

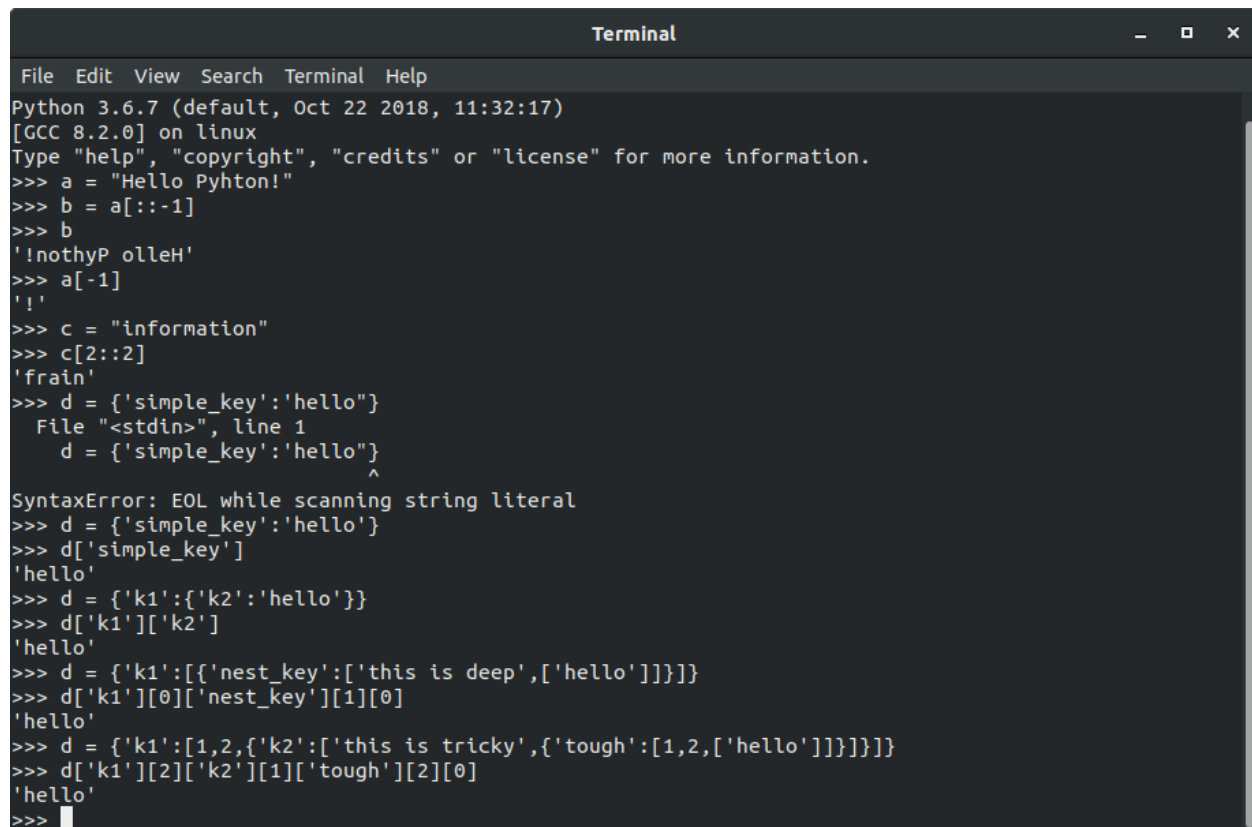
1. Given string `my_string = 'Hello Python!'`, Reverse the string using slicing, print `'!'` using indexing
2. Use slicing to get word "frain" from "information".
3. Using keys and indexing, grab the 'hello' from the following dictionaries:

```
d = {'simple_key':'hello'}
```

```
d = {'k1':{'k2':'hello'}}
```

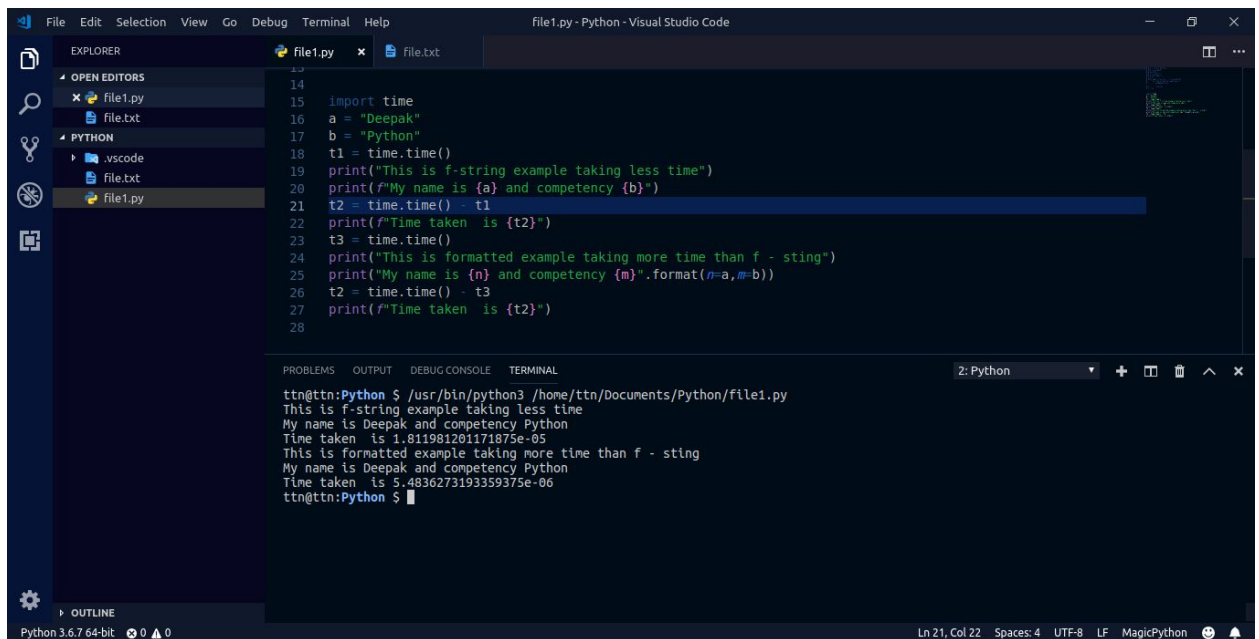
```
d = {'k1':[{'nest_key':['this is deep',['hello']]]]}
```

```
d = {'k1':[1,2,{ 'k2':['this is tricky',{'tough':[1,2,['hello']]}]}]}
```



```
Terminal
File Edit View Search Terminal Help
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a = "Hello Python!"
>>> b = a[::-1]
>>> b
'!nohtyP olleH'
>>> a[-1]
'!'
>>> c = "information"
>>> c[2::2]
'frain'
>>> d = {'simple_key':'hello'}
File "<stdin>", line 1
    d = {'simple_key':'hello'}
          ^
SyntaxError: EOL while scanning string literal
>>> d = {'simple_key':'hello'}
>>> d['simple_key']
'hello'
>>> d = {'k1':{'k2':'hello'}}
>>> d['k1']['k2']
'hello'
>>> d = {'k1':[{'nest_key':['this is deep',['hello']]}]}
>>> d['k1'][0]['nest_key'][1][0]
'hello'
>>> d = {'k1':[1,2,{ 'k2':['this is tricky',{'tough':[1,2,['hello']]}]}]}
>>> d['k1'][2]['k2'][1]['tough'][2][0]
'hello'
>>>
```

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



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the file structure with 'file1.py' selected. The main editor area displays the following Python code:

```
14
15 import time
16 a = "Deepak"
17 b = "Python"
18 t1 = time.time()
19 print("This is f-string example taking less time")
20 print(f"My name is {a} and competency {b}")
21 t2 = time.time() - t1
22 print(f"Time taken is {t2}")
23 t3 = time.time()
24 print("This is formatted example taking more time than f - sting")
25 print("My name is {n} and competency {m}".format(n=a,m=b))
26 t2 = time.time() - t3
27 print(f"Time taken is {t2}")
28
```

The TERMINAL panel at the bottom shows the output of running the script:

```
ttn@ttn:Python $ /usr/bin/python3 /home/ttn/Documents/Python/file1.py
This is f-string example taking less time
My name is Deepak and competency Python
Time taken is 1.811981201171875e-05
This is formatted example taking more time than f - sting
My name is Deepak and competency Python
Time taken is 5.4836273193359375e-06
ttn@ttn:Python $
```

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.

```
Terminal
File Edit View Search Terminal Help
ttn@ttn:~$ python3
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = {'a':4,'z':1,'f':7,'g':10}
>>> sorted(a.keys)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'builtin_function_or_method' object is not iterable
>>> sorted(a.keys())
['a', 'f', 'g', 'z']
>>> sorted(a.values())
[1, 4, 7, 10]
>>> print("But if we try to sort the dictionary we get error as ")
But if we try to sort the dictionary we get error as
>>> sorted(a)
['a', 'f', 'g', 'z']
>>> a.sort()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'dict' object has no attribute 'sort'
>>>
```