Datatypes and Its Conversions C--> Functional Programming C++ --> Both object oriented and functional prgramming Java--> Approx 100% Object oriented(Not Totally 100%) Python --> Both object oriented and functional programming In [1]: #What is Datatypes? Data + Types -->Represents the type of data that we are using inside a variable or program. Note: In python we need **not** to define the datatype explictly Internally PVM will automatically Considered the datatype while runtime. In [18]: #PVM --> PYTHON VIRTUAL MACHINE(INTERPRETOR/COMPILER) print(type(x)) print(id(x)) x="Pratyush" print(type(x)) <class 'int'> 2467019975248 <class 'str'> In []: type() --> to check the type of data id()--> return the address of the object In [2]: #How many datatypes we have in Python? Five types of Datatype we have **in** python: Numeric Datatype --> Int , Float , Complex Sequence Datatype --> String , List, Tuple Boolean Datatype --> Dictionary --> Set --> In [3]: #FlowChart of Datatypes **image** In [4]: # Integer Datatype: iNTERGER DATATYPE REPRESENTS THE INTEGRAL VALUE Two types of Integer: Positive integer --> 0,1,2,3,4,5,6,7,8,9,.... Negative Integer --> -1, -2, -3, -4, -5, -6, -7..... If we have any value without decimal point then that value is treated as integer #In python there is no limit for declaring an integer datatype/ Float datatype In [35]: type(x) int Out[35]: In [24]: x=200 type(x) int Out[24]: In [23]: x = -10type(x) Out[23]: int #Float Datatype:If we want to represent any value in decimal(point) format then we should go eith float datatype Float positive datatype: 1.2, 2.3..... Float Negative Datatype: -1.2, -9.8.... In [36]: x=10.98type(x) float Out[36]: In [30]: x=-10.989249323456 type(x) float Out[30]: In [32]: x = 0.0type(x) float Out[32]: In [6]: #Complex Number: A complex number is in of the form a+bj a--> Real part b--> Imaginary part j^2=-1 $j=(-1)^1/2$ In [40]: x=2+3j #real part : 2 #imaginary part : 3 type(x) complex Out[40]: In [38]: # Complex datatypes are used in scientific applications (Scientific Computing). Thatswhy nasa and drdo python **as** first priority Scipy--> Scientific Python Out[38]: In [7]: #Boolean Datatype: if we want to represnt datatype in boolean value format True --> 1 False -->0 In [41]: x=True type(x) bool Out[41]: In [42]: x=False type(x) bool Out[42]: In [46]: x=False+True-True+False-False+False Out[46]: In [8]: #String Datatype: String is a collection or a sequence of characters within single quotes or double quote. In python we don't have any datatype like char we can represnt char datatype as string in object in c++/c and Java Char='a' Short cut for running a cell : Shift+enter In [50]: x="True" type(x) str Out[50]: In [52]: x="10.5" type(x) Out[52]: str In [9]: #Standard Datatypes --> int , float , complex , boolean and string In []: In [10]: #Conversion of one datatype to another In []: These conversions are also known **as** typecasting if you want to convert any datatype into int for that we have int() In [59]: #converison float to int x=123.456y=int(x)print(type(y)) <class 'int'> In [57]: #Conversion complex number to int is not possible x=1+2j y=int(x)print(type(y)) Traceback (most recent call last) ~\AppData\Local\Temp/ipykernel_2716/723847106.py in <module> 1 #Conversion complex number to int 2 x**=1+2**j ----> 3 y=int(x) 4 print(type(y)) TypeError: can't convert complex to int In [60]: #Conversion of boolean to int is possible print(type(int(x))) <class 'int'> #Conversion of String with Integer inside quotes into int is possible print(type(int(x))) <class 'int'> In [63]: #Conversion of String with float inside quotes into int is not possible print(type(int(x))) Traceback (most recent call last) ~\AppData\Local\Temp/ipykernel_2716/1242230805.py in <module> 1 #Conversion of String with float inside quotes into int is not possible 2 x="10.5" ----> 3 print(type(int(x))) ValueError: invalid literal for int() with base 10: '10.5' In [64]: #Conversion of String with charcters inside quotes into int is not possible x="ten" print(type(int(x))) ValueError Traceback (most recent call last) ~\AppData\Local\Temp/ipykernel_2716/756630424.py in <module> 1 #Conversion of String with charcters inside quotes into int is not possible ----> 3 print(type(int(x))) ValueError: invalid literal for int() with base 10: 'ten' In [68]: #Float --#Convert any datatype into float we have float() #Conversion of int datatype into float is possible x=10 float(x) 10.0 Out[68]: In [67]: #Conversion of complex datatype into float is not possible x=10+10j type(float(x)) Traceback (most recent call last) ~\AppData\Local\Temp/ipykernel_2716/2970346466.py in <module> 1 #Conversion of complex datatype into float is not possible 2 x = 10 + 10j----> **3** type(float(x)) TypeError: can't convert complex to float In [69]: #Conversion of Boolean into float is possible x=True type(float(x)) float In [71]: #Conversion of String with float inside quotes into float is possible x="10.5" print(type(float(x))) <class 'float'> #Conversion of String with int inside quotes into float is possible x="10" print(type(float(x))) <class 'float'> In [73]: #Conversion of String with character inside quotes into float is not possible x="ten point 5" print(type(float(x))) ValueError Traceback (most recent call last) ~\AppData\Local\Temp/ipykernel_2716/1986004905.py in <module> 1 #Conversion of String with character inside quotes into float is not possible 2 x="ten point 5" ----> 3 print(type(float(x))) ValueError: could not convert string to float: 'ten point 5' Complex: For converting any datatype into complex we should use complex function In [74]: #Conversion of Int into complex is possible type(complex(x)) (10+0j) Out[74]: In [75]: #Conversion of float into complex is possible x=10.5type(complex(x)) complex Out[75]: In [77]: #Conversion of bool into complex is possible x=False complex(x) Out[77]: In [78]: #Conversion of String with character into complex is not possible x="Helloworld" type(complex(x)) ______ Traceback (most recent call last) ~\AppData\Local\Temp/ipykernel_2716/1991498921.py in <module> 1 #Conversion of String with character into complex is not possible 2 x="Helloworld" ----> 3 type(complex(x)) ValueError: complex() arg is a malformed string In [79]: #Conversion of string within Int inside quotes into complex is possible type(complex(x)) ${\tt complex}$ Out[79]: In [81]: #Conversion of string within float inside quotes into complex is possible x="10.5" complex(x) (10.5+0j) Out[81]: Boolean conversion 0--> False Other than 0 or "" we will get true In [82]: x=bool(0.0)-->0False Out[82]: In [83]: x=bool("") False Out[83]: In [93]: x=bool(0+0j) False Out[93]: In [94]: x=bool(0j) False Out[94]: In [96]: x=bool(-2)True Out[96]: In [98]: x=bool("0") True Out[98]: In [99]: #String for str() x=10.5str(x) '10.5' Out[99]: In [100... x=True str(x) 'True' Out[100. In [103.. x=10+2j str(x) '(10+2j)' Out[103.. x=10,20,"ele",500 In [109... #List--> it is a collection of dissimilar data. We can change the element of a list(Mutable) #lists is mutable --> we can change the element of the list list is also an ordered collection x=[10,20,"Hello",10.5,20] x[0]=200x[0] 200 Out[109... In []: In []: #Tuple--> it is also a collection of dissimilar data. We cannot change the element of a tuple (we cannot change the lement of the tuple that means it is immuatble). Tuple is a ordered collection of element Tuple index is important In [108... x=(10,20,"Hello",10.5,20) Out[108... In []: #Set --> Set is also a collection of element but duplicates are not allowed in set.And index is not important In [111... $x=\{10, 20, 30, 40, 50, 5, 102, 23404, 10, 20\}$ {5, 10, 20, 30, 40, 50, 102, 23404} Out[111... In []: #Dictionary --> it is also a collection of element in form of key and value pairs. Index is not important conclusion we can access the value woth the help of key Key will always be unique In [114... x={"Search":"Google", 2:"paytm", 3:"Yahoo"} x[2] 'paytm' Out[114... In []: List is Mutable --> we can change the values Tuples are immutable --> we cannot change the values Set is Mutable -->.we can change the values of set Dictionary are mutable --> you can change the values of dictionary In []: Conversion int(int()) --> Float int(int()) --> Boolean int(int())- string with integr value Float(float())--> Int Float(float())boolean Float (float())--> string with either integer or float value boolean(bool()) --> We can convert any datatype into boolean if the datatype has value as 0 or " " the the output will always be False else you will get output as true String str()--> Any datatype into string you can convert In [121... x=int(input()) y=int(input()) print(type(x)) print(type(y)) x**+**y 10 20 <class 'int'> <class 'int'> Out[121... 30 In [13]: In [14]: