

Custom Learnings

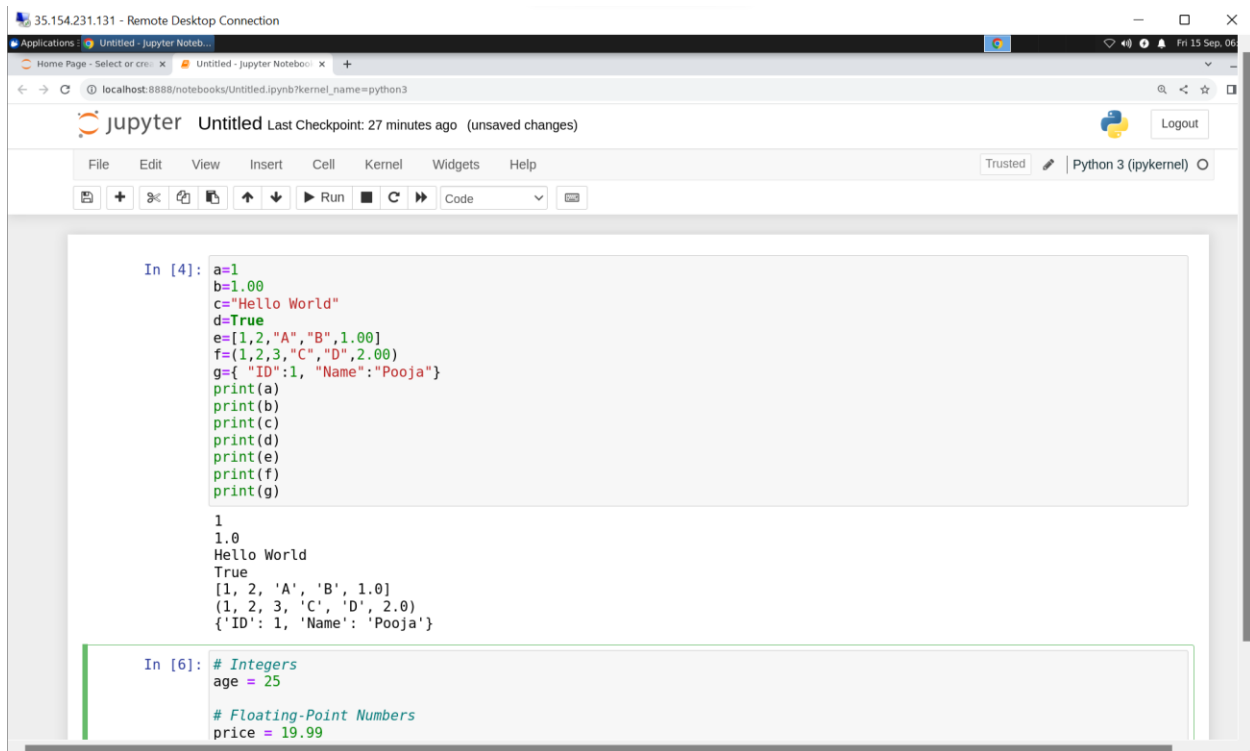
Day 13

Python

Data Types:

- Integer
- Boolean
- String
- Float
- List
- Dictionary
- Tuple

Creation of Data types in python:



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar. The notebook is titled 'Untitled' and shows the last checkpoint as '27 minutes ago (unsaved changes)'. The kernel is 'Python 3 (ipykernel)'. The code in the first cell (In [4]) defines variables a, b, c, d, e, f, and g, and prints their values. The output shows the values of these variables. The second cell (In [6]) shows comments and variable assignments for age and price.

```
In [4]: a=1
        b=1.00
        c="Hello World"
        d=True
        e=[1,2,"A","B",1.00]
        f=(1,2,3,"C","D",2.00)
        g={"ID":1, "Name":"Pooja"}
        print(a)
        print(b)
        print(c)
        print(d)
        print(e)
        print(f)
        print(g)

1
1.0
Hello World
True
[1, 2, 'A', 'B', 1.0]
(1, 2, 3, 'C', 'D', 2.0)
{'ID': 1, 'Name': 'Pooja'}
```

```
In [6]: # Integers
        age = 25

        # Floating-Point Numbers
        price = 19.99
```

Different Operators:

- Arithmetic operations: plus, minus, multiply, divide, modulus
- Logical operations: and, or, not
- Membership operations: in

```
name = person["name"]
math_grade = grades["math"]
```

```
In [5]: #Arithmetic Operator
a = 10
b = 5

# Addition
addition = a + b # Result: 15

# Subtraction
subtraction = a - b # Result: 5

# Multiplication
multiplication = a * b # Result: 50

# Division
division = a / b # Result: 2.0 (Python 3 always returns a float in division)

# Modulus (remainder)
modulus = a % b # Result: 0

# Exponentiation
exponentiation = a ** b # Result: 100000

#Membership Operator

my_list = [1, 2, 3, 4, 5]

# Membership - in operator
is_in_list = 3 in my_list # Result: True

# Membership - not in operator
is_not_in_list = 6 not in my_list # Result: True

print(addition)
print(subtraction)
print(multiplication)
print(division)
print(modulus)
print(exponentiation)
print(is_in_list)
print(is_not_in_list)

15
5
50
2.0
0
100000
True
True
```

Practice Question:

```
In [7]: sp_stations = [
    {
        "Region": "North",
        "Stations": ["A", "B", "C", "D"],
        "Profit": True,
        "Low Fuel Price": True,
        "User Satisfaction": True
    },
    {
        "Region": "South",
        "Stations": ["E", "F", "G", "H"],
        "Profit": True,
        "Low Fuel Price": False,
        "User Satisfaction": True
    },
    {
        "Region": "East",
        "Stations": ["I", "J", "K", "L"],
        "Profit": True,
        "Low Fuel Price": True,
        "User Satisfaction": False
    },
    {
        "Region": "West",
        "Stations": ["M", "N", "O", "P"],
        "Profit": False,
        "Low Fuel Price": True,
        "User Satisfaction": True
    }
]

for i in sp_stations:
    expansion = i["Profit"] and (i["Low Fuel Price"] or i["User Satisfaction"])
    if expansion:
        i["Stations"].append("S")

print(sp_stations)

[{'Region': 'North', 'Stations': ['A', 'B', 'C', 'D', 'S'], 'Profit': True, 'Low Fuel Price': True, 'User Satisfaction': True}, {'Region': 'South', 'Stations': ['E', 'F', 'G', 'H', 'S'], 'Profit': True, 'Low Fuel Price': False, 'User Satisfaction': True}, {'Region': 'East', 'Stations': ['I', 'J', 'K', 'L', 'S'], 'Profit': True, 'Low Fuel Price': True, 'User Satisfaction': False}, {'Region': 'West', 'Stations': ['M', 'N', 'O', 'P'], 'Profit': False, 'Low Fuel Price': True, 'User Satisfaction': True}]
```

- Zip Function
- Counter Function
- Enumerate Function
- String Interpolation
- Functions

```
In [13]: list1 = [1, 2, 3]
list2 = ['a', 'b', 'c', 'd']

def zip_function(a, b):
    zipped = zip(list1, list2)
    print(zipped)
    for item in zipped:
        print(item)

zip_function(list1, list2)

<zip object at 0x7f76d5700800>
(1, 'a')
(2, 'b')
(3, 'c')
```