

# 12 Prepare: Preparation Material

## Overview

---

In this lesson, you will continue applying all the topics of this semester as you work with data in files.

## Preparation Material

---

For the most part, you know all the topics you need to complete this lesson. But, there is one additional strategy that may be helpful.

### I. FINDING THE MAX OR MIN

Recall from previous lessons that a way to find the largest number in a list is to keep track of the largest number found to date, and then update that value if you find a new one that is larger. For example:

```
numbers = [42, 25, 18, 83, 23, 85, 38, 2]

largest_so_far = numbers[0]

for number in numbers:
    if number > largest_so_far:
        # This number is larger than the largest we had seen so far

        # So it is now the largest we've seen
        largest_so_far = number

# Now, after the loop we can display it:
print(f"The largest is: {largest_so_far}")
```

This produces the following output:

```
The largest is: 85
```

It is important to remember to start the `largest_so_far` variable at either the first value (assuming it's not an empty list) or else something that you know will be smaller than everything in your list. Otherwise, you won't consider some numbers. For example, if you started it at 10, then you would never consider anything less than 10.

## II. KEEPING TRACK OF THE ITEM THAT IS THE MAX OR MIN

What if you want to know not only which number is the largest, but also the item that is the largest? For example, you may want to know not only the most expensive price in your shopping cart, but the actual product that is the most expensive. The logic above will only find the largest value. To find the largest item, you'll also need to keep track of its name or value along the way.

Consider the following list of products in a shopping cart, along with their prices:

```
shopping_cart = [  
    ["Chips", 2.99],  
    ["Bread", 2.50],  
    ["Milk", 3.19],  
    ["Ice Cream", 6.99],  
    ["Cheese", 5.99],  
    ["Candy bar", 0.99]  
]
```

In order to find the most expensive price, we can iterate through it as before:

```
max_price = -1  
  
for item in shopping_cart:  
    price = item[1] # The price is the second part of the item
```

```
if price > max_price:
    # This is the new max
    max_price = price

print(f"The maximum price is: {max_price}")
```

This produces the following output:

```
The maximum price is: 6.99
```

If we want to find the name of the item that is the most expensive, we need to declare a variable (for example, `max_product`) before the loop to keep track of the *product* that had the maximum price. Then, whenever we update the `max_price` to save a new price, we also update the `max_product` to be the corresponding product name. For example:

```
max_price = -1
max_product = "" # It doesn't matter what you set this to, it just needs to be declared

for item in shopping_cart:
    product_name = item[0] # Product name is the first part
    price = item[1]

    if price > max_price:
        # This is the new max
        max_price = price

        # Also save this product name as the max product
        max_product = product_name

print(f"The maximum price is: {max_price}")
print(f"The product with the maximum price is: {max_product}")
```

This produces the following output:

```
The maximum price is: 6.99
The product with the maximum price is: Ice Cream
```

### III. OTHER CODE WITH LOOPS AND FILES

In addition to the code samples you've seen here, keep in mind that you can use all of the other components you have learned throughout the semester in conjunction with files. For example, if you wanted to restrict your analysis of products to only those that had a price over a certain amount, or that had a name matching certain criteria, you can include an if statement in the middle of the loop that is iterating through the file.

In many ways, this lesson is the culmination of all the building blocks that you've learned this semester and brings them all together.