**Variable Types**

## Assigning Values to Variables

#!/usr/bin/python3

counter = 100 # An integer assignment

miles = 1000.0 # A floating point

name = "John" # A string

print (counter)

print (miles)

print (name)

## Multiple Assignment

a = b = c = 1

a, b, c = 1, 2, "john"

## Python Strings

#!/usr/bin/python3

str = 'Hello World!'

print (str) # Prints complete string

print (str[0]) # Prints first character of the string

print (str[2:5]) # Prints characters starting from 3rd to 5th

print (str[2:]) # Prints string starting from 3rd character

print (str \* 2) # Prints string two times

print (str + "TEST") # Prints concatenated string

## Python Lists

#!/usr/bin/python3

list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]

tinylist = [123, 'john']

print (list) # Prints complete list

print (list[0]) # Prints first element of the list

print (list[1:3]) # Prints elements starting from 2nd till 3rd

print (list[2:]) # Prints elements starting from 3rd element

print (tinylist \* 2) # Prints list two times

print (list + tinylist) # Prints concatenated lists

## Python Tuples

#!/usr/bin/python3

tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )

tinytuple = (123, 'john')

print (tuple) # Prints complete tuple

print (tuple[0]) # Prints first element of the tuple

print (tuple[1:3]) # Prints elements starting from 2nd till 3rd

print (tuple[2:]) # Prints elements starting from 3rd element

print (tinytuple \* 2) # Prints tuple two times

print (tuple + tinytuple) # Prints concatenated tuple

#!/usr/bin/python3

tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )

list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]

tuple[2] = 1000 # Invalid syntax with tuple//không hợp lệ

list[2] = 1000 # Valid syntax with list//hợp lệ

## Python Dictionary

#!/usr/bin/python3

dict = {}

dict['one'] = "This is one"

dict[2] = "This is two"

tinydict = {'name': 'john','code':6734, 'dept': 'sales'}

print (dict['one']) # Prints value for 'one' key

print (dict[2]) # Prints value for 2 key

print (tinydict) # Prints complete dictionary

print (tinydict.keys()) # Prints all the keys

print (tinydict.values()) # Prints all the values

## Data Type Conversion

|  |  |
| --- | --- |
| **Sr.No.** | **Function & Description** |
| 1 | **int(x [,base])**  Converts x to an integer. The base specifies the base if x is a string. |
| 2 | **float(x)**  Converts x to a floating-point number. |
| 3 | **complex(real [,imag])**  Creates a complex number. |
| 4 | **str(x)**  Converts object x to a string representation. |
| 5 | **repr(x)**  Converts object x to an expression string. |
| 6 | **eval(str)**  Evaluates a string and returns an object. |
| 7 | **tuple(s)**  Converts s to a tuple. |
| 8 | **list(s)**  Converts s to a list. |
| 9 | **set(s)**  Converts s to a set. |
| 10 | **dict(d)**  Creates a dictionary. d must be a sequence of (key,value) tuples. |
| 11 | **frozenset(s)**  Converts s to a frozen set. |
| 12 | **chr(x)**  Converts an integer to a character. |
| 13 | **unichr(x)**  Converts an integer to a Unicode character. |
| 14 | **ord(x)**  Converts a single character to its integer value. |
| 15 | **hex(x)**  Converts an integer to a hexadecimal string. |
| 16 | **oct(x)**  Converts an integer to an octal string. |