Jack Johanneson

11/08/23

IT FDN 110

Assignment04

Error Handling and Dictionaries

Introduction

This week's coursework covers error handling and dictionaries. Errors handling is an important skill in managing your code and how other people interact with it. Dictionaries are a useful dataset that is more readable to humans, which is important in making your code readable to people less familiar to our specific script.

Error Handling

Error Handling can be used to prevent unwanted behavior. When running a piece of code that is prone to errors you can input code to help work around this error rather than it being a dead end in your code. Another use case is if your code has a very specific way it must operate you can build in an error to ensure that the input and operation is behaving as you want it to.

```
# Input user data
if menu_choice == "1":  # This will not work if it is an integer!
        try:
            student_first_name = input("Enter the student's first name: ")
            if not student_last_name.isalpha(): #check if the name is only letters
                raise ValueError("The first name should not contain numbers") #error if non letters in name
```

Figure 1. Example of an imposed error. The first name of a student here should only contain alphabetic characters, if non alphabetic characters are present, we voluntarily throw a 'ValueError' error.

Dictionaries

Dictionaries are similar to lists, but they allow us to use 'keys' to identify index locations. Keys can be more intuitive because they can be represented as a string rather than an integer. This makes referencing the data more intuitive and less dependent on memorizing what a given index is associated to. These can be used in very similar ways to lists.

```
course_name = input("Please enter the name of the course: ")
student_data = {'first_name': student_first_name, 'last_name': student_last_name, 'course_name': course_name}
students.append(student_data)
```

Figure 2 Example of a dictionary. The keys in quotes allow more intuitive access to the variables they are associated to.

Summary

The most important lesson of this week is learning how to handle errors, this is a skill that will save time and effort, in addition to making your programs more robust. This combined with debugging techniques previously covered will make programming much less infuriating. The dictionary variable is a nice quality of life tool, while they are never mandatory to use, they can often make your life easier.