

**Lesson 1.5 Reference Card Tkinter**

**Section 1. GUI component classes in the Tkinter library**

|  |  |
| --- | --- |
| **Class** | **Description** |
| Tk | Only create one. It is the top level GUI window. |

|  |  |
| --- | --- |
| * Create one instance of a Tk. * Pass the Tk object as the parent argument of components you want to place in the window. * Call mainloop() after the GUI is built. | Tk |
|  |
| mainloop() |

|  |  |
| --- | --- |
| **Widget Class** | **Description** |
| Canvas | For drawing |
| Scale | A slider |
| Text | A text editor; also useful for scrolling text output |
| Label | A small area to show text |
| Button | A box with optional text and mouse click handler |
| Radiobutton | One button in a group of buttons where only one can be selected at a time |
| Checkbutton | A check box that toggles on/off when clicked |

|  |  |
| --- | --- |
| * All widget classes have a grid method as shown here. * Constructors connect the widget to a Tk variable (see Reference Card Section 4) and a widget handler (a function). These are passed as variable and command arguments in the constructor. * Additional handlers can be subscribed using bind().   See A1.4.4 Part III. | Canvas |
|  |
| grid()  bind() |

**Section 2. Variables playing fixed role in the Tkinter library**

|  |  |  |  |
| --- | --- | --- | --- |
| NW | N | NE |  |
| W |  | E | These constants are used by the sticky argument in grid(). Example:  widget.grid(row=0,colummn=0, sticky=Tkinter.W) |
| SW | S | SE |  |
| HORIZONTAL  VERTICAL  END | | | These fixed-value variables in Tkinter are used by the Scale() and Text() classes, as well as others. |

**Section 3. Methods of the Canvas class**

|  |  |
| --- | --- |
| Example constructor call:  **Canvas**(root, width=300, height=300, background="#FFFFFF")   * coords() and itemconfig() can get or set an item's coordinates or attributes. The methods get coordinates or attributes if only the item argument is provided.   coords(*item*)  coords(*item*, *x1*, *y1*, *x2*, *y2*)  itemconfig(*item*)  itemconfig(*item*, *keyword*=*value*)   * move() can also change an item's coordinates   move(*item, dx, dy*)   * The create\_*item* methods all return an item number that should be assigned to a variable for later use. * The create\_*item* methods all take x,y coordinates:   create\_arc(*x1, y1, x2, y2 [, options]*)  create\_image(*x1, y1, image, [, options]*)  create\_rectangle(*x1, y1, x2, y2 [, options]*)  create\_text(*x1, y1 [, options]*)  create\_oval(*x1, y1, x2, y2 [, options]*)   * Options for some of these methods include   fill=’red’, outline='black',  font=(’Arial’, -100)  Color argument values can also be specified as a string concatenating # with two hexadecimal digits for red, green, and blue: fill=’#FA1234’   * after() will call *handler* when a timer reaches *msec* * Use in an indirect tail recursive call to create animation   after(*msec, handler*)   * See online documentation for other methods. | Canvas |
|  |
| \_\_init\_\_()  coords()  itemconfig()  move()  create\_oval()  create\_text()  create\_rectangle()  create\_image()  create\_arc()  after()  winfo\_width() |

|  |
| --- |
| **Section 4. Variable classes in the Tkinter library** |

|  |  |
| --- | --- |
| **Class** | **Description** |
| IntVar | Connects an int to a widget |
| BooleanVar | Connects a bool to a widget |
| DoubleVar | Connects a float to a widget |
| StringVar | Connects a str to a widget |

|  |  |
| --- | --- |
| * All four classes have get and set methods as shown here. * Objects in these classes are used for the variable argument of widget constructors. | IntVar |
|  |
| get()  set() |

**Section 5. Event names in the Tkinter library**

|  |  |
| --- | --- |
| * Events are passed to handlers, which can access the event's attributes. * Events have a name that is a string.\* The bind() methods of the widgets take a string that must match this string to catch an event. A few example name attributes:   "<Button-1>" for mouse click  "<ButtonRelease-1>" for mouse release  "<r>" for keypress of key "r"   * These attributes report the coordinates within the widget where the event occurred. | Event |
| name  x  y |
|  |

\*The Event has a type attribute that is an int. A dictionary transforms this code to the event string name that must be used by the bind() method.