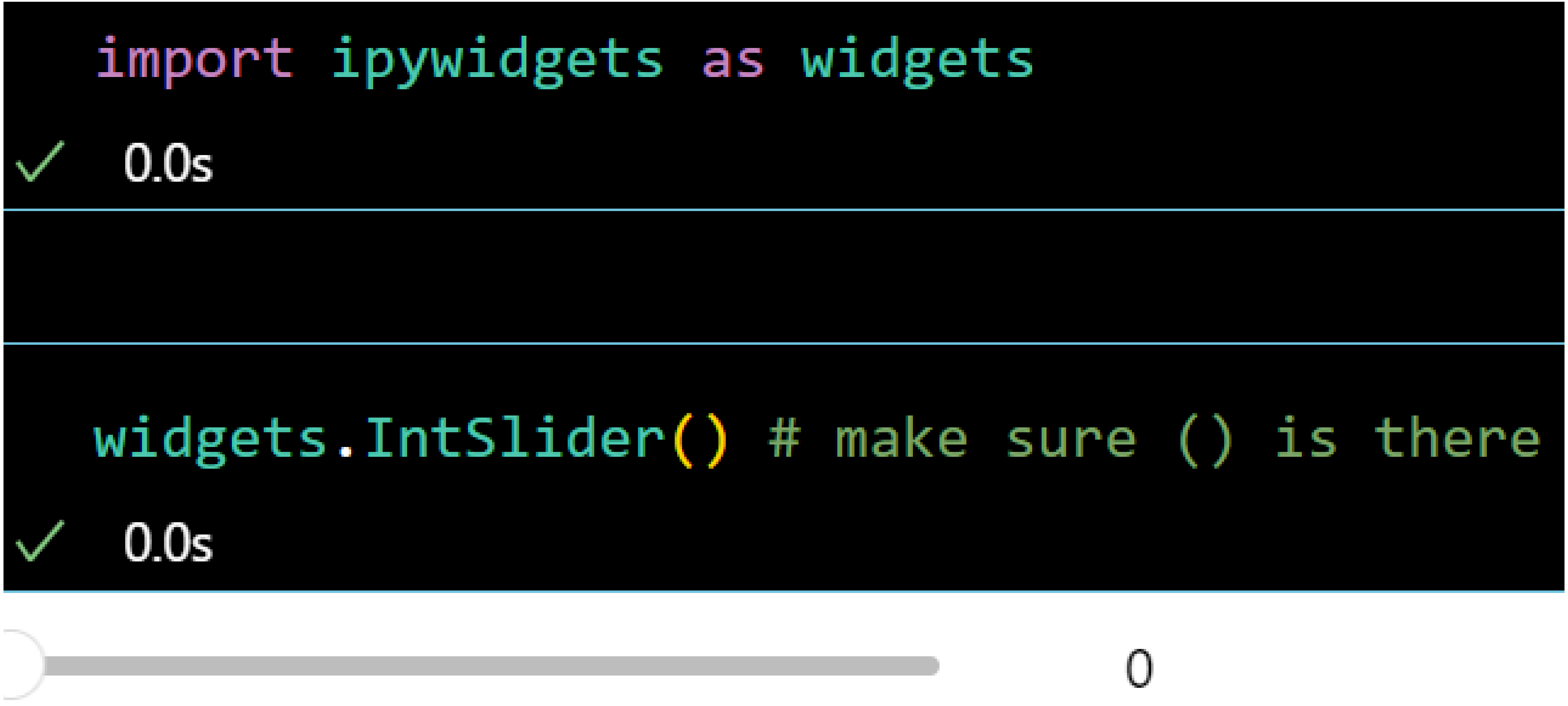


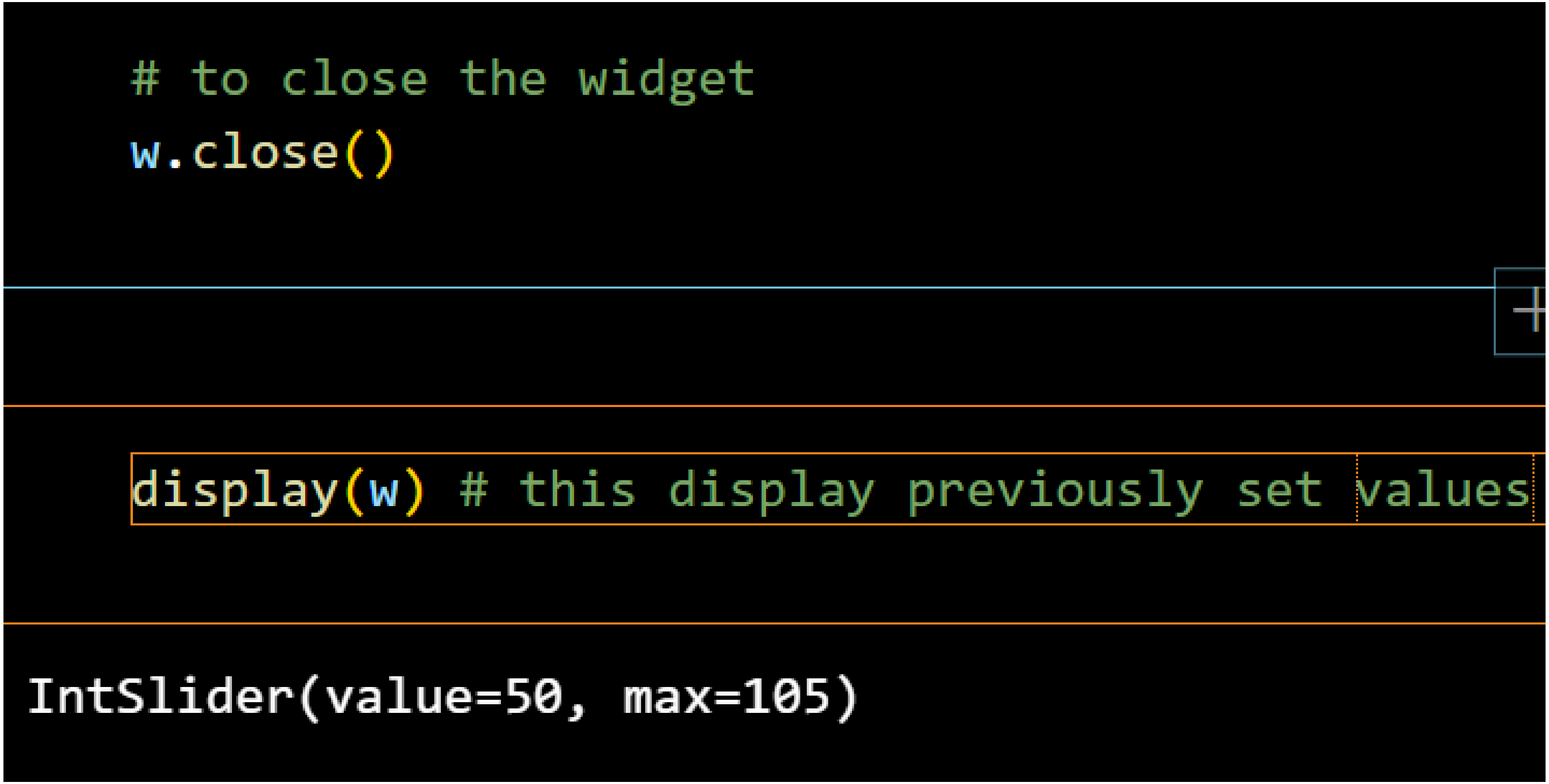
```
import ipywidgets as widgets
```

✓ 0.0s

```
widgets.IntSlider() # make sure () is there
```

✓ 0.0s

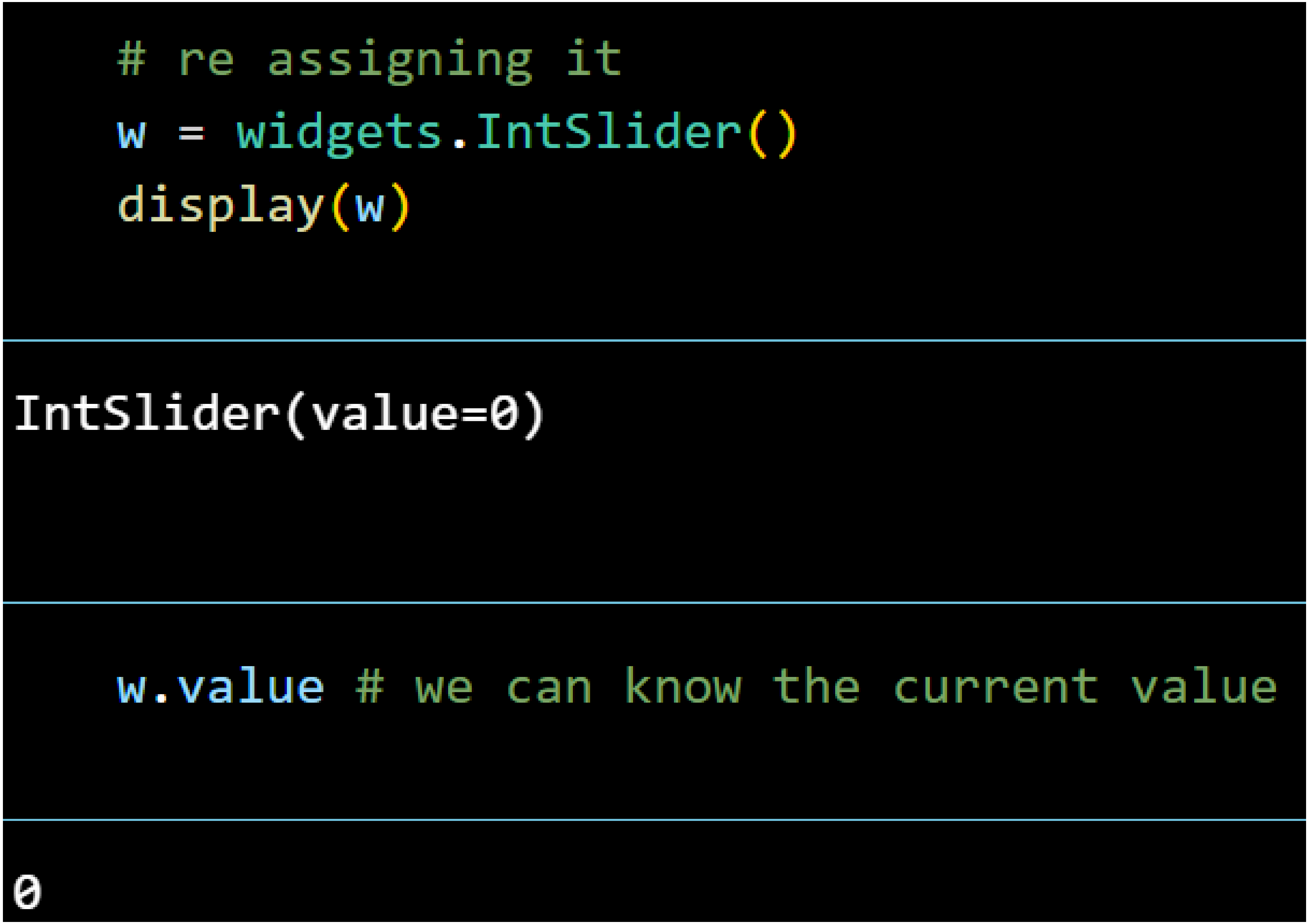
0



```
# to close the widget
w.close()
```

```
display(w) # this display previously set values
```

```
IntSlider(value=50, max=105)
```



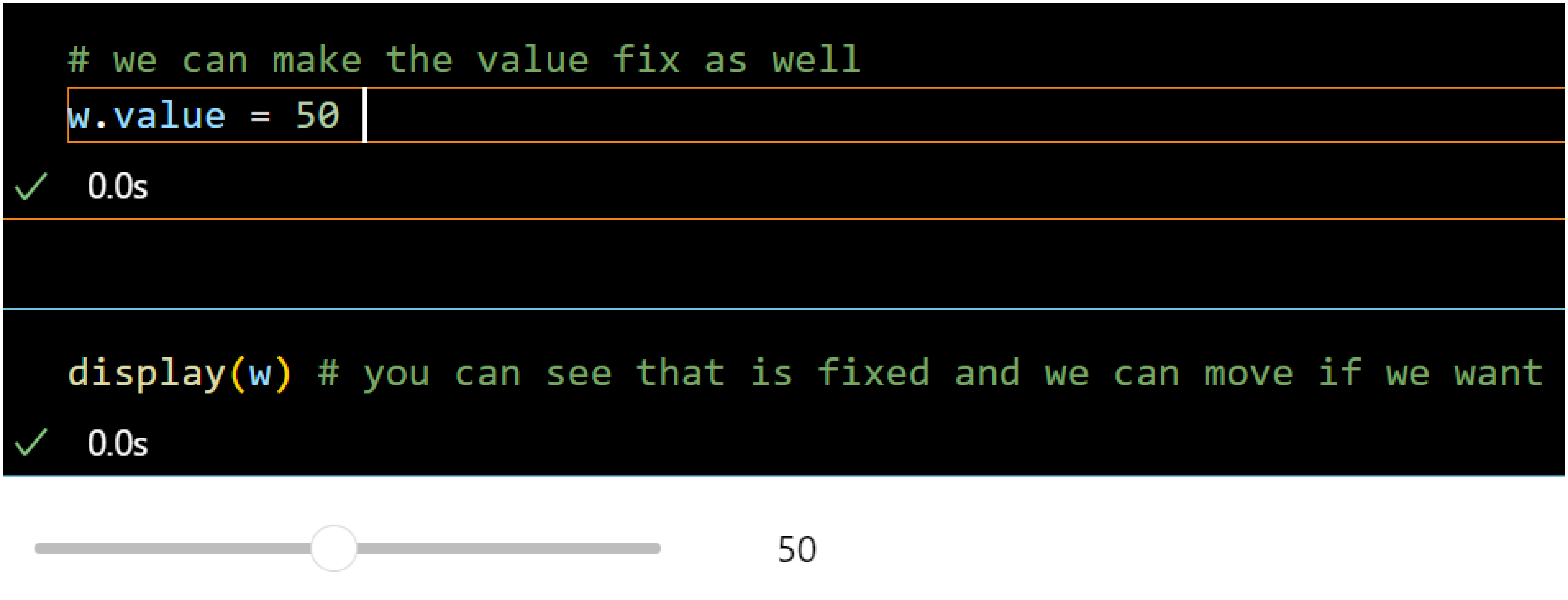
```python
# re assigning it
w = widgets.IntSlider()
display(w)
```

```
IntSlider(value=0)
```

```python
w.value # we can know the current value
```

```
0
```

[redacted]



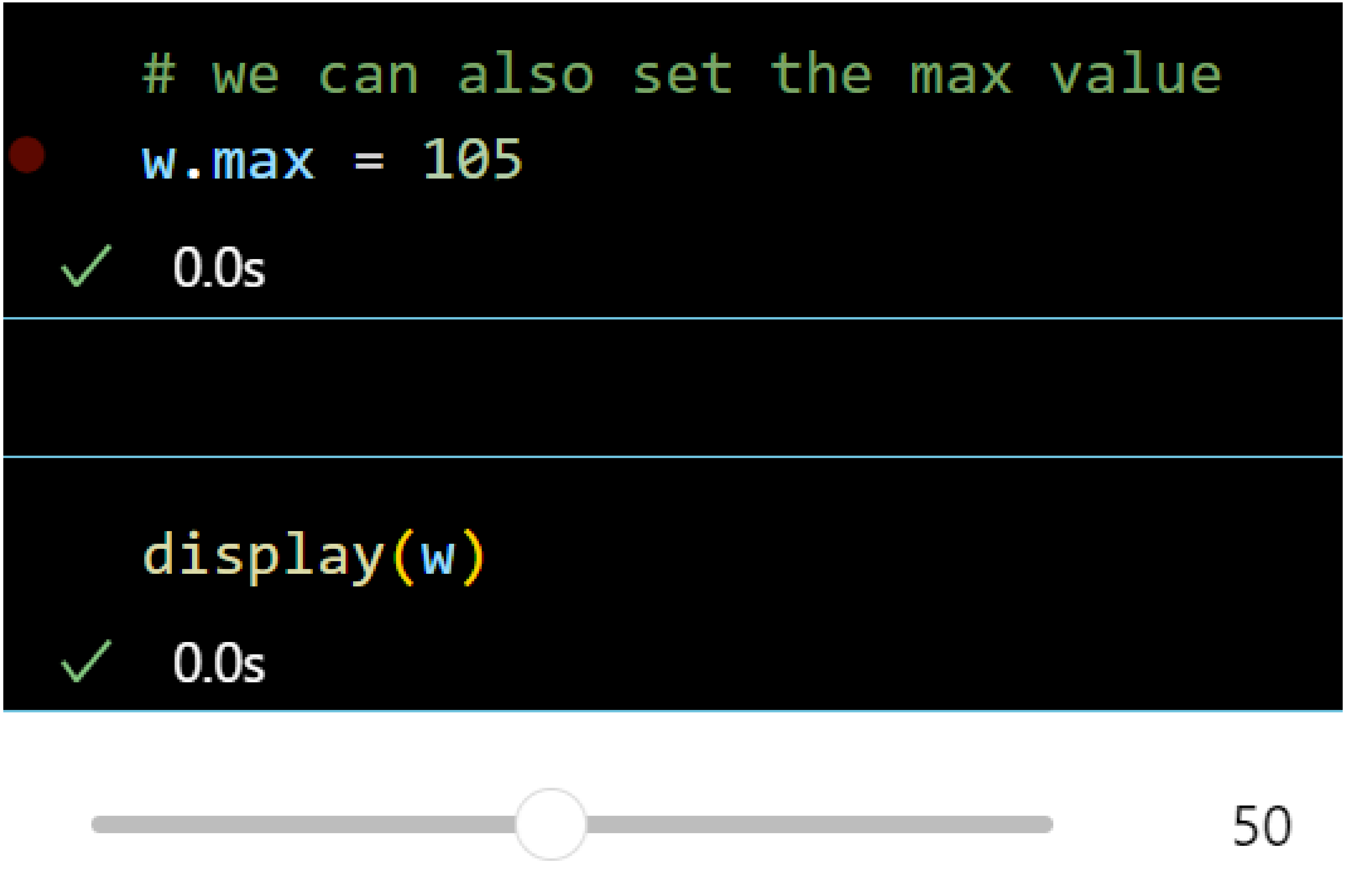
```
# we can make the value fix as well
w.value = 50
```

✓ 0.0s

```
display(w) # you can see that is fixed and we can move if we want
```

✓ 0.0s

```
50
```

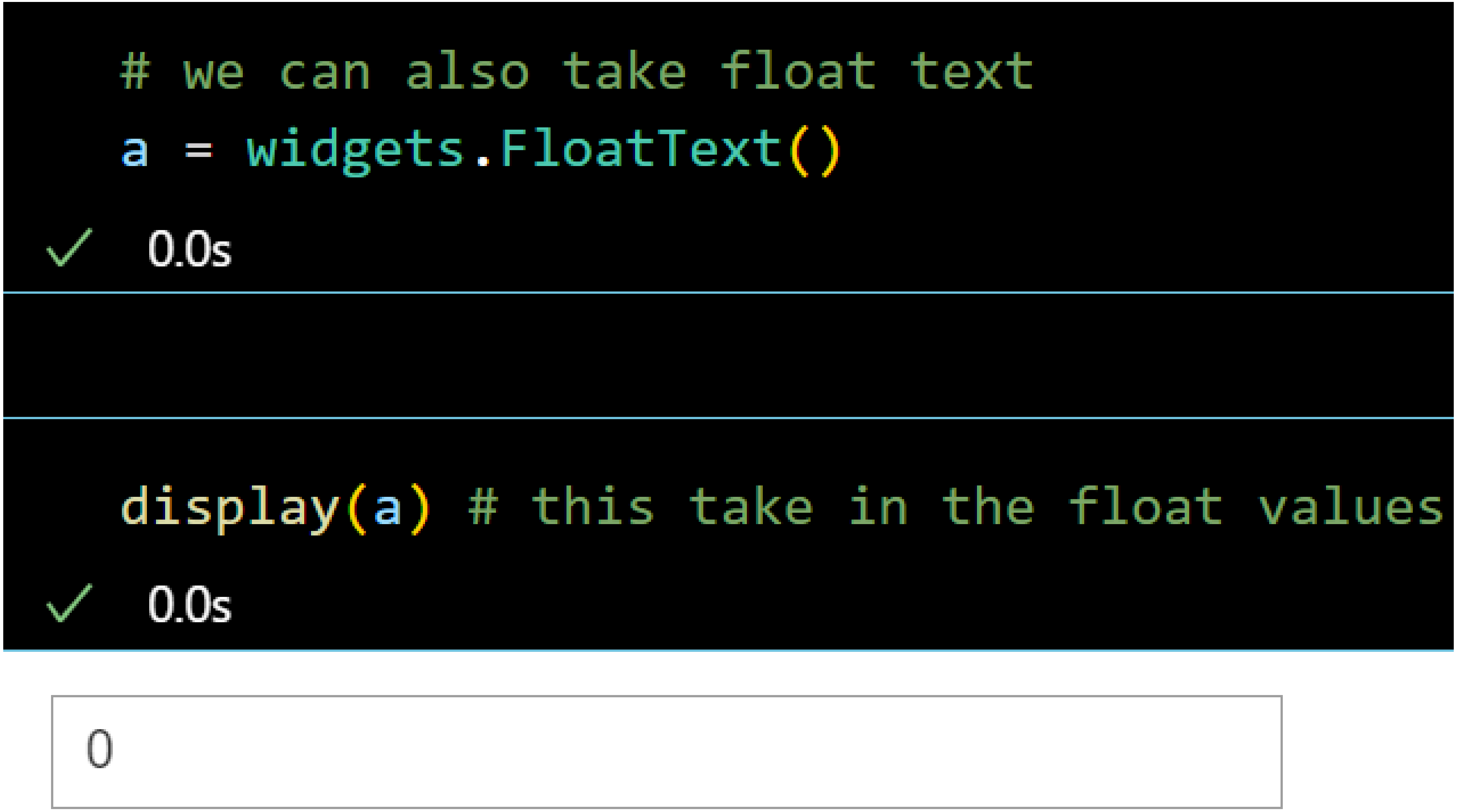


```
# we can also set the max value
w.max = 105
```

✓ 0.0s

```
display(w)
```

✓ 0.0s



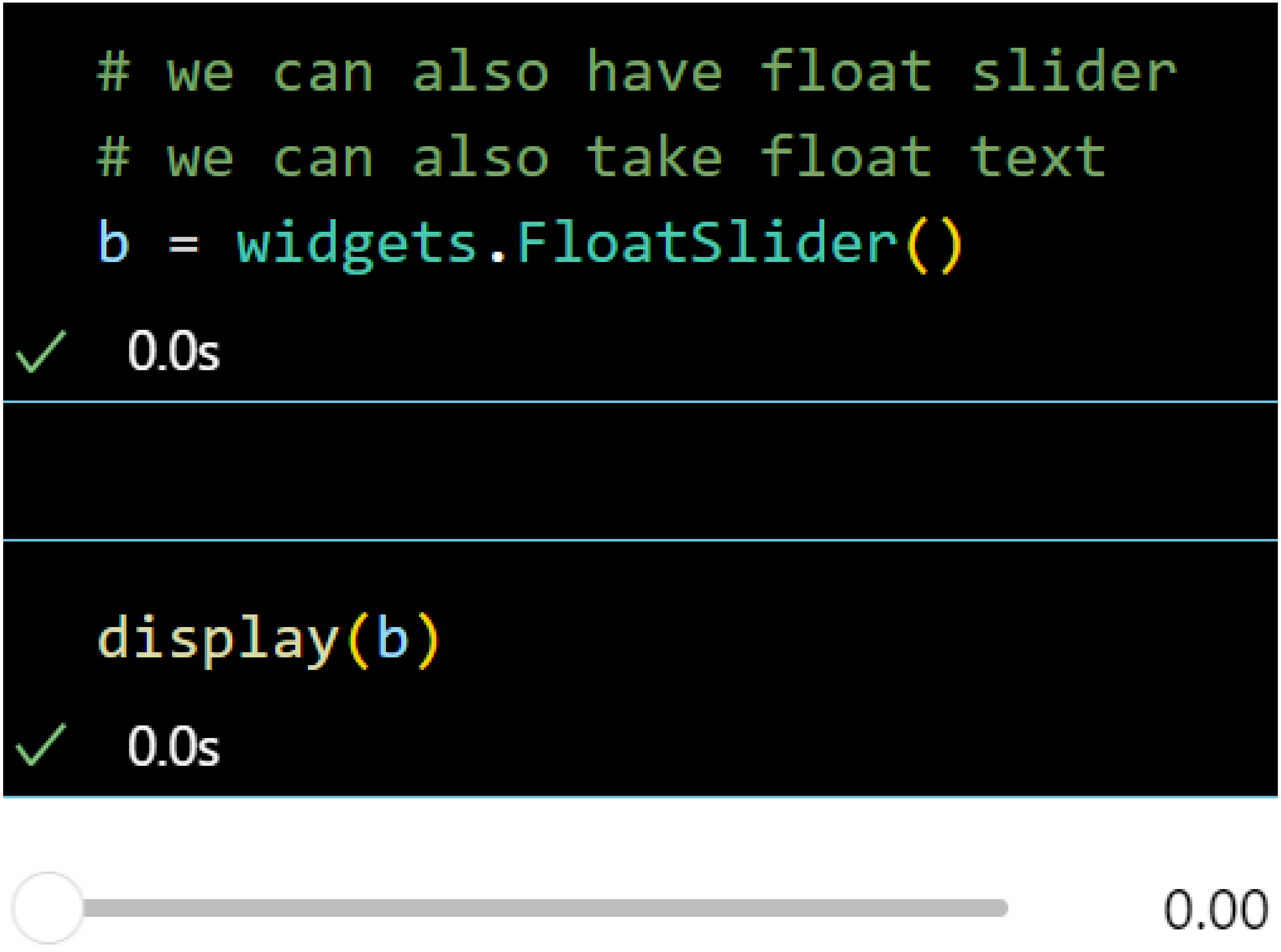
```
# we can also take float text
a = widgets.FloatText()
```

✓ 0.0s

```
display(a) # this take in the float values
```

✓ 0.0s

```
0
```

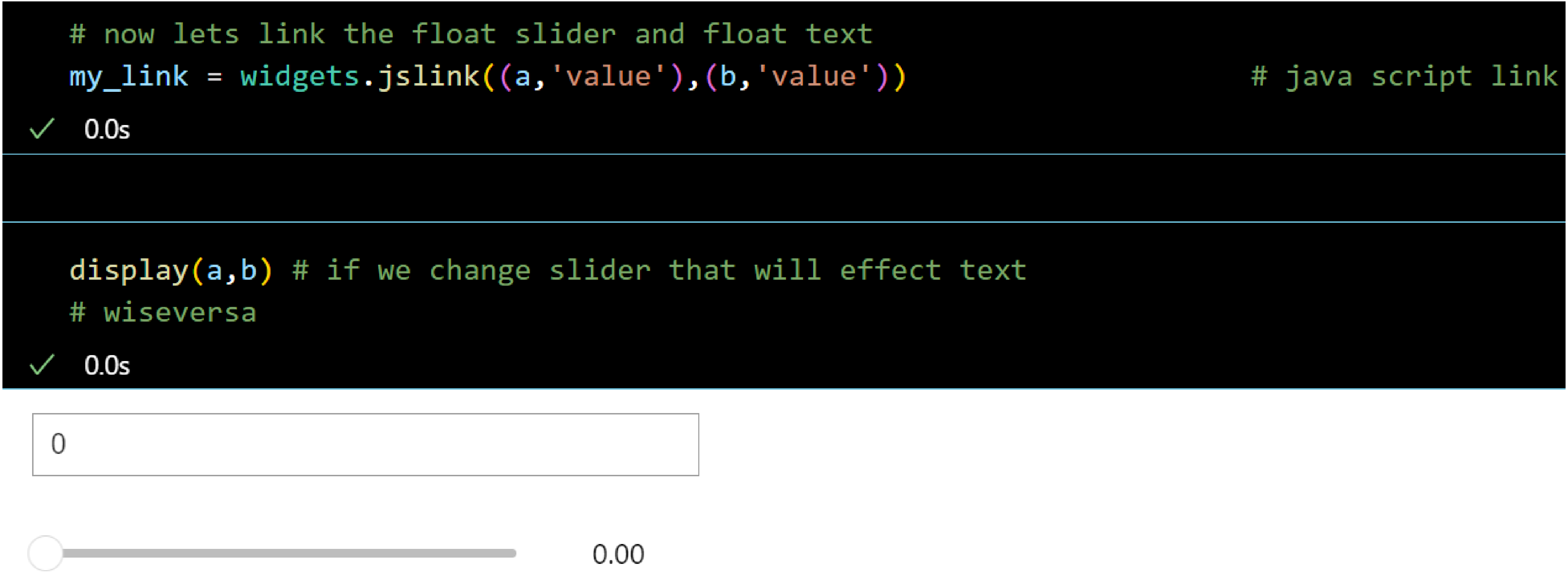


```
# we can also have float slider
# we can also take float text
b = widgets.FloatSlider()
```

✓ 0.0s

```
display(b)
```

✓ 0.0s



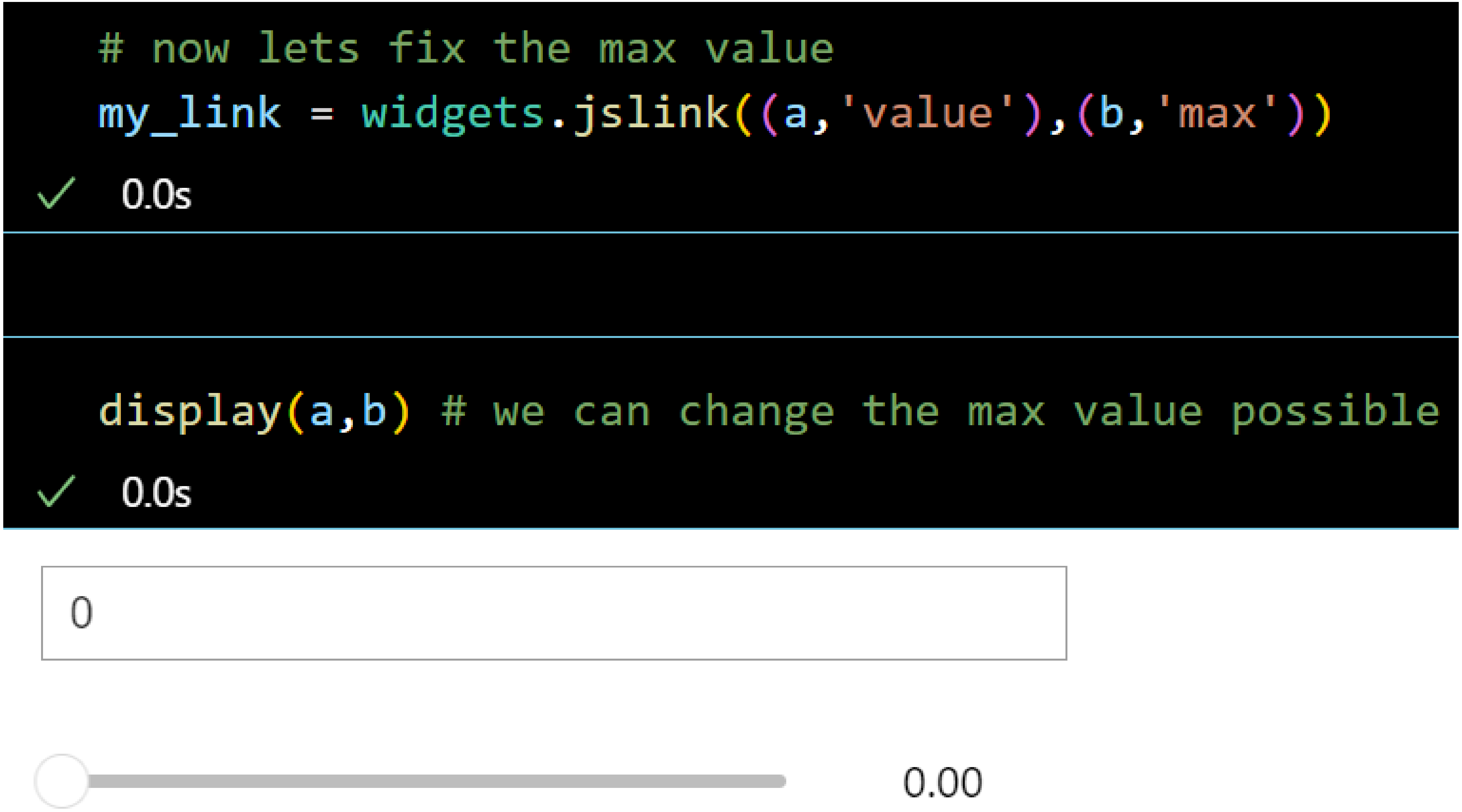
```
# now lets link the float slider and float text
my_link = widgets.jslink((a,'value'),(b,'value'))                    # java script link
```

```
✓ 0.0s
```

```
display(a,b) # if we change slider that will effect text
# wiseversa
```

```
✓ 0.0s
```

```
0

0.00
```



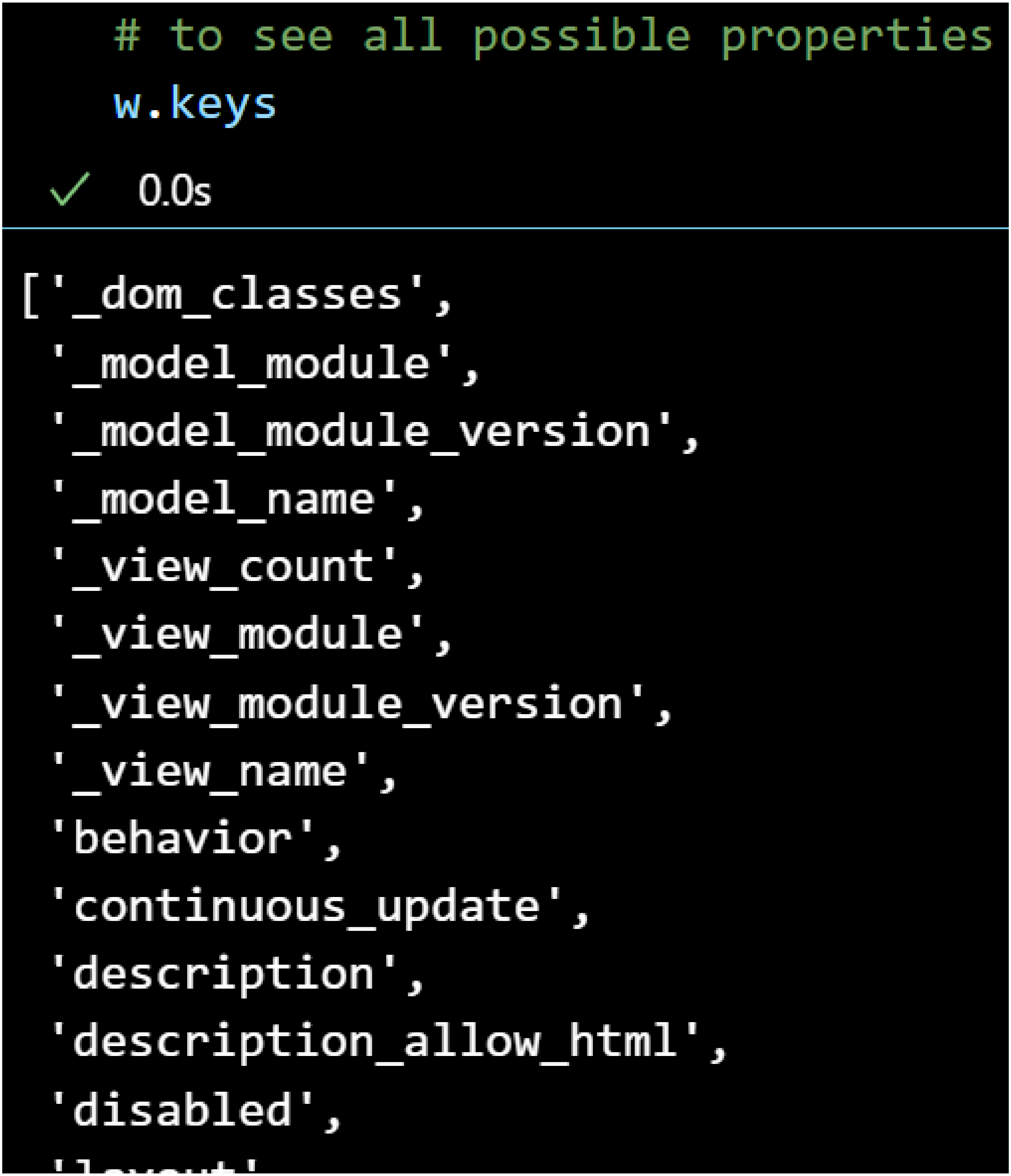
```
# now lets fix the max value
my_link = widgets.jslink((a,'value'),(b,'max'))
```

✓ 0.0s

```
display(a,b) # we can change the max value possible
```

✓ 0.0s

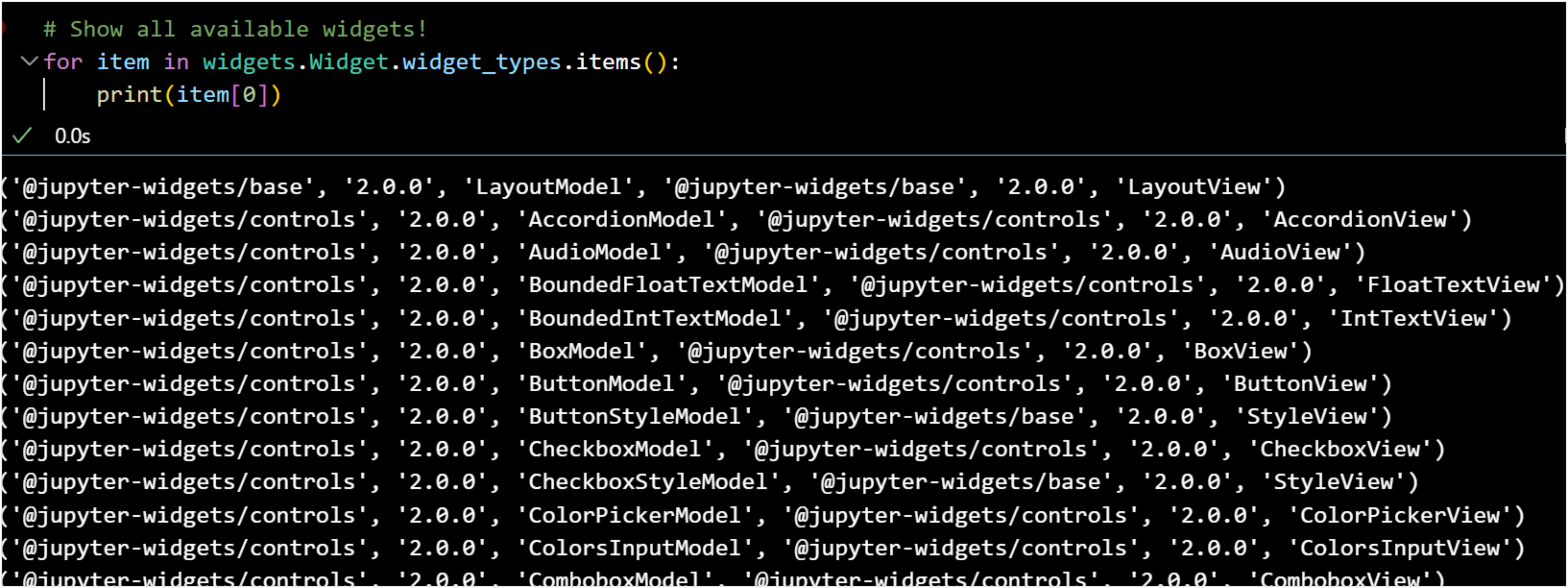
```
0

0.00
```



```
    # to see all possible properties
    w.keys
```

```
 ✓   0.0s
```

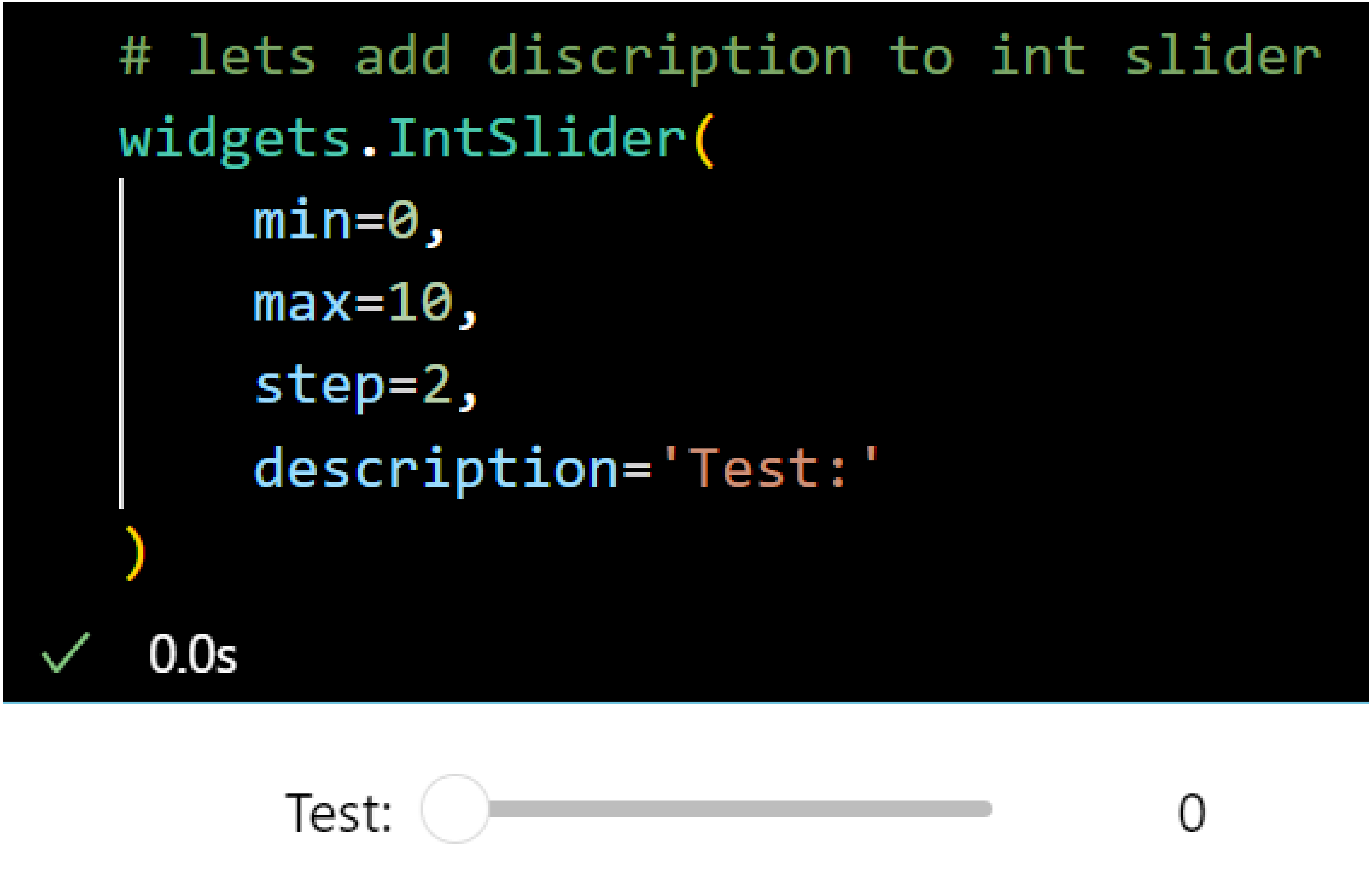
```
['_dom_classes',
 '_model_module',
 '_model_module_version',
 '_model_name',
 '_view_count',
 '_view_module',
 '_view_module_version',
 '_view_name',
 'behavior',
 'continuous_update',
 'description',
 'description_allow_html',
 'disabled',
```



```python
# Show all available widgets!
for item in widgets.Widget.widget_types.items():
    print(item[0])
```

✓ 0.0s

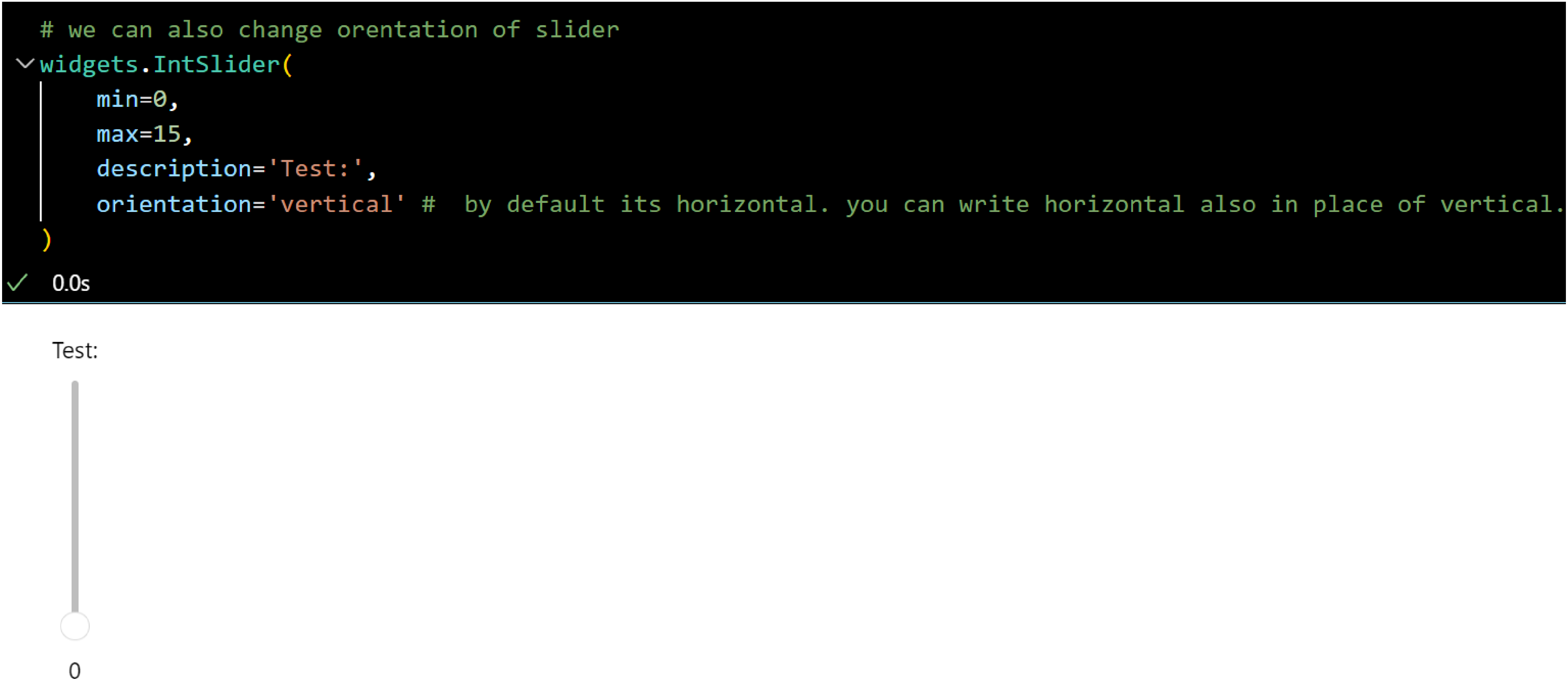
```
('@jupyter-widgets/base', '2.0.0', 'LayoutModel', '@jupyter-widgets/base', '2.0.0', 'LayoutView')
('@jupyter-widgets/controls', '2.0.0', 'AccordionModel', '@jupyter-widgets/controls', '2.0.0', 'AccordionView')
('@jupyter-widgets/controls', '2.0.0', 'AudioModel', '@jupyter-widgets/controls', '2.0.0', 'AudioView')
('@jupyter-widgets/controls', '2.0.0', 'BoundedFloatTextModel', '@jupyter-widgets/controls', '2.0.0', 'FloatTextView')
('@jupyter-widgets/controls', '2.0.0', 'BoundedIntTextModel', '@jupyter-widgets/controls', '2.0.0', 'IntTextView')
('@jupyter-widgets/controls', '2.0.0', 'BoxModel', '@jupyter-widgets/controls', '2.0.0', 'BoxView')
('@jupyter-widgets/controls', '2.0.0', 'ButtonModel', '@jupyter-widgets/controls', '2.0.0', 'ButtonView')
('@jupyter-widgets/controls', '2.0.0', 'ButtonStyleModel', '@jupyter-widgets/base', '2.0.0', 'StyleView')
('@jupyter-widgets/controls', '2.0.0', 'CheckboxModel', '@jupyter-widgets/controls', '2.0.0', 'CheckboxView')
('@jupyter-widgets/controls', '2.0.0', 'CheckboxStyleModel', '@jupyter-widgets/base', '2.0.0', 'StyleView')
('@jupyter-widgets/controls', '2.0.0', 'ColorPickerModel', '@jupyter-widgets/controls', '2.0.0', 'ColorPickerView')
('@jupyter-widgets/controls', '2.0.0', 'ColorsInputModel', '@jupyter-widgets/controls', '2.0.0', 'ColorsInputView')
('@jupyter-widgets/controls', '2.0.0', 'ComboboxModel', '@jupyter-widgets/controls', '2.0.0', 'ComboboxView')
```



```
# lets add discription to int slider
widgets.IntSlider(
        min=0,
        max=10,
        step=2,
        description='Test:'
)
```

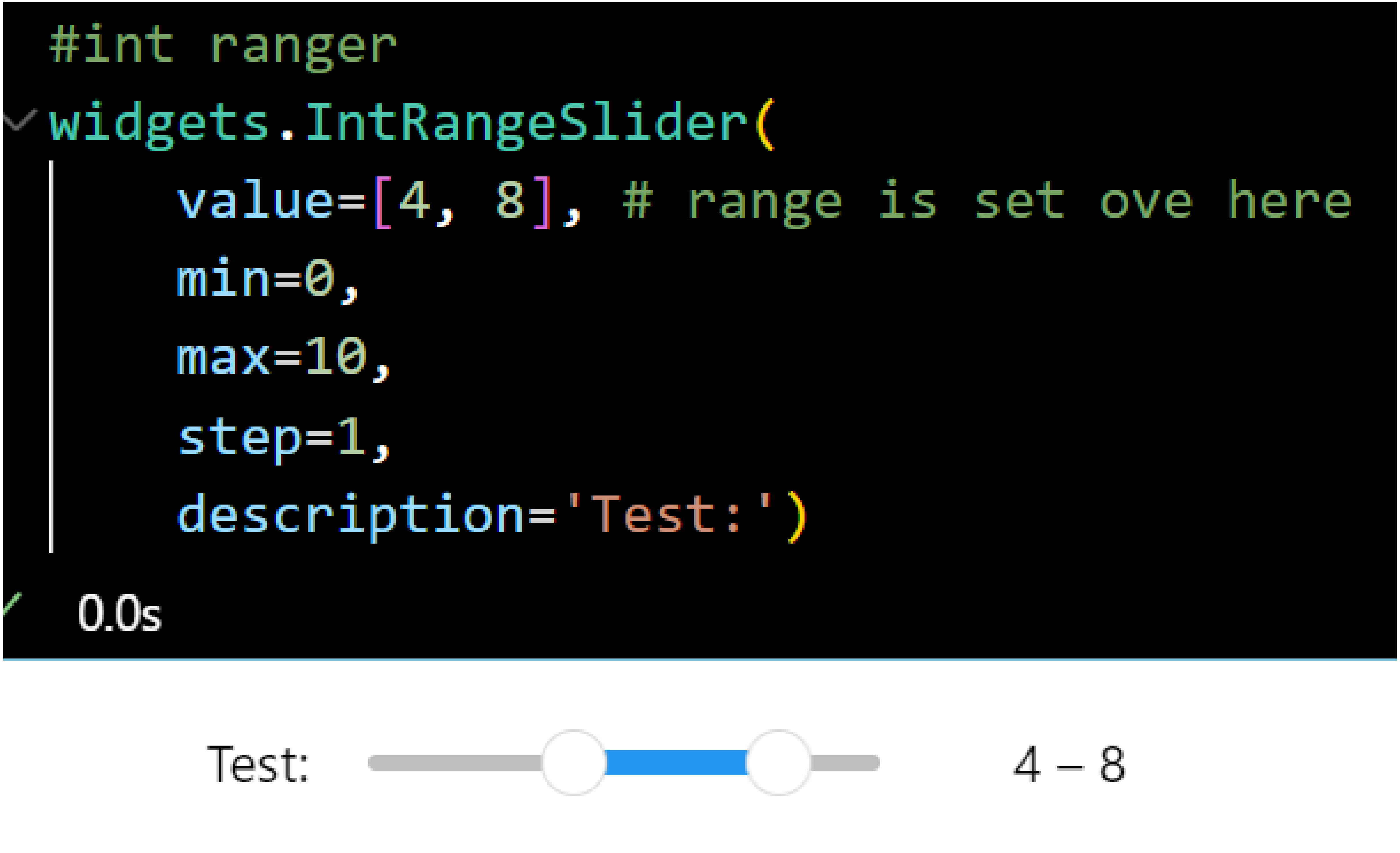
✓ 0.0s

Test: 0



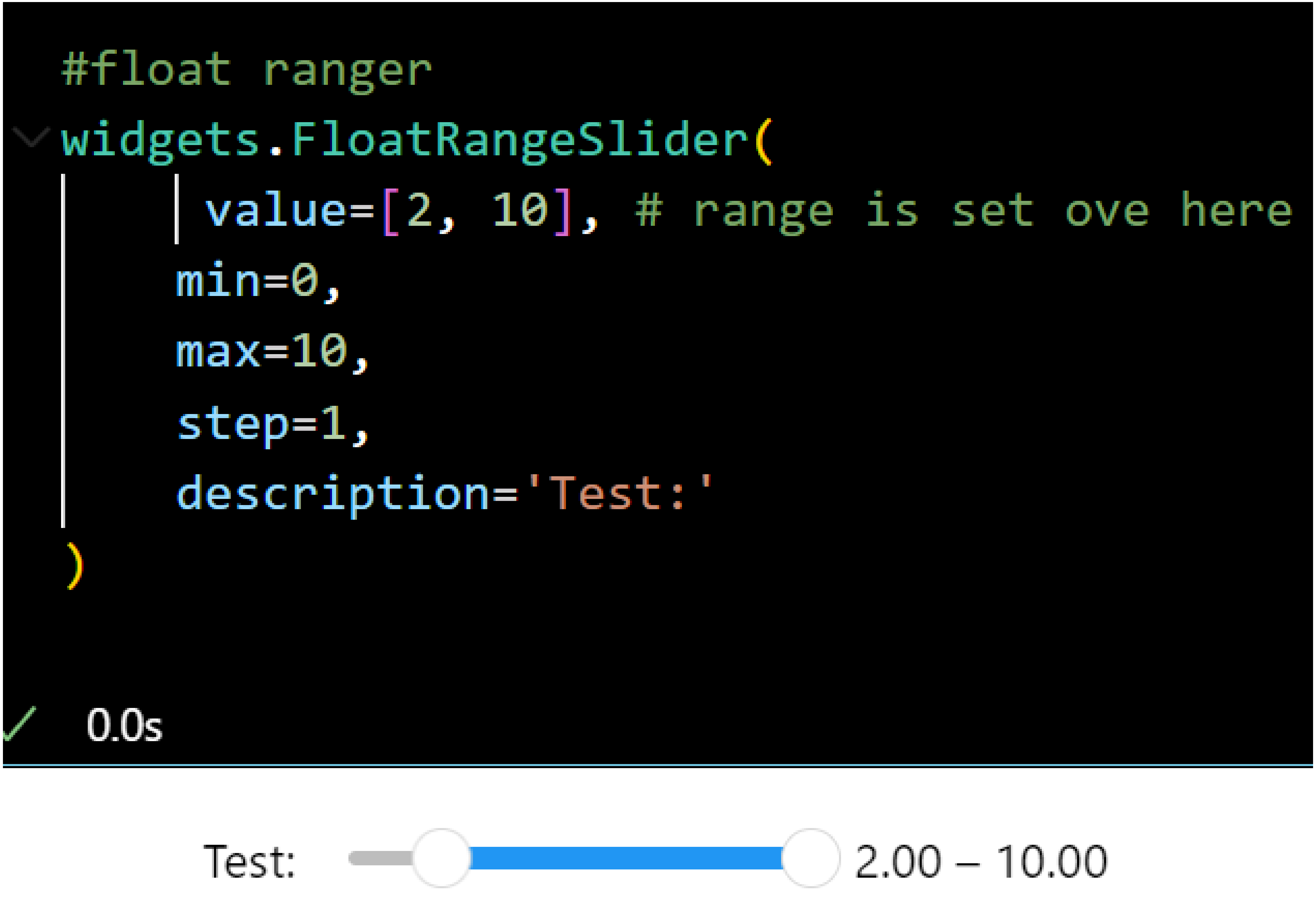
```
# we can also change orentation of slider
widgets.IntSlider(
    min=0,
    max=15,
    description='Test:',
    orientation='vertical' #  by default its horizontal. you can write horizontal also in place of vertical.
)
```

✓ 0.0s



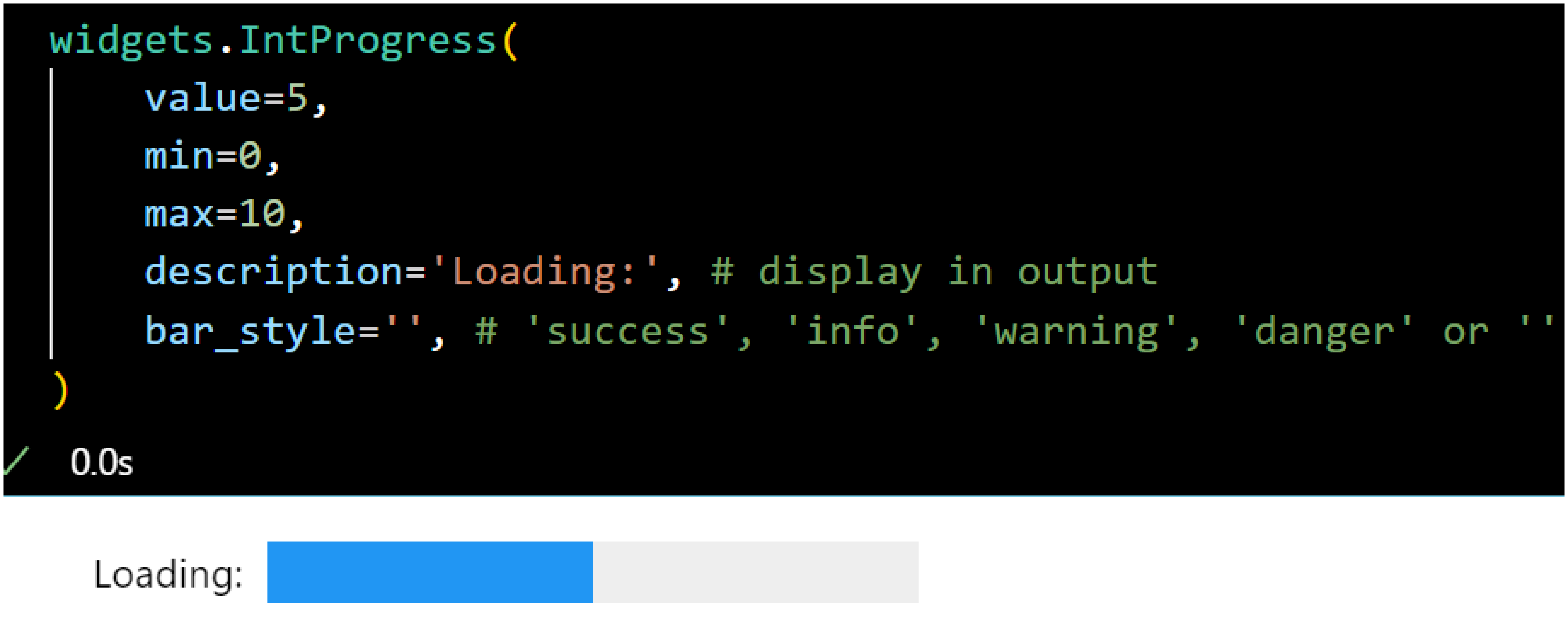
```
#int ranger
widgets.IntRangeSlider(
    value=[4, 8], # range is set ove here
    min=0,
    max=10,
    step=1,
    description='Test:')
```

```
✓ 0.0s
```



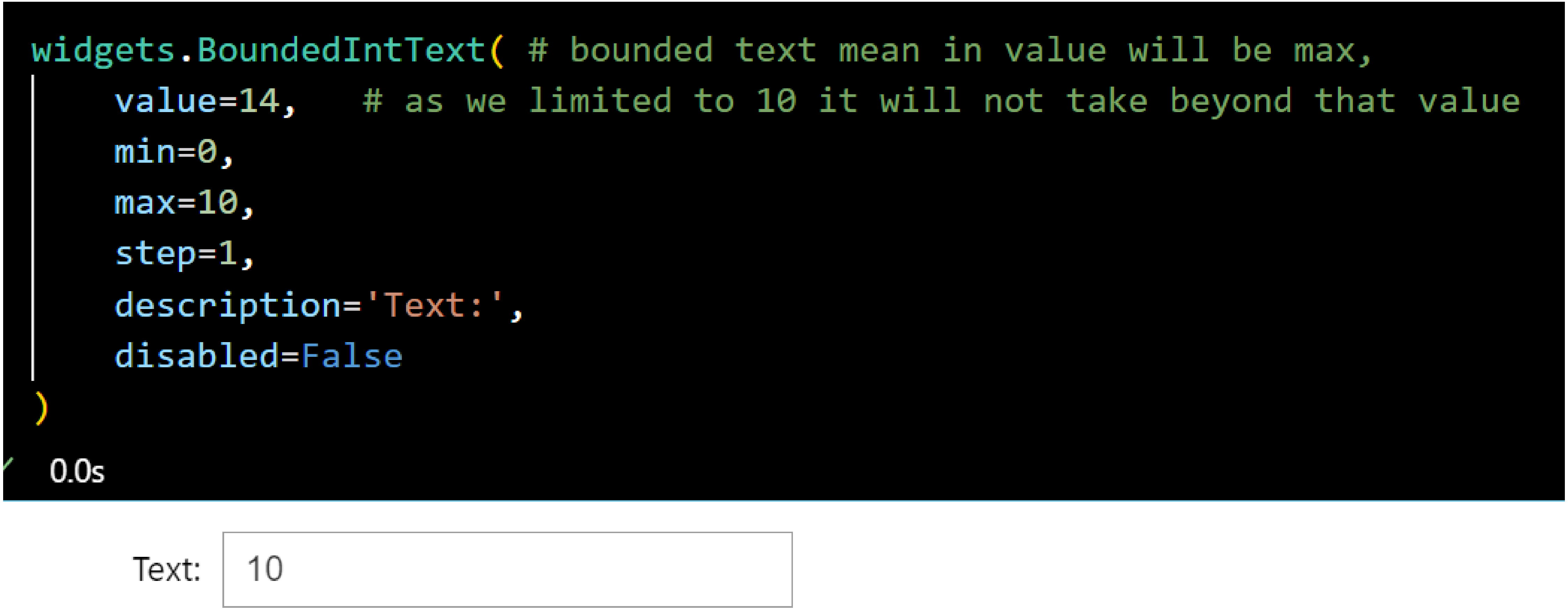
```
#float ranger
widgets.FloatRangeSlider(
        value=[2, 10], # range is set ove here
        min=0,
        max=10,
        step=1,
        description='Test:'
)
```

```
✓ 0.0s
```



```
widgets.IntProgress(
    value=5,
    min=0,
    max=10,
    description='Loading:', # display in output
    bar_style='', # 'success', 'info', 'warning', 'danger' or ''
)
```

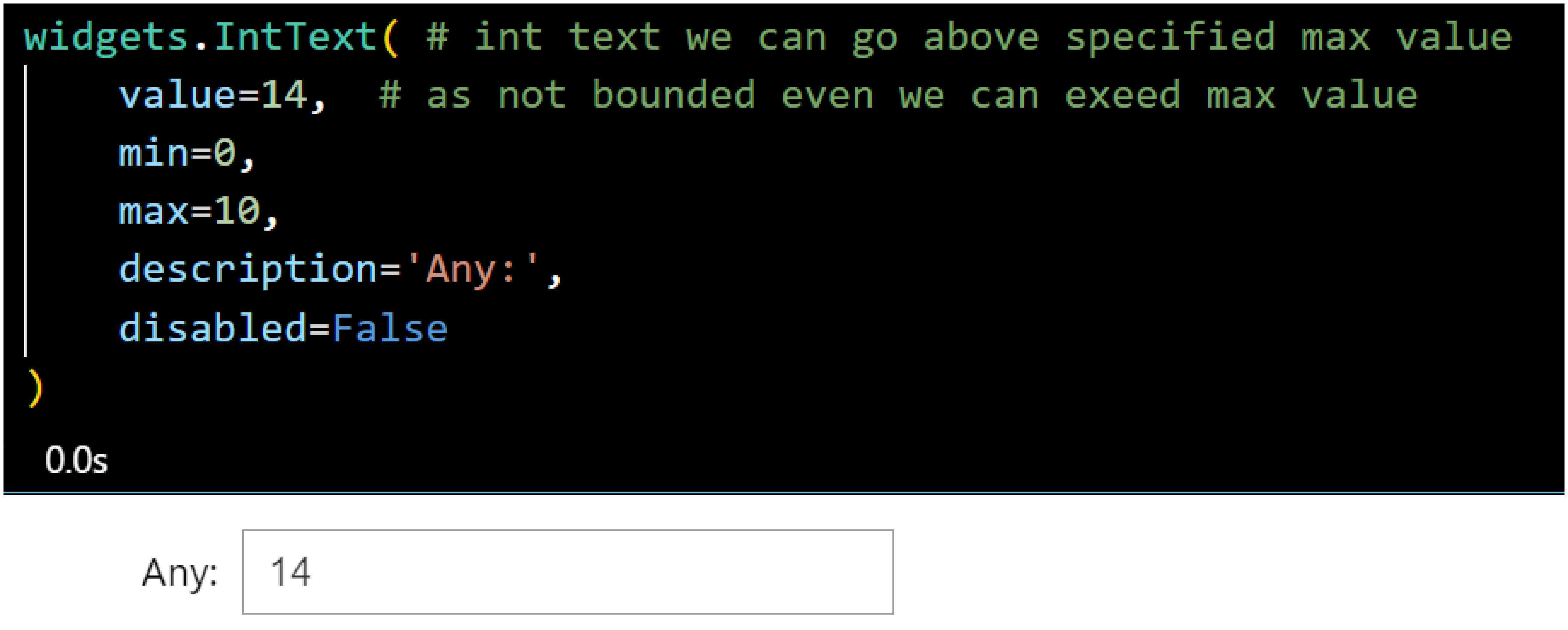
✓ 0.0s



```python
widgets.BoundedIntText( # bounded text mean in value will be max,
    value=14,   # as we limited to 10 it will not take beyond that value
    min=0,
    max=10,
    step=1,
    description='Text:',
    disabled=False
)
```

✓ 0.0s

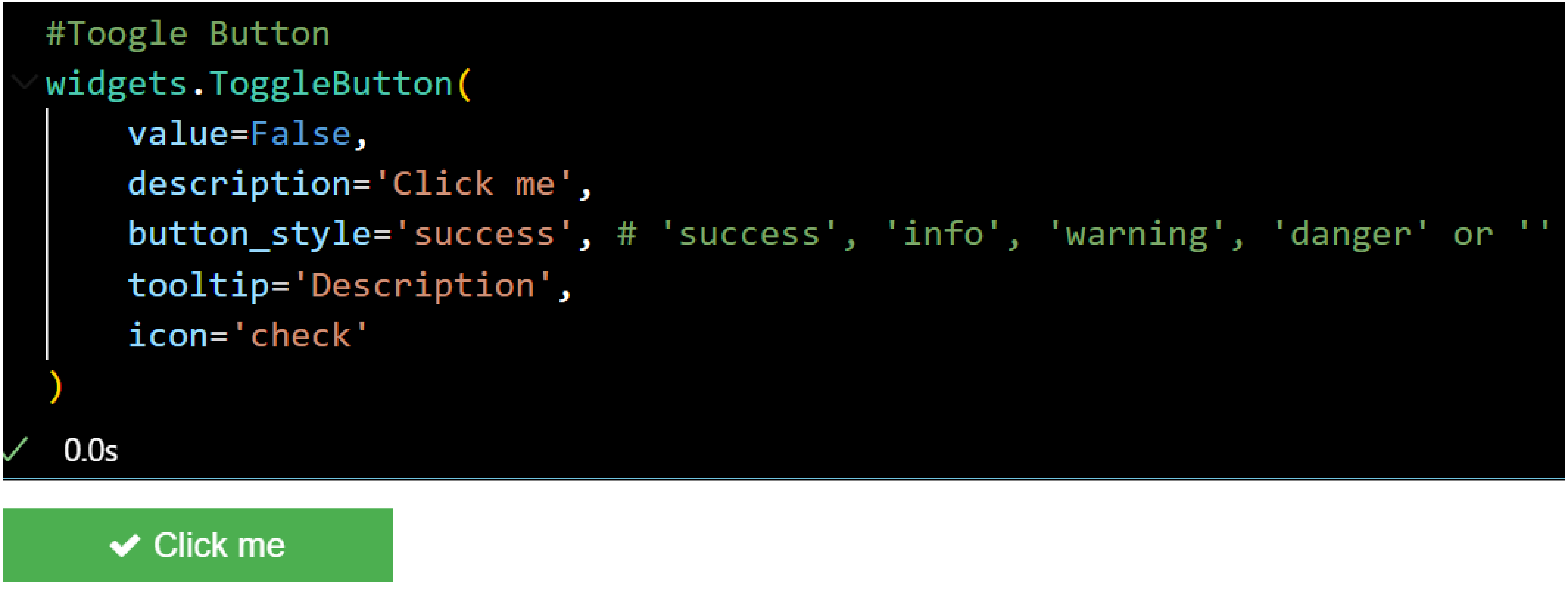
```
Text: 10
```



```
widgets.IntText( # int text we can go above specified max value
    value=14,  # as not bounded even we can exeed max value
    min=0,
    max=10,
    description='Any:',
    disabled=False
)
```

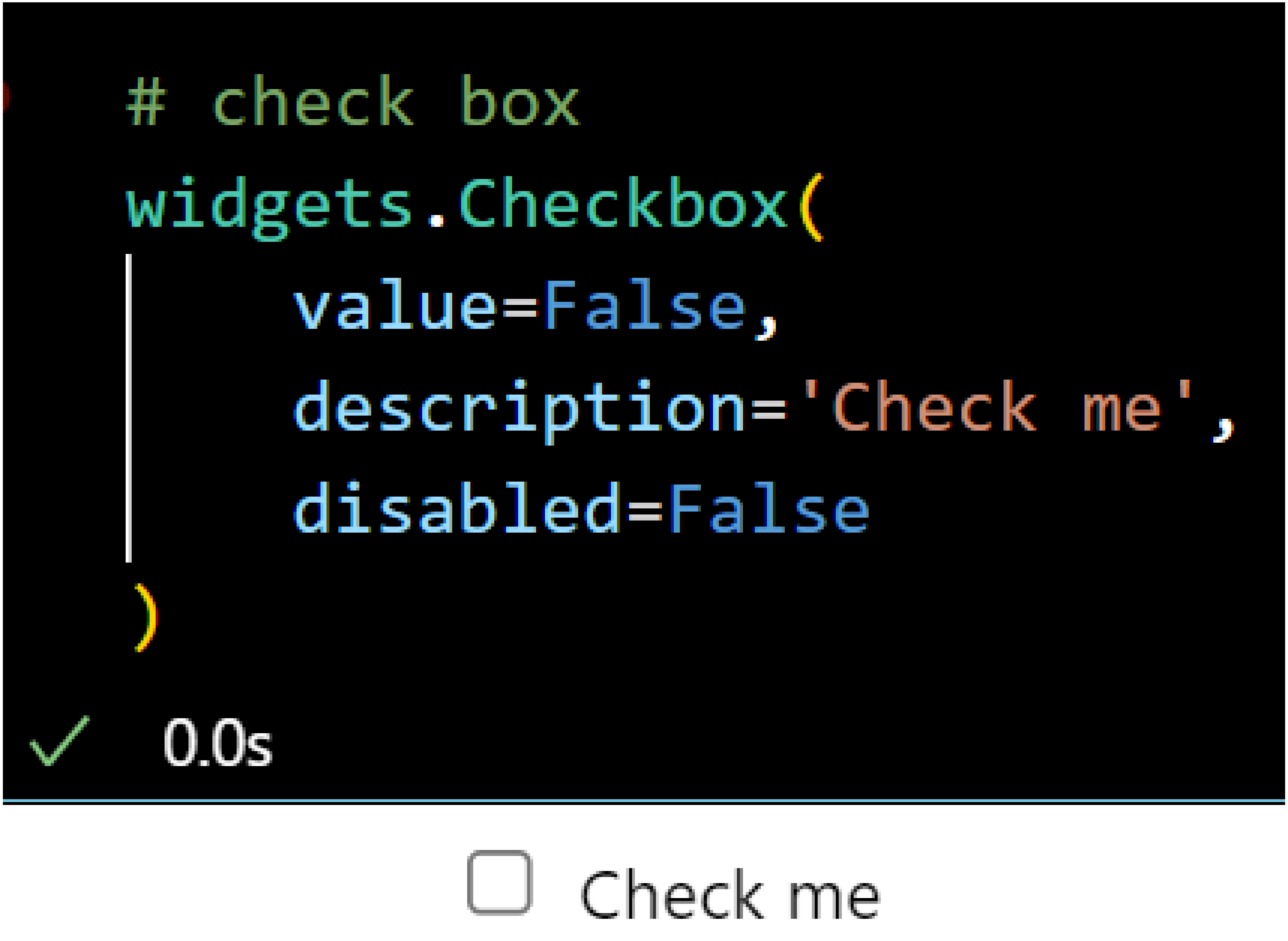
0.0s

Any: 14



```python
#Toogle Button
widgets.ToggleButton(
    value=False,
    description='Click me',
    button_style='success', # 'success', 'info', 'warning', 'danger' or ''
    tooltip='Description',
    icon='check'
)
```

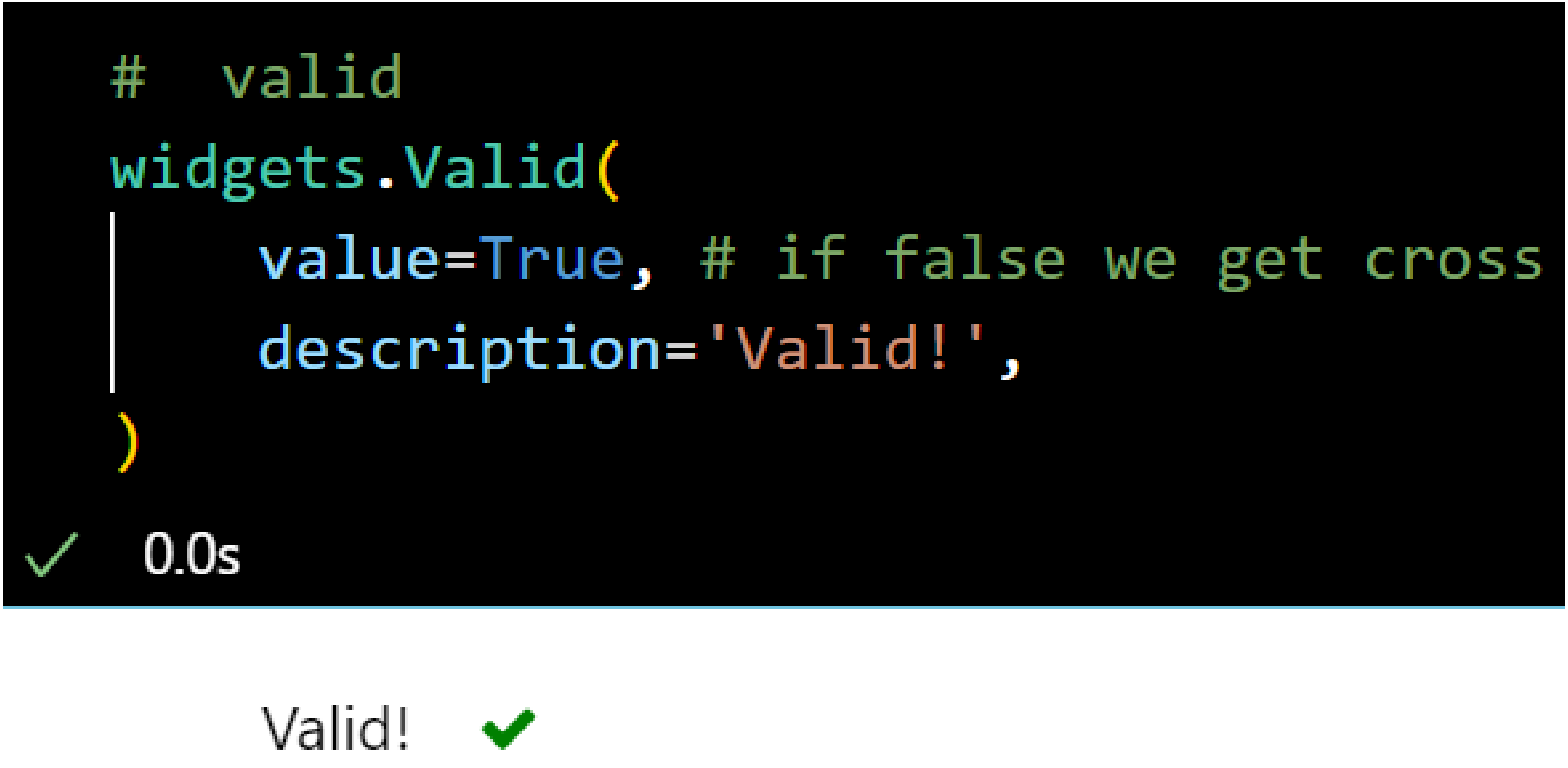
✓ 0.0s



```
# check box
widgets.Checkbox(
        value=False,
        description='Check me',
        disabled=False
)
```

✓ 0.0s

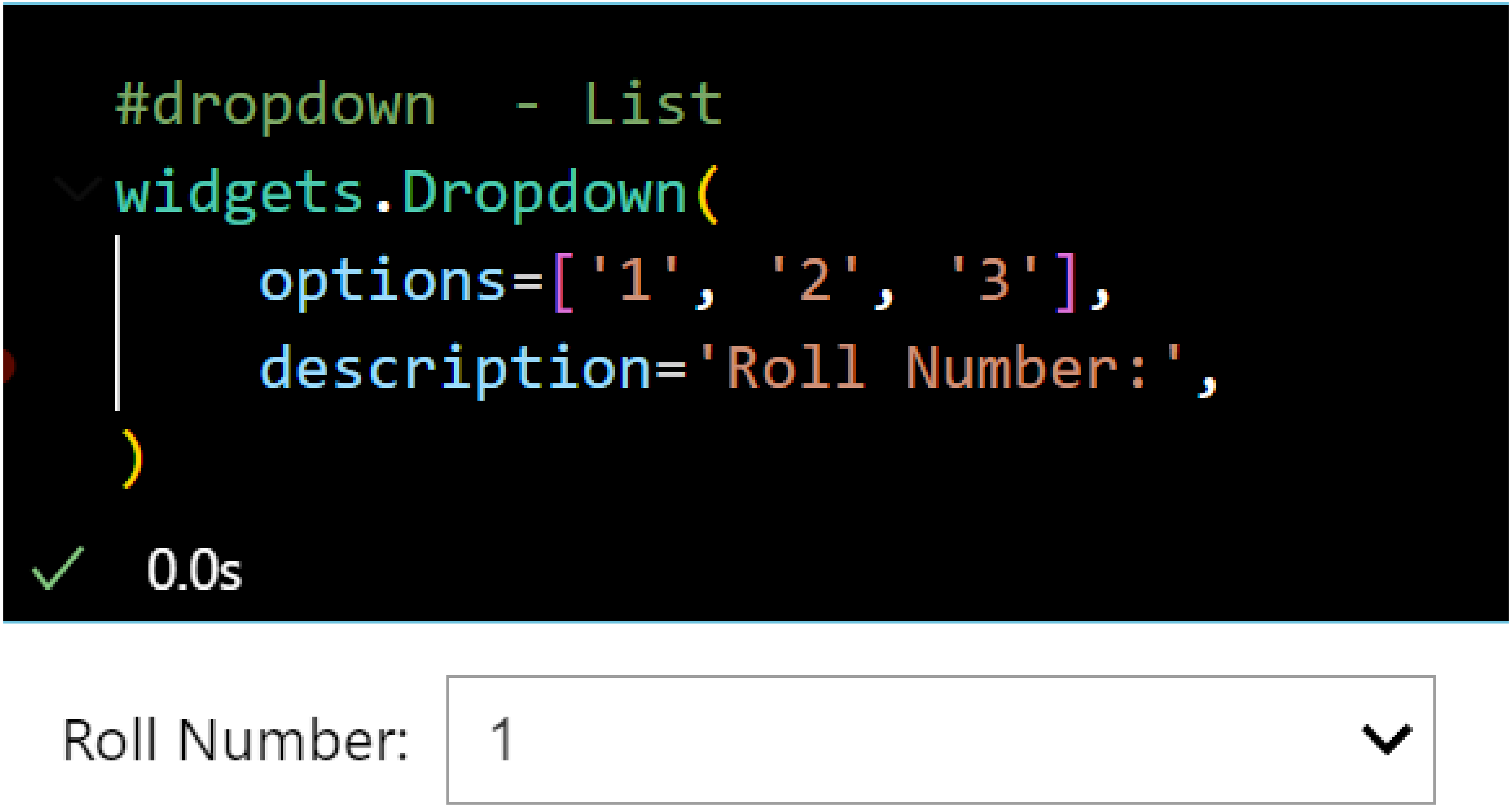
☐ Check me



```
#  valid
widgets.Valid(
    value=True, # if false we get cross
    description='Valid!',
)
```

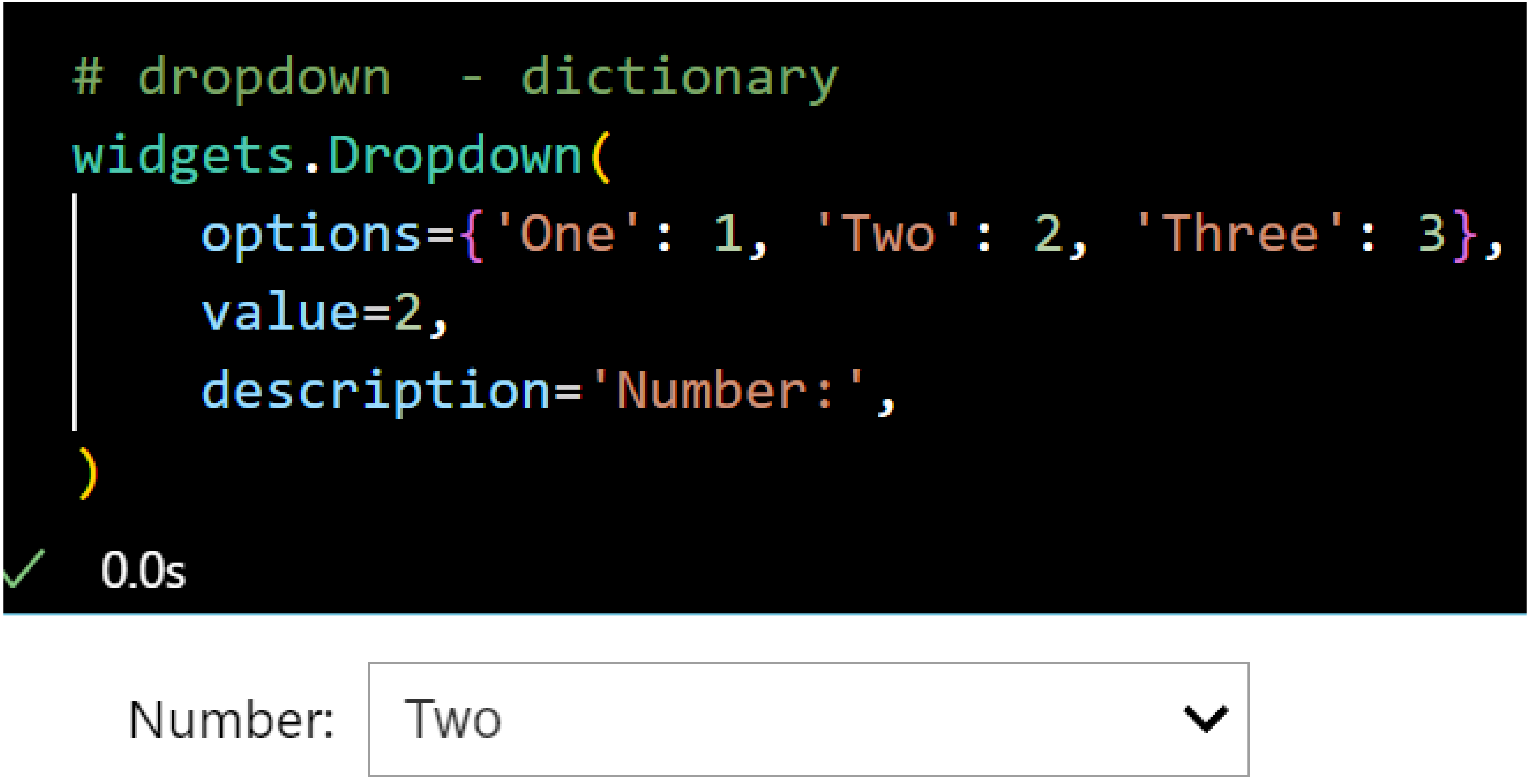
✓ 0.0s

```
Valid! ✔
```



```
#dropdown  - List
widgets.Dropdown(
    options=['1', '2', '3'],
    description='Roll Number:',
)
```

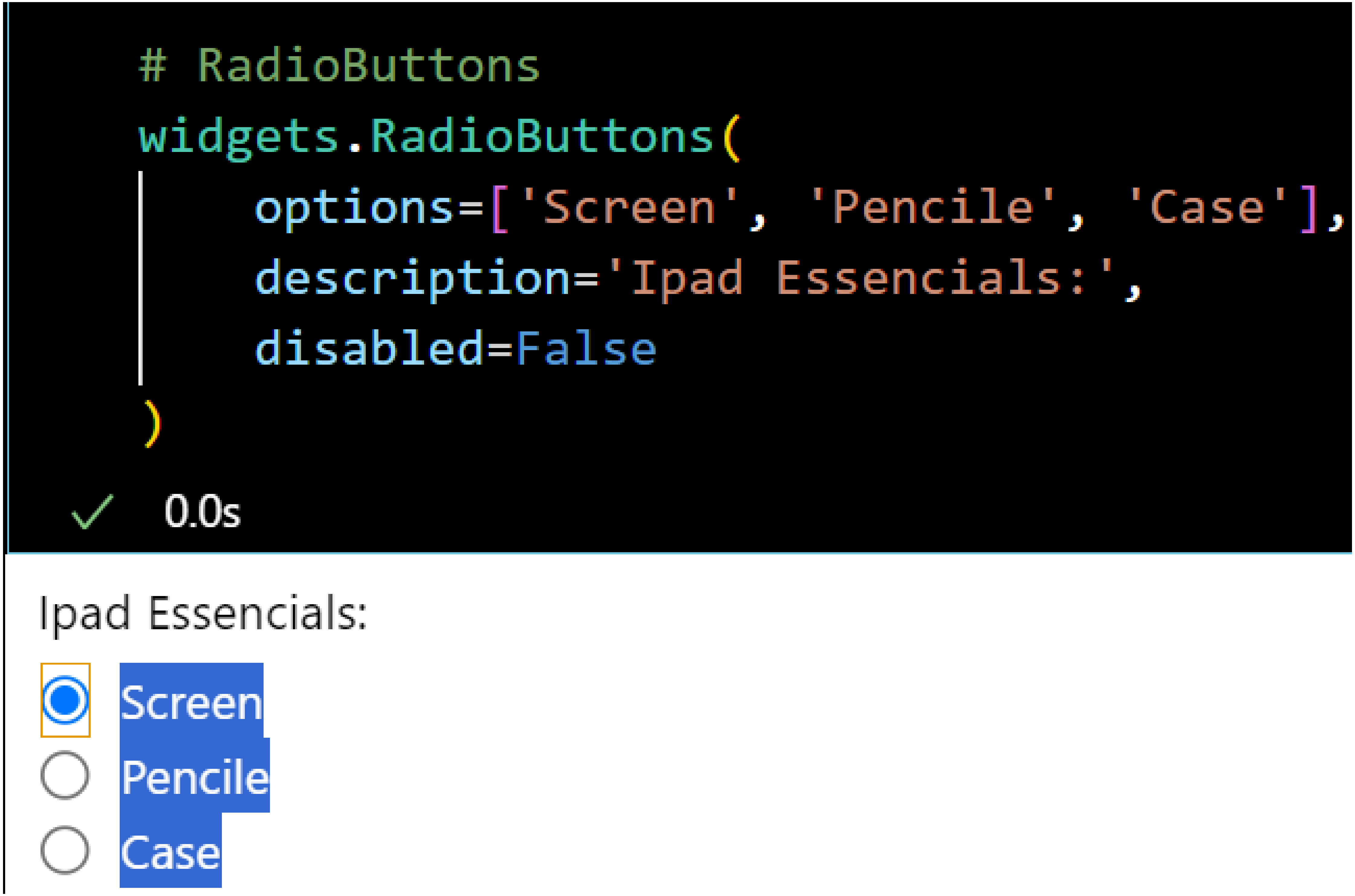
✓ 0.0s



```
# dropdown  - dictionary
widgets.Dropdown(
    options={'One': 1, 'Two': 2, 'Three': 3},
    value=2,
    description='Number:',
)
```

✓ 0.0s

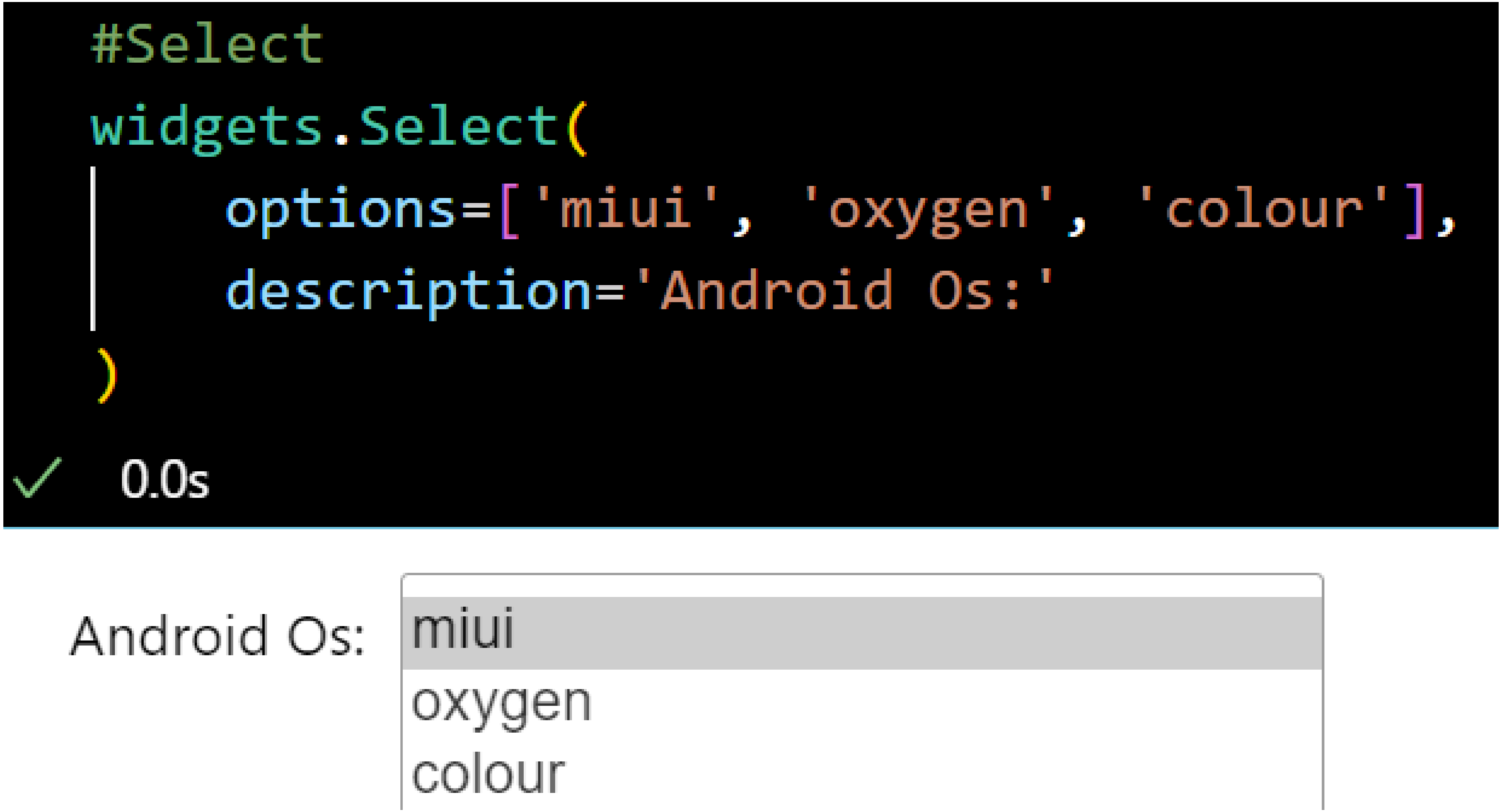
```
Number: [Two ▾]
```



```
# RadioButtons
widgets.RadioButtons(
    options=['Screen', 'Pencile', 'Case'],
    description='Ipad Essencials:',
    disabled=False
)
```

✓ 0.0s

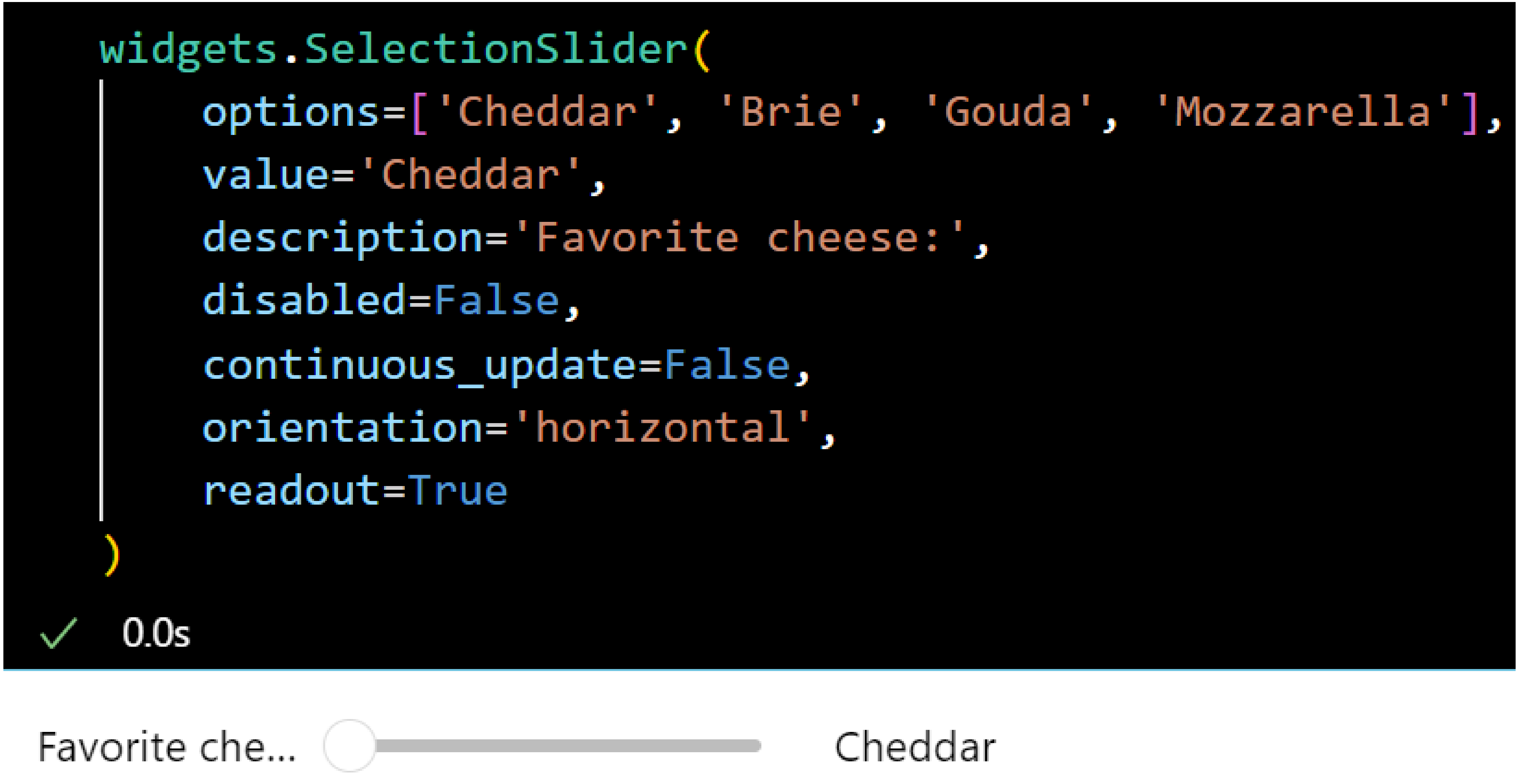
```
Ipad Essencials:
● Screen
○ Pencile
○ Case
```



```
#Select
widgets.Select(
    options=['miui', 'oxygen', 'colour'],
    description='Android Os:'
)
```

✓ 0.0s

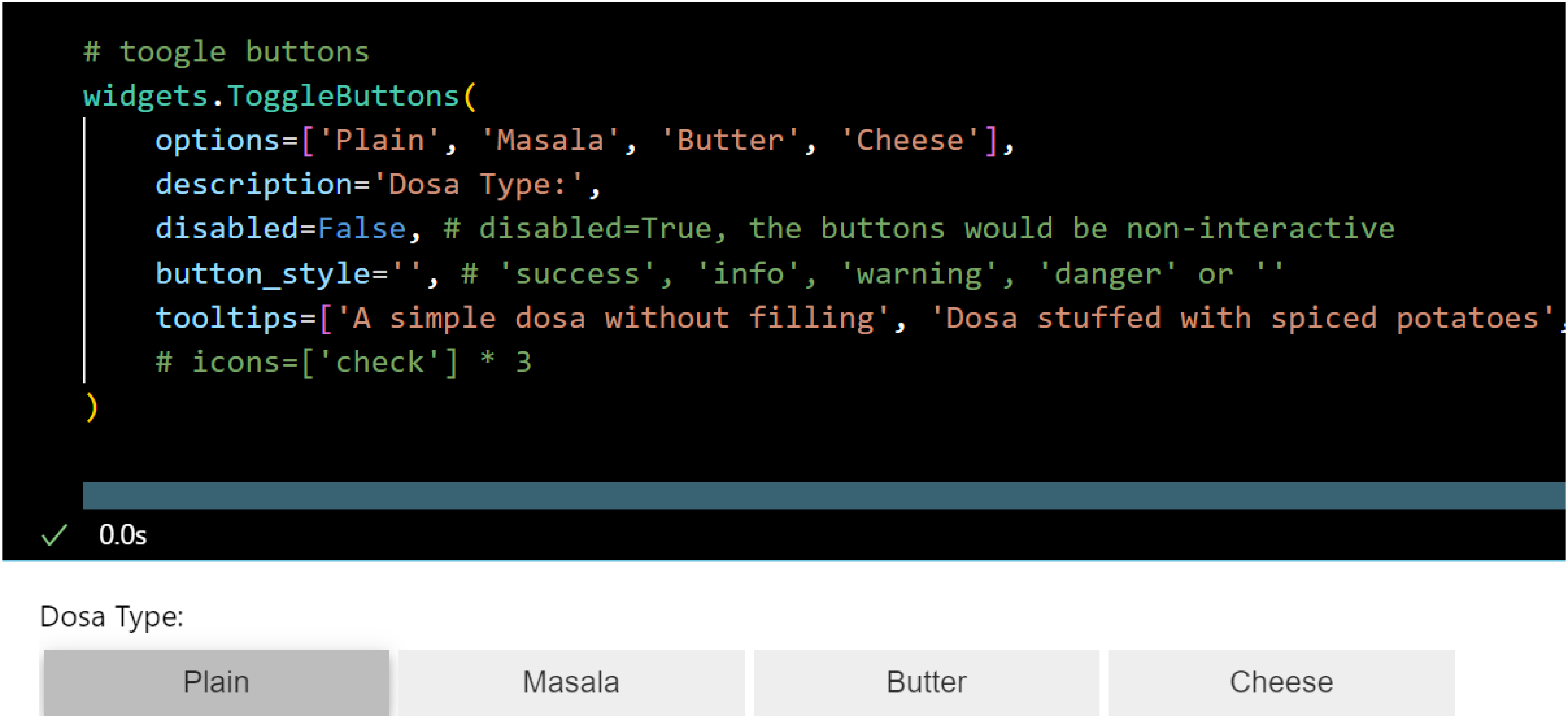
```
Android Os: miui
            oxygen
            colour
```



```
widgets.SelectionSlider(
    options=['Cheddar', 'Brie', 'Gouda', 'Mozzarella'],
    value='Cheddar',
    description='Favorite cheese:',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True
)
```

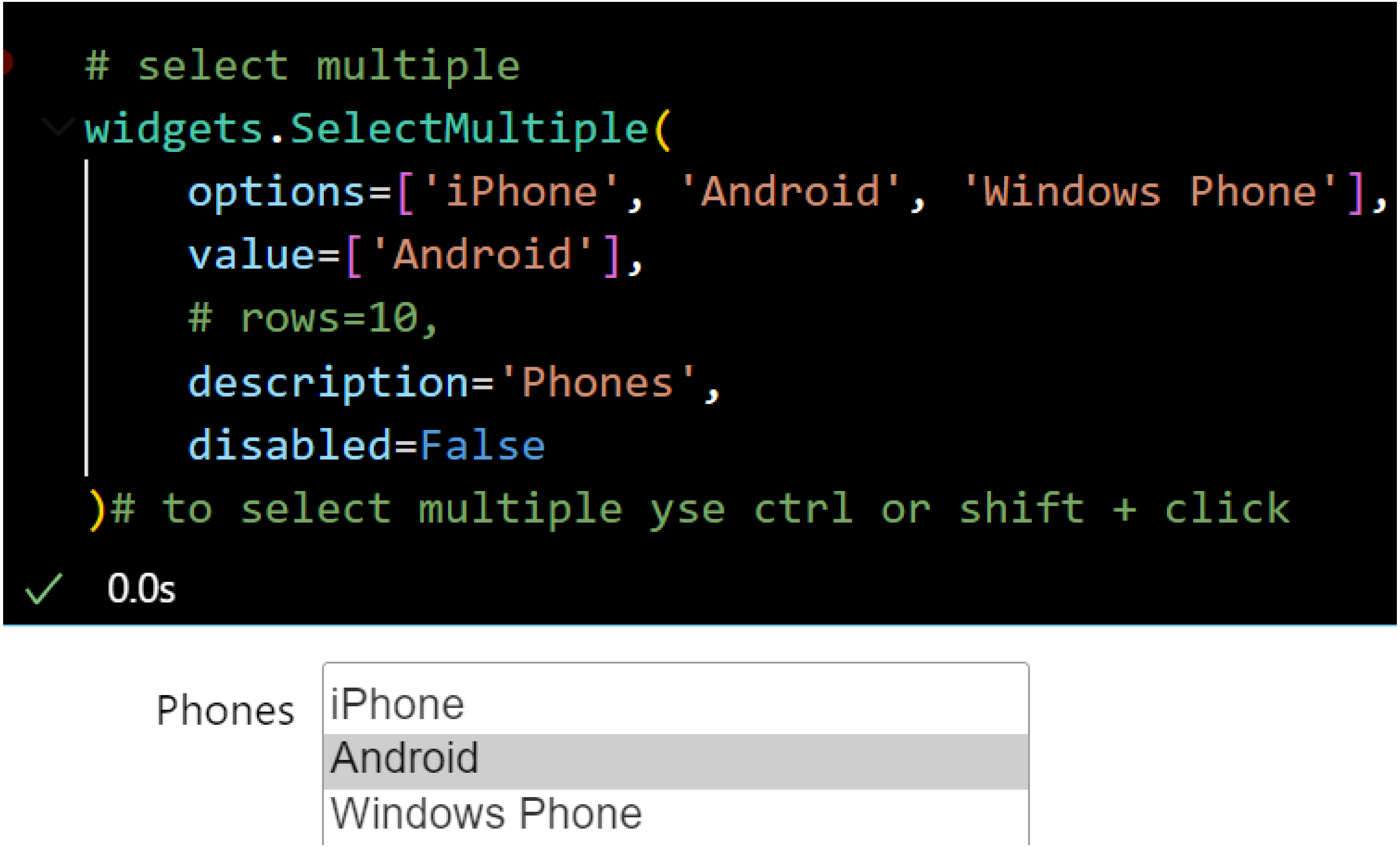
✓ 0.0s

Favorite che... Cheddar



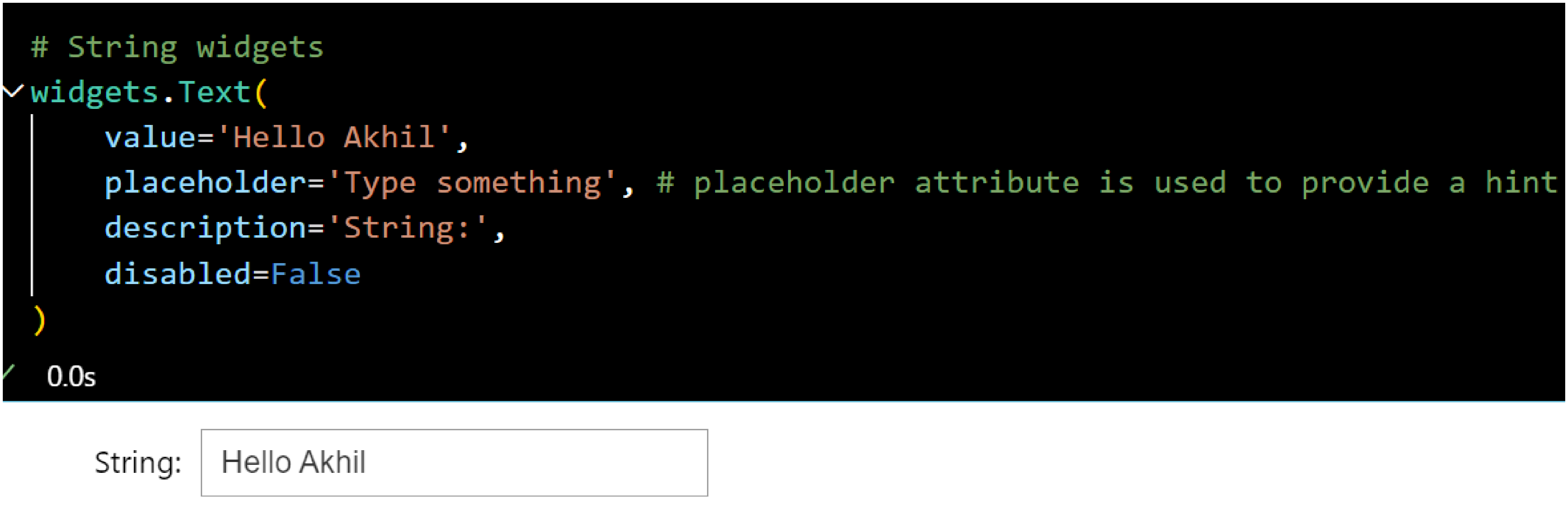
```
# toogle buttons
widgets.ToggleButtons(
    options=['Plain', 'Masala', 'Butter', 'Cheese'],
    description='Dosa Type:',
    disabled=False, # disabled=True, the buttons would be non-interactive
    button_style='', # 'success', 'info', 'warning', 'danger' or ''
    tooltips=['A simple dosa without filling', 'Dosa stuffed with spiced potatoes'
    # icons=['check'] * 3
)
```

✓ 0.0s



```
# select multiple
widgets.SelectMultiple(
    options=['iPhone', 'Android', 'Windows Phone'],
    value=['Android'],
    # rows=10,
    description='Phones',
    disabled=False
)# to select multiple yse ctrl or shift + click
```

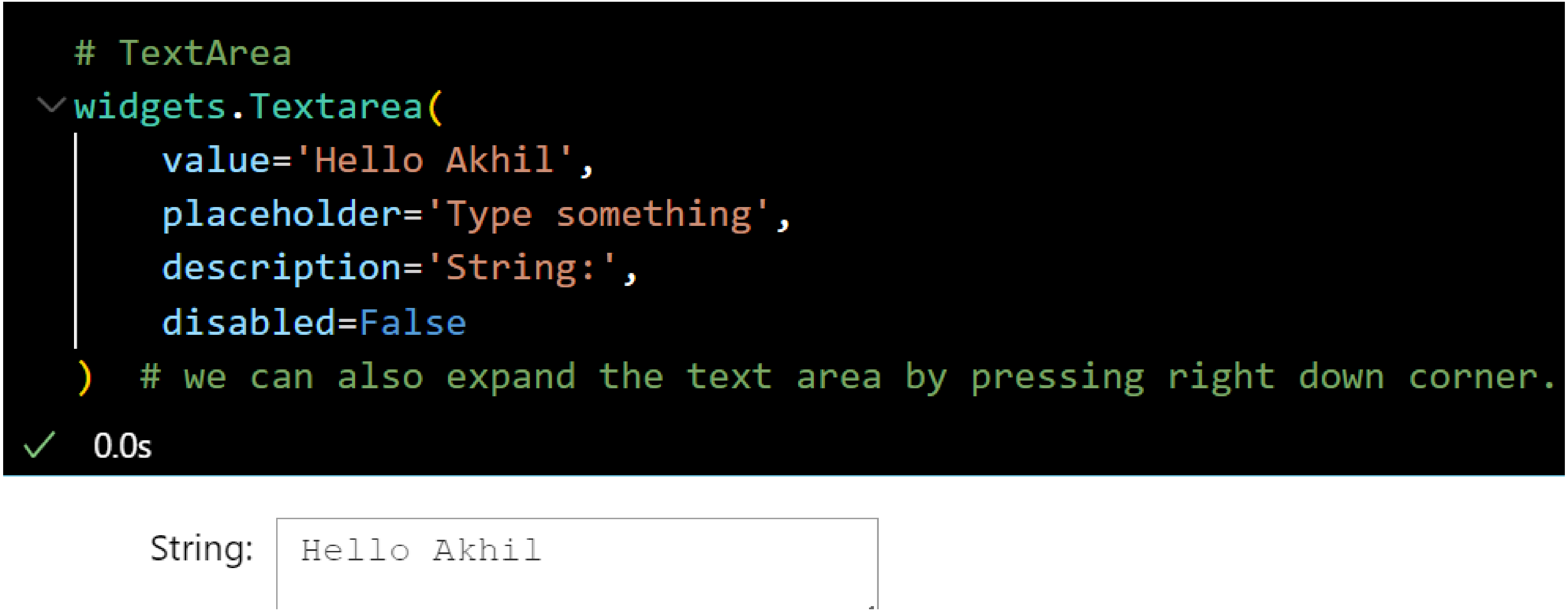
✓ 0.0s



```python
# String widgets
widgets.Text(
    value='Hello Akhil',
    placeholder='Type something', # placeholder attribute is used to provide a hint
    description='String:',
    disabled=False
)
```

0.0s

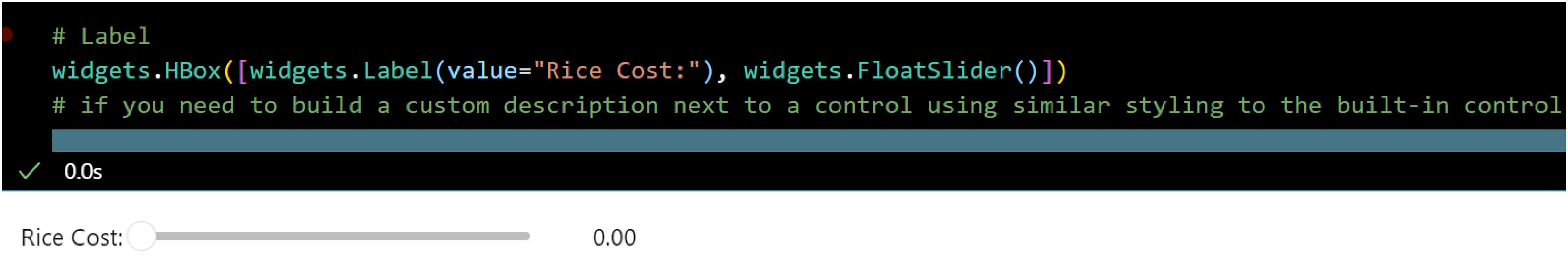
String: Hello Akhil



```
# TextArea
widgets.Textarea(
    value='Hello Akhil',
    placeholder='Type something',
    description='String:',
    disabled=False
)  # we can also expand the text area by pressing right down corner.
```

✓ 0.0s

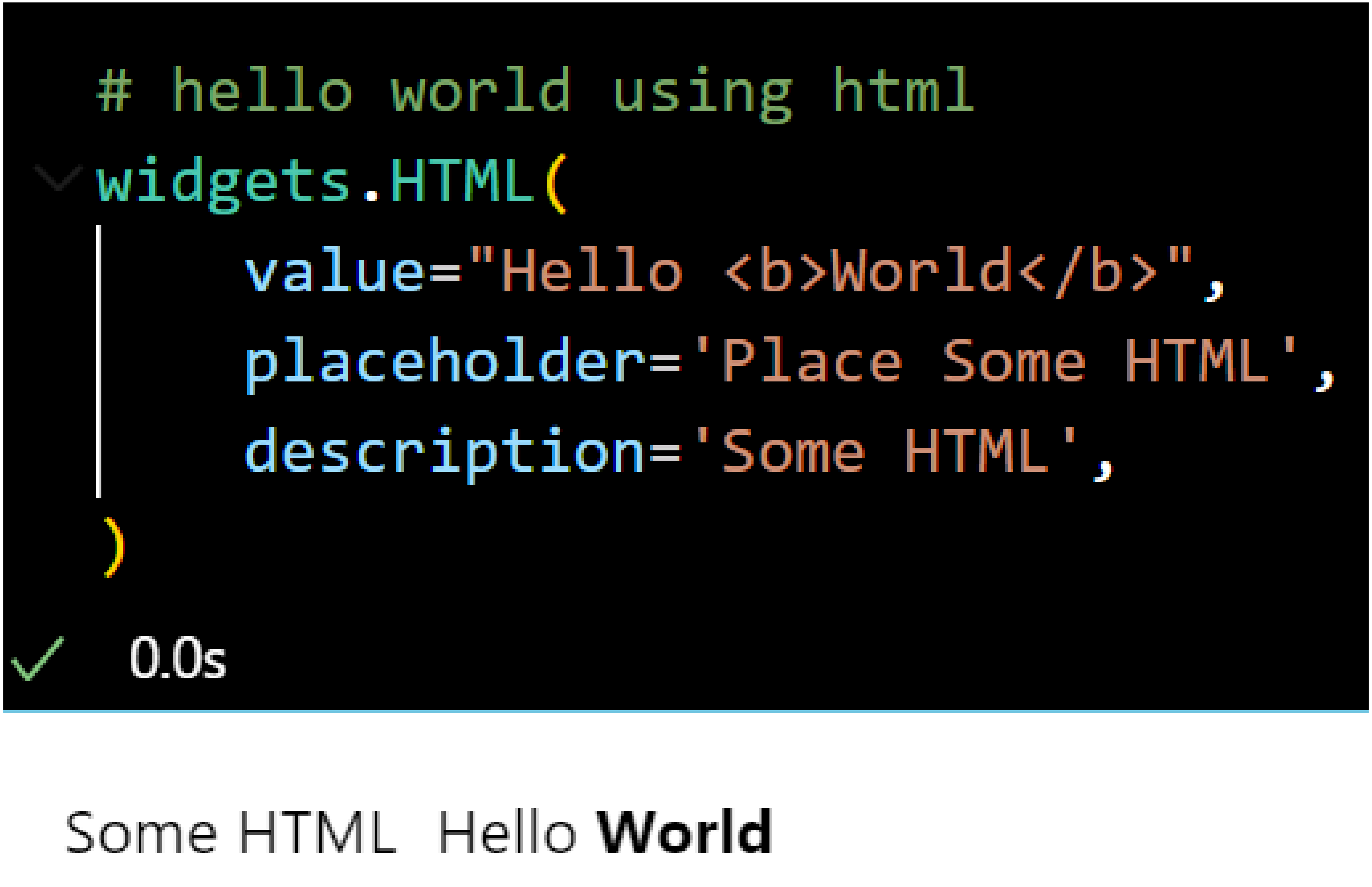
```
String: Hello Akhil
```



```python
# Label
widgets.HBox([widgets.Label(value="Rice Cost:"), widgets.FloatSlider()])
# if you need to build a custom description next to a control using similar styling to the built-in control
```

✓ 0.0s

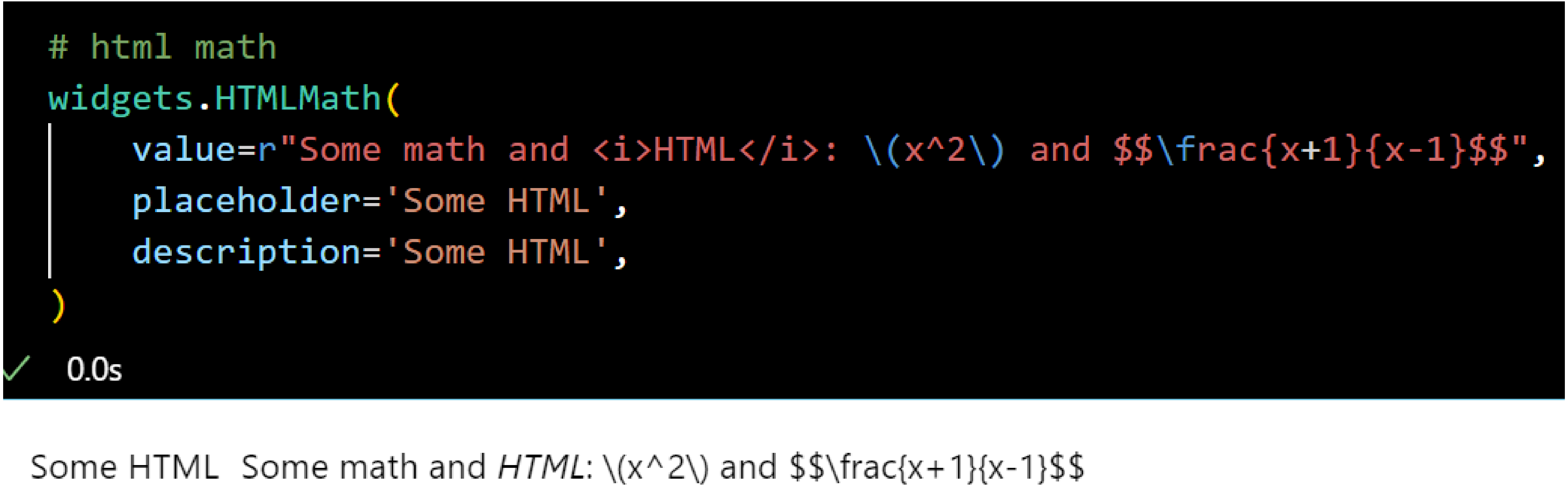
```
Rice Cost:                              0.00
```



```
# hello world using html
widgets.HTML(
        value="Hello <b>World</b>",
        placeholder='Place Some HTML',
        description='Some HTML',
)
```

✓ 0.0s

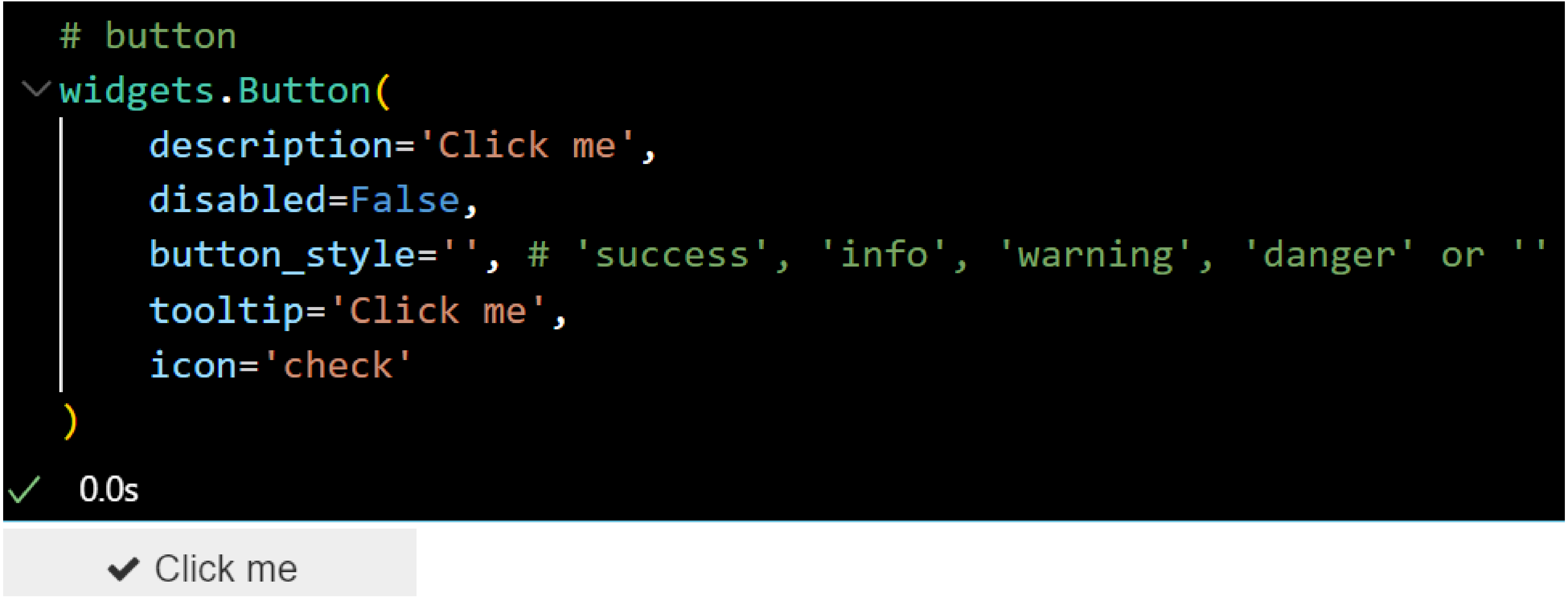
Some HTML Hello **World**



```python
# html math
widgets.HTMLMath(
    value=r"Some math and <i>HTML</i>: \(x^2\) and $$\frac{x+1}{x-1}$$",
    placeholder='Some HTML',
    description='Some HTML',
)
```

✓ 0.0s

```
Some HTML  Some math and HTML: \(x^2\) and $$\frac{x+1}{x-1}$$
```



```python
# button
widgets.Button(
    description='Click me',
    disabled=False,
    button_style='', # 'success', 'info', 'warning', 'danger' or ''
    tooltip='Click me',
    icon='check'
)
```

✓ 0.0s