```
# Using the 'main' function, builds the window including all the necessary buttons of a GUI based program.
# Also creates all the necessary functions to run a functioning POS System.

# Sources:
"""
- Menu items copyright American Dairy Queen.
- Button color names from http://www.tcl.tk/man/tcl8.6/TkCmd/colors.htm.
- 'import tkinter.messagebox as box' from Coding for Beginners by Mike McGrath.
- 'box.askyesno' from Coding for Beginners by Mike McGrath.
- 'for item in range(num_items)' from https://stackoverflow.com/a/12171382/6806860.
- 'order_items.pop(-1)' & 'order_prices.pop(-1)' from https://stackoverflow.com/a/18173414/6806860.
- 'width = window.winfo_width(), height = window.winfo_height()' from https://stackoverflow.com/a/4065851/6806860.
- 'lambda:' from https://stackoverflow.com/a/890188/6806860.
- 'value.get())' from https://stackoverflow.com/a/52792094/6806860 & https://stackoverflow.com/a/50438744/6806860.
- 'back.destroy()' from https://stackoverflow.com/q/21634906/6806860.
- 'items.configure(text = str(item_list))' & 'total.configure(text = "$ " + str(total_price))' from https://stackoverflow.com/a/56929450/6806860.
- 'pady = (20, 2)' from https://stackoverflow.com/a/4178084/6806860.
- '"${:0.2f}\n".format(total_price)' from https://stackoverflow.com/a/20457284/6806860.
"""

# Import and Assign Tk() Variable
import time
from tkinter import *
import tkinter.messagebox as box
window = Tk()

# List to Hold ORDER ITEMS
order_items = []

# List to Hold PRICES
order_prices = []

# Discounts for Open Discount Function and Meal Deal
discount = 0
meal_deal = False

# Total Order Price
total_price = 0

# Total Up Price, Update Order List and Present User with Total
def total_up(items_list, prices):
    # Make Global Variable Available
    global items
    global total
    global meal_deal
    global total_price
    # Make Sure Meal Deal Can Only Be Redeemed Once
    if not(meal_deal):
        # Cheeseburger Meal Deal
        if "Cheeseburger" in order_items:
            # Ask to Save Money
            deal = box.askyesno("2 for $5 Deal", "Get 2 Cheeseburgers for $5?")
            # Save Money if Yes
            if deal == 1:
```

```
                order_items.append("Cheeseburger")
                order_prices.append(1.5)
                print(order_items)
                print(order_prices)
                meal_deal = True
            # Don't Save Money if No
            else:
                meal_deal = False
        # Chicken Wrap Meal Deal
        elif "Chicken Wrap" in order_items:
            # Ask to Save Money
            deal = box.askyesno("2 for $5 Deal", "Get 2 Chicken Wraps for $5?")
            # Save Money if Yes
            if deal == 1:
                order_items.append("Chicken Wrap")
                order_prices.append(2)
                print(order_items)
                print(order_prices)
                meal_deal = True
            # Don't Save Money if No
            else:
                meal_deal = False
    # Turn the Item List into a String
    num_items = len(items_list)
    item_list = ""
    for item in range(num_items):
        item_list += " "
        item_list += order_items[(item)]
        item_list += " "
        item_list += "\n"
    # Respond & Debug
    items.configure(text = str(item_list))
    print(item_list)
    # Add Up Prices
    total_price = sum(order_prices)
    discount_price = (total_price*discount)
    total_price = total_price-discount_price
    # Respond & Debug
    total.configure(text = str("${:0.2f}\n".format(total_price)))
    print(total_price)

# Get Final Total to Take Payment
def final_total(total):
    box.showinfo("Order Total", str("Your order total was:
${:0.2f}\n".format(total_price)))
    exit()

# Grill Items Backend
def grill(item, price):
    # Add Item to Lists
    order_items.append(item)
    order_prices.append(price)
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Blizzard Items Backend
def blizzard(size, item, price):
    # Add Item to Lists
```

```
    item = str(size) + " " + str(item)
    order_items.append(item)
    order_prices.append(price)
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Chill Items Backend
def chill(item, price):
    # Add Item to Lists
    order_items.append(item)
    order_prices.append(price)
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Drink Items Backend
def drink(size, item, price):
    # Add Item to Lists
    item = str(size) + " " + str(item)
    order_items.append(item)
    order_prices.append(price)
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Beverage Items Backend
def beverage(item, price):
    # Add Item to Lists
    order_items.append(item)
    order_prices.append(price)
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Cake Items Backend
def cake(item, price):
    # Add Item to Lists
    order_items.append(item)
    order_prices.append(price)
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Delete Item in Order List
def delete_item():
    # Get Length of Lists
    num_items = len(order_items)
    num_prices = len(order_prices)
    # Check to Make Sure There are Actually Items to Delete
    if num_items and num_prices >= 1:
        # Remove Last Item & Price
        order_items.pop(-1)
        order_prices.pop(-1)
    else:
```

```
        print("Cannot remove from list.")
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)

# Backend for Open Food or Open Discount Button
def open(type, amount):
    # Make Global Variables Available
    global back
    global title
    global value
    global enter
    global price
    # Command for Open Food
    if type == "food":
        order_items.append("Open Food ($" + str(amount) + ")")
        order_prices.append(float(amount))
    # Command for Open Discount
    elif type == "disc":
        global discount
        discount = (float(amount)/100)
        print("Discount: " + str(discount))
    # Respond & Debug
    total_up(order_items, order_prices)
    print(order_items)
    print(order_prices)
    # Remove Entry Prompt
    back.destroy()
    title.destroy()
    value.destroy()
    enter.destroy()

# Open Food Button Backend
def open_food_ask():
    # Make Global Variables Available
    global back
    global title
    global value
    global enter
    global price
    # Ask for the Value to Add
    price = StringVar()
    back = Canvas(window, width = window.winfo_width(), height =
window.winfo_height())
    title = Label(window, text='Enter Value to Add:')
    value = Entry(window, textvariable = price)
    enter = Button(window, text = 'Accept', command = lambda: open("food",
value.get()))
    # Position on Window
    back.grid(row = 1, rowspan = 8, column = 1, columnspan = 5)
    title.grid(row = 3, column = 3)
    value.grid(row = 4, column = 3)
    enter.grid(row = 5, column = 3)

# Open Discount Button Backend
def open_disc_ask():
    # Make Global Variables Available
    global back
    global title
```

```
    global value
    global enter
    global price
    # Ask for the Discount Value
    price = StringVar()
    back = Canvas(window, width = window.winfo_width(), height =
window.winfo_height())
    title = Label(window, text='Enter Price to Discount:')
    value = Entry(window, textvariable = price)
    enter = Button(window, text = 'Accept', command = lambda: open("disc",
value.get()))
    # Position on Window
    back.grid(row = 1, rowspan = 8, column = 1, columnspan = 5)
    title.grid(row = 3, column = 3)
    value.grid(row = 4, column = 3)
    enter.grid(row = 5, column = 3)

# Builds the window, including buttons and order item list.
def main():
    # Make Global Variables Available
    global items
    global total
    # Window Heading
    window.title("POS GUI")
    # Title Information
    version = Label(window, text = "GUI POS CREATE TASK - Version 1.0.0")
    print("GUI POS CREATE TASK - Version 1.0.0\n")
    timer = Label(window, text = (time.strftime("%H:%M:%S")))
    # GRILL Buttons
    grill1 = Button(window, text = "Bacon Cheese Burger", bg = 'orange', command =
lambda: grill("Bacon Cheese Burger", 9))
    grill2 = Button(window, text = "Ultimate Burger", bg = 'orange', command =
lambda: grill("Ultimate Burger", 6))
    grill3 = Button(window, text = "Cheeseburger", bg = 'orange', command = lambda:
grill("Cheeseburger", 3.5))
    grill4 = Button(window, text = "Hamburger", bg = 'orange', command = lambda:
grill("Hamburger", 3))
    grill5 = Button(window, text = "Crispy Chicken", bg = 'orange', command =
lambda: grill("Crispy Chicken", 7))
    grill6 = Button(window, text = "Chicken Wrap", bg = 'orange', command = lambda:
grill("Chicken Wrap", 3))
    # CHILL Buttons
    chill1 = Button(window, text = "Mini Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Mini", "Blizzard", 4.5))
    chill2 = Button(window, text = "Small Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Small", "Blizzard", 6))
    chill3 = Button(window, text = "Medium Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Medium", "Blizzard", 8))
    chill4 = Button(window, text = "Large Blizzard", bg = 'dodgerblue', command =
lambda: blizzard("Large", "Blizzard", 9))
    chill5 = Button(window, text = "PB Parfait", bg = 'dodgerblue', command =
lambda: chill("PB Parfait", 6))
    chill6 = Button(window, text = "Banana Split", bg = 'dodgerblue', command =
lambda: chill("Banana Split", 6))
    # BEVERAGE Buttons
    bev1 = Button(window, text = "Small Drink", bg = 'green', command = lambda:
drink("Small", "Pop", 2))
    bev2 = Button(window, text = "Medium Drink", bg = 'green', command = lambda:
drink("Medium", "Pop", 3.5))
```

```
    bev3 = Button(window, text = "Large Drink", bg = 'green', command = lambda:
drink("Large", "Pop", 5))
    bev4 = Button(window, text = "Fruit Smoothie", bg = 'green', command = lambda:
beverage("Fruit Smoothie", 4))
    bev5 = Button(window, text = "Orange Julius", bg = 'green', command = lambda:
beverage("Orange Julius", 4))
    bev6 = Button(window, text = "Banana", bg = 'green', command = lambda:
beverage("Banana", 2))
    # CAKE Buttons
    cake1 = Button(window, text = "8\" Blizzard", bg = 'white', command = lambda:
cake("8\" Blizzard", 33))
    cake2 = Button(window, text = "10\" Blizzard", bg = 'white', command = lambda:
cake("10\" Blizzard", 38))
    cake3 = Button(window, text = "8\" Cake", bg = 'white', command = lambda:
cake("8\" Cake", 23))
    cake4 = Button(window, text = "10\" Cake", bg = 'white', command = lambda:
cake("10\" Cake", 28))
    cake5 = Button(window, text = "12pk Dilly Bars", bg = 'white', command = lambda:
cake("12pk Dilly Bars", 19))
    cake6 = Button(window, text = "12pk Sandwiches", bg = 'white', command = lambda:
cake("12pk Sandwiches", 19))
    # Order Items
    order = Frame(window)
    items = Label(order, text = " Order Items Here ", borderwidth = 2, relief =
"groove")
    total = Label(order, text = "$" + "0.00")
    # Special Buttons
    delete = Button(window, text = "Delete Last", command = delete_item)
    edit = Button(window, text = "Edit Last", command = delete_item)
    open_food = Button(window, text = "Open Price", command = open_food_ask)
    open_disc = Button(window, text = "Open Discount", command = open_disc_ask)
    payment = Button(window, text = "Take Payment", command = lambda:
final_total(total_price))
    # Position on Window
    version.grid(row = 1, column = 1, columnspan = 4, padx = 10)
    timer.grid(row = 1, column = 5, columnspan = 2, padx = 10)
    grill1.grid(row = 2, column = 1, padx = 10, pady = 5)
    grill2.grid(row = 3, column = 1, padx = 10, pady = 5)
    grill3.grid(row = 4, column = 1, padx = 10, pady = 5)
    grill4.grid(row = 5, column = 1, padx = 10, pady = 5)
    grill5.grid(row = 6, column = 1, padx = 10, pady = 5)
    grill6.grid(row = 7, column = 1, padx = 10, pady = 5)
    chill1.grid(row = 2, column = 2, padx = 10, pady = 5)
    chill2.grid(row = 3, column = 2, padx = 10, pady = 5)
    chill3.grid(row = 4, column = 2, padx = 10, pady = 5)
    chill4.grid(row = 5, column = 2, padx = 10, pady = 5)
    chill5.grid(row = 6, column = 2, padx = 10, pady = 5)
    chill6.grid(row = 7, column = 2, padx = 10, pady = 5)
    bev1.grid(row = 2, column = 3, padx = 10, pady = 5)
    bev2.grid(row = 3, column = 3, padx = 10, pady = 5)
    bev3.grid(row = 4, column = 3, padx = 10, pady = 5)
    bev4.grid(row = 5, column = 3, padx = 10, pady = 5)
    bev5.grid(row = 6, column = 3, padx = 10, pady = 5)
    bev6.grid(row = 7, column = 3, padx = 10, pady = 5)
    cake1.grid(row = 2, column = 4, padx = 10, pady = 5)
    cake2.grid(row = 3, column = 4, padx = 10, pady = 5)
    cake3.grid(row = 4, column = 4, padx = 10, pady = 5)
    cake4.grid(row = 5, column = 4, padx = 10, pady = 5)
    cake5.grid(row = 6, column = 4, padx = 10, pady = 5)
    cake6.grid(row = 7, column = 4, padx = 10, pady = 5)
```

```
    order.grid(row = 2, rowspan = 6, column = 5, columnspan = 2, padx = 10)
    items.pack(side = TOP)
    total.pack(side = BOTTOM)
    delete.grid(row = 8, column = 1, padx = 10, pady = (20, 2))
    edit.grid(row = 8, column = 2, padx = 10, pady = (20, 2))
    open_food.grid(row = 8, column = 3, padx = 10, pady = (20, 2))
    open_disc.grid(row = 8, column = 4, padx = 10, pady = (20, 2))
    payment.grid(row = 8, column = 5, padx = 10, pady = (20, 2))
    # Sustain Window
    window.mainloop()

# Build Window
main()

```