My Project

Generated by Doxygen 1.8.12

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 Axes Namespace Reference

Functions

- def get_axes (window, markings, scale_x, scale_y)

  On a Canvas object, constructs X and Y axes appropriately placed.

Variables

- dictionary x_coordinates = { }
- dictionary y_coordinates = { }

4.1.1 Function Documentation

4.1.1.1 get_axes()

```
def Axes.get_axes (
              window,
              markings,
              scale_x,
              scale_y )
```

On a Canvas object, constructs X and Y axes appropriately placed.

Returns a new canvas with this done. Vertical line: Start from the middle with respect to x, draw a line from top to bottom with respect to y. Horizontal line: Start from the middle with respect to y, draw a line from left to right with respect to x. Then calls _get_labels() to add the labels to the two lines.

Parameters

window	The window to draw on.
markings	The number of labels to put on each axis.
scale_x	The scale value to use on the x axis.
scale_y	The scale value to use on the y axis.

4.1.2 Variable Documentation

4.1.2.1 x_coordinates

```
dictionary Axes.x_coordinates = { }
```

4.1.2.2 y_coordinates

```
dictionary Axes.y_coordinates = { }
```

4.2 Graph Namespace Reference

Classes

- class Graph

  xyGraph constructing class
  Author: Hatim Rehman
  This class represents a graph object that can plot points with data contained within a list of tuples
  [(x1, y1), (x2, y2), ... , (xn, yn)]

Functions

- def coord (x, y)

Variables

- Graph = Graph(6 , [(-4,4), (-2,1), (-1,1) , (1,1), (2,2), (4,5), (5,3)])

4.2.1 Function Documentation

4.2.1.1 coord()

```
def Graph.coord (
              x,
              y )
```

4.2.2 Variable Documentation

4.2.2.1 Graph

```
Graph.Graph = Graph( 6 , [ (-4,4), (-2,1), (-1,1) , (1,1), (2,2), (4,5), (5,3) ])
```

4.3 InputParser Namespace Reference

Functions

- def clean_data (data)

  This function breaks apart the user input and assigns the coordinate pairs to a dictionary (data structure).
- def checktype (x, y)

  This function checks if the data input type is correct(i.e.

4.3.1 Function Documentation

4.3.1.1 checktype()

```
def InputParser.checktype (
              x,
              y )
```

This function checks if the data input type is correct(i.e.

integers or floats). If data type is not correct function outputs an exception message.
Author: Sarthak Desai

Parameters

x	is a list of x values from all the co-ordinate pairs.
y	is a list of y values from all the co-ordinate pairs.

4.3.1.2 clean_data()

```
def InputParser.clean_data (
              data )
```

This function breaks apart the user input and assigns the coordinate pairs to a dictionary (data structure).

If there are inconsistencies in the user input (for example a missing y value)the function prints an exception message.
Author: Sarthak Desai

Parameters

data	is the list of co-ordinate pairs inputted by the user.

4.4 Scale Namespace Reference

Functions

- def get_scale (data_set, round_to=0)

  get_scale
  Author: Louis Bursey
  Function to get the appropriate max and mins for the scale of the graph
  Use round_to to create padding for the graph, ie if the graph goes in increments of 5, set round_to to 5 to keep the graph neat.

4.4.1 Function Documentation

4.4.1.1 get_scale()

```
def Scale.get_scale (
              data_set,
              round_to = 0 )
```

get_scale
Author: Louis Bursey
Function to get the appropriate max and mins for the scale of the graph
Use round_to to create padding for the graph, ie if the graph goes in increments of 5, set round_to to 5 to keep the graph neat.

Returns a tuple in the format [x,y]

Parameters

data_set,the	set of data being graphed, in a (x,y) dictionary
round_to,an	optional parameter to make the scale a multiple of round_to

Chapter 5

Class Documentation

5.1 Graph.Graph Class Reference

xyGraph constructing class
Author: Hatim Rehman
This class represents a graph object that can plot points with data contained within a list of tuples
[(x1, y1), (x2, y2), ... , (xn, yn)]

Public Member Functions

- def __init__ (self, n, data=None)

  The constructor method Takes in 3 parameters.
- def plot_point (self, coord, kwargs)

  Plot points method Takes in a coordinate to plot, with keyword args that may be used to style the coordinate.
- def plot_points (self, data=None, kwargs)

  Plot points method takes in multiple coordinates and calls the plot_point() method from its own API on each method.
- def plot_points_with_line (self, data, kwargs)

  Plot points with line method takes in multiple coordinates and calls the plot_point() method from its own API on each method.
- def plot_function (self, func, x_interval=None, kwargs)

  Plots a python function onto the graph Outputs a graph with the func parameter plotted.

Public Attributes

- data
- markings
- scale
- scale_x
- scale_y
- master
- graph
- graph_height
- graph_width
- x_offset
- y_offset
- dx
- dy

5.1.1 Detailed Description

xyGraph constructing class
Author: Hatim Rehman
This class represents a graph object that can plot points with data contained within a list of tuples
[(x1, y1), (x2, y2), ... , (xn, yn)]

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
def Graph.Graph.__init__ (
              self,
              n,
              data = None )
```

The constructor method Takes in 3 parameters.

Outputs a graph with plotted data that is entered as a parameter.

Parameters

self	The object pointer.
n	The number of markings to appear on the x and y axes
data	The data to plot

5.1.3 Member Function Documentation

5.1.3.1 plot_function()

```
def Graph.Graph.plot_function (
              self,
              func,
              x_interval = None,
              kwargs )
```

Plots a python function onto the graph Outputs a graph with the func parameter plotted.

Parameters

self	The object pointer.
func	A python function that takes in a double value and returns a double value
x_interval	A list of double values that the function should pass through [x1, ...\, xn]
kwargs	Keyword arguments for Tkinter's create_circle() method

5.1.3.2 plot_point()

```
def Graph.Graph.plot_point (
```

```
                self,
                coord,
                kwargs )
```

Plot points method Takes in a coordinate to plot, with keyword args that may be used to style the coordinate.

Outputs a graph with the coord parameter plotted.

Parameters

self	The object pointer.
coord	A dictionary with the format { 'x': *value*, 'y': *value* }
kwargs	Keyword arguments for Tkinter's create_circle() method

5.1.3.3 plot_points()

```
def Graph.Graph.plot_points (
                self,
                data = None,
                kwargs )
```

Plot points method takes in multiple coordinates and calls the plot_point() method from its own API on each method.

Outputs a graph with the data parameter plotted.

Parameters

self	The object pointer.
data	A list of tuples [(x1, y1), (x2, y2), ... , (xn, yn)]
kwargs	Keyword arguments for Tkinter's create_circle() method

5.1.3.4 plot_points_with_line()

```
def Graph.Graph.plot_points_with_line (
                self,
                data,
                kwargs )
```

Plot points with line method takes in multiple coordinates and calls the plot_point() method from its own API on each method.

Then calls its private function generating method that creates a polynomial that passes through each point using the Lagrange polynomial interpolation theorem. Outputs a graph with the data parameter plotted.

Parameters

self	The object pointer.
data	A list of tuples [(x1, y1), (x2, y2), ... , (xn, yn)]
kwargs	Keyword arguments for Tkinter's create_circle() method

5.1.4 Member Data Documentation

5.1.4.1 data

Graph.Graph.data

5.1.4.2 dx

Graph.Graph.dx

5.1.4.3 dy

Graph.Graph.dy

5.1.4.4 graph

Graph.Graph.graph

5.1.4.5 graph_height

Graph.Graph.graph_height

5.1.4.6 graph_width

Graph.Graph.graph_width

5.1.4.7 markings

Graph.Graph.markings

5.1.4.8 master

Graph.Graph.master

5.1.4.9 scale

Graph.Graph.scale

5.1.4.10 scale_x

Graph.Graph.scale_x

5.1.4.11 scale_y

Graph.Graph.scale_y

5.1.4.12 x_offset

Graph.Graph.x_offset

5.1.4.13 y_offset

Graph.Graph.y_offset

The documentation for this class was generated from the following file:

- C:/Users/sarth/Desktop/All code/Graph.py

Chapter 6

File Documentation

6.1 C:/Users/sarth/Desktop/All code/Axes.py File Reference

Namespaces

- Axes

Functions

- def Axes.get_axes (window, markings, scale_x, scale_y)

  On a Canvas object, constructs X and Y axes appropriately placed.

Variables

- dictionary Axes.x_coordinates = { }
- dictionary Axes.y_coordinates = { }

6.2 C:/Users/sarth/Desktop/All code/Graph.py File Reference

Classes

- class Graph.Graph

  xyGraph constructing class
  Author: Hatim Rehman
  This class represents a graph object that can plot points with data contained within a list of tuples
  [(x1, y1), (x2, y2), ... , (xn, yn)]

Namespaces

- Graph

Functions

- def Graph.coord (x, y)

Variables

- Graph.Graph = Graph(6 , [(-4,4), (-2,1), (-1,1) , (1,1), (2,2), (4,5), (5,3)])

6.3 C:/Users/sarth/Desktop/All code/InputParser.py File Reference

Namespaces

- InputParser

Functions

- def InputParser.clean_data (data)

  This function breaks apart the user input and assigns the coordinate pairs to a dictionary (data structure).
- def InputParser.checktype (x, y)

  This function checks if the data input type is correct(i.e.

6.4 C:/Users/sarth/Desktop/All code/Scale.py File Reference

Namespaces

- Scale

Functions

- def Scale.get_scale (data_set, round_to=0)

  get_scale
  Author: Louis Bursey
  Function to get the appropriate max and mins for the scale of the graph
  Use round_to to create padding for the graph, ie if the graph goes in increments of 5, set round_to to 5 to keep the graph neat.

Index