|  |  |
| --- | --- |
| **Event** | **Description** |
| <Button-1> | Button 1 is the leftmost button, button 2 is the middle button(where available), and button 3 the rightmost button.  <Button-1>, <ButtonPress-1>, and <1> are all synonyms.  For mouse wheel support under Linux, use Button-4 (scroll up) and Button-5 (scroll down) |
| <B1-Motion> | The mouse is moved, with mouse button 1 being held down (use B2 for the middle button, B3 for the right button). |
| <ButtonRelease-1> | Button 1 was released. This is probably a better choice in most cases than the Button event, because if the user accidentally presses the button, they can move the mouse off the widget to avoid setting off the event. |
| <Double-Button-1> | Button 1 was double clicked. You can use Double or Triple as prefixes. |
| <Enter> | The mouse pointer entered the widget (this event doesn't mean that the user pressed the Enter key!). |
| <Leave> | The mouse pointer left the widget. |
| <FocusIn> | Keyboard focus was moved to this widget, or to a child of this widget. |
| <FocusOut> | Keyboard focus was moved from this widget to another widget. |
| <Return> | The user pressed the Enter key. For an ordinary 102-key PC-style keyboard, the special keys are Cancel (the Break key), BackSpace, Tab, Return(the Enter key), Shift\_L (any Shift key), Control\_L (any Control key), Alt\_L (any Alt key), Pause, Caps\_Lock, Escape, Prior (Page Up), Next (Page Down), End, Home, Left, Up, Right, Down, Print, Insert, Delete, F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12, Num\_Lock, and Scroll\_Lock. |
| <Key> | The user pressed any key. The key is provided in the char member of the event object passed to the callback (this is an empty string for special keys). |
| a | The user typed an "a". Most printable characters can be used as is. The exceptions are space (<space>) and less than (<less>). Note that 1 is a keyboard binding, while <1> is a button binding. |
| <Shift-Up> | The user pressed the Up arrow, while holding the Shift key pressed. You can use prefixes like Alt, Shift, and Control. |
| <Configure> | The widget changed size (or location, on some platforms). The new size is provided in the width and height attributes of the event object passed to the callback. |
| <Activate> | A widget is changing from being inactive to being active. This refers to changes in the state option of a widget such as a button changing from inactive (grayed out) to active. |
| <Deactivate> | A widget is changing from being active to being inactive. This refers to changes in the state option of a widget such as a radiobutton changing from active to inactive (grayed out). |
| <Destroy> | A widget is being destroyed. |
| <Expose> | This event occurs whenever at least some part of your application or widget becomes visible after having been covered up by another window. |
| <KeyRelease> | The user let up on a key. |
| <Map> | A widget is being mapped, that is, made visible in the application. This will happen, for example, when you call the widget's .grid() method. |
| <Motion> | The user moved the mouse pointer entirely within a widget. |
| <MouseWheel> | The user moved the mouse wheel up or down. At present, this binding works on Windows and MacOS, but not under Linux. |
| <Unmap> | A widget is being unmapped and is no longer visible. |
| <Visibility> | Happens when at least some part of the application window becomes visible on the screen. |

<https://stackoverflow.com/questions/32289175/list-of-all-tkinter-events>