```
#import all of ipywidgets with widgets as callable abreviation
import ipywidgets as widgets

#import relevant widgets
from ipywidgets import Button, Layout, jslink, IntText, IntSlider

#Function to create a button
def create_expanded_button(description, button_style):
    return Button(description=description, button_style=button_style, layout=Layout(height='auto',
width='auto'))

#Create functions for on_click handlers
def topleft(b):
    print("top left")
def topright(b):
    print("top right")
def bottomleft(b):
    print("bottom left")
def bottomright(b):
    print("bottomright")

#create buttons from your create button function

top_left_button = create_expanded_button("Top left", 'info')
top_right_button = create_expanded_button("Top right", 'success')
bottom_left_button = create_expanded_button("Bottom left", 'danger')
bottom_right_button = create_expanded_button("Bottom right", 'warning')

#creating on_click
top_left_button.on_click(topleft)
top_right_button.on_click(topright)
bottom_left_button.on_click(bottomleft)
bottom_right_button.on_click(bottomright)

from ipywidgets import TwoByTwoLayout
#Create two by two layout of buttons
TwoByTwoLayout(top_left=top_left_button,
               top_right=top_right_button,
               bottom_left=bottom_left_button,
               bottom_right=bottom_right_button)
```

```
import ipywidgets as widgets

#create a series of checkboxes
data = ["data1", "data2", "data3", "data4"]
checkboxes = [widgets.Checkbox(value=False, description=label) for label in data]
output = widgets.VBox(children=checkboxes)
display(output)

selected_data = []
for i in range(0, len(checkboxes)):
    if checkboxes[i].value == True:
        selected_data = selected_data + [checkboxes[i].description]
print(selected_data)

#Display all items from checked checkboxes
selected_data = []
for i in range(0, len(checkboxes)):
    if checkboxes[i].value == True:
        selected_data = selected_data + [checkboxes[i].description]
print(selected_data)

#Toggle Button Syntax
#widgets.ToggleButton(
    #value=False,
    #description='Click me',
    #disabled=False,
    #button_style='', # 'success', 'info', 'warning', 'danger' or ''
    #tooltip='Description',
    #icon='check' # (FontAwesome names without the `fa-` prefix)

#Create Radiobuttons
import ipywidgets as widgets
widgets.RadioButtons(
    options=['100% beef', 'veggie', 'turkey'],
#    value='100% beef', # Defaults to '100% beefe'
#    layout={'width': 'max-content'}, # If the items' names are long
    description='protein:',
    disabled=False
)
```

```
#GridspecLayout Allows you to arrange multiple widgets in a grid
from ipywidgets import GridspecLayout


#combo sizes
#All american $8.50
#Cheeseburger $7.50
#BaconBurger $7.50
#Medium +$1
#Large +$2

#create a grid and define its boundries
combogrid= GridspecLayout(3, 1)

#each widget on the grid has to be individually defined
combogrid[0,0]= widgets.ToggleButtons(options=['All American', 'Cheeseburger', 'Bacon Burger',
'Turkey_Deluxe', 'Chicken_club'], value="Cheeseburger", description='Combos', disabled=False,
button_style='warning' )
combogrid[1,0]= widgets.RadioButtons(options=['Small', 'Medium', 'Large'], description='Size',
disabled=False )
combogrid[2,0]= create_expanded_button("Add ", 'info')
#the name of each widget in a grid will be the grid name and it's location EX: combogrid[0,0] is
the first widget in a grid



#.value will return the selected option in a toggle button or radio button in you can set the default
value when defining the widget
def add(b):
    orderPrice=0
    if (combogrid[0,0].value == "All American" or combogrid[0,0].value == "Turkey_Deluxe" or
combogrid[0,0].value == "Chicken_club"):
        orderPrice+= 8.50
    if (combogrid[0,0].value == "Cheeseburger" or combogrid[0,0].value == "Bacon Burger"):
        orderPrice+= 7.50
    if (combogrid[1,0].value == "Medium"):
        orderPrice+= 1
    if (combogrid[1,0].value == "Large"):
        orderPrice+= 2
    print(combogrid[0,0].value + " " + combogrid[1,0].value + " " + str(orderPrice) )


combogrid[2,0].on_click(add)
```

#call your GUI
combogrid