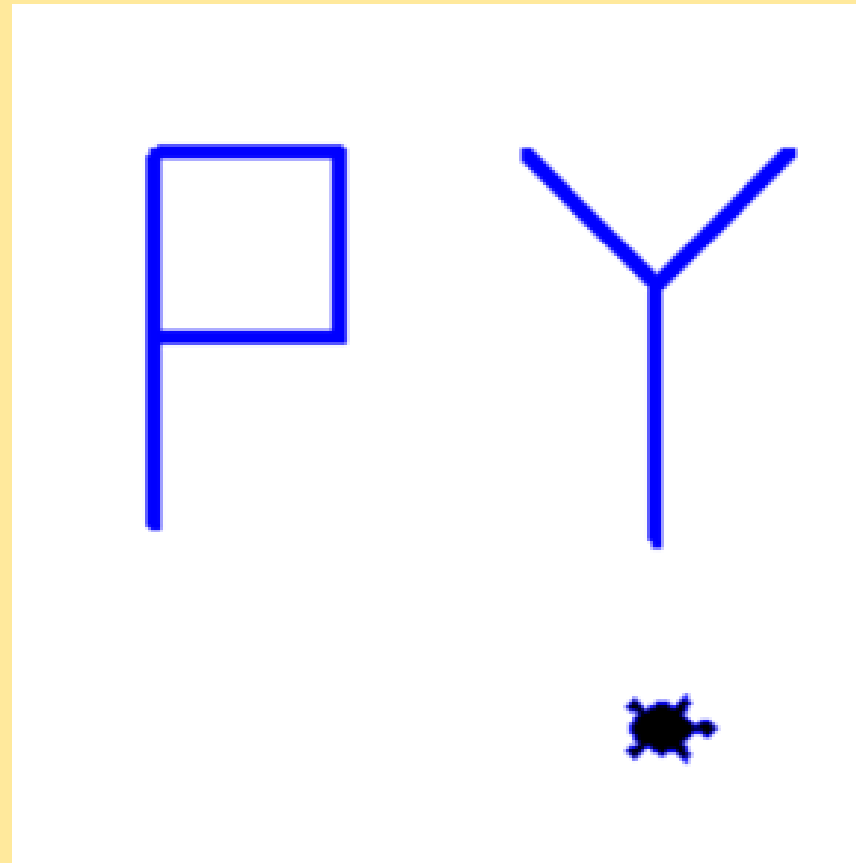
Here, on line 13 we can use 'pencolor' to determine the colour that the letters will be, for example: blue. Following this, by using delay() we can change the speed of the turtle to go as fast or as slow as you prefer.



# What it should look like...

{4}

Python & Java 4 Teachers



# Final product:



## Python & Java 4 Teachers



```
1 #Writing PY in turtle graphics as level 1a
2 #Importing Turtle library
3
4 from turtle import *
5
6 #Creating the turtle shape cursor
7 shape("turtle")
8
9 #pensize changes the thickness of the line
10 pensize("4")
11
12 #EXTENSION
13 pencolor("blue")
14 #delay changes the speed of the turtle
15 delay(0)
16
17 #Main code
18 #Capital P instructions
19 lt(90)
20 fd(100)
21 rt(90)
22 fd(50)
23 rt(90)
24 fd(50)
25 rt(90)
26 fd(50)
27 #Finished Letter p
28 #start lifting up pen to move for the Y
29 pu()
30 rt(180)
31 fd(100)
32 lt(90)
33 fd(50)
34 pd()
35 #Finished moving now to write the capital Y
36 #A 'Y' moves in a diagonal direction so we turn the turtle 135°.
37 rt(135)
38 fd(50)
39 lt(90)
40 fd(50)
41 #need to turn back onself to complete the letter
42 lt(180)
43 fd(50)
44 lt(45)
45 fd(70)
46 #Finished the letter 'Y'
```

# Conclusion

## Learning Outcomes:

- ✓ You should have confidently been able to import a library into Python,
- ✓ You should be comfortable using angles to navigate the turtle along,
- ✓ You should be confident in manipulating the speed and colour within a turtle graphics program.



# Congratulations!

You have created your initials in Turtle Graphics!



{4}

Python & Java 4 Teachers