

Torrodjae Somerville

CTEC: 298-101 Symbolic Computation Using Big Data

Dr. Bemley

11/1/2025 : These are the proofs of my .ipynb files working in Jupyter Notebook with the correct tutorials

FAX.PLUS GoToAssist HR Microsoft Sagely TripLog Ubiquiti Wufoo Zoho Creator TJ's Technology To... scannerhowto Login - Larmax Assi... Dell Outle

Jupyter LearnPython Part2 Last Checkpoint: 14 minutes ago Trusted  
File Edit View Run Kernel Settings Help  
JupyterLab Python [conda env:base] \* Anaconda Toolbox

```
[1]: # Conditions Tutorial Completion
number = 17
second_number = 0
first_array = [1,2,3]
second_array = [1,2]
name = "Torrodjae Somerville"

if number > 15:
    print("1")
if first_array:
    print("2")
if len(second_array) == 2:
    print("3")
if len(first_array) + len(second_array) == 5:
    print("4")
if first_array and first_array[0] == 1:
    print("5")
if not second_number:
    print("6")
print(name)

1
2
3
4
5
6
Torrodjae Somerville
```

```
[2]: #Loops Tutorial Completion
name = "Torrodjae Somerville"
```

Jupyter LearnPython Part2 Last Checkpoint: 14 minutes ago Trusted  
File Edit View Run Kernel Settings Help  
JupyterLab Python [conda env:base] \* Anaconda Toolbox

```
[2]: #Loops Tutorial Completion
name = "Torrodjae Somerville"
numbers = [
    951, 402, 984, 651, 360, 69, 408, 319, 601, 485, 980, 507, 725, 547, 544,
    615, 83, 165, 141, 501, 263, 617, 865, 575, 219, 390, 984, 592, 236, 105, 942, 941,
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,
    815, 67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445, 742, 717,
    958, 609, 842, 451, 688, 753, 854, 685, 93, 857, 440, 380, 126, 721, 328, 753, 470,
    743, 527
]

# your code goes here
for number in numbers:
    if number == 237:
        break
    if number % 2 == 1:
        continue
    print(number)
print(name)
```

```
402
Torrodjae Somerville
984
Torrodjae Somerville
360
Torrodjae Somerville
408
Torrodjae Somerville
980
Torrodjae Somerville
544
Torrodjae Somerville
390
Torrodjae Somerville
984
```



```
[3]: #Functions Tutorial Completion
name= "Torrodjae Somerville"
def list_benefits():
    return "More organized code", "More readable code", "Easier code reuse", "Allowing programmers to share and connect code together"

# Modify this function to concatenate to each benefit - " is a benefit of functions!"
def build_sentence(benefit):
    return "%s is a benefit of functions!" % benefit

def name_the_benefits_of_functions():
    list_of_benefits = list_benefits()
    for benefit in list_of_benefits:
        print(build_sentence(benefit))

name_the_benefits_of_functions()
print(name)
```

More organized code is a benefit of functions!  
More readable code is a benefit of functions!  
Easier code reuse is a benefit of functions!  
Allowing programmers to share and connect code together is a benefit of functions!  
Torrodjae Somerville

```
[4]: # Classes and Object Tutorial Completion
name="Torrodjae"
```



```
[4]: # Classes and Object Tutorial Completion
name="Torrodjae"
class Vehicle:
    name = ""
    kind = "car"
    color = ""
    value = 100.00
    def description(self):
        desc_str = "%s is a %s %s worth $%.2f." % (self.name, self.color, self.kind, self.value)
        return desc_str
# your code goes here
car1 = Vehicle()
car1.name = "Fer"
car1.color = "red"
car1.kind = "convertible"
car1.value = 60000.00

car2 = Vehicle()
car2.name = "Jump"
car2.color = "blue"
car2.kind = "van"
car2.value = 10000.00
# test code
print(car1.description())
print(car2.description())
print(name)
```

Fer is a red convertible worth \$60000.00.  
Jump is a blue van worth \$10000.00.  
Torrodjae

```
[5]: #Dictionaries Tutorial Completion
```



```
Fer is a red convertible worth $60000.00.  
Jump is a blue van worth $10000.00.  
Torrodjae
```

```
[5]: #Dictionaries Tutorial Completion  
name="Torrodjae Somerville"  
phonebook = {  
    "John" : 938477566,  
    "Jack" : 938377264,  
    "Jill" : 947662781  
}  
# your code goes here  
phonebook["Jake"] = 938273443  
del phonebook["Jill"]  
# testing code  
if "Jake" in phonebook:  
    print("Jake is listed in the phonebook.")  
  
if "Jill" not in phonebook:  
    print("Jill is not listed in the phonebook.")  
print(name)
```

```
Jake is listed in the phonebook.  
Jill is not listed in the phonebook.  
Torrodjae Somerville
```

```
[6]: #Modules Packages Tutorial Completion
```



```
if "Jake" in phonebook:  
    print("Jake is listed in the phonebook.")  
  
if "Jill" not in phonebook:  
    print("Jill is not listed in the phonebook.")  
print(name)
```

```
Jake is listed in the phonebook.  
Jill is not listed in the phonebook.  
Torrodjae Somerville
```

```
[6]: #Modules Packages Tutorial Completion  
import re  
name="Torrodjae Somerville"  
# Your code goes here  
find_members = []  
for member in dir(re):  
    if "find" in member:  
        find_members.append(member)  
  
print(sorted(find_members))  
print(name)
```

```
['findall', 'finditer']  
Torrodjae Somerville
```

```
[ ]:
```

