Chapter 11
Advanced GUI Programming Using Tkinter

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

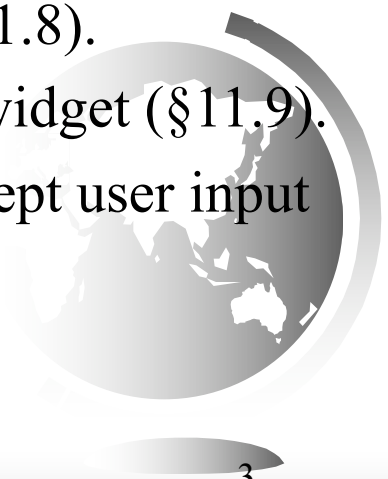
Motivations

The preceding chapter introduced basic GUI programming using Tkinter. This chapter introduces combo boxes, menus, handling mouse and key events, animations, scrollbars, and dialog boxes.

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Objectives

- To create a combo box for selecting a single item using **OptionMenu**. (§11.2).
- To create applications that contain menus (§11.3).
- To create applications that contain popup menus (§11.4).
- To bind mouse and key events on a widget to a callback function for processing events (§11.5).
- To write a GUI program for finding and displaying the points that are nearest to each other (§11.6).
- To develop animations (§11.7).
- To write a GUI program for animating bouncing balls (§11.8).
- To use scroll bars to scroll through the contents of a text widget (§11.9).
- To use standard dialog boxes to display messages and accept user input (§11.10).



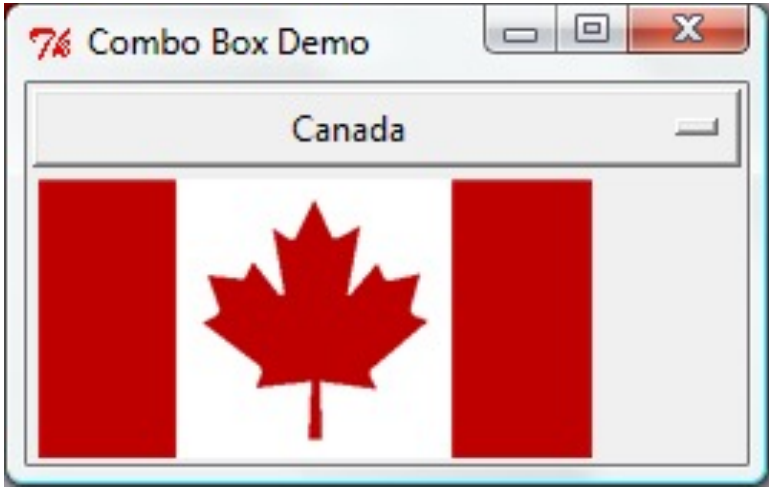
© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

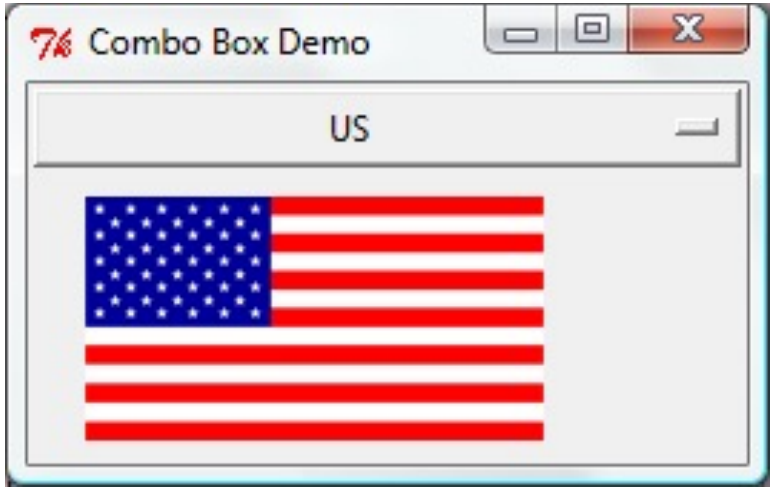
Combo Boxes

- A *combo box*, also known as a *choice list* or *drop-down list*, contains a list of items from which the user can choose. A combo box is useful for limiting a user's range of choices and avoids the cumbersome validation of data input. Tkinter uses the **OptionMenu** class to create a combo box.

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Combo Boxes



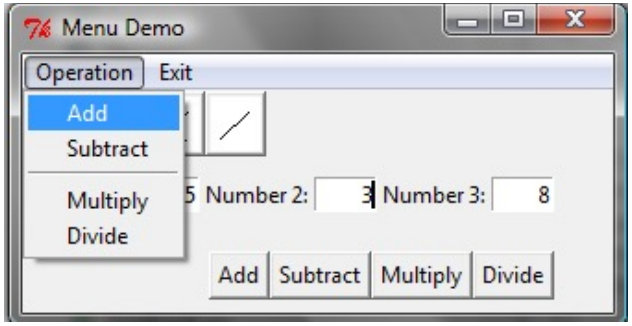


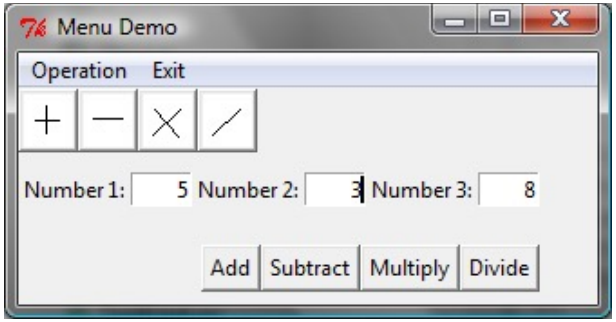
ComboBoxDemo

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Menus

Tkinter provides a comprehensive solution for building graphical user interfaces.

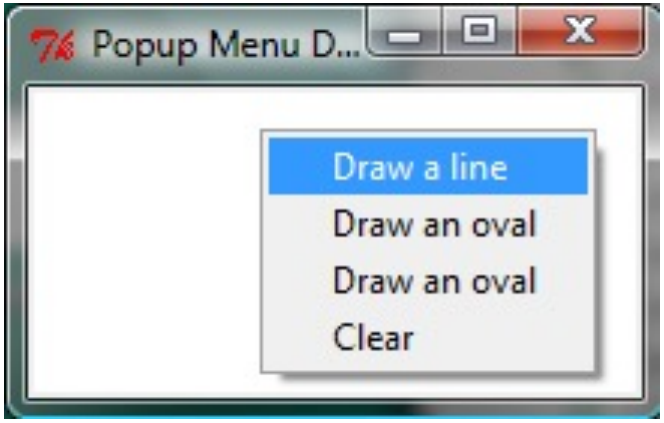


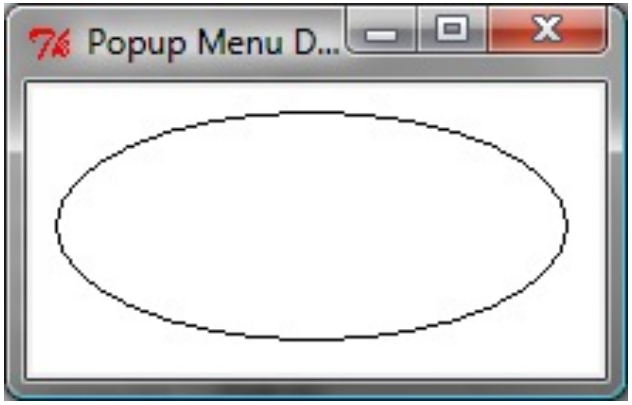


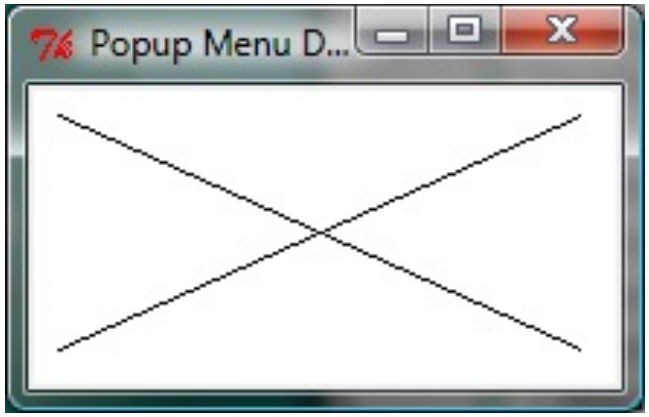
MenuDemo

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Popup Menus







PopupMenuDemo

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Mouse and Key Events

```
widget.bind(event, handler)
```

```
def popup(event):
    menu.post(event.x_root, event.y_root)
```

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Events

Event	Description
<Button-*i*>	Button-1, Button-2, and Button-3 are for left, middle, or right buttons. When a mouse button is pressed over the widget, Tkinter automatically grabs the mouse pointer location. ButtonPressed-i is synonymous to Button-i.
<B*i*-Motion>	An event occurs, when a mouse button is moved while being held down on the widget.
<ButtonReleased-*i*>	An event occurs, when a mouse button is released.
<Double-Button-*i*>	An event occurs, when a mouse button is double-clicked.
<Triple-Button-*i*>	An event occurs, when a mouse button is triple-clicked.
<Enter>	An event occurs, when a mouse pointer enters the widget.
<Leave>	An event occurs, when a mouse pointer leaves the widget.
<Return>	An event occurs, when the Enter key is pressed. You can bind any key such as <A>, <B>, <Up>, <Down>, <Left>, <Right> in the keyboard with an event.
<Key>	An event occurs, when a key is pressed.
<Shift-A>	An event occurs, when the Shift+A keys are pressed. You use Alt, Shift, and Control to combine with other keys.

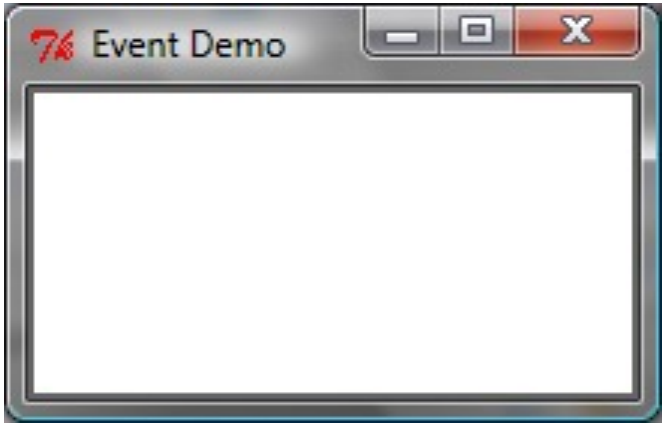
© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

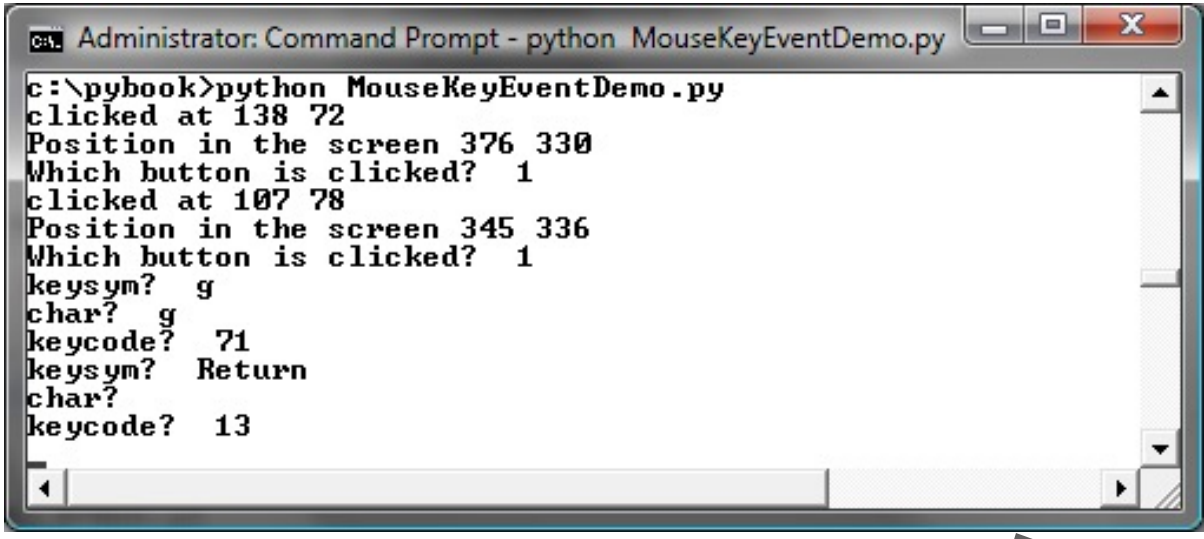
Event Properties

Event	Description
widget	The widget object that fires this event.
x and y	The current mouse location in the widget pixels.
x__root and y_root	The current mouse position relative to the upper left corner of the screen, in pixels.
num	The button number (1, 2, 3), indicating which mouse button was clicked.
char	The char entered from the keyboard for key events.
keysym	The key symbol for the key entered from the keyboard for key events.
keycode	The key code for the key entered from the keyboard for key events.

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Mouse Key Event Demo

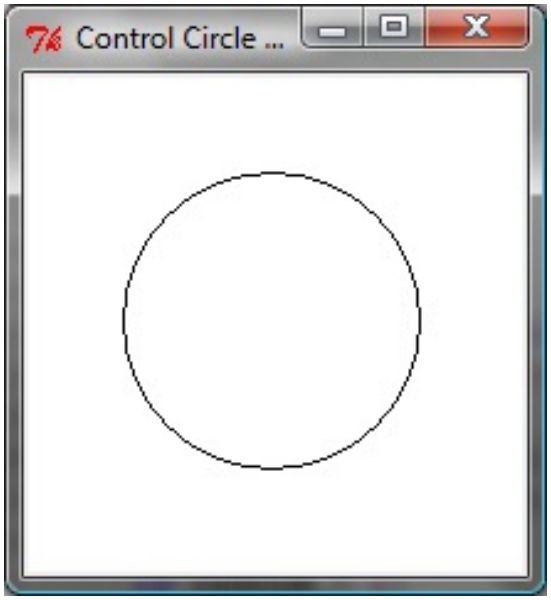


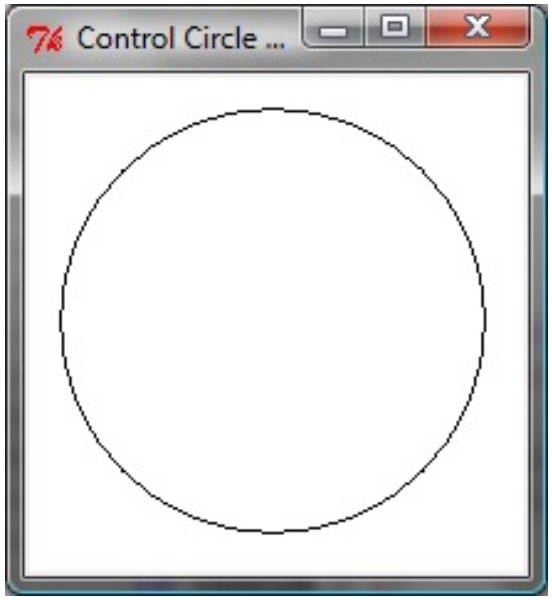


MouseKeyEventDemo

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Control Circle Demo

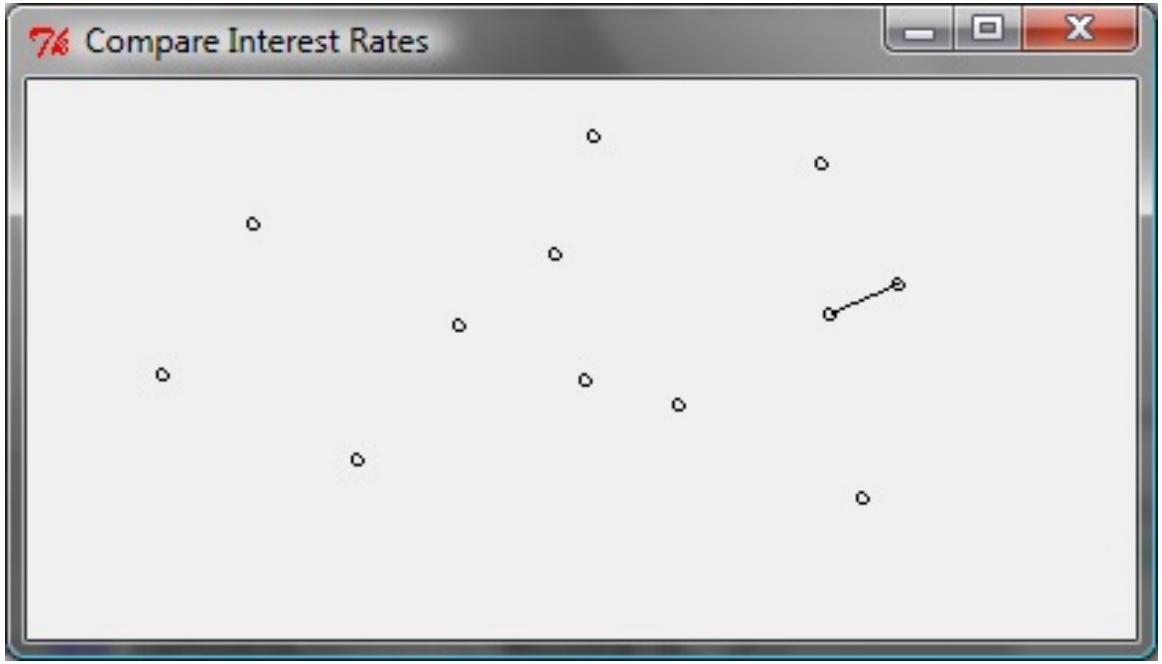




EnlargeShrinkCircle

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

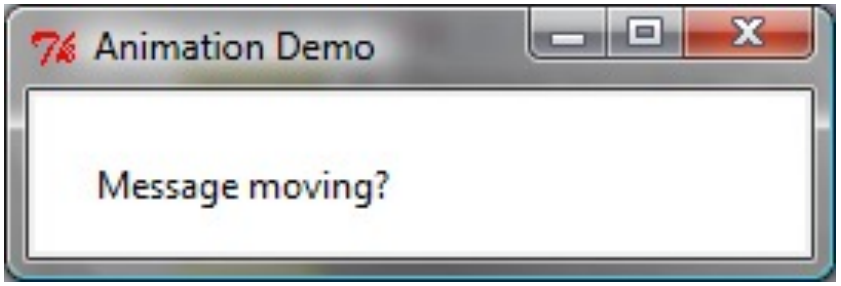
GUI: Finding Two Points Nearest to Each Other

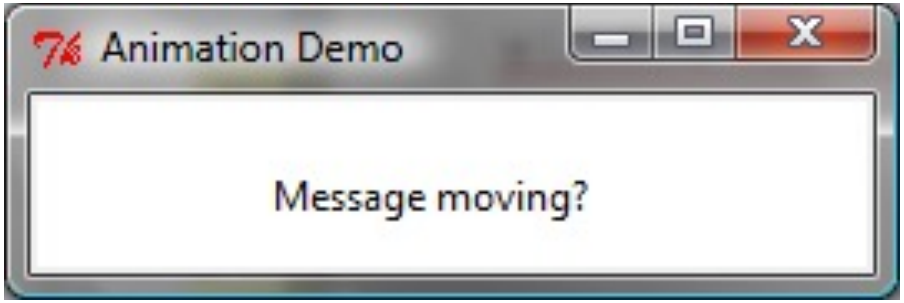


NearestPoints

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Animations

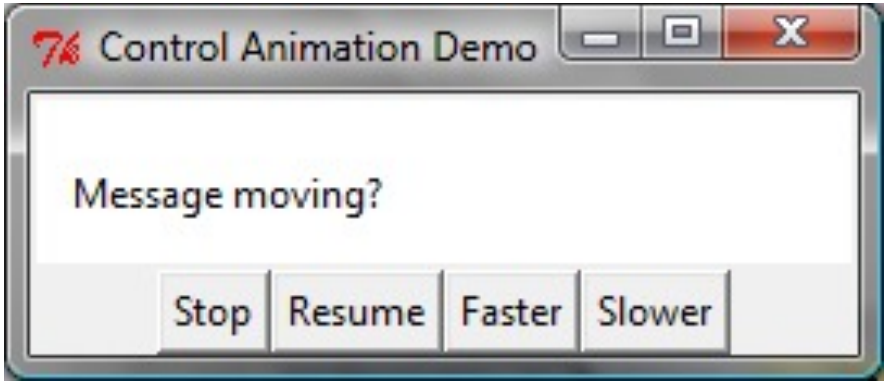


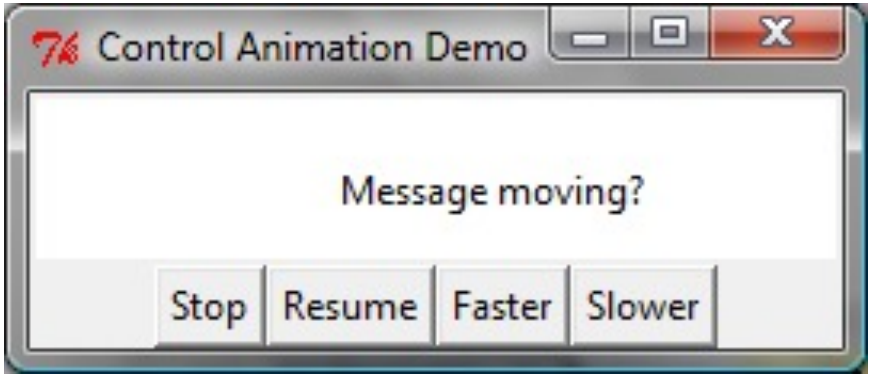


AnimationDemo

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Control Animations





ControlAnimation

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Case Studies: Bouncing Balls

Ball
x: int y: int
dx: int dy: int
color: Color
radius: int

The x-, y-coordinates for the center of the ball. By default, it is (0, 0).

dx and dy are the increment for (x, y).

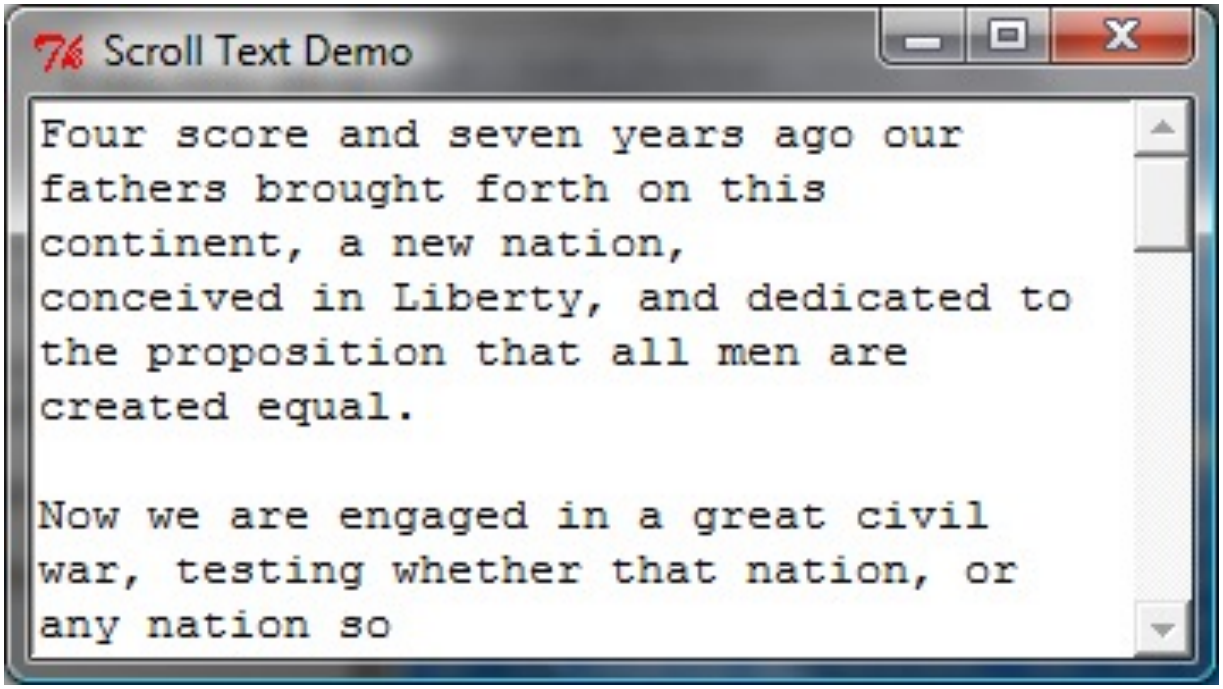
The color of the ball.

The radius of the ball.

BounceBalls

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

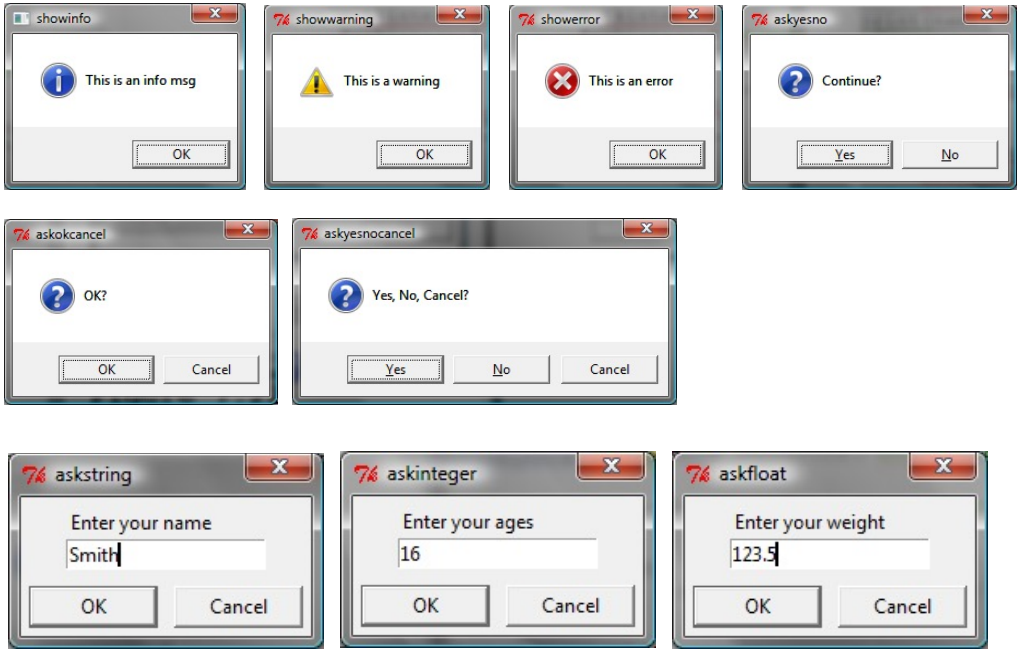
Scrollbar



ScrollText

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.

Standard Dialogs



DialogDemo

© Copyright 2018 by Pearson Education, Inc. All Rights Reserved.