

■ Oscilloscope

kobayashikorio@gmail.com

2016/10/31

Oscilloscope is a OOP style programming tool for monitoring the variables in the program. With OOP style, encapsulation is performed. Oscilloscope has two types of windows. One is variable viewer, and one is variable trend viewer.

```
PROBE - CLASS;
scope[nam_] := Module[{dx, xrange = 50, data = ConstantArray[0, 50], ps = 0},

  PROBE - METHOD;
  probe[nam[x_]] ^= (
    If[ps == Infinity, Return[]];
    data = RotateLeft[data];
    data[[-1]] = x;
    dx = x;
    Pause[ps]);

  SHOW - METHOD;
  showBox[nam] ^= CreateDocument[{Column[{Pane[Row[{nam, " -> ", Dynamic[dx]},
    BaseStyle -> {FontSize -> 16, FontFamily -> "Times"}],
    ImageSize -> {350, 320}],
    Pane[Row[{"Pausing-time ", SetterBar[Dynamic[ps], {Infinity, 0, 0.2, 1}]}],
    ImageMargins -> 10]
  }
  ]},
  WindowTitle -> nam,
  WindowElements -> {"MagnificationPopUp"}, WindowSize -> {400, 315}];

  SHOW - PLOT - METHOD;
  showPlotBox[nam] ^=
    CreateDocument[{Column[{Pane[Row[{nam, " -> ", Dynamic[dx]},
      BaseStyle -> {FontSize -> 16, FontFamily -> "Times"}],
      ImageSize -> {400, 30}],
      Pane[Dynamic[ListLinePlot[data, PlotRange -> {{0, xrange}, Automatic}]]],
      Pane[Row[{"Pausing-time ", SetterBar[Dynamic[ps], {Infinity, 0, 0.2, 1}]}],
      ImageMargins -> 10]
    }
  ]},
  WindowTitle -> nam,
  WindowElements -> {"MagnificationPopUp"}, WindowSize -> {400, 315}];
];
```

Before the Oscilloscope running, OOP instances should be constructed. Arbitrary number of Oscilloscope widows can be create.

```
objectList = {$x, $y};
Map[scope, objectList];
showPlotBox[$x]; showBox[$x]; showPlotBox[$y];
```

following is a sample program to monitor the variable,

```
Do[probe[$x[Random[]]]; probe[$y[Random[]]]; Pause[0.2], {100}];
```

Because the simbol “probe” is always global, upper sample program will run without evaluation of “scope” class, and also yo can close windows any time.

? probe

```
Global`probe
```