Event types

There are several event types available with Tkinter, including:

KeyPress : Activated when a keyboard button has been pressed, the Key event can also be used for this.

KeyRelease : Activated when a keyboard button is released.

Button : Activated when a mouse button has been clicked.

ButtonRelease : Activated when a mouse button has been released.

Motion : Activated when the mouse cursor moves across the designated widget.

Enter : Activated when the mouse cursor enters the designated widget.

Leave : Activated when the mouse cursor leaves the designated widget.

MouseWheel : Activated when the mouse wheel is scrolled.

FocusIn : Activated when the designated widget gains focus through user input such as the mouse clicking on it.

FocusOut : Activated when the designated widget loses focus.

Configure : Activated when the designated widget's configurations have changes such as its width being adjusted by the user or its border being adjusted.

Event modifiers

An event modifier can alter the circumstances in which an event's handler is activated, for example, some modifiers will require another button to be depressed while the event occurs.

Control : Requires that the control button is being pressed while the event is occurring.

Alt : Requires that the alt button is being pressed while the event is occurring.

Shift : Requires that the shift button is being pressed while the event is occurring.

Lock : Requires that caps lock is activated when the event occurs.

Double : Requires that the given event happens twice in quick succession (such as a double click)

Triple : Requires that the given event happens three times in quick succession

Quadruple : Requires that the given event happens four times in quick succession

For example:

```
    label.bind( "<Double-Button>", mouseClick )
```

Event details

The detail section of the event string allows us to specify a more specific event such as only a certain key on the keyboard being pressed or only a certain mouse being being pressed.

-When using Button or ButtonRelease we can give a numeric detail from 1 to 5 which represents the specific mouse button you wish to have the handler trigger from.

-When using Key, KeyPress or KeyRelease we can give the ASCII value of the specific key we wish to trigger the event.

As an example, let's create an event handler that only activates on the double click of the left mouse button.

```
label.bind( "<Double-Button-1>", mouseClick )
```

The event object

The event object that is passed to the handler when the event is triggered can be used to collect and use information about the event that has occurred. The event object has a number of useful properties such as:

keysym : Returns the name of the key (space, e, return) that triggered a keyboard based event such as KeyPress, Key or KeyRelease.

keycode : Returns the code of the key that triggered a keyboard based event.

button : Returns the mouse button (1-5) that triggered a mouse based event.

x : Returns the x coordinate of where events such as Button occur.

y : Returns the y coordinate of where events such as Button occur.

width : Returns the current width of the widget associated with the event.

height : Returns the current height of the widget associated with the event.

As an example, let's create an event handler that prints the x and y coordinates of a mouse click event to the console.

```
def mouseClick( event ):
  print(   "mouse clicked at x=" + event.x + " y=" + event.y  )

label = Label(base)
label.bind( "<Button>", mouseClick )
```