

Home C2_DataStructures +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

Checkpoint created: 10:47:41 Not Trusted Python 3

Out[38]: 3

In [42]: `# Lets working with string
greeting = 'hello'
name = "NARESHIT aspired data scientitst"`

In [3]: `# Lets working with string
greeting = 'hello!!'
name = "First DS Practice!!!!"`

In [4]: `message = greeting + name
message`

Out[4]: 'hello!! First DS Practice!!!!'

In [40]: `name = "Naresh"
message = greeting + name
message`

Out[40]: 'hello NARESH'

In [5]: `name = "Arshad"
message = greeting + name
message`

Out[5]: 'hello!!Arshad'

In [44]: `# ===== BOOL VARIABLES AND OPERATORS =====
4<5`

Home C2_DataStructures +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Out[40]: 'hello NARESH'

In [5]: `name = "Arshad"
message = greeting + name
message`

Out[5]: 'hello!!Arshad'

In [44]: `# ===== BOOL VARIABLES AND OPERATORS =====
4<5`

Out[44]: True

In [45]: `10>100`

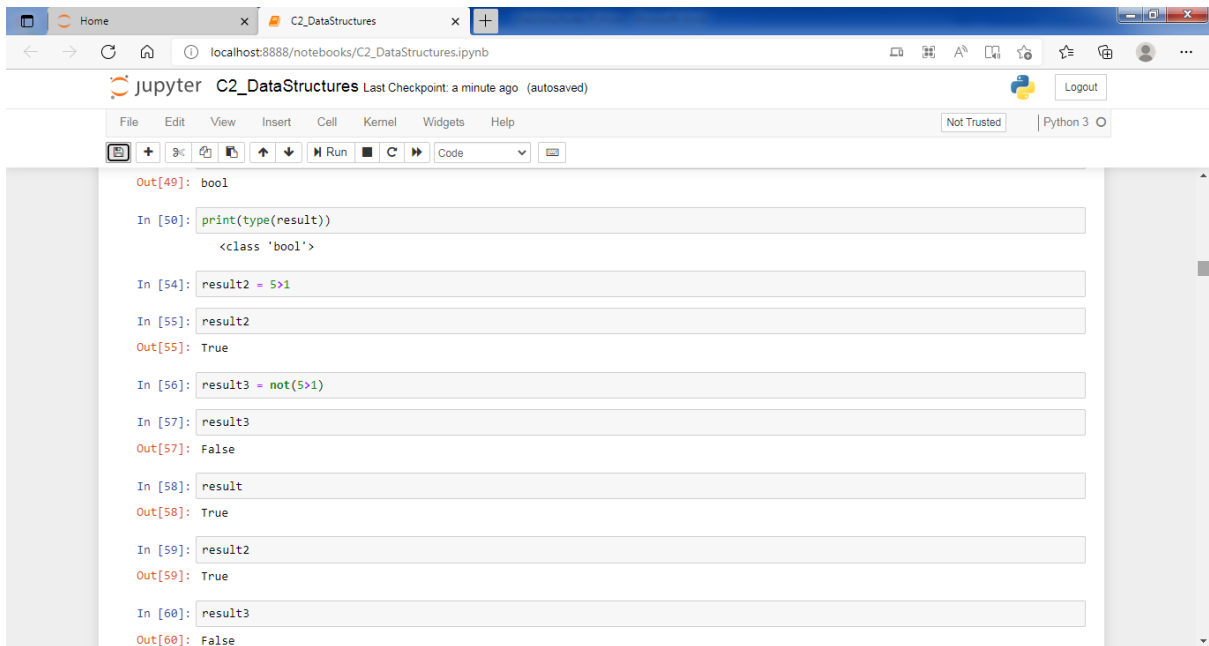
Out[45]: False

In []: `# <
>
<=
>=
and
or
not`

In [46]: `result = 4<5`

In [53]: `result`

Out[53]: True



Out[49]: bool

In [50]: `print(type(result))`
<class 'bool'>

In [54]: `result2 = 5>1`

In [55]: `result2`
Out[55]: True

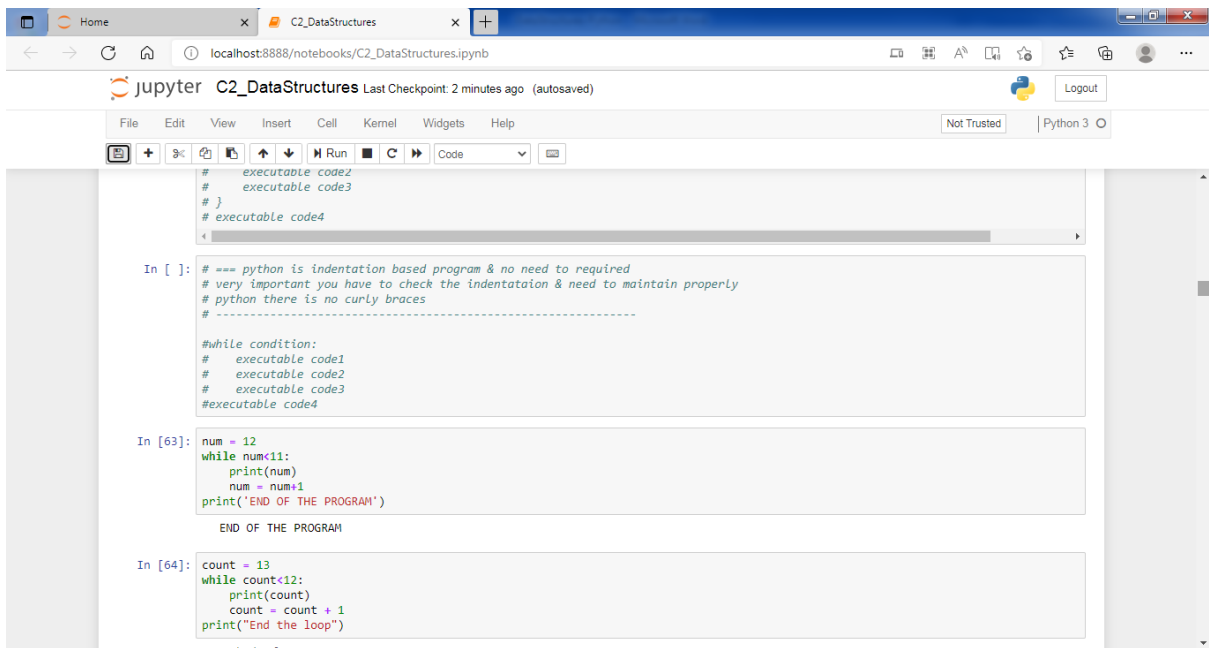
In [56]: `result3 = not(5>1)`

In [57]: `result3`
Out[57]: False

In [58]: `result`
Out[58]: True

In [59]: `result2`
Out[59]: True

In [60]: `result3`
Out[60]: False



executable code2
executable code3
}
executable code4

In []: `# == python is indentation based program & no need to required`
`# very important you have to check the indentataion & need to maintain properly`
`# python there is no curly braces`
`# -----`

`#while condition:`
`# executable code1`
`# executable code2`
`# executable code3`
`#executable code4`

In [63]: `num = 12`
`while num<11:`
 `print(num)`
 `num = num+1`
`print('END OF THE PROGRAM')`

END OF THE PROGRAM

In [64]: `count = 13`
`while count<12:`
 `print(count)`
 `count = count + 1`
`print("End the loop")`

End the loop

A Jupyter Notebook interface with the title 'C2_DataStructures'. The browser address bar shows 'localhost:8888/notebooks/C2_DataStructures.ipynb'. The notebook has three code cells. The first cell (In [8]) contains a for loop that iterates over the range(3) and prints 'Hello python:' followed by the index i, and 'welcome to DATA SCIENCE' on the next line. The output shows three iterations. The second cell (In [6]) contains a for loop that iterates over the range(10) and prints 'Hello python:' followed by the index i, and '#print("Hello python:", i)' on the next line. The output shows ten iterations. The third cell (In [68]) contains a comment '#Another way' and a list creation statement 'mylist = [10, 100, 1000, 2000]'. The output shows the list [10, 100, 1000, 2000]. The fourth cell (In [69]) contains a slice operation 'mylist[::-1]' and the output shows the reversed list [2000, 1000, 100, 10].

```
In [8]: for i in range(3):
        print("Hello python:", i)
        print('welcome to DATA SCIENCE')

Hello python: 0
welcome to DATA SCIENCE
Hello python: 1
welcome to DATA SCIENCE
Hello python: 2
welcome to DATA SCIENCE

In [6]: for i in range(10):
        print("Hello python:", i)
        #print("Hello python:", i)

Hello python: 0
Hello python: 1
Hello python: 2
Hello python: 3
Hello python: 4
Hello python: 5
Hello python: 6
Hello python: 7
Hello python: 8
Hello python: 9

In [68]: #Another way
mylist = [10, 100, 1000, 2000]

In [69]: mylist[::-1]
```

Lists

A Jupyter Notebook interface with the title 'C2_DataStructures'. The browser address bar shows 'localhost:8888/notebooks/C2_DataStructures.ipynb'. The notebook has several code cells. The first cell (In [68]) contains a comment '#Another way' and a list creation statement 'mylist = [10, 100, 1000, 2000]'. The output shows the list [10, 100, 1000, 2000]. The second cell (In [69]) contains a slice operation 'mylist[::-1]' and the output shows the reversed list [2000, 1000, 100, 10]. The third cell (In [75]) contains a slice operation 'mylist[::-1]' and the output shows the reversed list [10, 100, 1000]. The fourth cell (In [80]) contains a slice operation 'mylist[::-2]' and the output shows the list [10, 100]. The fifth cell (In [81]) contains a list access 'mylist' and the output shows the list [10, 100, 1000, 2000]. The sixth cell (In [83]) contains a list access 'mylist[3]' and the output shows the value 2000. The seventh cell (In [84]) contains a for loop that iterates over the list 'mylist' and prints 'jj is equal to :', j) for each element. The output shows four iterations.

```
In [68]: #Another way
mylist = [10, 100, 1000, 2000]

In [69]: mylist[::-1]
Out[69]: [2000, 1000, 100, 10]

In [75]: mylist[::-1]
Out[75]: [10, 100, 1000]

In [80]: mylist[::-2]
Out[80]: [10, 100]

In [81]: mylist
Out[81]: [10, 100, 1000, 2000]

In [83]: mylist[3]
Out[83]: 2000

In [84]: for j in mylist:
        print('jj is equal to :', j)
```

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [83]: mylist[3]
Out[83]: 2000

In [84]: for j in mylist:
          print('jj is equal to :', j)

          jj is equal to : 10
          jj is equal to : 100
          jj is equal to : 1000
          jj is equal to : 2000

In [ ]: range(5) #very quick way to creat a set of 5 numbers

In [9]: list(range(5))
Out[9]: [0, 1, 2, 3, 4]

In [1]: list(range(10)) #what actually range 5 does
Out[1]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [2]: # ===== IF STATEMENT =====
          # ----(-2),(-1),(0),(1),(2)-----

          import numpy as np # numpy is package is a scientific computation
          from numpy.random import randn
```

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [2]: # ===== IF STATEMENT =====
          # ----(-2),(-1),(0),(1),(2)-----

          import numpy as np # numpy is package is a scientific computation
          from numpy.random import randn

In [6]: randn()
Out[6]: -0.008993219757061173

In [7]: # ===== IF-ELSE STATEMENT =====
          # ----(-2),(-1),(0),(1),(2)-----

          #answer = None
          x = randn()
          if x>1: # if is the keyword then use condition & if the condition is true then code got executed and keep going down
              answer = "Greater than 1"
          else:
              answer = "Less than 1"
          print(x)
          print(answer)

          -0.4774162551104668
          Less than 1

what we learned so far -

1.Type of variable 2.Using variable 3.Logical variable and operator 4.while loop 5.For loop 6.if statement, if _ else statement, elif statement 7.random variables: randn() 8.introduce to numpy
```

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

1.Type of variable 2.Using variable 3.Logical variable and operator 4.while loop 5.For loop 6.If statement, If _else statement, elif statement 7.random variables: randn() 8.introduce to numpy

Fundamentals of python which are used in datascience -

```
In [8]: # List - ordered sequence of elements & always start with 0
# you can declare the mixed datatype variables
# List closed in [] bracket
MyFirstList = [5, 50, 67, 2020]
```

```
In [9]: MyFirstList
Out[9]: [5, 50, 67, 2020]
```

```
In [10]: MyFirstListP = [5, 50, 67, 2022]
```

```
In [11]: MyFirstListP
Out[11]: [5, 50, 67, 2022]
```

```
In [10]: type(MyFirstList)
Out[10]: list
```

```
In [11]: MysecondList = [2, 5.0, 'AMXHAM', True, 40+67j] #multiple datatypes you can declared in the List
```

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
Out[10]: list
```

```
In [11]: MysecondList = [2, 5.0, 'AMXHAM', True, 40+67j] #multiple datatypes you can declared in the List
```

```
In [12]: MysecondList
Out[12]: [2, 5.0, 'AMXHAM', True, (40+67j)]
```

```
In [13]: type(MysecondList)
Out[13]: list
```

```
In [14]: amx = ['how are you?', 55, (3,4,5)] #List inside the list
```

```
In [15]: amx
Out[15]: ['how are you?', 55, (3, 4, 5)]
```

```
In [12]: amx = [55, (3,4,5), 'how are you?'] #List inside the List
```

```
In [13]: amx
Out[13]: [55, (3, 4, 5), 'how are you?']
```

```
In [16]: fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']
```

```
In [ ]:
```

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```
In [12]: amx = [55, (3,4,5), 'how are you?'] #List inside the list
In [13]: amx
Out[13]: [55, (3, 4, 5), 'how are you?']

In [15]: fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']
In [ ]:

In [17]: fruits.count('apple')
Out[17]: 2

In [16]: fruits.count('kiwi')
Out[16]: 1

In [18]: range(15)
Out[18]: range(0, 15)

In [19]: list(range(15))
Out[19]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

In [20]: x = [1,2,3,4,5,6,7]
x
```

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

```
In [19]: list(range(15))
Out[19]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

In [20]: x = [1,2,3,4,5,6,7]
x
Out[20]: [1, 2, 3, 4, 5, 6, 7]

In [21]: y = list(range(8))
In [22]: y
Out[22]: [0, 1, 2, 3, 4, 5, 6, 7]

In [29]: z = list(range(0,8))
In [30]: z
Out[30]: [0, 1, 2, 3, 4, 5, 6, 7]

In [27]: a1 = list(range(5,8))
a1
Out[27]: [5, 6, 7]

In [28]: a2 = list(range(100,120))
a2
```

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
x = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
x
Out[17]: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']

In [68]: x[2:8] #Left side part always begins with 0 but right side always have to (no.-1)// 7-1
Out[68]: ['c', 'd', 'e', 'f', 'g', 'h']

In [69]: x[3:6]
Out[69]: ['d', 'e', 'f']

In [60]: x
Out[60]: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']

In [61]: x[-5:-3]
Out[61]: ['d', 'e']

In [18]: x[-2:-6]
Out[18]: []

In [63]: x[-5:-2]
Out[63]: ['d', 'e', 'f']

In [64]: x
```

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

```
In [18]: x[-2:-6]
Out[18]: []

In [63]: x[-5:-2]
Out[63]: ['d', 'e', 'f']

In [64]: x
Out[64]: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']

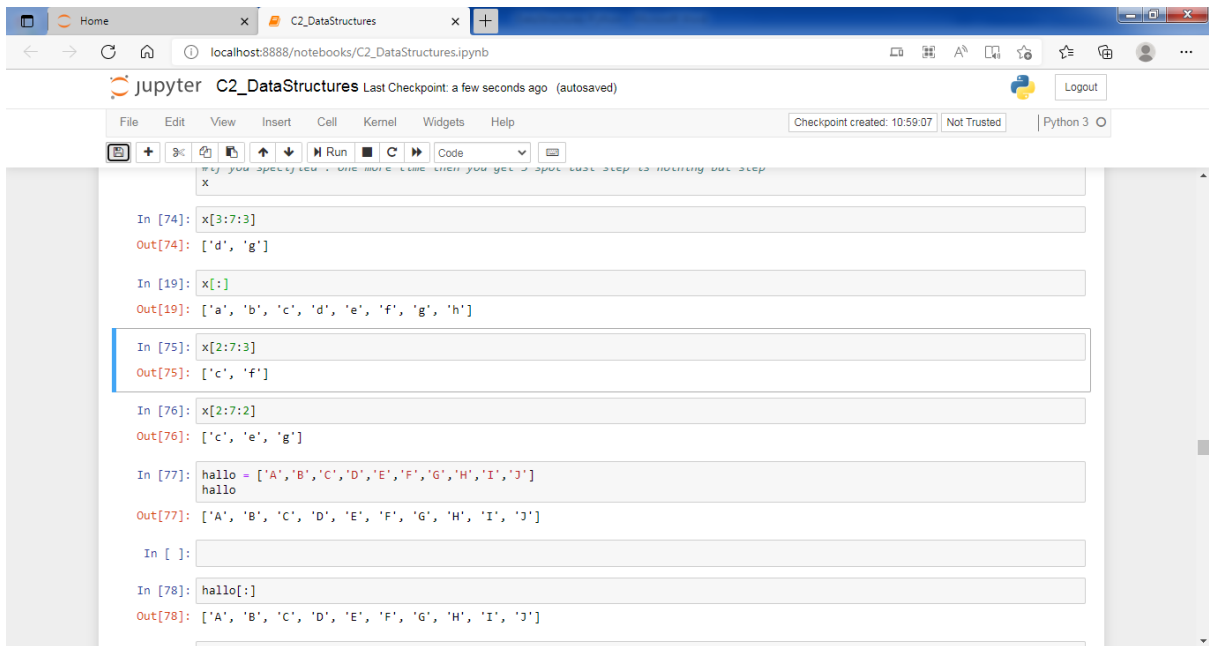
In [65]: x[2:7]
Out[65]: ['c', 'd', 'e', 'f', 'g']

In [70]: x[1:5]
Out[70]: ['b', 'c', 'd', 'e']

In [ ]: x

In [71]: x[-6:-1]
Out[71]: ['c', 'd', 'e', 'f', 'g']

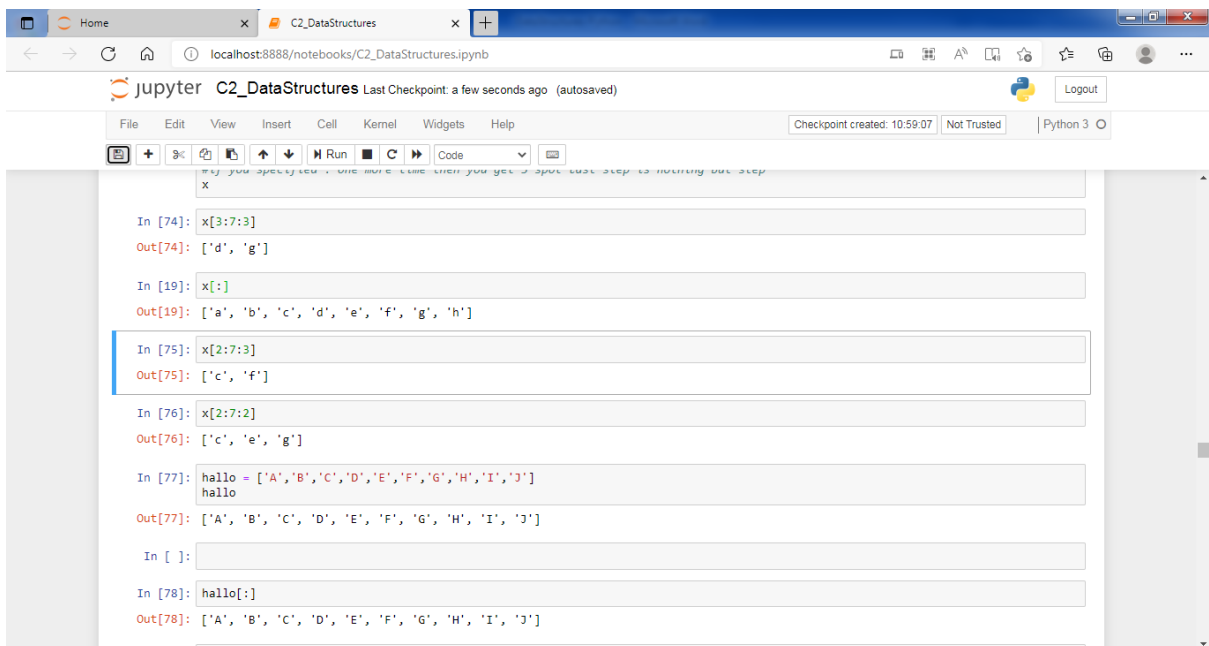
In [ ]: #advanced slicing - [: : ] starting,ending,step
#if you specified : one more time then you get 3 spot last step is nothing but step
x
```



The screenshot shows a Jupyter Notebook window titled "C2_DataStructures" with a URL of "localhost:8888/notebooks/C2_DataStructures.ipynb". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, running, and other actions. The notebook content consists of several code cells, each with an input prompt (In [n]:) and an output (Out[n:]). The code cells are as follows:

- Cell 74: `x[3:7:3]` outputs `['d', 'g']`.
- Cell 19: `x[:]` outputs `['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']`.
- Cell 75: `x[2:7:3]` outputs `['c', 'f']`.
- Cell 76: `x[2:7:2]` outputs `['c', 'e', 'g']`.
- Cell 77: `hallo = ['A','B','C','D','E','F','G','H','I','J']` followed by `hallo` outputs `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`.
- Cell 78: `hallo[:]` outputs `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`.

The notebook also shows a "Checkpoint created: 10:59:07" status and a "Not Trusted" warning. The Python version is 3.



This screenshot is identical to the one above, showing the same Jupyter Notebook interface with the same code cells and outputs. The notebook title is "C2_DataStructures" and the URL is "localhost:8888/notebooks/C2_DataStructures.ipynb". The code cells and their outputs are the same as in the first screenshot.

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

Checkpoint created: 10:59:37 Not Trusted Python 3

In [79]: `hallo[1]`
Out[79]: `'B'`

In [80]: `hallo[5]`
Out[80]: `'F'`

In [81]: `hallo`
Out[81]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [5]: `hallo[2:]`
Out[5]: `['C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [82]: `hallo[4:]`
Out[82]: `['E', 'F', 'G', 'H', 'I', 'J']`

In [6]: `hallo`
Out[6]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [83]: `hallo[:4]`
Out[83]: `['A', 'B', 'C', 'D']`

In [84]: `hallo[:3]`
Out[84]: `['A', 'B', 'C']`

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

In [5]: `hallo[2:]`
Out[5]: `['C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [82]: `hallo[4:]`
Out[82]: `['E', 'F', 'G', 'H', 'I', 'J']`

In [6]: `hallo`
Out[6]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [83]: `hallo[:4]`
Out[83]: `['A', 'B', 'C', 'D']`

In [84]: `hallo[:3]`
Out[84]: `['A', 'B', 'C']`

In [8]: `hallo`
Out[8]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [85]: `hallo[2:10:2]`
Out[85]: `['C', 'E', 'G', 'I']`

In [86]: `hallo[2:10:3]`
Out[86]: `['C', 'E', 'G']`

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

In [8]: `hello`
Out[8]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [85]: `hello[2:10:2]`
Out[85]: `['C', 'E', 'G', 'I']`

In [86]: `hello[2:10:3]`
Out[86]: `['C', 'F', 'I']`

In [10]: `hello`
Out[10]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [87]: `hello[-7:5]`
Out[87]: `['D', 'E']`

In [88]: `hello[-7:4]`
Out[88]: `['D']`

In [89]: `hello[-7:2]`
Out[89]: `[]`

In [90]: `hello[-8:-5]`

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a minute ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Out[86]: `['C', 'F', 'I']`

In [10]: `hello`
Out[10]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [87]: `hello[-7:5]`
Out[87]: `['D', 'E']`

In [88]: `hello[-7:4]`
Out[88]: `['D']`

In [89]: `hello[-7:2]`
Out[89]: `[]`

In [90]: `hello[-8:-5]`
Out[90]: `['C', 'D', 'E']`

In [13]: `hello`
Out[13]: `['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']`

In [14]: `# --- advanced slicing`
`hello[2:9:2]`
Out[14]: `['C', 'E', 'G', 'I']`

Home C2_DataStructures +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Out[90]: ['C', 'D', 'E']

In [13]: `hello`

Out[13]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [14]: `# --- advanced slicing`
`hello[2:9:2]`

Out[14]: ['C', 'E', 'G', 'I']

In [15]: `hello`

Out[15]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [91]: `hello[::3]`

Out[91]: ['A', 'D', 'G', 'J']

In [92]: `hello[::4]`

Out[92]: ['A', 'E', 'I']

In [19]: `hello`

Out[19]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [93]: `hello[::2]`

Out[93]: ['A', 'C', 'E', 'G', 'I']

Home C2_DataStructures +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Out[90]: ['C', 'D', 'E']

In [92]: `hello[::4]`

Out[92]: ['A', 'E', 'I']

In [19]: `hello`

Out[19]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [93]: `hello[::2]`

Out[93]: ['A', 'C', 'E', 'G', 'I']

In [96]: `hello[2::]`

Out[96]: ['C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [95]: `hello[2:]`

Out[95]: ['C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [21]: `hello`

Out[21]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [22]: `hello[1:-3]`

Out[22]: ['B', 'G', 'D', 'A']

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Out[93]: ['C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [21]: hallo

Out[21]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [22]: hallo[::-3]

Out[22]: ['J', 'G', 'D', 'A']

In [97]: hallo[::-3]

Out[97]: ['A', 'D', 'G', 'J']

In [24]: hallo

Out[24]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [99]: hallo[::-3]

Out[99]: ['A', 'B', 'C', 'D', 'E', 'F', 'G']

In [25]: hallo

Out[25]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [26]: hallo[:3]

Out[26]: ['A', 'B', 'C']

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Checkpoint created: 11:03:12 Not Trusted Python 3

Out[26]: ['A', 'B', 'C']

In [27]: hallo

Out[27]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [100]: hallo[6:1:-1]

Out[100]: ['G', 'F', 'E', 'D', 'C']

In [29]: hallo

Out[29]: ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J']

In [30]: hallo[6:2:-2]

Out[30]: ['G', 'E']

In [101]: hallo[6:1:-2]

Out[101]: ['G', 'E', 'C']

In [31]: *#Tuples -- Immutable list or immutable sequence of values*

t = (10,20,90)

In [32]: t

Out[32]: (10, 20, 90)

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a minute ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Out[57]: [23245, 27, 546, 215]

In [58]: 1

Out[58]: [23245, 27, 546, 215]

In [59]: a

Out[59]: array([23245, 27, 546, 215, -1234])

In [60]: a.mean()

Out[60]: 4559.8

In [63]: a.mode()

AttributeError Traceback (most recent call last)
<ipython-input-63-dd961ae6378b> in <module>
----> 1 a.mode()

AttributeError: 'numpy.ndarray' object has no attribute 'mode'

In [64]: a.median()

AttributeError Traceback (most recent call last)
<ipython-input-64-4f9e0b44009a> in <module>
----> 1 a.median()

AttributeError: 'numpy.ndarray' object has no attribute 'median'

Home x C2_DataStructures x +

localhost:8888/notebooks/C2_DataStructures.ipynb

jupyter C2_DataStructures Last Checkpoint: a few seconds ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Checkpoint created: 11:12:33 Not Trusted Python 3

In [67]: l[1:]

Out[67]: [27, 546, 215]

In [68]: l[0:2]

Out[68]: [23245, 27]

In [26]: l

Out[26]: [23245, 27, 546, 215, -1234]

In [24]: l[::4]

Out[24]: [23245, -1234]

In [28]: l[::-2]

Out[28]: [-1234, 546, 23245]

In [27]: l[::3]

Out[27]: [23245, 215]

In [73]: a

Out[73]: array([23245, 27, 546, 215, -1234])

A Jupyter Notebook interface with a browser window titled 'C2_DataStructures'. The address bar shows 'localhost:8888/notebooks/C2_DataStructures.ipynb'. The notebook has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar. The status bar indicates 'Not Trusted' and 'Python 3'. The notebook content shows a series of code cells and their outputs:

```
In [112]: c = a.copy() # when you create a copy of numpy array, modification can not be done both object array but in list it
Out[112]: array([23245, 27, 546, 215, -1234])

In [113]: c
Out[113]: array([23245, 27, 546, 215, -1234])

In [114]: a
Out[114]: array([23245, 27, 546, 215, -1234])

In [115]: c[0]
Out[115]: 23245

In [116]: c[0] = 0
Out[116]: 0

In [83]: c
Out[83]: array([ 0, 27, 546, 215, -1234])

In [84]: a
Out[84]: array([23245, 27, 546, 215, -1234])

In [117]: a[0]
Out[117]: 23245
```

A Jupyter Notebook interface with a browser window titled 'C2_DataStructures'. The address bar shows 'localhost:8888/notebooks/C2_DataStructures.ipynb'. The notebook has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar. The status bar indicates 'Not Trusted' and 'Python 3'. The notebook content shows a series of code cells and their outputs:

```
In [118]: m = [1,2,3]
          m[0]
Out[118]: 1

In [88]: m[0] = 81
Out[88]: 81

In [89]: m
Out[89]: [81, 2, 3]

In [119]: n = list.copy(m) # when you create a copy of list, modification will done both object
Out[119]: [81, 2, 3]

In [120]: n=m.copy()
Out[120]: [81, 2, 3]

In [121]: n
Out[121]: [81, 2, 3]

In [ ]: m
```

what we learned so far --

" 1> list 2> create some list[] range() 3> using [] bracket 4> slicing list 5> Tuples 6> functions 7> packages 8> Numpys & Arrays 9> slicing arrays"

BYTHOU END