Home | Using return statement | Untitled17

localhost:8888/notebooks/Untitled17.ipynb

jupyter Untitled17 Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[3]:
```
#using the return ststement
def cube(num):
    return num **3
num=int(input("Enter the number"))
result=cube(num)
print(" the cube is:",result)
```

```
Enter the number 23
 the cube is: 12167
```

[1]:
```
#using Break statement
while True:
    num=int(input("Enter the number"))
    if num <0:
            print("negative number enter Stop excecution")
            break
    print(" you entered:",num)
```

```
Enter the number 23
 you enntered: 23
Enter the number -3
negative number enter Stop excecution
```

[5]:
```
#using continue loop
for num in range(1,21):
 if num % 2 != 0:
     continue
 print(num)
```

Type here to search

ENG 12:07 PM 21-03-2025

Home | Using return statement | Untitled17

localhost:8888/notebooks/Untitled17.ipynb

jupyter Untitled17 Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
print(num)
```

```
2
4
6
8
10
12
14
16
18
20
```

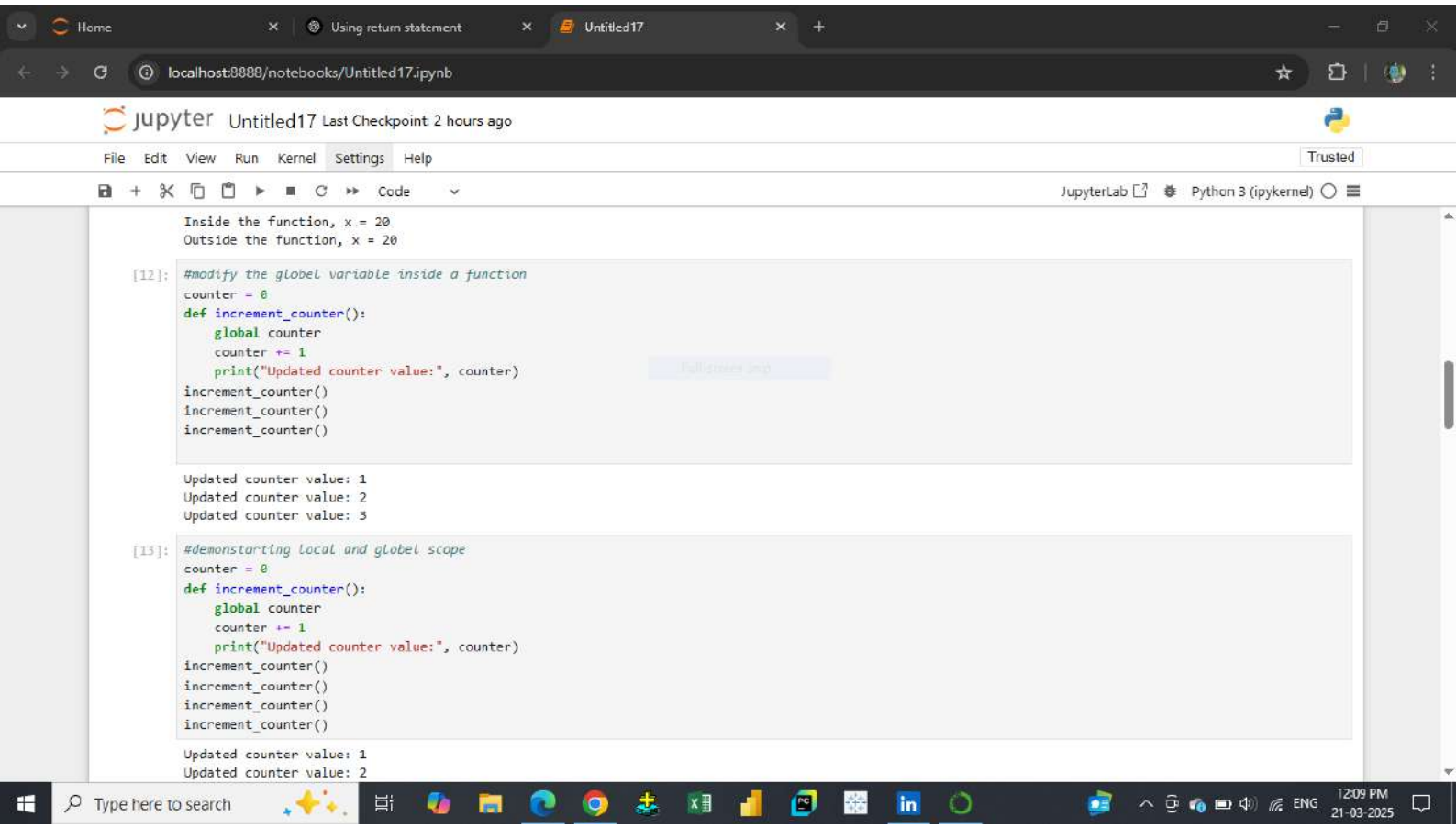
[10]:
```
# using Break and continue Loop
for num in range(1,21):
    if num > 15:
        break
    if num % 2 != 0:
     continue
print(num)
```

```
16
```

[11]:
```
#local scope in function
x = 20
def my_function():
    print("Inside the function, x =", x)
my_function()
print("Outside the function, x =", x)
```

Type here to search

ENG 12:08 PM 21-03-2025


Home
Using return statement
Untitled17
localhost:8888/notebooks/Untitled17.ipynb
jupyter Untitled17 Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Inside the function, x = 20
Outside the function, x = 20
[12]:
#modify the globel variable inside a function
counter = 0
def increment_counter():
global counter
counter += 1
print("Updated counter value:", counter)
increment_counter()
increment_counter()
increment_counter()
Updated counter value: 1
Updated counter value: 2
Updated counter value: 3
[13]:
#demonstarting local and globel scope
counter = 0
def increment_counter():
global counter
counter += 1
print("Updated counter value:", counter)
increment_counter()
increment_counter()
increment_counter()
increment_counter()
Updated counter value: 1
Updated counter value: 2
Type here to search
ENG
12:09 PM
21-03-2025

Home | Untitled17

localhost:8888/notebooks/Untitled17.ipynb

jupyter Untitled17 Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

[16]:
```
#working with tuples in python
fruits = ("apple", "banana", "cherry")
fruits_list = list(fruits)
fruits_list[1] = "orange"
fruits = tuple(fruits_list)
print("Modified tuple:", fruits)
print("Second element:", fruits[1])
fruits_list = list(fruits)
fruits_list[1] = "orange"
fruits = tuple(fruits_list)
print("Modified tuple:", fruits)
```

```
Modified tuple: ('apple', 'orange', 'cherry')
Second element: orange
Modified tuple: ('apple', 'orange', 'cherry')
```

[17]:
```
#working with sets
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
# Union
union_set = set1.union(set2)
print("Union of set1 and set2:", union_set)
# Intersection
intersection_set = set1.intersection(set2)
print("Intersection of set1 and set2:", intersection_set)
# Difference
difference_set1 = set1.difference(set2)
print("Difference (set1 - set2):", difference_set1)
difference_set2 = set2.difference(set1)
print("Difference (set2 - set1):", difference_set2)
```

Type here to search

ENG 12:10 PM 21-03-2025

Home | Untitled17

localhost:8888/notebooks/Untitled17.ipynb

jupyter Untitled17 Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

```
print("Difference (set2 - set1):", difference_set2)
set1.add(9)
print("Set1 after adding 9:", set1)
set2.remove(5)
print("Set2 after removing 5:", set2)
num = 3
if num in set1:
    print(f"{num} is present in set1.")
else:
    print(f"{num} is not present in set1.")
```

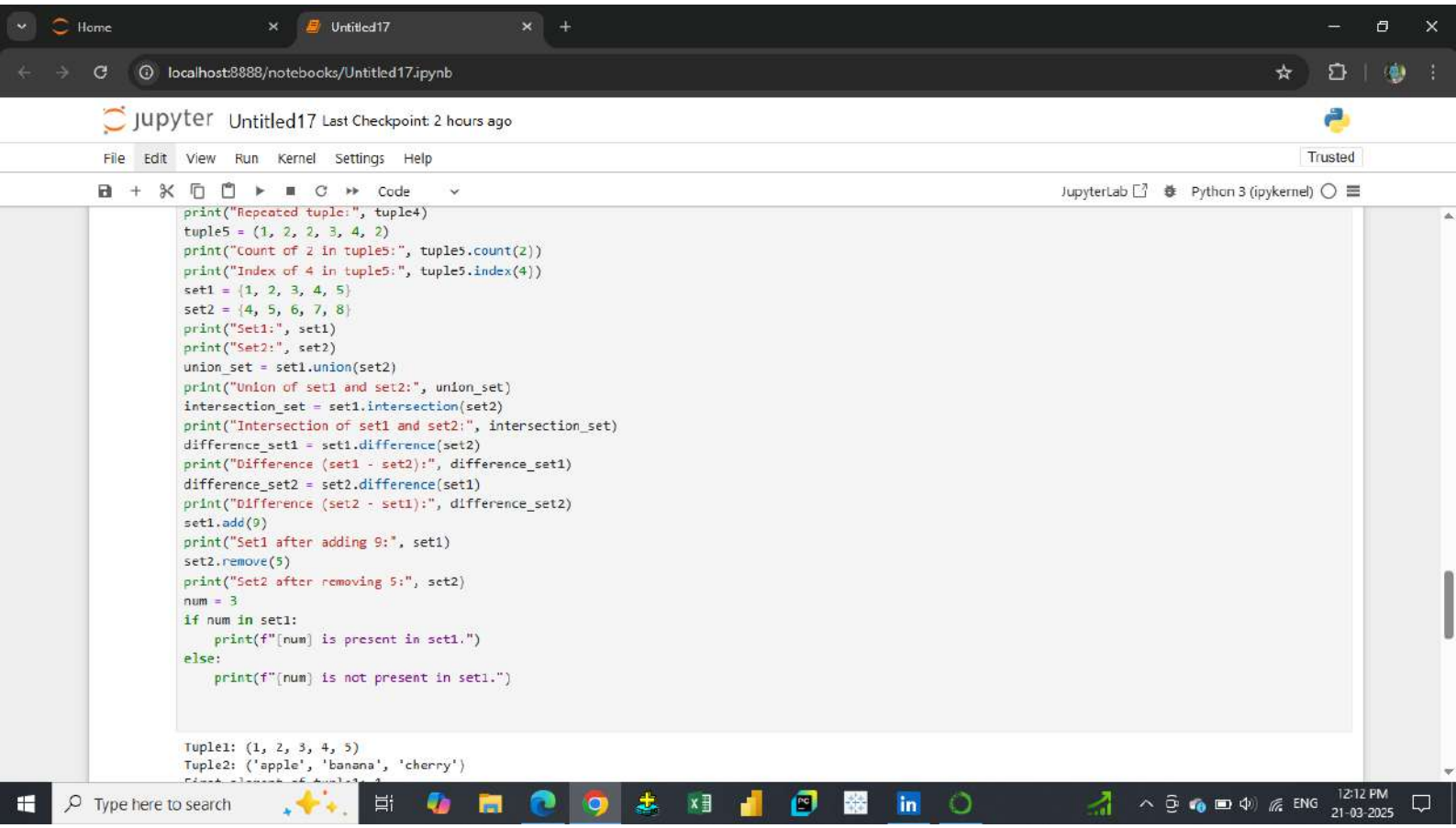
```
Union of set1 and set2: {1, 2, 3, 4, 5, 6, 7, 8}
Intersection of set1 and set2: {4, 5}
Difference (set1 - set2): {1, 2, 3}
Difference (set2 - set1): {8, 6, 7}
Set1 after adding 9: {1, 2, 3, 4, 5, 9}
Set2 after removing 5: {4, 6, 7, 8}
3 is present in set1.
```

[23]:

```
#tuple and set operation
tuple1 = (1, 2, 3, 4, 5)
tuple2 = ("apple", "banana", "cherry")
print("Tuple1:", tuple1)
print("Tuple2:", tuple2)
print("First element of tuple1:", tuple1[0])
print("Last element of tuple2:", tuple2[-1])
print("Slicing tuple1 from index 1 to 3:", tuple1[1:4])
tuple3 = tuple1 + tuple2
print("Concatenated tuple:", tuple3)
tuple4 = tuple1 * 2
```

Type here to search

ENG 12:11 PM 21-03-2025


Home
Untitled17
localhost:8888/notebooks/Untitled17.ipynb
jupyter Untitled17 Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
print("Repeated tuple:", tuple4)
tuple5 = (1, 2, 2, 3, 4, 2)
print("Count of 2 in tuple5:", tuple5.count(2))
print("Index of 4 in tuple5:", tuple5.index(4))
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
print("Set1:", set1)
print("Set2:", set2)
union_set = set1.union(set2)
print("Union of set1 and set2:", union_set)
intersection_set = set1.intersection(set2)
print("Intersection of set1 and set2:", intersection_set)
difference_set1 = set1.difference(set2)
print("Difference (set1 - set2):", difference_set1)
difference_set2 = set2.difference(set1)
print("Difference (set2 - set1):", difference_set2)
set1.add(9)
print("Set1 after adding 9:", set1)
set2.remove(5)
print("Set2 after removing 5:", set2)
num = 3
if num in set1:
print(f"{num} is present in set1.")
else:
print(f"{num} is not present in set1.")
Tuple1: (1, 2, 3, 4, 5)
Tuple2: ('apple', 'banana', 'cherry')
Type here to search
ENG
12:12 PM
21-03-2025

Home | Untitled17

localhost:8888/notebooks/Untitled17.ipynb

jupyter Untitled17 Last Checkpoint: 2 hours ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab | Python 3 (ipykernel)

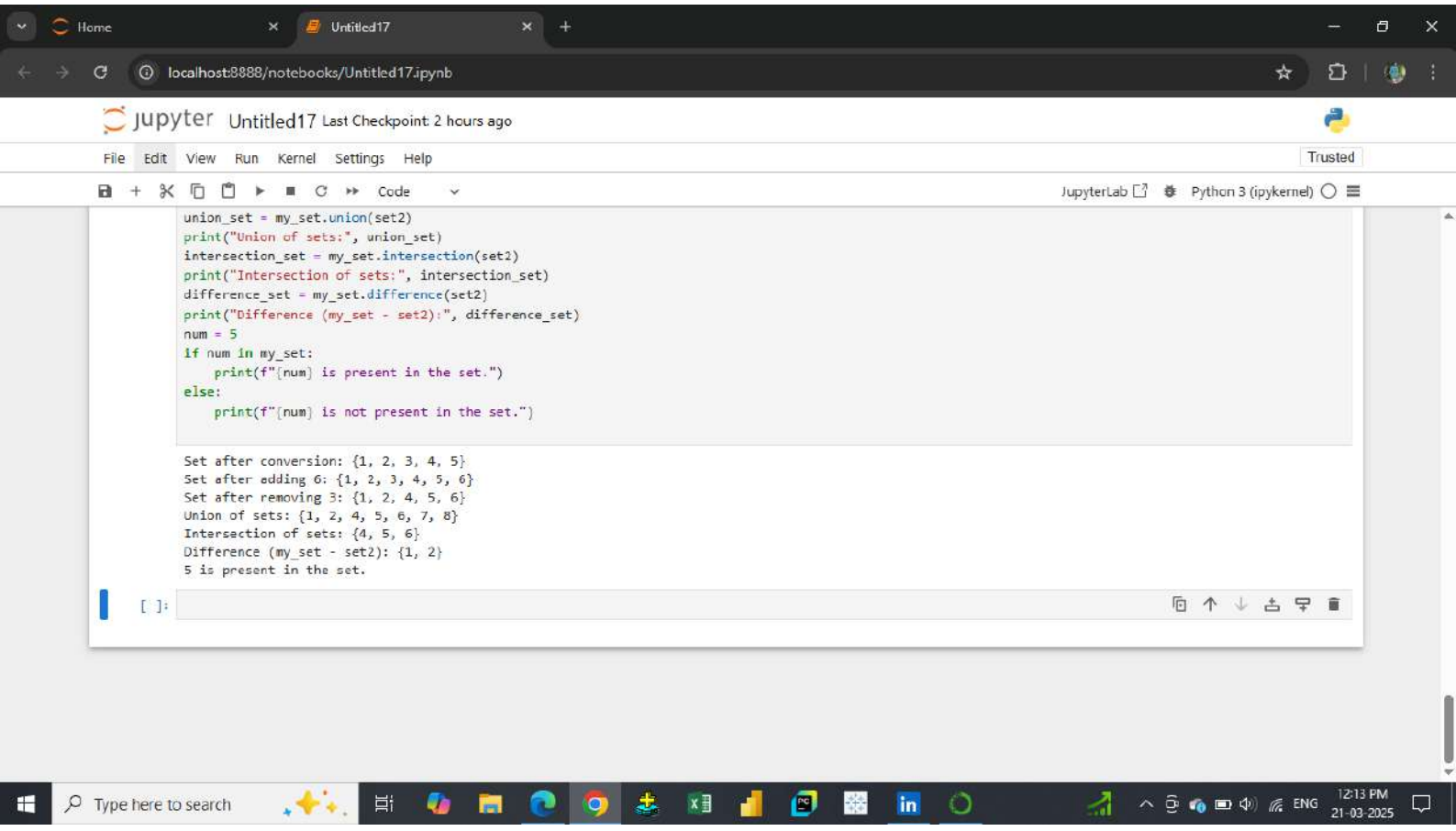
```
First element of tuple1: 1
Last element of tuple2: cherry
Slicing tuple1 from index 1 to 3: (2, 3, 4)
Concatenated tuple: (1, 2, 3, 4, 5, 'apple', 'banana', 'cherry')
Repeated tuple: (1, 2, 3, 4, 5, 1, 2, 3, 4, 5)
Count of 2 in tuple5: 3
Index of 4 in tuple5: 4
Set1: {1, 2, 3, 4, 5}
Set2: {4, 5, 6, 7, 8}
Union of set1 and set2: {1, 2, 3, 4, 5, 6, 7, 8}
Intersection of set1 and set2: {4, 5}
Difference (set1 - set2): {1, 2, 3}
Difference (set2 - set1): {8, 6, 7}
Set1 after adding 9: {1, 2, 3, 4, 5, 9}
Set2 after removing 5: {4, 6, 7, 8}
3 is present in set1.
```

[24]:

```
#list to set conversion and basic set operaton
my_list = [1, 2, 3, 4, 5, 3, 2, 1]
my_set = set(my_list)
print("Set after conversion:", my_set)
my_set.add(6)
print("Set after adding 6:", my_set)
my_set.remove(3)
print("Set after removing 3:", my_set)
set2 = {4, 5, 6, 7, 8}
union_set = my_set.union(set2)
print("Union of sets:", union_set)
intersection_set = my_set.intersection(set2)
print("Intersection of sets:", intersection_set)
difference_set = my_set.difference(set2)
```

Type here to search

ENG 12:13 PM 21-03-2025


Home
Untitled17
localhost:8888/notebooks/Untitled17.ipynb
jupyter Untitled17 Last Checkpoint: 2 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
union_set = my_set.union(set2)
print("Union of sets:", union_set)
intersection_set = my_set.intersection(set2)
print("Intersection of sets:", intersection_set)
difference_set = my_set.difference(set2)
print("Difference (my_set - set2):", difference_set)
num = 5
if num in my_set:
print(f"{num} is present in the set.")
else:
print(f"{num} is not present in the set.")
Set after conversion: {1, 2, 3, 4, 5}
Set after adding 6: {1, 2, 3, 4, 5, 6}
Set after removing 3: {1, 2, 4, 5, 6}
Union of sets: {1, 2, 4, 5, 6, 7, 8}
Intersection of sets: {4, 5, 6}
Difference (my_set - set2): {1, 2}
5 is present in the set.
[ ]:
Type here to search
ENG
12:13 PM
21-03-2025