



SESSION - 5

MULTI LAYER GRAPH



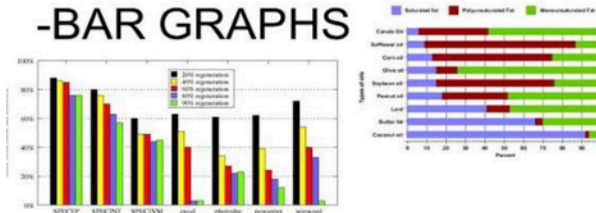
Learning Outcomes:

- **Remember:** The students will recall about Coordinate System .
- **Understand:** They will focus on understanding :
 - Different types of graph
 - Multi layer representation
 - Data Handling
- **Apply:** They will learn to apply the Library - pygal to generate graphs in the output .
- **Analyze:** They will check their understanding by developing a code .
- **Create:** They will create the code in EduBlocks

In Multilevel Graph, each layer of the graph is a subset of the original network (nodes and intra-edges)

The inter-edges only connect the same nodes on different layers.

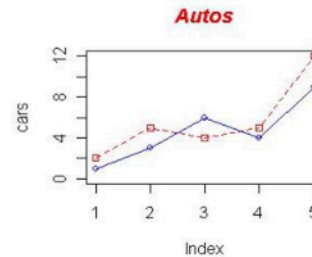
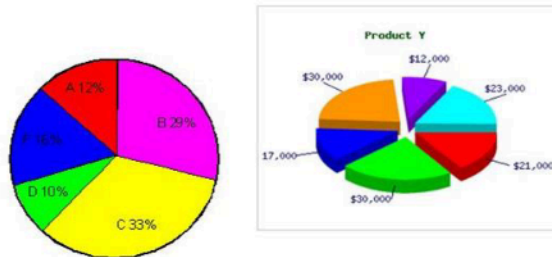
-BAR GRAPHS



-LINE GRAPHS



-PIE CHARTS





Apply & Create


TASK 01:-


</> WRITE A CODE TO CREATE A LINE GRAPH


Program


 Imports

 Variables

 Statements

 Text

 Math

 Logic

Imports

import time

import math

import random

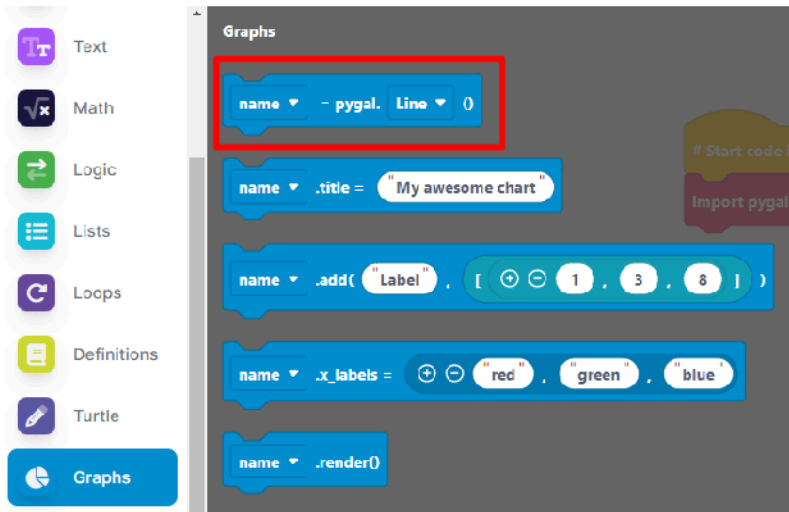
import pygal

Start code here

import pygal

```
1 #Start code here
2 import pygal
3
```

Program



The image shows a block editor interface with a sidebar on the left containing icons for Text, Math, Logic, Lists, Loops, Definitions, Turtle, and Graphs. The main workspace is titled "Graphs" and contains a sequence of blue blocks. The first block, "name ▾ = pygal. Line ▾ 0", is highlighted with a red rectangle. The second block is "name ▾ .title = 'My awesome chart'". The third block is "name ▾ .add('Label' , [⊕ ⊖ 1 . 3 . 8])". The fourth block is "name ▾ .x_labels = ⊕ ⊖ 'red' 'green' 'blue'". The fifth block is "name ▾ .render()".



The image shows a simplified block editor with three blocks: a yellow block "# Start code here", a pink block "import pygal", and a blue block "name ▾ = pygal. Line ▾ 0".

```
1 #Start code here
2 import pygal
3 name = pygal.Line()
4
```

Program

The image shows a block editor interface with a sidebar on the left containing icons for Text, Math, Logic, Lists, Loops, Definitions, Turtle, and Graphs. The main workspace is titled "Graphs" and contains several blue blocks. The first block is "name = pygal. Line ()". The second block, ".title = 'My awesome chart'", is highlighted with a red rectangle. The third block is ".add('Label' , [1 , 3 , 8])". The fourth block is ".x_labels = ['red' , 'green' , 'blue']". The fifth block is ".render()".

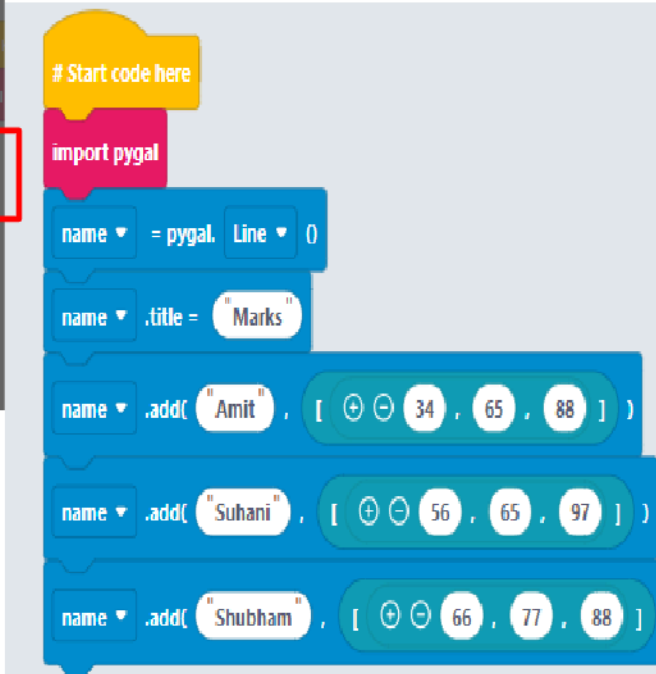
The image shows a simplified block editor interface. It starts with a yellow block "# Start code here", followed by a pink block "import pygal". Then there are two blue blocks: "name = pygal. Line ()" and "name .title = 'Marks'".

```
1 #Start code here
2 import pygal
3 name = pygal.Line()
4 name.title = "Marks"
5
```

Program



A screenshot of a Scratch-like block palette and workspace. The palette on the left has categories: Text, Math, Logic, Lists, Loops, Definitions, and Turtle. The 'Graphs' category is selected. The workspace shows a 'name' variable connected to 'pygal. Line()' and '.title = "My awesome chart"'. A red box highlights the 'name.add()' block with arguments 'Label', '1', '3', and '8'. Below it is an 'x_labels' block with 'red', 'green', and 'blue' labels. At the bottom is a '.render()' block.



A screenshot of a Scratch-like block code area. It starts with a yellow 'Start code here' block, followed by a pink 'import pygal' block. Then, a blue 'name = pygal. Line()' block, a blue 'name.title = "Marks"' block, and three blue 'name.add()' blocks. The first 'name.add()' block has arguments 'Amit', '34', '65', and '88'. The second has 'Suhani', '56', '65', and '97'. The third has 'Shubham', '66', '77', and '88'.

Code

```
1 #Start code here
2 import pygal
3 name = pygal.Line()
4 name.title = "Marks"
5 name.add("Amit", [34, 65, 88])
6 name.add("Suhani", [56, 65, 97])
7 name.add("Shubham", [66, 77, 88])
8
```


Program

Start code here

import pygal

name = pygal. Line 0

name .title = "Marks"

name .add("Amit" , [+ - 34 , 65 , 88])

name .add("Suhani" , [+ - 56 , 65 , 97])

name .add("Shubham" , [+ - 66 , 77 , 88])

name .x_labels = + - "English" , "Maths" , "Science"

name .render()

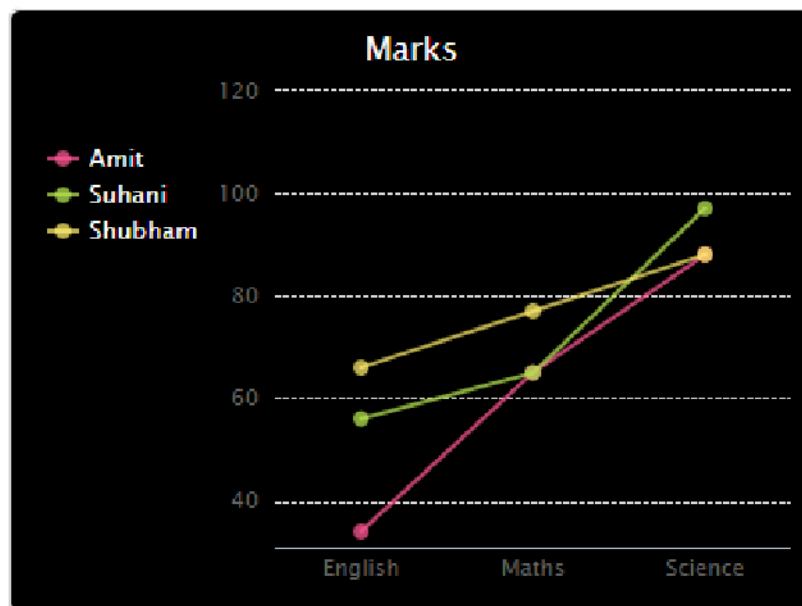
Code

```
1 #Start code here
2 import pygal
3 name = pygal.Line()
4 name.title = "Marks"
5 name.add("Amit", [34, 65, 88])
6 name.add("Suhani", [56, 65, 97])
7 name.add("Shubham", [66, 77, 88])
8 name.x_labels = "English", "Maths", "Science"
9 name.render()
10
```

Output

Code

Output



Homework

Plot the graph for the following data

Crop	Economic produce	Area (Mha)			Crop production (Mt)		
		2010/11	2020/21	2030/31	2010/11	2020/21	2030/31
Rice	Foodgrains	42.9	48.1	50.3	96.0	109.9	123.2
Wheat	Foodgrains	29.1	33.7	36.6	87.0	108.2	121.1
Jowar (Sweet Sorghum)	Foodgrains	7.4	5.2	3.4	7.0	6.0	5.7
Bajra	Foodgrains	9.6	9.3	8.8	10.4	11.4	12.3
Maize	Foodgrains	8.6	8.4	9.0	21.7	24.8	28.3
Other cereals	Foodgrains	2.9	2.1	1.5	4.6	3.9	3.8
Gram	Foodgrains	9.2	8.9	8.7	8.2	8.4	8.6
Tur (Arhar)	Foodgrains	4.4	4.4	4.7	2.9	3.1	3.3
Lentil (Masur)	Foodgrains	1.6	1.7	1.9	0.9	1.2	1.4
Other pulses	Foodgrains	11.2	12.8	13.2	6.2	6.3	6.8