

# Grocery List Generator

Code in Place 2021 Final Project

Shelby Bachman

# The grocery list generator

This program creates a formatted, organized grocery list from user input.

It has a number of features, including:

- the use of strings, lists, dictionaries & control flow
- the reading of files & drawing of images
- creation of a grocery list in multiple formats
- multiple customization options

Let's see how it works!

# Running the program

Start the program via the command line: `python grocery_list.py`

A welcome message appears:

```
$ python grocery_list.py
-----
Welcome to the grocery list creator!
-----
A few reminders:
- Use singular nouns only
- Enter items separately from quantities
-----
Enter the item to buy: █
```

# Entering list items

When prompted, enter each item you wish to buy, one at a time

- A current limitation of the program is that the user must enter a **singular nouns**
- Behind the scenes, the program is trying to identify the item in its database
- The program will prompt to correct the entry if the item is not found

After each item, the program will prompts you for a quantity

- Quantity must be convertable to an integer, otherwise the program will prompt for another quantity

```
$ python grocery_list.py
-----
Welcome to the grocery list creator!
-----
A few reminders:
- Use singular nouns only
- Enter items separately from quantities
-----
Enter the item to buy: banana
Enter the integer quantity of banana: 5
-----
Enter the item to buy: kiwi
Enter the integer quantity of kiwi: 3
-----
Enter the item to buy: lemon
Enter the integer quantity of lemon: 4
-----
Enter the item to buy: lime
Enter the integer quantity of lime: 2
-----
Enter the item to buy: milk
Enter the integer quantity of milk: 1
-----
Enter the item to buy: egg
Enter the integer quantity of egg: 1█
```

# The grocery list

Once you enter all list item, the program displays a grocery list

- The list is organized by category
- Contains all inputted items and quantities

The list is displayed in the console

- Can be easily copied to list management software of choice

```
grocery list 2021-05-26

baking
[] baking powder x1
[] baking soda x1
[] chocolate chip x1
[] gluten-free flour x1
[] sugar x1

canned_foods
[] caper x1
[] chickpea x1
[] olive x1

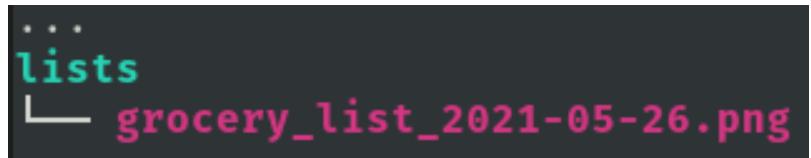
condiments
[] mayonnaise x1
[] mustard x1

dairy
[] egg x1
[] milk x1
[] yogurt x1
```

# The grocery list

The list is also saved as a .png image

- Dimensions scaled for a smartphone
- Current date included
- Can be emailed or shared to the cloud
- Saved to the `lists/` subdirectory in the same folder as the program



grocery list	
2021-05-26	
<b>baking</b>	<b>vegetables</b>
baking powder x1	eggplant x2
baking soda x1	lettuce x2
chocolate chip x1	mushroom x2
gluten-free flour x1	potato x5
sugar x1	spinach x1
	tomato x3
	zucchini x3
<b>canned_foods</b>	
caper x1	
chickpea x1	
olive x1	
<b>condiments</b>	
mayonnaise x1	
mustard x1	
<b>dairy</b>	
egg x1	
milk x1	
yogurt x1	

# How it works

User inputs are checked against a dictionary of food items, created from .csv files

- At the start of the program, these .csv files are read into a single dictionary, with keys as categories and values as lists of foods

The food dictionary is used to organize inputs according to category

The .png image is created by "drawing" text on a blank image

data
baking.csv
beverages.csv
canned_foods.csv
condiments.csv
dairy.csv
fruits.csv
grains.csv
meat.csv
snacks.csv
spices.csv
vegetables.csv

# Customization

Food items in the database can be added to relevant .csv files in the data/ subdirectory

List image dimensions can be customized by updating two lines in the program code:

```
# image dimensions  
# (currently set for iphone 6)  
IMAGE_WIDTH = 750  
IMAGE_HEIGHT = 1334
```

Font family used in the list image can be customized via files in the utils/ subdirectory:

```
# files containing fonts  
# (light font will be used for date & items)  
# (bold font will be used for title & section headings)  
FONTPATH_LIGHT = 'utils/Roboto-Light.ttf'  
FONTPATH_BOLD = 'utils/Roboto-Bold.ttf'
```