Python_Practice_Day_2

December 29, 2021

1 Strings

[1]:
```python
# String is a collection of characters which is always placed inside "" or ''
```

[2]:
```python
name = "Harsha"
print(name)
print(type(name))
```

```
Harsha
<class 'str'>
```

[3]:
```python
name1 = 'Harsha'
print(name1)
print(type(name1))
```

```
Harsha
<class 'str'>
```

[4]:
```python
name2 = 'H'
print(name2)
print(type(name2))
```

```
H
<class 'str'>
```

[5]:
```python
# Index refers to a position in an ordered list. Python strings can be thought␣
 ↪of as lists of characters.
# Each character is given an index starting from 0.
```

[6]:
```python
name = "Harsha"
print(name[1])
```

```
a
```

[7]:
```python
name = "Intellipaat Training"
print(len(name))
```

```
20
```

[8]:
```python
name = "intellipaat training"
print(name.upper())
```

```
INTELLIPAAT TRAINING
```

[9]:
```python
name = "INTELLIPAAT TRAINING"
print(name.lower())
```

```
intellipaat training
```

[11]:
```python
name = "Harsha"
print(name.replace("H","h"))
```

```
harsha
```

2 String Slicing

[12]:
```python
name = "Sri Harsha"
print(name)
```

```
Sri Harsha
```

[13]:
```python
print(name[0])
```

```
S
```

[14]:
```python
print(name[0:3])
```

```
Sri
```

[15]:
```python
name = "Sri Harsha"
print(name[4:])
```

```
Harsha
```

[16]:
```python
# [starting index : ending index : skip value]
name = "Sri Harsha Akshintala"
print(len(name))
print(name[0:15])
```

```
21
Sri Harsha Aksh
```

[17]:
```python
print(name[0:21])
```

```
Sri Harsha Akshintala
```

[19]:
```python
print(name[0:21:2])
```

```
SiHrh khnaa
```

```
[20]: name = "Harsha"
      print(name[0:-1])
```

```
Harsh
```

```
[21]: print(name[-1])
```

```
a
```

```
[23]: print(name[::-1])
```

```
ahsraH
```

3 Python Operators

```
[24]: # We have following types of operators in Python \
      # Assignment Operators
      # Arithmetic Operators
      # Comparison Operators
      # Logical Operators
      # Identity Operators
```

```
[25]: # Assignment Operators:(=)
      num1 = 5
      # 5 is assigned to num1
      # In python, datatype is detected by interpreter, where as in other lang we␣
       ↪have to define datatype.
```

```
[26]: # Arithmetic Operators (+,_,*,/,//,%)
      num1 = 10
      num2 = 3
      print(num1 + num2)
      print(num1 - num2)
      print(num1 * num2)
      print(num1 / num2)
      print(num1 // num2) # // --> #Integer Division
      print(num1 % num2)
```

```
13
7
30
3.3333333333333335
3
1
```

```
[27]: # Comparison Operators (>,<,>=,<=,==,!=)
      # Result would be in True/False
```

[28]:
```
a = 50
b = 20
print(a > b)
print(a < b)
print(a == b)
print(a != b)
```

```
True
False
False
True
```

[29]:
```
# Logical Operator (and, or, not)
# and --> all condition has to be true
# or --> atleast one has to be true
# not --> Inverts
```

[30]:
```
a = 50
b = 20
print(a > b and b < a )
```

```
True
```

[31]:
```
print(a > b or b > a)
```

```
True
```

[32]:
```
print(not(a > b or b > a))
```

```
False
```

[36]:
```
# Identity Opeartor (is , is not)
a = 20
b = 10
print(a is b)
print(a is not b)
```

```
False
True
```

4 List

[ ]:
```
# List is a data structure in python which is mutable and ordered sequence of␣
↪elements.
# List allows duplicated elements.
# Denoted by []

marks = [40,35,49,33,22]
print(marks)
```

```
[2]: name = ["Harsha","Charan","Sinju",8]
     print(name)
```

```
['Harsha', 'Charan', 'Sinju', 8]
```

```
[3]: name[1] = "Pithani Charan"
     print(name)
```

```
['Harsha', 'Pithani Charan', 'Sinju', 8]
```

```
[4]: list1 = [1,"harsha",10.5,10+8j]
     print(list1)
```

```
[1, 'harsha', 10.5, (10+8j)]
```

```
[5]: list1[2] = 10+8j
     print(list1)
```

```
[1, 'harsha', (10+8j), (10+8j)]
```

5 Tuple

```
[40]: # Tuple is a collection of objects which are ordered and immutable.
      # Allows duplicate elements
      # Denoted by ()
```

```
[41]: name = ("Harsha","Charan","Sinju",8)
      print(name)
```

```
('Harsha', 'Charan', 'Sinju', 8)
```

```
[42]: print(type(name))
```

```
<class 'tuple'>
```

```
[43]: tup1 = ("Harsha","Charan","Sinju",8)
      tup1[1] = "Pithani"
      print(tup1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_10204/1637024170.py in <module>
      1 tup1 = ("Harsha","Charan","Sinju",8)
----> 2 tup1[1] = "Pithani"
      3 print(tup1)

TypeError: 'tuple' object does not support item assignment
```

```
[44]: tup1 = ("Harsha","Charan","Sinju",8)
      tup1(1) = "Pithani"
      print(tup1)
```

```
  File "C:\Users\harsh\AppData\Local\Temp/ipykernel_10204/4170756673.py", line
    tup1(1) = "Pithani"
    ^
SyntaxError: cannot assign to function call
```

```
[45]: tup1 = ("Harsha","Charan","Sinju",8)
      tup1[1] = "Pithani"
      print(tup1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_10204/1637024170.py in <module>
      1 tup1 = ("Harsha","Charan","Sinju",8)
----> 2 tup1[1] = "Pithani"
      3 print(tup1)

TypeError: 'tuple' object does not support item assignment
```

6 Dictionary

```
[55]: # Dictionary is also a data structurewhich stores key value Pairs.
      # does not allow duplicate keys.
```

```
[6]: dict1 = {"Harsha": 10000,"Charan":20000,"Sinju":25000}
     print(dict1)
```

```
{'Harsha': 10000, 'Charan': 20000, 'Sinju': 25000}
```

```
[11]: dict2 = {'Name':'Harsha','State':'Andhra','Country':'India'}
      print(dict2)
```

```
{'Name': 'Harsha', 'State': 'Andhra', 'Country': 'India'}
```

```
[14]: dict2 = {'Name':'Harsha','State':'Andhra','Country':'India'}
      print(dict2)
```

```
{'Name': 'A', 'State': 'Andhra', 'Country': 'India'}
```

```
[15]: dict2 = {'Name':'Harsha','State':'Andhra','Country':'India'}
      print(dict2)
```

```
{'Name': 'A', 'State': 'Andhra', 'Country': 'India'}
```

[16]:
```
dict2 = {'Name':'Harsha','State':'Andhra','Country':'India'}
print(dict2)
```

```
{'Name': 'Harsha', 'State': 'Andhra', 'Country': 'India'}
```

7 Set

[57]:
```
# Set is mutable i.e. we can make any changes in set. But elements are not␣
 ↪duplicated.
# Set is unordered.
```

[58]:
```
set1 = {1,1,2,2,3,3,4,4,5,5}
print(set1)
```

```
{1, 2, 3, 4, 5}
```

8 Conditional Statements

[46]:
```
# Here we would be dealing with "if, else, elif"
```

[47]:
```
num1 = 50
num2 = 100
if(num1 > num2):
    print("num1 is greater")
```

[48]:
```
num1 = 500
num2 = 100
if(num1 > num2):
    print("num1 is greater")
```

```
num1 is greater
```

[49]:
```
num1 = 50
num2 = 100
if(num1 > num2):
    print("num1 is greater")
else:
    print("num2 is greater")
```

```
num2 is greater
```

[52]:
```
marks = int(input("Enter the marks of a student"))
if(marks >= 80):
    print("Student got A grade")
elif(marks >= 60 and marks < 80):
    print("Student got B grade")
```

```python
elif(marks >= 40 and marks < 60):
    print("Student got C grade")
else:
    print("Student Failed")
```

```
Enter the marks of a student3
Student Failed
```

[ ]:
```python
#Task -- Create atleast 5 different scenarios to show conditional statements
#Revise each and every content for next one week
```