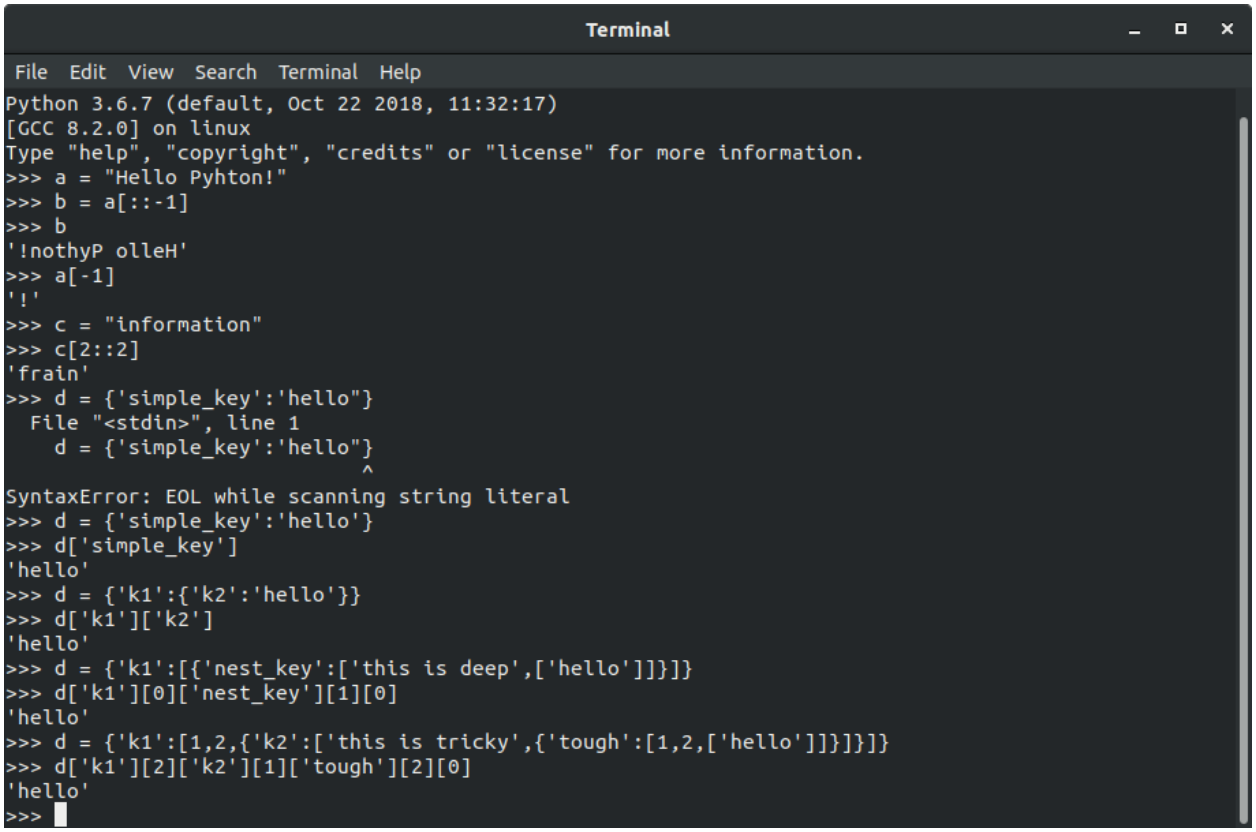
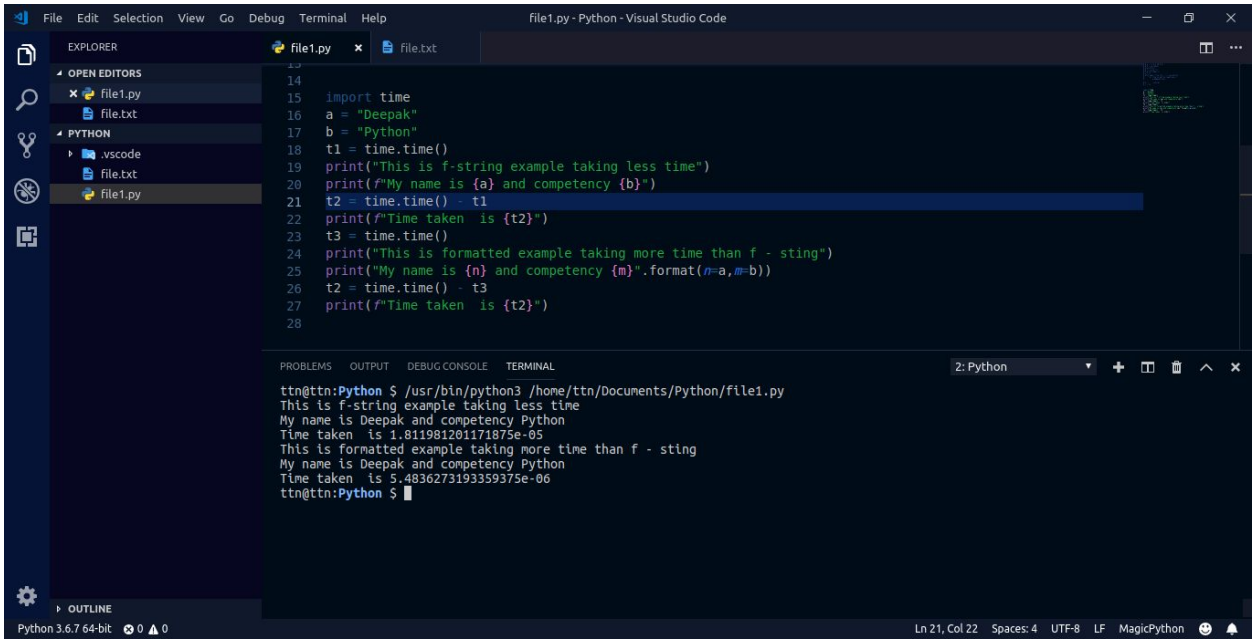
1. **Given string my_string = ‘Hello Python!’, Reverse the string using slicing, print ’!’ using indexing**
2. **Use slicing to get word “frain” from “information”.**
3. **Using keys and indexing, grab the 'hello' from the following dictionaries:**

**d = {'simple_key':'hello'}**
**d = {'k1':{'k2':'hello'}}**
**d = {'k1':[{'nest_key':['this is deep',['hello']]}]}**
**d = {'k1':[1,2,{'k2':['this is tricky',{'tough':[1,2,['hello']]}]}]}**




**4. Using examples explain string.format and f-strings**




Code using **str.format()** is much more easily readable than code using %-formatting, but **str.format()** can still be quite verbose when you are dealing with multiple parameters and longer strings.

Also called “formatted string literals,” **f-strings** are string literals that have an f at the beginning and curly braces containing expressions that will be replaced with their values. The expressions are evaluated at runtime and then formatted using the __format__ protocol.

**5. Can we sort a dictionary? Why or why not?**

**Dictionaries can't be sorted -- a mapping has no ordering! -- so, when you feel the need to sort one, you no doubt want to sort its *keys* (in a separate list). Sorting (key,value) pairs (items) is simplest, but not fastest.**

**The concept of 'sort' applies only to a collection which has _order_ -- a sequence; a mapping (e.g. a dictionary) has NO order, thus it cannot be sorted. Still, its keys can be extracted as a list, which can then be sorted.**