Lista.Widgets

October 26, 2019

Para entrar no modo apresentação, execute a seguinte célula e pressione -

[1]:
```
%cd ..
%reload_ext slide
```

1 Lista de Widgets

Foram apresentados `IntSlider`, `Output`, `VBox` e `Button` até agora. No restante deste notebook, vou apresentar outros widgets que existem na biblioteca `ipywidgets`

Parte do material tirado de https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20List.html

1.0.1 Widgets de texto

Label Apenas um label somente leitura

[2]:
```
from ipywidgets import Label
Label("Texto")
```

Text Campo de texto

[3]:
```
from ipywidgets import Text
Text(
    value='Hello World',
    placeholder='Type something',
    description='String:',
    disabled=False
)
```

Textarea Área de texto

[4]:
```
from ipywidgets import Textarea
Textarea(
    value='Hello World',
    placeholder='Type something',
```

```
    description='String:',
    disabled=False
)
```

Combobox Combobox com autocomplete

[5]:
```
from ipywidgets import Combobox
Combobox(
    # value='John',
    placeholder='Choose Someone',
    options=['Paul', 'John', 'George', 'Ringo'],
    description='Combobox:',
    ensure_option=True,
    disabled=False
)
```

HTML HTML somente leitura

[6]:
```
from ipywidgets import HTML
HTML(
    value="Hello <b>World</b>",
    placeholder='Some HTML',
    description='Some HTML',
)
```

```
HTML(value='Hello <b>World</b>', description='Some HTML', placeholder='Some HTML')
```

HTMLMath HTML somente leitura com fórmulas

[7]:
```
from ipywidgets import HTMLMath
HTMLMath(
    value=r"Some math and <i>HTML</i>: \(x^2\) and $$\frac{x+1}{x-1}$$",
    placeholder='Some HTML',
    description='Some HTML',
)
```

```
HTMLMath(value='Some math and <i>HTML</i>: \\(x^2\\) and $$\\frac{x+1}{x-1}$$', description='S
```

1.0.2 Widgets numéricos

FloatSlider Semelhante a `IntSlider`, mas para `float`

[8]:
```
from ipywidgets import FloatSlider
FloatSlider(
    value=7.5,
    min=0,
    max=10.0,
    step=0.1,
    readout_format='.1f'
)
```

```
FloatSlider(value=7.5, max=10.0, readout_format='.1f')
```

FloatLogSlider `FloatSlider` com escala logaritimica

[9]:
```
from ipywidgets import FloatLogSlider
FloatLogSlider(
    value=10,
    base=10,
    min=-10, # max exponent of base
    max=10, # min exponent of base
    step=0.2, # exponent step
    description='Log Slider'
)
```

```
FloatLogSlider(value=10.0, description='Log Slider', max=10.0, min=-10.0, step=0.2)
```

IntRangeSlider, FloatRangeSlider Sliders com dois valores

[10]:
```
from ipywidgets import IntRangeSlider
IntRangeSlider(
    value=[5, 7],
    min=0,
    max=10,
    step=1,
)
```

```
IntRangeSlider(value=(5, 7), max=10)
```

[11]:
```
_.value
```

[11]:
```
(5, 7)
```

IntProgress, FloatProgress Widgets que representam barra de progresso

[12]:
```
from ipywidgets import IntProgress
IntProgress(
    value=7,
    min=0,
    max=10,
    step=1,
    description='Loading:',
    bar_style='', # 'success', 'info', 'warning', 'danger' or ''
    orientation='horizontal'
)
```

```
IntProgress(value=7, description='Loading:', max=10)
```

IntText, FloatText Campos de texto numéricos

[13]:
```
from ipywidgets import IntText
IntText(
    value=7,
    description='Any:',
    disabled=False
)
```

```
IntText(value=7, description='Any:')
```

BoundedIntText, BoundedFloatText Campos de texto numéricos limitados

[14]:
```
from ipywidgets import BoundedFloatText
BoundedFloatText(
    value=7.5,
    min=0,
    max=10.0,
    step=0.1,
    description='Text:',
)
```

```
BoundedFloatText(value=7.5, description='Text:', max=10.0, step=0.1)
```

1.0.3 Widgets booleanos

ToggleButton Botão com estado booleano

[15]:
```
from ipywidgets import ToggleButton
ToggleButton(
```

```
    value=False,
    description='Click me',
    button_style='', # 'success', 'info', 'warning', 'danger' or ''
    tooltip='Description',
    icon='check'
)
```

```
ToggleButton(value=False, description='Click me', icon='check', tooltip='Description')
```

Checkbox

[16]:
```
from ipywidgets import Checkbox
Checkbox(
    value=False,
    description='Check me',
)
```

```
Checkbox(value=False, description='Check me')
```

Valid Indicador somente leitura

[17]:
```
from ipywidgets import Valid
Valid(
    value=False,
    description='Valid!',
)
```

```
Valid(value=False, description='Valid!')
```

1.0.4 Widgets de seleção

Dropdown Widget para selecionar elementos de uma lista

[18]:
```
from ipywidgets import Dropdown
Dropdown(
    options=['1', '2', '3'], # [('One', 1), ('Two', 2), ('Three', 3)]
    value='2',
    description='Number:',
    disabled=False,
)
```

```
Dropdown(description='Number:', index=1, options=('1', '2', '3'), value='2')
```

RadioButtons Selecionar usando radio buttons

[19]:
```
from ipywidgets import RadioButtons
RadioButtons(
    options=['One', 'Two', 'Three'],
    description='Number:',
    disabled=False
)
```

```
RadioButtons(description='Number:', options=('One', 'Two', 'Three'), value='One')
```

Select Selecionar usando uma lista visível

[20]:
```
from ipywidgets import Select
Select(
    options=['Linux', 'Windows', 'OSX'],
    value='OSX',
    # rows=10,
    description='OS:',
    disabled=False
)
```

```
Select(description='OS:', index=2, options=('Linux', 'Windows', 'OSX'), value='OSX')
```

SelectionSlider Slider para seleção de campos nominais

[21]:
```
from ipywidgets import SelectionSlider
SelectionSlider(
    options=['mal passada', 'ao ponto', 'bem passada'],
    value='ao ponto',
    description='Carne ...',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True
)
```

```
SelectionSlider(continuous_update=False, description='Carne ...', index=1, options=('mal passa
```

SelectionRangeSlider Slider para seleção de intervalo nominal

[22]:
```
import datetime
from ipywidgets import SelectionRangeSlider
dates = [datetime.date(2019,i,1) for i in range(1,13)]
```

```
options = [(i.strftime('%b'), i) for i in dates]
SelectionRangeSlider(
    options=options,
    index=(0,11),
    description='2019',
    disabled=False
)
```

```
SelectionRangeSlider(description='2019', index=(0, 11), options=(('Jan', datetime.date(2019, 1
```

ToggleButtons `ToggleButton` para escolher um único elemento de lista

[23]:
```
from ipywidgets import ToggleButtons
ToggleButtons(
    options=['Slow', 'Regular', 'Fast'],
    description='Speed:',
    disabled=False,
    button_style='', # 'success', 'info', 'warning', 'danger' or ''
    tooltips=['Description of slow', 'Description of regular', 'Description of␣
 ↪fast'],
#     icons=['check'] * 3
)
```

```
ToggleButtons(description='Speed:', options=('Slow', 'Regular', 'Fast'), tooltips=('Descriptio
```

SelectMultiple Seleção de vários elementos

[24]:
```
from ipywidgets import SelectMultiple
SelectMultiple(
    options=['Apples', 'Oranges', 'Pears'],
    value=['Oranges'],
    #rows=10,
    description='Fruits',
    disabled=False
)
```

```
SelectMultiple(description='Fruits', index=(1,), options=('Apples', 'Oranges', 'Pears'), value=
```

1.0.5 Widgets de estrutura

HBox Semelhante ao `VBox`, mas exibe widgets na horizontal ao invés de na vertical

[25]:
```
from ipywidgets import HBox
HBox([Label("a"), Label("b")])
```

```
HBox(children=(Label(value='a'), Label(value='b')))
```

GridBox Semelhante a `VBox` e `HBox`, mas usa `HTML Grid` para fazer a exibição

Aqui estamos usando Layout também para definir atributos do `CSS`

[26]:
```
from ipywidgets import GridBox, Layout
items = [Label(str(i)) for i in range(8)]
GridBox(items, layout=Layout(grid_template_columns="repeat(3, 100px)"))
```

```
GridBox(children=(Label(value='0'), Label(value='1'), Label(value='2'), Label(value='3'), Label
```

Accordion Exibe widgets em páginas diferentes de Accordion

[27]:
```
from ipywidgets import Accordion
accordion = Accordion([Label("a"), Label("b")])
accordion.set_title(0, 'Page 0')
accordion.set_title(1, 'Page 1')
accordion
```

```
Accordion(children=(Label(value='a'), Label(value='b')), _titles={'0': 'Page 0', '1': 'Page 1'
```

Tab Exibe widgets em abas diferentes

[28]:
```
from ipywidgets import Tab
tab = Tab([Label("a"), Label("b")])
tab.set_title(0, 'Page 0')
tab.set_title(1, 'Page 1')
tab
```

```
Tab(children=(Label(value='a'), Label(value='b')), _titles={'0': 'Page 0', '1': 'Page 1'})
```

1.0.6 Outros widgets

Play Widget útil para controlar animações

[29]:
```
from ipywidgets import Play, jslink, IntSlider
play = Play(
#     interval=10,
```

```
    value=50,
    min=0,
    max=100,
    step=1,
    description="Press play",
    disabled=False
)
slider = IntSlider()
jslink((play, 'value'), (slider, 'value'))
HBox([play, slider])
```

```
HBox(children=(Play(value=50, description='Press play'), IntSlider(value=0)))
```

DatePicker Widget para escolher datas

[30]:
```
from ipywidgets import DatePicker
DatePicker(
    description='Pick a Date',
    disabled=False
)
```

```
DatePicker(value=None, description='Pick a Date')
```

ColorPicker Widget para escolher cor

[31]:
```
from ipywidgets import ColorPicker
ColorPicker(
    concise=False,
    description='Pick a color',
    value='blue',
    disabled=False
)
```

```
ColorPicker(value='blue', description='Pick a color')
```

FileUpload Widget para fazer upload de arquivos e receber em bytes

[32]:
```
from ipywidgets import FileUpload
FileUpload(
    accept='',  # Accepted file extension e.g. '.txt', '.pdf', 'image/*',␣
 ↪'image/*,.pdf'
    multiple=False  # True to accept multiple files upload else False
)
```

```
FileUpload(value={}, description='Upload')
```

Image Widget para visualizar imagem

[33]:
```
from ipywidgets import Image
file = open("images/jupyter.png", "rb")
image = file.read()
Image(
    value=image,
    format='png',
    width=50,
    height=50,
)
```

```
Image(value=b'\x89PNG\r\n\x1a\n\x00\x00\x00\rIHDR\x00\x00\x00X\x00\x00\x00f\x08\x06\x00\x00\x0
```

Controller Widget para usar controle de jogo como entrada

[34]:
```
from ipywidgets import Controller
Controller(
    index=0,
)
```

```
Controller()
```