Junyan Lu

May 10, 2023

Foundation of Programming: Python

Assignment 05

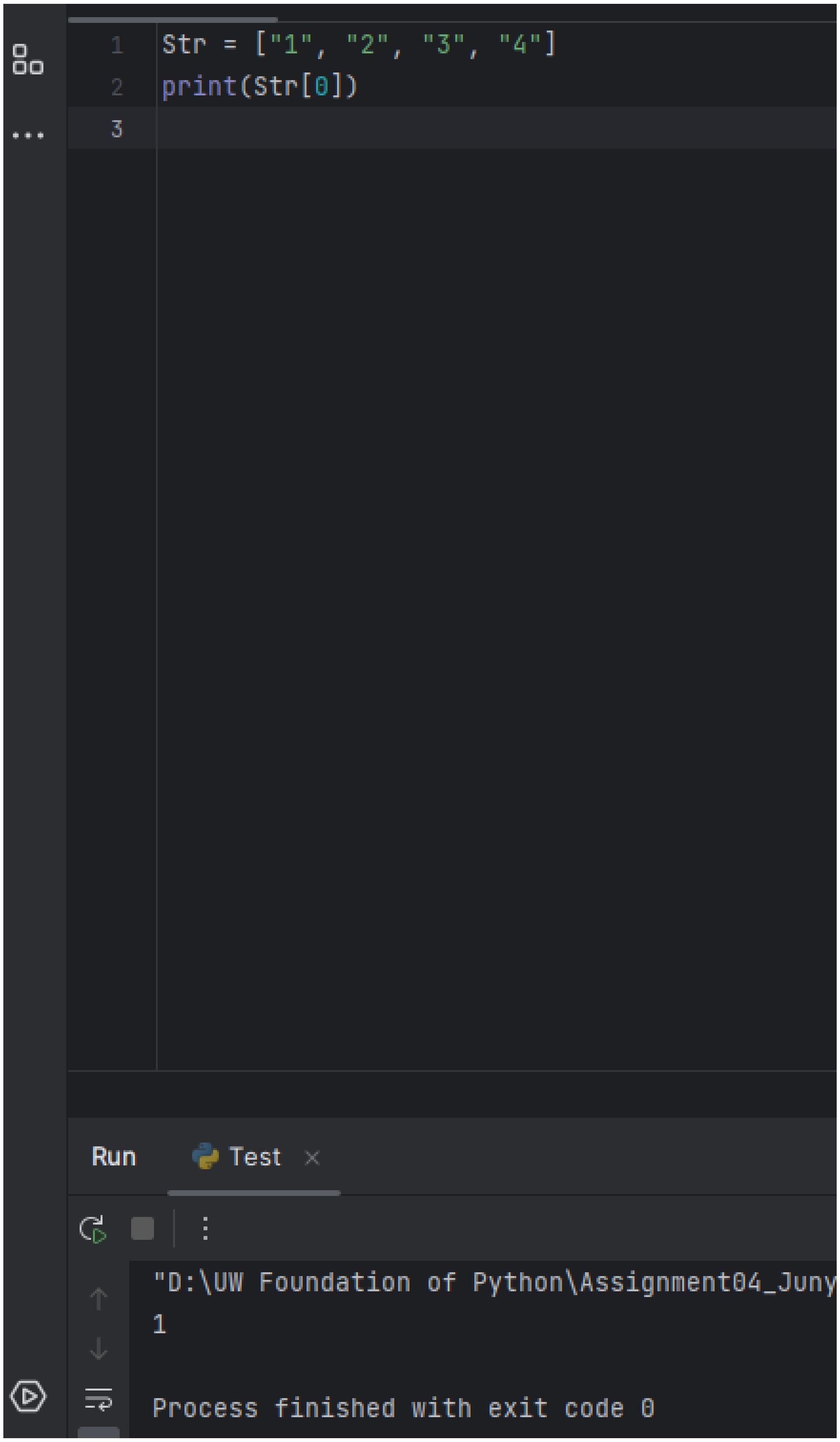
Using Lists and Dictionary to store data

Introduction

I learned about List and Dictionary as data storage structures. Lists are ordered collections of elements that can be accessed by their index. Dictionaries, on the other hand, store data as key-value pairs, where each key is unique. Both List and Dictionary are powerful tools for storing and managing data efficiently in various programming languages.

List

List is an ordered collection of elements enclosed in square brackets “[ ]”, allowing for easy storage, organization, and modification of data. It supports indexing, where elements are assigned numerical positions, starting from 0. Here is an example: (Figure 1)



```
Str = ["1", "2", "3", "4"]
print(Str[0])
```

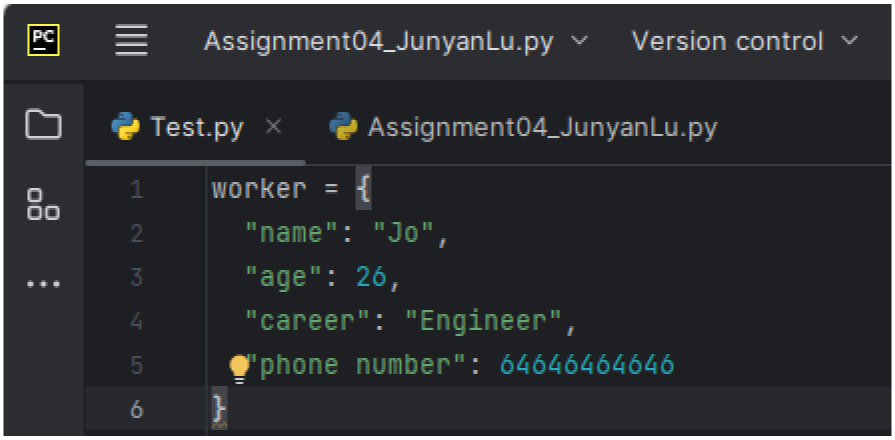
```
"D:\UW Foundation of Python\Assignment04_Juny
1

Process finished with exit code 0
```

Figure 1. List

In this example, str[0] would give result "1"

Dictionary

Dictionary is like a real-life dictionary, where you can look up a word (key) to find its definition (value). It's a way to store and find information quickly using unique identifiers. Here's an example of dictionary: (Figure 2)



```
worker = {
  "name": "Jo",
  "age": 26,
  "career": "Engineer",
  "phone number": 64646464646
}
```

Figure 2. Dictionary

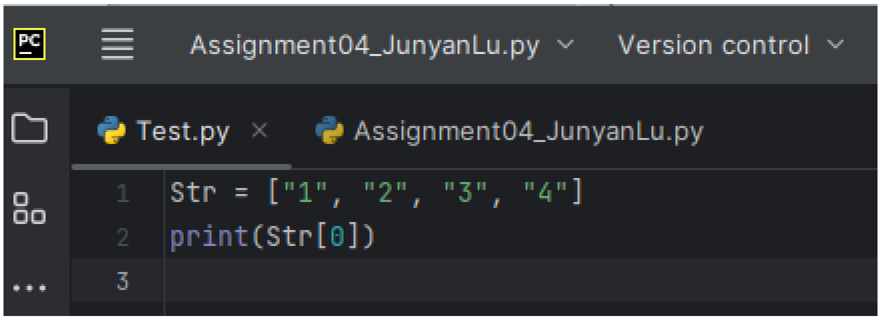
In this example, the dictionary called “worker” that stores information. The keys are "name", "age", "career", and "phone number", and the corresponding values are "Jo", 26, "Engineer", and 646464646

Difference between List and Dictionary

Lists are best for storing and managing ordered collections, while dictionaries are ideal for organizing and retrieving data based on specific identifiers or labels.

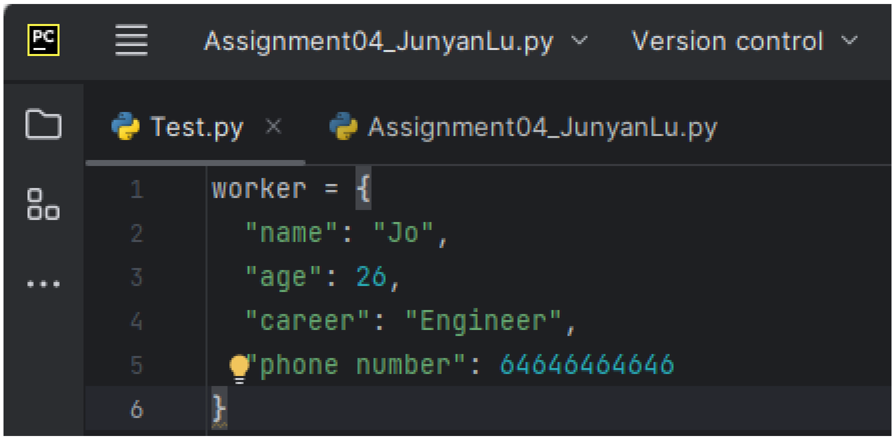
Difference between Index and Key

The index relies on the element's position, starting from 0, while the key provides a specific label or identifier to retrieve associated values. Here's an example to illustrate the difference between an index and a key:



```python
Str = ["1", "2", "3", "4"]
print(Str[0])

```

For instance, Str[0] refers to the element "1", Str[1] refers to "2", and Str[2] refers to "3".



```python
worker = {
  "name": "Jo",
  "age": 26,
  "career": "Engineer",
  "phone number": 64646464646
}
```

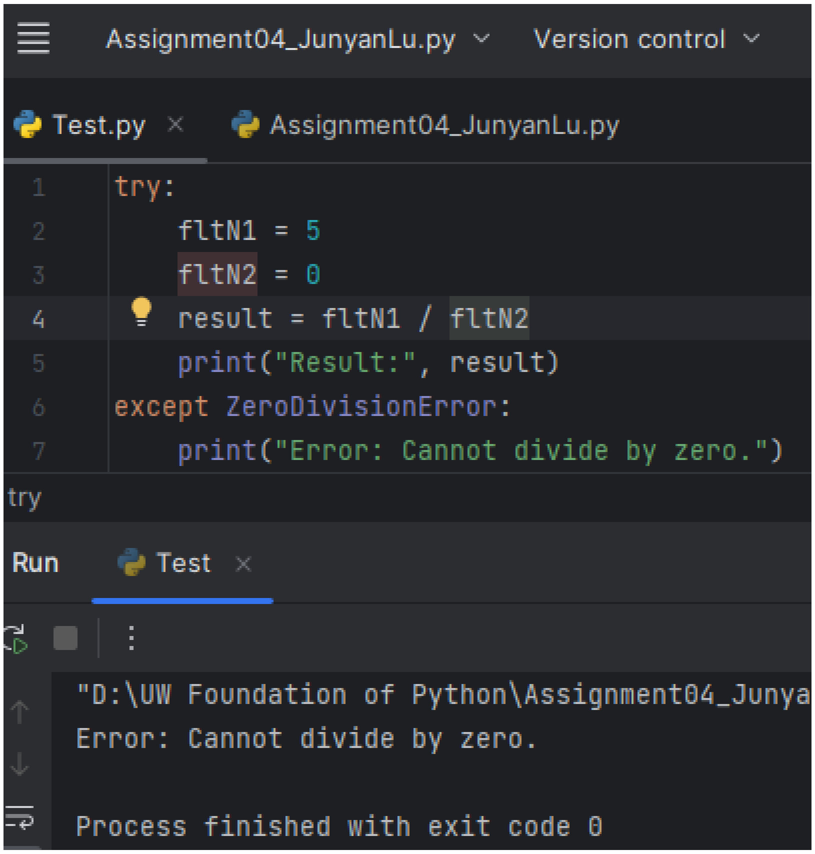
In this dictionary, to access a specific value, you use the associated key. For example, worker["name"] retrieves the value "Jo", worker["age"] retrieves 26, and worker["career"] retrieves "Engineer".

Script template

The script template is useful because it provides a predefined structure and framework for writing code. Also script templates help maintain consistency, organize code better, and encourage best practices, making it easier to understand, maintain, and collaborate on code projects.

Try-Except

Using Try-Except for error handling is recommended because it allows us to anticipate and handle potential errors or exceptions that may occur during the execution of code.Here's an example to demonstrate error handling using Try-Except in Python:



```
try:
    fltN1 = 5
    fltN2 = 0
    result = fltN1 / fltN2
    print("Result:", result)
except ZeroDivisionError:
    print("Error: Cannot divide by zero.")
```

```
"D:\UW Foundation of Python\Assignment04_Junya
Error: Cannot divide by zero.

Process finished with exit code 0
```

GitHub

GitHub is a web-based platform. It allows developers to store their code, track changes, and collaborate with others on software projects.

Summary

Lists and dictionaries are used in programming to store and organize data. A list is an ordered collection, while a dictionary uses unique keys to access values. Indexes are used for lists, and keys are used for dictionaries. Script templates provide a pre-defined structure for writing code, promoting consistency. Try-Except is used for handling errors in a controlled manner. GitHub is a platform for managing code, enabling collaboration and version control.