

```

1 # Using the 'main' function, builds the window including all the necessary buttons
  of a GUI based program.
2 # Also creates all the necessary functions to run a functioning POS System.
3
4 # Sources:
5 """
6 - Menu items copyright American Dairy Queen.
7 - Button color names from http://www.tcl.tk/man/tcl8.6/TkCmd/colors.htm.
8 - 'import tkinter.messagebox as box' from Coding for Beginners by Mike McGrath.
9 - 'box.askyesno' from Coding for Beginners by Mike McGrath.
10 - 'for item in range(num_items)' from https://stackoverflow.com/a/12171382/6806860.
11 - 'order_items.pop(-1)' & 'order_prices.pop(-1)' from
    https://stackoverflow.com/a/18173414/6806860.
12 - 'width = window.winfo_width(), height = window.winfo_height()' from
    https://stackoverflow.com/a/4065851/6806860.
13 - 'lambda:' from https://stackoverflow.com/a/890188/6806860.
14 - 'value.get()' from https://stackoverflow.com/a/52792094/6806860 &
    https://stackoverflow.com/a/50438744/6806860.
15 - 'back.destroy()' from https://stackoverflow.com/q/21634906/6806860.
16 - 'items.configure(text = str(item_list))' & 'total.configure(text = "$ " +
    str(total_price))' from https://stackoverflow.com/a/56929450/6806860.
17 - 'pady = (20, 2)' from https://stackoverflow.com/a/4178084/6806860.
18 - '"${:0.2f}\n".format(total_price)' from
    https://stackoverflow.com/a/20457284/6806860.
19 """
20
21 # Import and Assign Tk() Variable
22 import time
23 from tkinter import *
24 import tkinter.messagebox as box
25 window = Tk()
26
27 # List to Hold ORDER ITEMS
28 order_items = []
29
30 # List to Hold PRICES
31 order_prices = []
32
33 # Discounts for Open Discount Function and Meal Deal
34 discount = 0
35 meal_deal = False
36
37 # Total Order Price
38 total_price = 0
39
40 # Total Up Price, Update Order List and Present User with Total
41 def total_up(items_list, prices):
42     # Make Global Variable Available
43     global items
44     global total
45     global meal_deal
46     global total_price
47     # Make Sure Meal Deal Can Only Be Redeemed Once
48     if not(meal_deal):
49         # Cheeseburger Meal Deal
50         if "Cheeseburger" in order_items:
51             # Ask to Save Money
52             deal = box.askyesno("2 for $5 Deal", "Get 2 Cheeseburgers for $5?")
53             # Save Money if Yes
54             if deal == 1:

```

```

55         order_items.append("Cheeseburger")
56         order_prices.append(1.5)
57         print(order_items)
58         print(order_prices)
59         meal_deal = True
60         # Don't Save Money if No
61         else:
62             meal_deal = False
63         # Chicken Wrap Meal Deal
64         elif "Chicken Wrap" in order_items:
65             # Ask to Save Money
66             deal = box.askyesno("2 for $5 Deal", "Get 2 Chicken Wraps for $5?")
67             # Save Money if Yes
68             if deal == 1:
69                 order_items.append("Chicken Wrap")
70                 order_prices.append(2)
71                 print(order_items)
72                 print(order_prices)
73                 meal_deal = True
74             # Don't Save Money if No
75             else:
76                 meal_deal = False
77         # Turn the Item List into a String
78         num_items = len(items_list)
79         item_list = ""
80         for item in range(num_items):
81             item_list += " "
82             item_list += order_items[(item)]
83             item_list += " "
84             item_list += "\n"
85         # Respond & Debug
86         items.configure(text = str(item_list))
87         print(item_list)
88         # Add Up Prices
89         total_price = sum(order_prices)
90         discount_price = (total_price*discount)
91         total_price = total_price-discount_price
92         # Respond & Debug
93         total.configure(text = str("${:0.2f}\n".format(total_price)))
94         print(total_price)
95
96     # Get Final Total to Take Payment
97     def final_total(total):
98         box.showinfo("Order Total", str("Your order total was:
99         ${:0.2f}\n".format(total_price)))
100         exit()
101
102     # Grill Items Backend
103     def grill(item, price):
104         # Add Item to Lists
105         order_items.append(item)
106         order_prices.append(price)
107         # Respond & Debug
108         total_up(order_items, order_prices)
109         print(order_items)
110         print(order_prices)
111
112     # Blizzard Items Backend
113     def blizzard(size, item, price):
114         # Add Item to Lists

```

```

114     item = str(size) + " " + str(item)
115     order_items.append(item)
116     order_prices.append(price)
117     # Respond & Debug
118     total_up(order_items, order_prices)
119     print(order_items)
120     print(order_prices)
121
122 # Chill Items Backend
123 def chill(item, price):
124     # Add Item to Lists
125     order_items.append(item)
126     order_prices.append(price)
127     # Respond & Debug
128     total_up(order_items, order_prices)
129     print(order_items)
130     print(order_prices)
131
132 # Drink Items Backend
133 def drink(size, item, price):
134     # Add Item to Lists
135     item = str(size) + " " + str(item)
136     order_items.append(item)
137     order_prices.append(price)
138     # Respond & Debug
139     total_up(order_items, order_prices)
140     print(order_items)
141     print(order_prices)
142
143 # Beverage Items Backend
144 def beverage(item, price):
145     # Add Item to Lists
146     order_items.append(item)
147     order_prices.append(price)
148     # Respond & Debug
149     total_up(order_items, order_prices)
150     print(order_items)
151     print(order_prices)
152
153 # Cake Items Backend
154 def cake(item, price):
155     # Add Item to Lists
156     order_items.append(item)
157     order_prices.append(price)
158     # Respond & Debug
159     total_up(order_items, order_prices)
160     print(order_items)
161     print(order_prices)
162
163 # Delete Item in Order List
164 def delete_item():
165     # Get Length of Lists
166     num_items = len(order_items)
167     num_prices = len(order_prices)
168     # Check to Make Sure There are Actually Items to Delete
169     if num_items and num_prices >= 1:
170         # Remove Last Item & Price
171         order_items.pop(-1)
172         order_prices.pop(-1)
173     else:

```

```

174         print("Cannot remove from list.")
175     # Respond & Debug
176     total_up(order_items, order_prices)
177     print(order_items)
178     print(order_prices)
179
180 # Backend for Open Food or Open Discount Button
181 def open(type, amount):
182     # Make Global Variables Available
183     global back
184     global title
185     global value
186     global enter
187     global price
188     # Command for Open Food
189     if type == "food":
190         order_items.append("Open Food ($" + str(amount) + ")")
191         order_prices.append(float(amount))
192     # Command for Open Discount
193     elif type == "disc":
194         global discount
195         discount = (float(amount)/100)
196         print("Discount: " + str(discount))
197     # Respond & Debug
198     total_up(order_items, order_prices)
199     print(order_items)
200     print(order_prices)
201     # Remove Entry Prompt
202     back.destroy()
203     title.destroy()
204     value.destroy()
205     enter.destroy()
206
207 # Open Food Button Backend
208 def open_food_ask():
209     # Make Global Variables Available
210     global back
211     global title
212     global value
213     global enter
214     global price
215     # Ask for the Value to Add
216     price = StringVar()
217     back = Canvas(window, width = window.winfo_width(), height =
window.winfo_height())
218     title = Label(window, text='Enter Value to Add:')
219     value = Entry(window, textvariable = price)
220     enter = Button(window, text = 'Accept', command = lambda: open("food",
value.get()))
221     # Position on Window
222     back.grid(row = 1, rowspan = 8, column = 1, columnspan = 5)
223     title.grid(row = 3, column = 3)
224     value.grid(row = 4, column = 3)
225     enter.grid(row = 5, column = 3)
226
227 # Open Discount Button Backend
228 def open_disc_ask():
229     # Make Global Variables Available
230     global back
231     global title

```

```

232     global value
233     global enter
234     global price
235     # Ask for the Discount Value
236     price = StringVar()
237     back = Canvas(window, width = window.winfo_width(), height =
window.winfo_height())
238     title = Label(window, text='Enter Price to Discount:')
239     value = Entry(window, textvariable = price)
240     enter = Button(window, text = 'Accept', command = lambda: open("disc",
value.get()))
241     # Position on Window
242     back.grid(row = 1, rowspan = 8, column = 1, columnspan = 5)
243     title.grid(row = 3, column = 3)
244     value.grid(row = 4, column = 3)
245     enter.grid(row = 5, column = 3)
246
247 # Builds the window, including buttons and order item list.
248 def main():
249     # Make Global Variables Available
250     global items
251     global total
252     # Window Heading
253     window.title("POS GUI")
254     # Title Information
255     version = Label(window, text = "GUI POS CREATE TASK - Version 1.0.0")
256     print("GUI POS CREATE TASK - Version 1.0.0\n")
257     timer = Label(window, text = (time.strftime("%H:%M:%S")))
258     # GRILL Buttons
259     grill1 = Button(window, text = "Bacon Cheese Burger", bg = 'orange', command =
lambda: grill("Bacon Cheese Burger", 9))
260     grill2 = Button(window, text = "Ultimate Burger", bg = 'orange', command =
lambda: grill("Ultimate Burger", 6))
261     grill3 = Button(window, text = "Cheeseburger", bg = 'orange', command = lambda:
grill("Cheeseburger", 3.5))
262     grill4 = Button(window, text = "Hamburger", bg = 'orange', command = lambda:
grill("Hamburger", 3))
263     grill5 = Button(window, text = "Crispy Chicken", bg = 'orange', command =
lambda: grill("Crispy Chicken", 7))
264     grill6 = Button(window, text = "Chicken Wrap", bg = 'orange', command = lambda:
grill("Chicken Wrap", 3))
265     # CHILL Buttons
266     chill1 = Button(window, text = "Mini Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Mini", "Blizzard", 4.5))
267     chill2 = Button(window, text = "Small Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Small", "Blizzard", 6))
268     chill3 = Button(window, text = "Medium Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Medium", "Blizzard", 8))
269     chill4 = Button(window, text = "Large Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Large", "Blizzard", 9))
270     chill5 = Button(window, text = "PB Parfait", bg = 'dodgerblue', command =
lambda: chill("PB Parfait", 6))
271     chill6 = Button(window, text = "Banana Split", bg = 'dodgerblue', command =
lambda: chill("Banana Split", 6))
272     # BEVERAGE Buttons
273     bev1 = Button(window, text = "Small Drink", bg = 'green', command = lambda:
drink("Small", "Pop", 2))
274     bev2 = Button(window, text = "Medium Drink", bg = 'green', command = lambda:
drink("Medium", "Pop", 3.5))

```

```

275     bev3 = Button(window, text = "Large Drink", bg = 'green', command = lambda:
drink("Large", "Pop", 5))
276     bev4 = Button(window, text = "Fruit Smoothie", bg = 'green', command = lambda:
beverage("Fruit Smoothie", 4))
277     bev5 = Button(window, text = "Orange Julius", bg = 'green', command = lambda:
beverage("Orange Julius", 4))
278     bev6 = Button(window, text = "Banana", bg = 'green', command = lambda:
beverage("Banana", 2))
279     # CAKE Buttons
280     cake1 = Button(window, text = "8\" Blizzard", bg = 'white', command = lambda:
cake("8\" Blizzard", 33))
281     cake2 = Button(window, text = "10\" Blizzard", bg = 'white', command = lambda:
cake("10\" Blizzard", 38))
282     cake3 = Button(window, text = "8\" Cake", bg = 'white', command = lambda:
cake("8\" Cake", 23))
283     cake4 = Button(window, text = "10\" Cake", bg = 'white', command = lambda:
cake("10\" Cake", 28))
284     cake5 = Button(window, text = "12pk Dilly Bars", bg = 'white', command = lambda:
cake("12pk Dilly Bars", 19))
285     cake6 = Button(window, text = "12pk Sandwiches", bg = 'white', command = lambda:
cake("12pk Sandwiches", 19))
286     # Order Items
287     order = Frame(window)
288     items = Label(order, text = " Order Items Here ", borderwidth = 2, relief =
"groove")
289     total = Label(order, text = "$" + "0.00")
290     # Special Buttons
291     delete = Button(window, text = "Delete Last", command = delete_item)
292     edit = Button(window, text = "Edit Last", command = delete_item)
293     open_food = Button(window, text = "Open Price", command = open_food_ask)
294     open_disc = Button(window, text = "Open Discount", command = open_disc_ask)
295     payment = Button(window, text = "Take Payment", command = lambda:
final_total(total_price))
296     # Position on Window
297     version.grid(row = 1, column = 1, columnspan = 4, padx = 10)
298     timer.grid(row = 1, column = 5, columnspan = 2, padx = 10)
299     grill1.grid(row = 2, column = 1, padx = 10, pady = 5)
300     grill2.grid(row = 3, column = 1, padx = 10, pady = 5)
301     grill3.grid(row = 4, column = 1, padx = 10, pady = 5)
302     grill4.grid(row = 5, column = 1, padx = 10, pady = 5)
303     grill5.grid(row = 6, column = 1, padx = 10, pady = 5)
304     grill6.grid(row = 7, column = 1, padx = 10, pady = 5)
305     chill1.grid(row = 2, column = 2, padx = 10, pady = 5)
306     chill2.grid(row = 3, column = 2, padx = 10, pady = 5)
307     chill3.grid(row = 4, column = 2, padx = 10, pady = 5)
308     chill4.grid(row = 5, column = 2, padx = 10, pady = 5)
309     chill5.grid(row = 6, column = 2, padx = 10, pady = 5)
310     chill6.grid(row = 7, column = 2, padx = 10, pady = 5)
311     bev1.grid(row = 2, column = 3, padx = 10, pady = 5)
312     bev2.grid(row = 3, column = 3, padx = 10, pady = 5)
313     bev3.grid(row = 4, column = 3, padx = 10, pady = 5)
314     bev4.grid(row = 5, column = 3, padx = 10, pady = 5)
315     bev5.grid(row = 6, column = 3, padx = 10, pady = 5)
316     bev6.grid(row = 7, column = 3, padx = 10, pady = 5)
317     cake1.grid(row = 2, column = 4, padx = 10, pady = 5)
318     cake2.grid(row = 3, column = 4, padx = 10, pady = 5)
319     cake3.grid(row = 4, column = 4, padx = 10, pady = 5)
320     cake4.grid(row = 5, column = 4, padx = 10, pady = 5)
321     cake5.grid(row = 6, column = 4, padx = 10, pady = 5)
322     cake6.grid(row = 7, column = 4, padx = 10, pady = 5)

```

```
323     order.grid(row = 2, rowspan = 6, column = 5, colspan = 2, padx = 10)
324     items.pack(side = TOP)
325     total.pack(side = BOTTOM)
326     delete.grid(row = 8, column = 1, padx = 10, pady = (20, 2))
327     edit.grid(row = 8, column = 2, padx = 10, pady = (20, 2))
328     open_food.grid(row = 8, column = 3, padx = 10, pady = (20, 2))
329     open_disc.grid(row = 8, column = 4, padx = 10, pady = (20, 2))
330     payment.grid(row = 8, column = 5, padx = 10, pady = (20, 2))
331     # Sustain Window
332     window.mainloop()
333
334 # Build Window
335 main()
336
```