appJar Study Guide:

The easiest way to create GUIs in Python.

Input Widgets:

Input widgets are used for capturing user interactions, either by clicking, typing or dragging.

They usually provide three functions:

- ***ADD** - this creates the widget*
- ***GET** - this gets the contents/state of the widget*
- ***SET** - this changes what's in the widget*

As well as options to change the way they look/act.

*For each of the above to work, we need to know which widget you are referring to - so every widget gets a **unique title**.*

If you want your input widget to have a label, there are some auto-label functions

If you want to get the contents of all widgets in the GUI as a single dictionary, use:
* `.getAllInputs(includeEmptyInputs=False)`
This will return the contents of all input fields as a dictionary.
Set `includeEmptyInputs` to True, if you want to include any empty input fields.

Study Guide Widgets List:

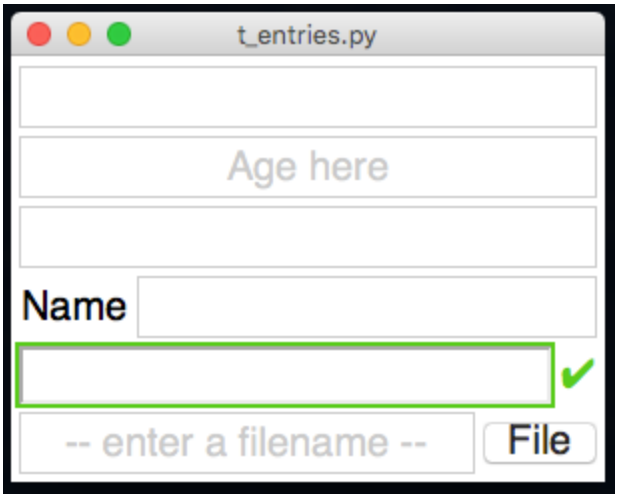
Entry:
TextArea:
Button:
RadioButton:
CheckBox:
ListBox:

Entry

Entries are used to capture typed input from the user. They take a single parameter - a title.

There are five special-case Entries:

- **NumericEntry** - this only allows numbers to be typed in - always returns a float (None if empty).
- **SecretEntry** - this will show stars, instead of the letters typed - useful for capturing passwords.
- **AutoEntry** - this takes a list of words to provide auto-completion.
- **ValidationEntry** - can be set to valid/invalid/waiting - will colour the border green/red/black and show a ✔/✘/★
- **OpenEntry/SaveEntry/DirectoryEntry** - provides a button to select a file/directory and auto-populates the Entry



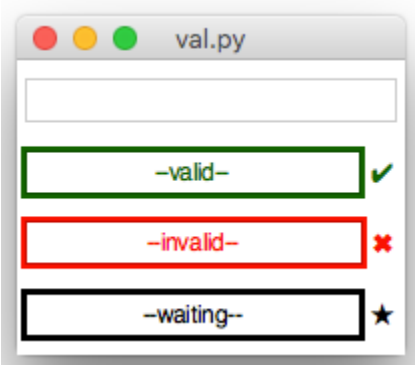
Add Entries

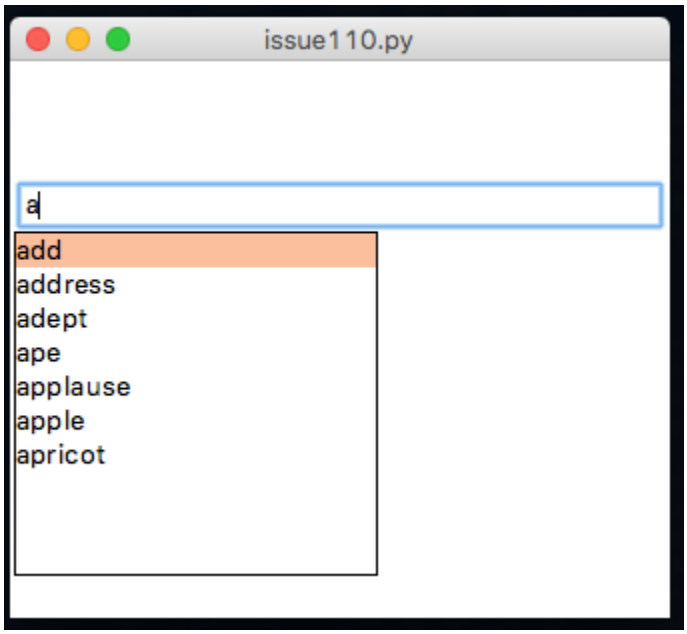
- `.addEntry(title)`
- `.addNumericEntry(title)`
- `.addSecretEntry(title)`
- `.addValidationEntry(title)`
- `.addOpenEntry(title)`
- `.addSaveEntry(title)`
- `.addDirectoryEntry(title)`
- `.addAutoEntry(title, words)`
  This also takes a `words` parameter, which must be a list, and is used to populate he drop-down.

Each of these will add the specified type of Entry, using the title provided.

Set Entries

- `.setEntry(title, text, callFunction=True)`
- `.setEntryDefault(title, text)`
- `.setEntryUpperCase(title)` & `.setEntryLowerCase(title)`
- `.setEntryMaxLength(title, maxLength)`
- `.setEntryValid(title)` & `.setEntryInvalid(title)` & `.setEntryWaitingValidation(title)`



- `.setValidationEntry(title, state="valid")`
- `.setValidationEntryLabelBg(title, bg)`
- `.setAutoEntryNumRows(title, rows)`
- 

- `.appendAutoEntry(title, value)`

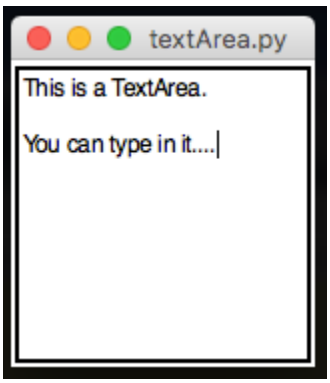
- `.removeAutoEntry(title, value)`
- `.changeAutoEntry(title, value)`
- `.clearEntry(title, callFunction=True)`
- `.clearAllEntries(callFunction=False)`
- `.setFocus(title)`

Get Entries

- `.getEntry(title)`
- `.getAllEntries()`

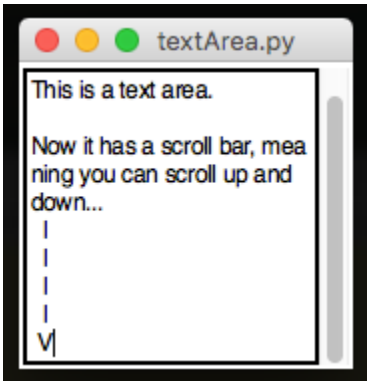
TextArea

Similar to an Entry, but allows you to type text over multiple lines.



Add TextAreas

- `.addTextArea(title, text=None)`
- `.addScrolledTextArea(title, text=None)`



Set TextAreas

- `.setTextArea(title, text, end=True, callFunction=True)`
- Set `callFunction` to be False, if you don't want to call any associated functions.
- `.clearTextArea(title, callFunction=True)`
- `.clearAllTextAreas(callFunction=False)`

- `.highlightTextArea(title, start, end=END)`

Get TextAreas

- `.getTextArea(title)`
- `.getAllTextAreas()`

Search TextAreas

- `.searchTextArea(title, pattern, start=None, stop=None, nocase=True, backwards=False)`

TextArea Fonts

Fonts are tricky on TextAreas, it's currently not possible to have different font sizes & families on different parts of a TextArea.
However, it is possible to make parts of the TextArea **bold**, **italic** or **underlined**.

- `.setTextAreaFont(title, **kwargs)`
- `.textAreaApplyFontRange(title, tag, start, end=END)`
- `.textAreaApplyFontSelected(title, tag)`
- `.textAreaToggleFontRange(title, tag, start, end=END)`
- `.textAreaToggleFontSelected(title, tag)`

Tag TextAreas

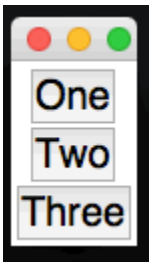
- `.textAreaCreateTag(title, **kwargs)`
- `.textAreaChangeTag(title, **kwargs)`
- `.textAreaDeleteTag(title, *tags)`
- `.textAreaTagPattern(title, tag, pattern, regexp=False)`
- `.textAreaTagRange(title, tag, start, end)`
- `.textAreaTagSelected(title, tag)`
- `.textAreaUntagRange(title, tag, start, end)`
- `.textAreaUntagSelected(title, tag)`
- `.textAreaToggleTagRange(title, tag, start, end)`

- `.textAreaToggleTagSelected(title, tag)`
- `.getTextAreaTags(title)`
- `.getTextAreaTag(title)`

Button

A clickable button, that will call a function.
These are the key to starting an interactive application.
The GUI is looping, waiting for something to happen.
A button click is the classic way to start interacting with a GUI.

Whenever any function is called by the GUI, the title of the widget that called it is passed as a parameter.
That way, multiple widgets can use the same function, but different actions can be performed, depending on the name passed as a parameter.




```

```

Add Buttons

- `.addButton(title, function)`
- `.addButtons(titles, functions)`
- `.addImageButton(title, function, imgFile, align=None)`
- `.addIconButton(title, function, iconName, align=None)`
- `.addNamedButton(name, title, function)`

Set Buttons

- `.setButton(name, text)`
- `.setButtonImage(title, image, align=None)`

RadioButton

A group of round boxes, only one of which can be selected.
These are great for getting a single value, for a multiple choice question.

Add RadioButtons

- `.addRadioButton(title, name)`

Set RadioButtons

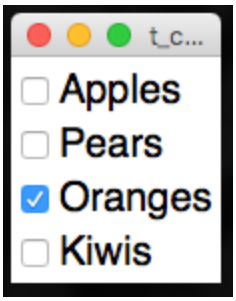
- `.setRadioButton(title, value, callFunction=True)`
- `.setRadioSquare(title, square=True)`
- `.clearAllRadioButtons(callFunction=False)`

Get RadioButtons

- `.getRadioButton(title)`

CheckBox

A simple tick-box, with a label, that can be either ON or OFF.



Add CheckBoxes

- `.addCheckBox(title)`
- `.addNamedCheckBox(name, title)`

Set CheckBoxes

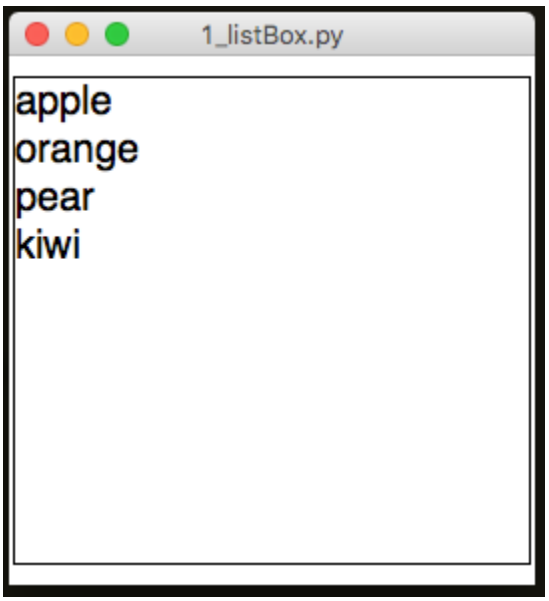
- `.setCheckBox(title, ticked=True, callFunction=True)`
- `.clearAllCheckBoxes(callFunction=False)`
- `.setCheckBoxText(title, text)`

Get CheckBoxes

- `.getCheckBox(title)`
- `.getAllCheckBoxes()`

ListBox

A box containing a list of items, single or multi-select



Add ListBoxes

- `.addListBox(title, values)`.
- `.addListItem(title, item)`
- `.addListItems(title, items)`

Set ListBoxes

- `.setListItem(title, item, newVal, first=False)`
  `.setListItemAtPos(title, pos, newVal)`
- `.removeListItem(title, item)`
  `.removeListItemAtPos(title, pos)`
- `.clearListBox(title, callFunction=True)`
- `.clearAllListBoxes(callFunction=False)`

- `.updateListBox(title, items, select=False, callFunction=True)`
- `.selectListItem(title, item, callFunction=True)`
  `.selectListItemAtPos(title, pos, callFunction=False)`
- `.deselectListItemAtPos(title, pos, callFunction=False)`
- `.deselectAllListItems(title, callFunction=False)`
- `.setListBoxRows(title, rows)`
- `.setListBoxMulti(list, multi=True)`
- `.setListBoxGroup(list, group=True)`
- `.setListItemBg(title, item, colour)` & `.setListItemFg(title, item, colour)`
  `.setListItemAtPosBg(title, item, colour)` & `.setListItemAtPosFg(title, item, colour)`

Get ListBoxes

- `.getListBox(title)`
- `.getAllListBoxes()`
- `.getAllListItems(title)`