Project 3:

Creating a Fanatasy Shop Simulator with an Inventory System

Goal:

Create a fantasy shop simulator where:

- The player can buy items from a shop, add them to their inventory, and manage their gold.
- The game loop is controlled by a **game_run** boolean to allow the player to continue playing until they decide to exit.

Game Requirements:

- The game should start with a welcome message and a brief introduction.
- The player should begin with an empty inventory and 100 gold.
- The player can view the shop to see available items and their prices.
- The player can buy items if they have enough gold, and the items will be added to their inventory.
- The game should keep track of the player's inventory and remaining gold.
- The player can choose to exit the shop, at which point the game will display their inventory and remaining gold.

Instructions:

Create a New Python File:

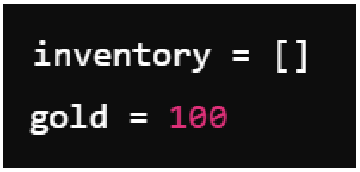
- In your "week 3" folder, create a new folder named "project3".
- Open your "project3" folder and create a Python file named "shop_sim.py".

Add an Intro Message:

- Let's make our game a bit more fun by starting with a welcome message for the player.
- Use the print( ) function to create a message at the top of your file.

Set Up the Player's Inventory and Gold:

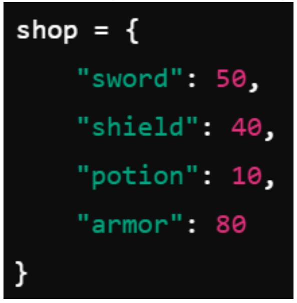
- Create variables to store the player’s inventory (an empty list or dictionary) and starting gold (100).
  Example:



```
inventory = []
gold = 100
```

Create the Shop Using a Dictionary:

- Create a dictionary for the shop with more items and their prices.
  Example:

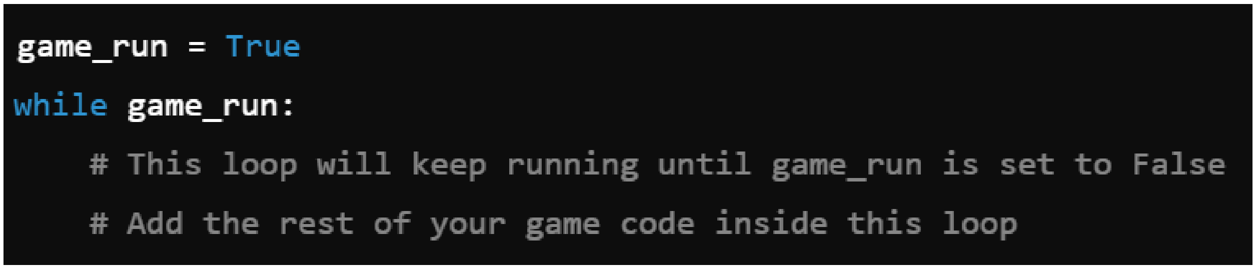


```
shop = {
    "sword": 50,
    "shield": 40,
    "potion": 10,
    "armor": 80
}
```

Create the Game Loop:

Build the Loop with a game_run Boolean:

- Create a game_run boolean variable to control whether the game continues running.
- Use this boolean to keep the game loop active until the player decides to exit the shop.

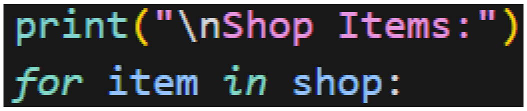


```
game_run = True
while game_run:
    # This loop will keep running until game_run is set to False
    # Add the rest of your game code inside this loop
```

Print Shop Items:

- Inside the loop, we want to print all the items in the shop so that the player can see what's available to buy.

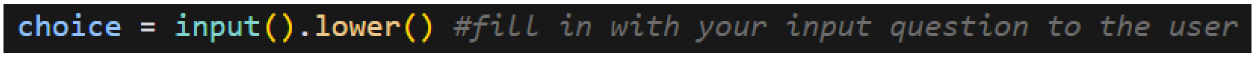
- Use a for loop to iterate through the shop dictionary and print each item with its price.



```python
print("\nShop Items:")
for item in shop:
```

Ask for Player Input:

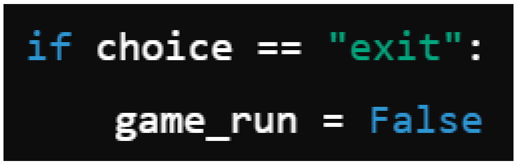
- After printing the shop items, prompt the player to enter the name of the item they want to buy or type "exit" if they want to leave the shop.
- Use the .lower() function to convert the user input to a lowercase string.



```python
choice = input().lower() #fill in with your input question to the user
```

Check If Player Wants to Exit Game:

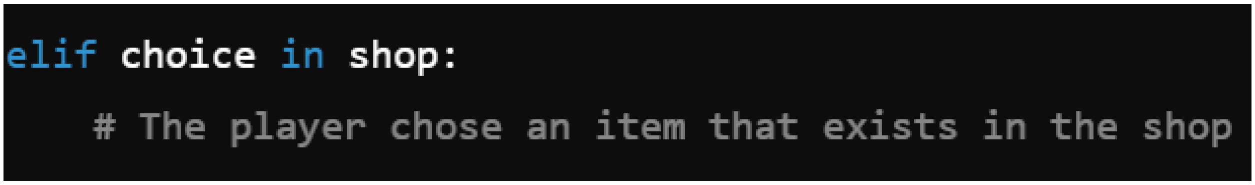
- If the player types "exit", we need to set game_run to False to stop the loop and exit the game.



```python
if choice == "exit":
    game_run = False
```

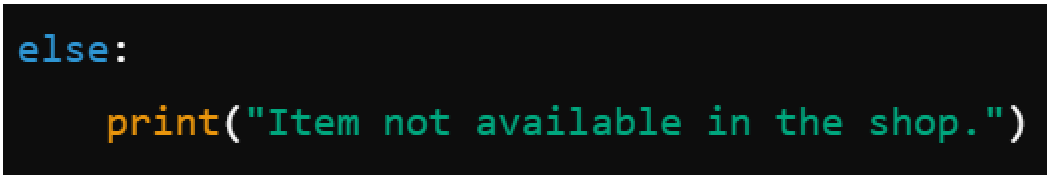
Check If Item is in the Shop:

- If the player doesn't type "exit", we need to check whether the item they want to buy is available in the shop.



```python
elif choice in shop:
    # The player chose an item that exists in the shop
```

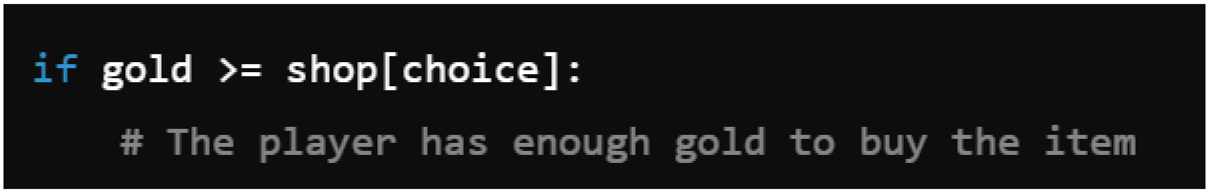
- If the Item doesn't exist in the shop, display a message telling them the item is not available.



```python
else:
    print("Item not available in the shop.")
```

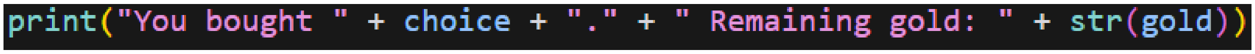
Check If Player Has Enough Gold to Buy:

- Before the player can buy the item, we need to check if they have enough gold to pay for it.



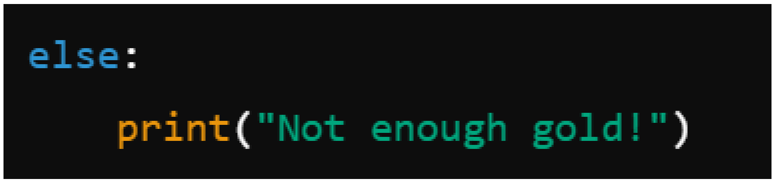
```
if gold >= shop[choice]:
    # The player has enough gold to buy the item
```

- If they do subtract the price of the item from the gold variable and add the item to the inventory variable using the .append() function.
- Then print a message telling them what they bought and how much gold they have left.



```
print("You bought " + choice + "." + " Remaining gold: " + str(gold))
```

- If they don't have enough gold just print a message saying they can't afford it.



```
else:
    print("Not enough gold!")
```

Run Your Game:

- Save your file and run it from the terminal. Remember to check if you're in the right folder, you should be in your project3 folder.
- Run your game by typing "python shop_sim.py".
- Test your game by buying different items from the shop and making sure the inventory and gold are being updated.

Bonus:

- **Sell Items Back to the Shop:** Allow players to sell items back to the shop for half their original price.
- **View Inventory:** Add a feature that lets players view their inventory at any time.