Widget List

This lecture will serve as a reference for widgets, providing a list of the GUI widgets available!

Complete list

For a complete list of the GUI widgets available to you, you can list the registered widget types. `Widget` and `DOMWidget`, not listed below, are base classes.

In [1]:

```
import ipywidgets as widgets

# Show all available widgets!
widgets.Widget.widget_types.values()
```

Out[1]:

```
[ipywidgets.widgets.widget_string.Text,
 ipywidgets.widgets.widget_box.Box,
 ipywidgets.widgets.widget_controller.Axis,
 ipywidgets.widgets.widget_bool.Checkbox,
 ipywidgets.widgets.widget_int.IntRangeSlider,
 ipywidgets.widgets.widget_selection.RadioButtons,
 ipywidgets.widgets.widget_string.HTML,
 ipywidgets.widgets.widget_float.FloatRangeSlider,
 ipywidgets.widgets.widget_box.PlaceProxy,
 ipywidgets.widgets.widget_selection.ToggleButtons,
 ipywidgets.widgets.widget_int.IntText,
 ipywidgets.widgets.widget_selection.Dropdown,
 ipywidgets.widgets.widget_bool.Valid,
 ipywidgets.widgets.widget_bool.ToggleButton,
 ipywidgets.widgets.widget_float.FloatSlider,
 ipywidgets.widgets.widget_int.IntProgress,
 ipywidgets.widgets.widget_selection.SelectMultiple,
 ipywidgets.widgets.widget_float.FloatProgress,
 ipywidgets.widgets.widget_string.Latex,
 ipywidgets.widgets.widget_box.FlexBox,
 ipywidgets.widgets.widget_string.Textarea,
 ipywidgets.widgets.widget_float.BoundedFloatText,
 ipywidgets.widgets.widget_controller.Button,
 ipywidgets.widgets.widget_selection.Select,
 ipywidgets.widgets.widget_selectioncontainer.Accordion,
 ipywidgets.widgets.widget_float.FloatText,
 ipywidgets.widgets.widget_image.Image,
 ipywidgets.widgets.widget_button.Button,
 ipywidgets.widgets.widget_int.BoundedIntText,
 ipywidgets.widgets.widget_box.Proxy,
 ipywidgets.widgets.widget_selectioncontainer.Tab,
 ipywidgets.widgets.widget_int.IntSlider,
 ipywidgets.widgets.widget_controller.Controller]
```

Numeric widgets

There are 8 widgets distributed with IPython that are designed to display numeric values. Widgets exist for displaying integers and floats, both bounded and unbounded. The integer widgets share a similar naming scheme to their floating point counterparts. By replacing `Float` with `Int` in the widget name, you can find the Integer equivalent.

FloatSlider

In [2]:
```
widgets.FloatSlider(
    value=7.5,
    min=5.0,
    max=10.0,
    step=0.1,
    description='Test:',
)
```

Sliders can also be **displayed vertically**.

In [3]:
```
widgets.FloatSlider(
    value=7.5,
    min=5.0,
    max=10.0,
    step=0.1,
    description='Test',
    orientation='vertical',
)
```

FloatProgress

In [4]:
```
widgets.FloatProgress(
    value=7.5,
    min=5.0,
    max=10.0,
    step=0.1,
    description='Loading:',
)
```

BoundedFloatText

In [5]:
```
widgets.BoundedFloatText(
    value=7.5,
    min=5.0,
    max=10.0,
    description='Text:',
)
```

FloatText

In [6]:

```
widgets.FloatText(
    value=7.5,
    description='Any:',
)
```

Boolean widgets

There are three widgets that are designed to display a boolean value.

ToggleButton

In [7]:

```
widgets.ToggleButton(
    description='Click me',
    value=False,
)
```

Checkbox

In [8]:

```
widgets.Checkbox(
    description='Check me',
    value=True,
)
```

Valid

The valid widget provides a read-only indicator.

In [9]:

```
widgets.Valid(
    value=True,
)
```

Selection widgets

There are four widgets that can be used to display single selection lists, and one that can be used to display multiple selection lists. All inherit from the same base class. You can specify the **enumeration of selectable options by passing a list**. You can **also specify the enumeration as a dictionary**, in which case the **keys will be used as the item displayed** in the list and the corresponding **value will be returned** when an item is selected.

Dropdown

```
In [10]: from IPython.display import display

         w = widgets.Dropdown(
             options=['1', '2', '3'],
             value='2',
             description='Number:',
         )
         display(w)
```

```
In [12]: # Show value
         w.value

Out[12]: '1'
```

The following is also valid:

```
In [13]: w = widgets.Dropdown(
             options={'One': 1, 'Two': 2, 'Three': 3},
             value=2,
             description='Number:')

         display(w)
```

```
In [14]: w.value

Out[14]: 2
```

RadioButtons

```
In [15]: widgets.RadioButtons(
             description='Pizza topping:',
             options=['pepperoni', 'pineapple', 'anchovies'],
         )
```

Select

```
 In [ ]: widgets.Select(
             description='OS:',
             options=['Linux', 'Windows', 'OSX'],
         )
```

ToggleButtons

In [ ]:

```
widgets.ToggleButtons(
    description='Speed:',
    options=['Slow', 'Regular', 'Fast'],
)
```

SelectMultiple

Multiple values can be selected with `shift` and/or `ctrl` (or `command`) pressed and mouse clicks or arrow keys.

In [16]:

```
w = widgets.SelectMultiple(
    description="Fruits",
    options=['Apples', 'Oranges', 'Pears'])

display(w)
```

In [ ]:

```
w.value
```

String widgets

There are 4 widgets that can be used to display a string value. Of those, the `Text` and `Textarea` widgets accept input. The `Latex` and `HTML` widgets display the string as either Latex or HTML respectively, but do not accept input.

Text

In [17]:

```
widgets.Text(
    description='String:',
    value='Hello World',
)
```

Textarea

In [18]:

```
widgets.Textarea(
    description='String:',
    value='Hello World',
)
```

Latex

In [22]:

```
widgets.Latex(
    value="$$\\frac{n!}{k!(n-k)!}$$",
)
```

HTML

In [23]:

```
widgets.HTML(
    value="Hello <b>World</b>"
)
```

Button

In [24]:

```
widgets.Button(description='Click me')
```

Conclusion

Use this as a future reference for yourself!