```
interact(fun,x=10) # we get a slide bar
# minimum is defined by x
# max is definde by 3 X
```

✓ 0.6s


```
10
```

```
def fun(x):
    return x**2  # this takes input x and return the square.
```

✓ 0.0s

+ Code + Markdown

```
interact(fun,x=10) # above is input and below is output.
```

✓ 0.1s


```
100
```

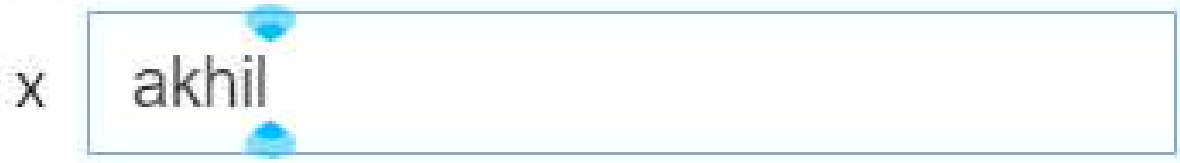
```
# lets check what  we get when we pass boolean.
interact(fun,x=True) # we get check box
```

✓ 0.0s


```
1
```

```
# for text box :
def fun(x):
    return x
interact(fun,x='akhil')
```

✓ 0.0s



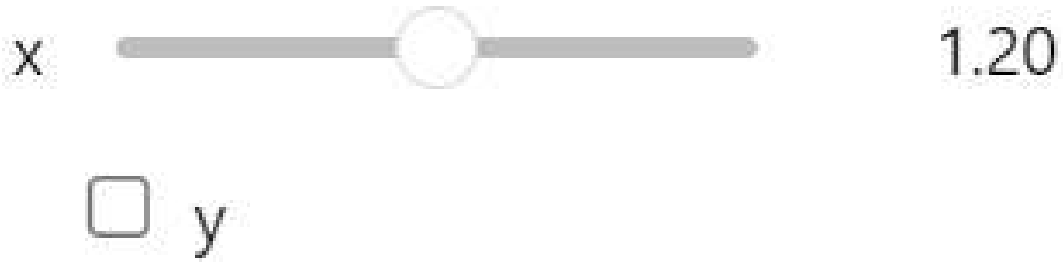

```
'akhil'
```

```
# for text box :
def fun(x):
    return x
interact(fun,x='akhil')
```


```
'akhil is python coder'
```

```python
# lets use decorator
@interact(x=1.2,y=False)
def a(x,y):
    return (x,y) # this return tuples
```

✓ 0.0s




```
(1.2, False)
```

```python
# to make value fixed
@interact(x=fixed(1.2),y=False)  # no more slider
def a(x,y):
    return (x,y) # this return tuples
```

✓ 0.0s

☐ y

```
(1.2, False)
```

```python
# to make value fixed
@interact(x=fixed(1.2),y=False)  # no more slider
def a(x,y):
    return (x,y) # this return tuples
```

✓ 0.0s

☐ y

```
(1.2, False)
```

```
# we can also fix the text box :
interact(fun,x=fixed('akhil'))  # now we cant modify this
```

✓ 0.0s

```
'akhil'
```

```
# to give cistom values to slider
interact(fun,x=widgets.IntSlider(min =-50,max=50,step=1,value=0))
# make sure IntSlider is correct casing
```

✓ 0.0s

0

```
# we can also write the above as
interact(fun,x=(-5,5,1))
```

✓ 0.0s

```
# we can also make floating point slider :
interact(fun,x=(-5.0,5.3,.1))
```


```
0.09999999999999964
```

```python
# also done using interative decorator
@interact(x=(-10,10,1)) # tuples are sliders
def h(x=5.0):
    return x
```

✓ 0.0s

5

```
#  widget abbreviation
interact(fun,x='Hello')
```

✓ 0.0s


```
'Hello'
```

```
# Drop down menu using list
interact(fun,x=['Hello','Akhil','Welcome'])
```

✓ 0.0s

```
interact(fun,x={'key1':'Value 1','key2':'Value 2'}) # in drop down we have key (above)
# value is given below
```

✓ 0.0s


```
'Value 1'
```

```
# working with interactive function :
from IPython.display import display

def f(a,b):
    display(a+b)
    return a+b
```

✓ 0.0s

```
w = interactive(f,a=10,b=30)
```

✓ 0.0s

```
# checking type
type(w)
```

✓ 0.0s

```
ywidgets.widgets.interaction.interactive
```

```
w.children
# the fist 2 are int sliders and output
```

✓ 0.0s

```
ntSlider(value=10, description='a', max=30, min=-10),
ntSlider(value=30, description='b', max=90, min=-30),
utput(outputs=({'output_type': 'display_data', 'data': {'text/plain': '40'}, 'metadata': {}},))
```

```
# to display sliders
display(w)
```

✓ 0.0s

a 10

b 30

40