

Code [Task 01]:

```
#Task 1: Dictionary Operations
capitals = {
    'Pakistan': 'Islamabad',
    'India': 'New Delhi',
    'UAE': 'Abu Dhabi',
    'Afghanistan': 'Kabul'
}

#print(capitals)

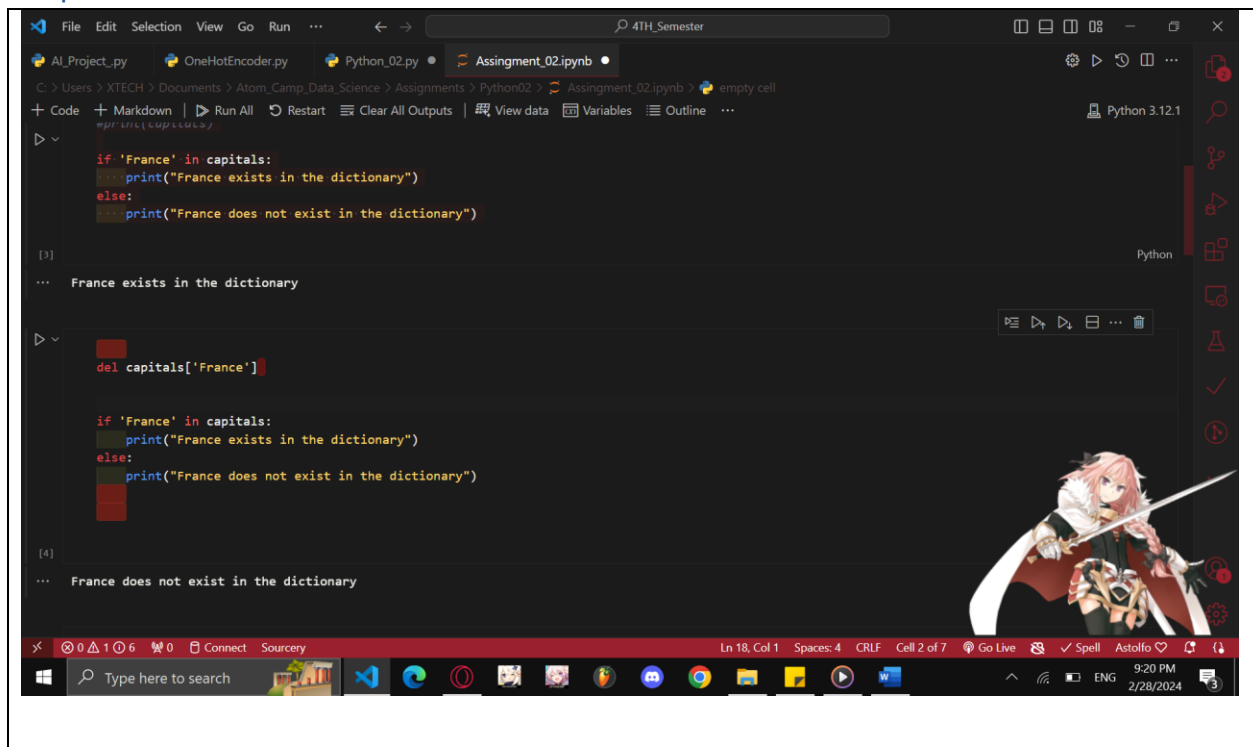
capitals['France'] = 'Paris'
#print(capitals)

if 'France' in capitals:
    print("France exists in the dictionary")
else:
    print("France does not exist in the dictionary")

del capitals['France']

if 'France' in capitals:
    print("France exists in the dictionary")
else:
    print("France does not exist in the dictionary")
```

Output:

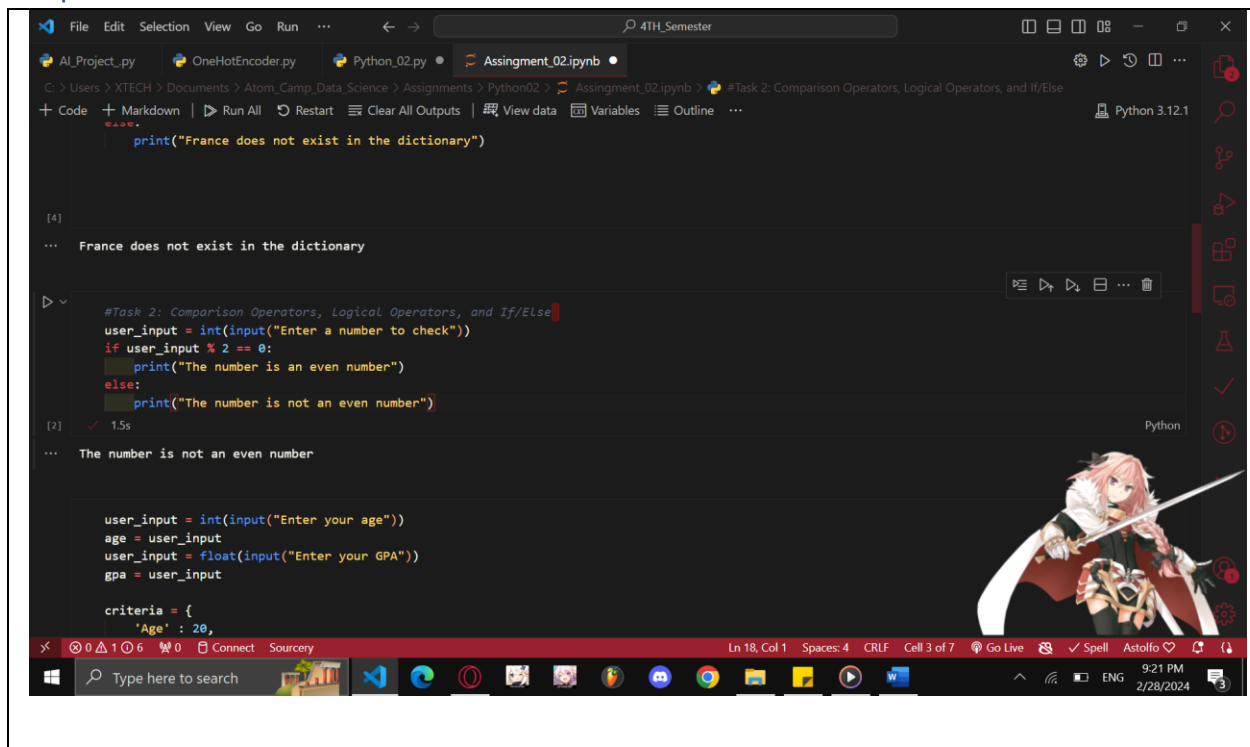


```
File Edit Selection View Go Run ...  
C:\Users\XTECH\Documents> Atom_Camp_Data_Science > Assignments > Python02 > Assingment_02.ipynb > empty cell  
+ Code + Markdown | ▶ Run All ⏮ Restart ⏻ Clear All Outputs | 📄 View data 📊 Variables 📖 Outline ...  
Python 3.12.1  
[3]  
... France exists in the dictionary  
[4]  
... France does not exist in the dictionary
```

Code [Task 02]:

```
#Task 2: Comparison Operators, Logical Operators, and If/Else  
user_input = int(input("Enter a number to check"))  
if user_input % 2 == 0:  
    print("The number is an even number")  
else:  
    print("The number is not an even number")
```

Output:



The screenshot shows a Jupyter Notebook window titled '4TH_Semester'. The notebook has several tabs: 'AI_Project.py', 'OneHotEncoder.py', 'Python_02.py', and 'Assingment_02.ipynb'. The active tab is 'Assingment_02.ipynb'. The notebook content includes a code cell with a print statement: `print("France does not exist in the dictionary")`. Below the code cell, the output is displayed: `... France does not exist in the dictionary`. The next code cell contains a task description: `#Task 2: Comparison Operators, Logical Operators, and If/Else`, followed by code for checking if a number is even or odd: `user_input = int(input("Enter a number to check"))`, `if user_input % 2 == 0:`, `print("The number is an even number")`, `else:`, `print("The number is not an even number")`. The output for this cell is: `... The number is not an even number`. The final code cell contains code for inputting age and GPA, and a dictionary definition: `user_input = int(input("Enter your age"))`, `age = user_input`, `user_input = float(input("Enter your GPA"))`, `gpa = user_input`, and `criteria = {'Age': 20, 'GPA': 3.00}`. The notebook interface includes a sidebar with icons for file operations, a top bar with menu options like 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a bottom status bar showing 'Ln 18, Col 1', 'Spaces: 4', 'CRLF', 'Cell 3 of 7', 'Go Live', 'Spell', 'Astolfo', and the date '2/28/2024'.

Code:

```
user_input = int(input("Enter your age"))
age = user_input
user_input = float(input("Enter your GPA"))
gpa = user_input

criteria = {
    'Age' : 20,
    'GPA' : 3.00
}

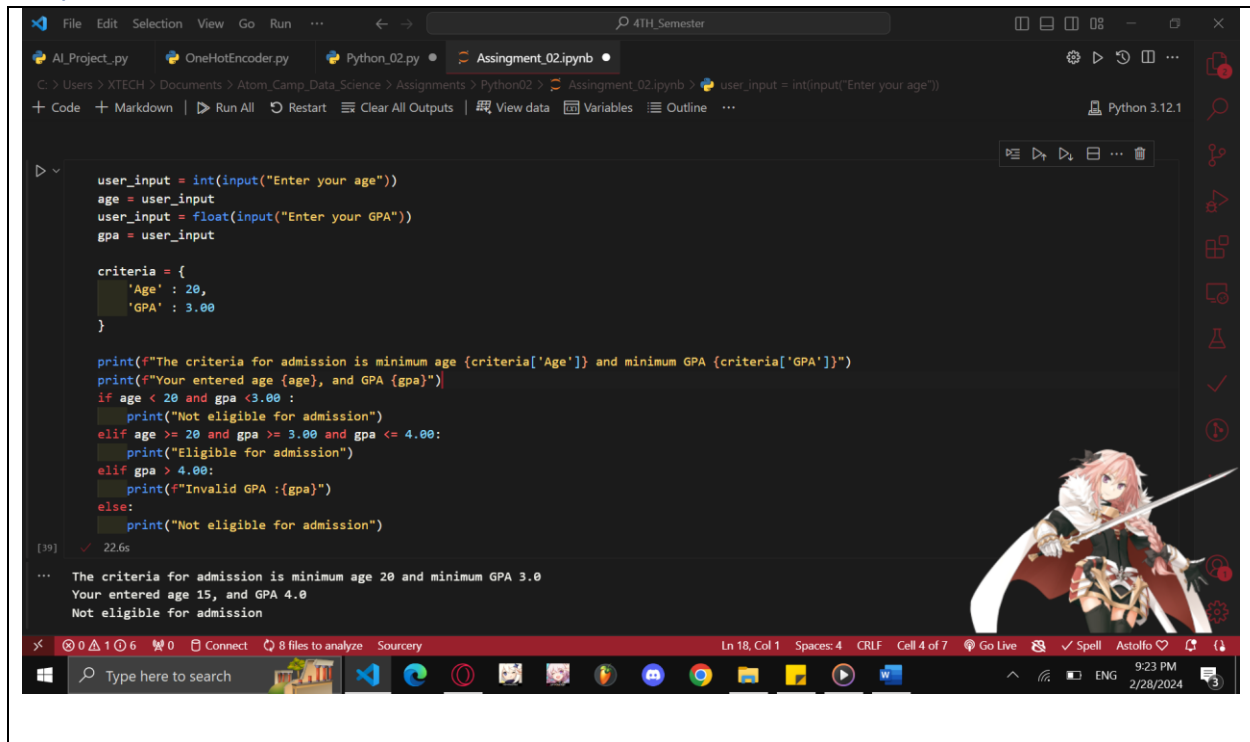
print(f"The criteria for admission is minimum age {criteria['Age']} and minimum GPA {criteria['GPA']}")
print(f"You entered age {age}, and GPA {gpa}")
if age < 20 and gpa < 3.00 :
    print("Not eligible for admission")
elif age >= 20 and gpa >= 3.00 and gpa <= 4.00:
    print("Eligible for admission")
elif gpa > 4.00:
```

```

        print(f"Invalid GPA :{gpa}")
    else:
        print("Not eligible for admission")

```

Output:



```

user_input = int(input("Enter your age"))
age = user_input
user_input = float(input("Enter your GPA"))
gpa = user_input

criteria = {
    'Age' : 20,
    'GPA' : 3.00
}

print(f"The criteria for admission is minimum age {criteria['Age']} and minimum GPA {criteria['GPA']}")
print(f"Your entered age {age}, and GPA {gpa}")
if age < 20 and gpa < 3.00 :
    print("Not eligible for admission")
elif age >= 20 and gpa >= 3.00 and gpa <= 4.00:
    print("Eligible for admission")
elif gpa > 4.00:
    print(f"Invalid GPA :{gpa}")
else:
    print("Not eligible for admission")

```

Output:

```

The criteria for admission is minimum age 20 and minimum GPA 3.0
Your entered age 15, and GPA 4.0
Not eligible for admission

```

Code [Task 03]:

```

#Task 3: Advanced Data Types
fruit_set_01 = {'Apple', 'Banana', 'Orange', 'Grapes'}
fruit_set_02 = {'Apple', 'Banana',}

print(f"Fruit Set 01 : {fruit_set_01}")
print(f"Fruit Set 02 : {fruit_set_02}")

union_set = fruit_set_01 | fruit_set_02
print(f"Union Set : {union_set}")

intersection_set = fruit_set_01 & fruit_set_02
print(f"intersection Set : {intersection_set}")

```

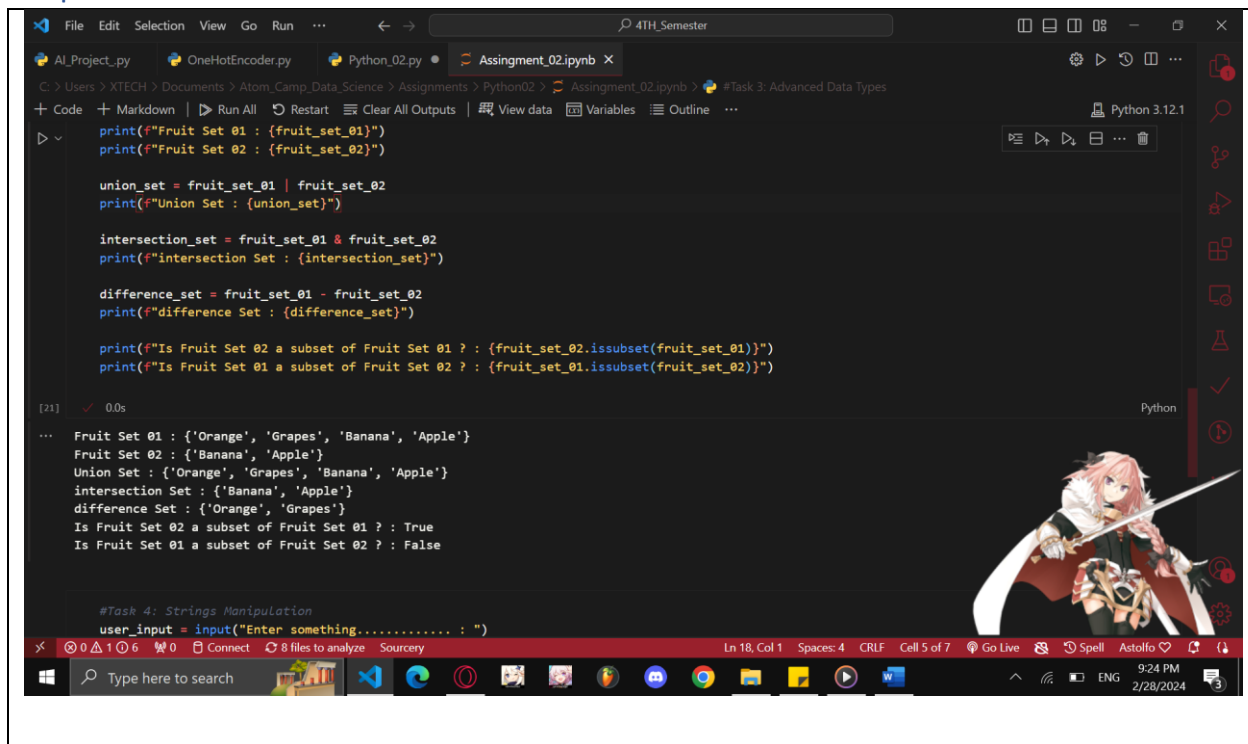
```

difference_set = fruit_set_01 - fruit_set_02
print(f"difference Set : {difference_set}")

print(f"Is Fruit Set 02 a subset of Fruit Set 01 ? :
{fruit_set_02.issubset(fruit_set_01)}")
print(f"Is Fruit Set 01 a subset of Fruit Set 02 ? :
{fruit_set_01.issubset(fruit_set_02)}")

```

Output:



```

File Edit Selection View Go Run ... 4TH_Semester
AI_Project.py OneHotEncoder.py Python_02.py Assingment_02.ipynb X
C:\Users\XTECH\Documents\Atom_Camp_Data_Science\Assignments\Python02> Assingment_02.ipynb #Task 3: Advanced Data Types
+ Code + Markdown | ▶ Run All ⏮ Restart ☰ Clear All Outputs | 🔍 View data 📄 Variables 📖 Outline ... Python 3.12.1
▶ ~
union_set = fruit_set_01 | fruit_set_02
print(f"Union Set : {union_set}")

intersection_set = fruit_set_01 & fruit_set_02
print(f"Intersection Set : {intersection_set}")

difference_set = fruit_set_01 - fruit_set_02
print(f"difference Set : {difference_set}")

print(f"Is Fruit Set 02 a subset of Fruit Set 01 ? : {fruit_set_02.issubset(fruit_set_01)}")
print(f"Is Fruit Set 01 a subset of Fruit Set 02 ? : {fruit_set_01.issubset(fruit_set_02)}")

[21] ✓ 0.0s
...
Fruit Set 01 : {'Orange', 'Grapes', 'Banana', 'Apple'}
Fruit Set 02 : {'Banana', 'Apple'}
Union Set : {'Orange', 'Grapes', 'Banana', 'Apple'}
Intersection Set : {'Banana', 'Apple'}
difference Set : {'Orange', 'Grapes'}
Is Fruit Set 02 a subset of Fruit Set 01 ? : True
Is Fruit Set 01 a subset of Fruit Set 02 ? : False

#Task 4: Strings Manipulation
user_input = input("Enter something..... : ")

```

Code [Task 04]:

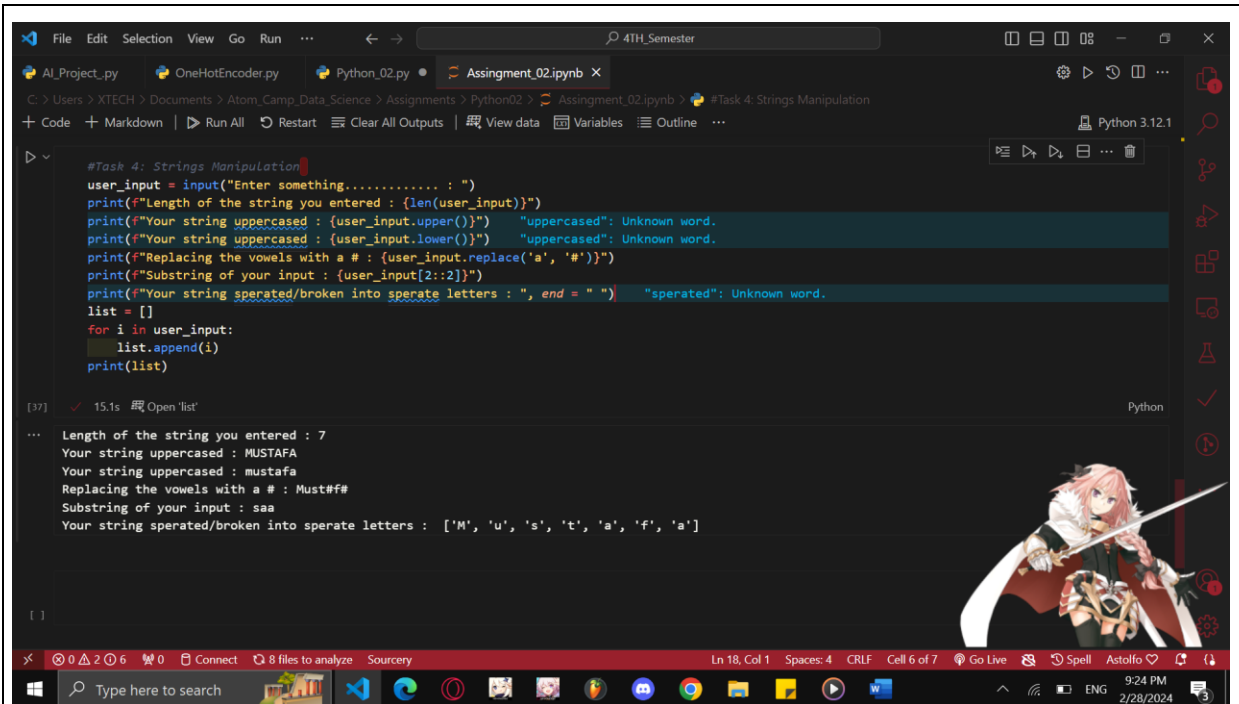
```

#Task 4: Strings Manipulation
user_input = input("Enter something..... : ")
print(f"Length of the string you entered : {len(user_input)}")
print(f"Your string uppercased : {user_input.upper()}")
print(f"Your string lowercased : {user_input.lower()}")
print(f"Replacing the vowels with a # : {user_input.replace('a', '#')}")
print(f"Substring of your input : {user_input[2::2]}")
print(f"Your string sperated/broken into sperate letters : ", end = " ")

```

```
list = []
for i in user_input:
    list.append(i)
print(list)
```

Output:



The screenshot shows a Jupyter Notebook window titled "Assingment_02.ipynb" with a Python 3.12.1 kernel. The code in the cell is as follows:

```
#Task 4: Strings Manipulation
user_input = input("Enter something..... : ")
print(f"Length of the string you entered : {len(user_input)}")
print(f"Your string uppercased : {user_input.upper()}") "uppercased": Unknown word.
print(f"Your string uppercased : {user_input.lower()}") "uppercased": Unknown word.
print(f"Replacing the vowels with a # : {user_input.replace('a', '#')}")
print(f"Substring of your input : {user_input[2::2]}")
print(f"Your string sperated/broken into sperate letters : ", end = " ") "sperated": Unknown word.
list = []
for i in user_input:
    list.append(i)
print(list)
```

The output of the code is displayed below the cell:

```
[37] ✓ 15.1s 🔍 Open 'list'

... Length of the string you entered : 7
Your string uppercased : MUSTAFA
Your string uppercased : mustafa
Replacing the vowels with a # : Must##
Substring of your input : saa
Your string sperated/broken into sperate letters : ['M', 'u', 's', 't', 'a', 'f', 'a']

[ ]
```

The interface includes a file explorer on the left, a toolbar with various icons, and a status bar at the bottom showing "Ln 18, Col 1" and "Spaces: 4". The Windows taskbar is visible at the very bottom with the date "2/28/2024".