# Python Workouts:

## How to Declare Variable:

|  |
| --- |
| **For int:**  X=100 🡪 python will automatically consider this as int  Y = ‘String’ -> based on the single quotes, python will consider this as string datatype  We can cast the data types as below in python  X= int(10) |

## Code 1:

Create 2 variables with **x** as 100 & **y** as 10 respectively and find the Multiplication and division of both and store in some val as **z** and **z1**.

|  |
| --- |
| x = 100;  y =10;  z = x\*y;  print (z)  z1 = x/y;  print (z1) |

## Code 2:

Create **a** as 2000 and find the division of **a** by **y** (created in step 1) and reassign **a** with the divided result (200).

|  |
| --- |
| a = 2000;  a=2000/10;  print (a) |

## Code 3:

Prove Python is Dynamically Typed Language: Create x:int=100, then assign the **x** to **y**, but the datatype of **y** has to be of type string

|  |
| --- |
| x:int = 100  y = str(x);  print (x);  print (y); |

## Code 5:

Prove Python is Strongly Typed Language

|  |
| --- |
| x = 7;  y = 'gokul'  z = 'formula1'  a = 7  print (y+' '+z); 🡪 this will work output will be gokul formula1  print (x+a) 🡪 this will work output will be 14  print (y+x+z); 🡪 this will throw error as datatype between the three variable is different |

## Case 6, Case 7 & Case 8:

6. Create variables a,b,c,d assigned with 10,20,30,40 respectively

7. Prove Python variables are case sensitive

8. Prove variable name can’t start with numbers or cannot contains special character other than \_

|  |
| --- |
| a = 10;  b = 20;  c = 30;  d = 40;  1a = 50; 🡪 it will throw error  print (b); 🡪 this will work  print (a); 🡪 this will work  print (A); 🡪 this wont work |

## Case 9:

Show some examples of when do we use single, double and triple (single/double) quotes

|  |
| --- |
| a = 'hello'  b = "world its's python"  c = "'language'"  d = 'program'  print (a+ ' '+b+' '+c+' '+d); |

## Case 10:

Show an examples to use arithmetic, comparison, relational and logical operators.

|  |
| --- |
| **Arithmetic operator:**  x = 100;  y =60;  print (x+y);  print (x-y);  print (x\*y);  print (x/y);  print (x%y);  **Comparison Operator/Relational operator:**  x = 100;  y =60;  if x==y:  print ("x and y is equal")  elif x!=y:  print ('x is not equal to y')  elif x>y:  print ("x is greater than y")  elif x<y:  print ('x is smaller than y')  else:  print ("x and y is not equal")  **Logical Operator:**  x = 100;  y =60;  z = 70;  a = 110;  if (x>y and x>z) or x>a:  print ("x and y is equal")  else:  print ("x and y is not equal") |

## Case 11:

Write a program to find the greatest of 3 numbers

|  |
| --- |
| x = 10  y = 25  z = 50  if x>y and x>z:  print ("x is greater number")  elif y>x and y>z:  print ("y is greater number")  else:  print ("z is greater number")  **with number as input:**  x = int(**input**("Enter First Number:"))  y = int(**input**("Enter Second Number:"))  z = int(**input**("Enter Third Number:"))  if x>y and x>z:  print ( str(x) + "is greater number")  elif y>x and y>z:  print (str(y) + "is greater number")  else:  print (str(z)+ "is greater number") |

## Case 12:

Write a single program to find the given number is even or whether it is negative

|  |
| --- |
| x = int(input("Enter Number:"))  if x%2==0 and x >0:  print (str(x)+" is even and positive number")  elif x%2==0 and x<0:  print (str(x)+" is even and negative number")  elif x%2!=0 and x >0:  print (str(x)+" is odd and positive number")  else:  print (str(x)+" number is neither even nor positive") |

## Case 13:

Write a nested if then else to print the course fees - check if student choosing bigdata, then fees is 25000, if student choosing spark then fees is 15000, if the student choosing datascience then check if machinelearning then 25000 or if deep learning then 45000 otherwise if both then 25000+25000.

|  |
| --- |
| x = str(input("Enter Course:"))  y = str(input("Enter the Secondary Course:"))  if x=='bigdata':  print (x +" course fees is 25,000")  elif x == 'spark':  print (x + " course fees is 15,000")  if (x == 'datascience'):  if y=='machinelearning':  print (y + " course if 25,000")  elif (y=='deeplearning'):  print (y + " course if 45,000")  elif(y != ''):  print (y + " course is 50,000")  else:  print ("Entered Course is invalid")  elif ((x!='bigdata' and y!='') or (x!='spark'and y!='')):  print ("Entered Course if invalid x") |

## Case 14:

Check whether the given string is palindrome or not (try to use some function like reverse). For eg: x="madam" and y="madam", if x matches with y then print as "palindrome" else "not a  palindrome".

|  |
| --- |
| x=input('Enter the Name:')  y = x [::-1]  if x==y:  print ("Given word is palindrome")  else:  print ("Given word is not palindrome") |

## Case 15:

Check whether the x=100 is an integer or string. (try to use some functions like str or upper function etc to execute this use case) or use isinstanceof(variablename,datatype) function.

|  |
| --- |
| **Approach 1:**  x = int(input("Enter the Input:"))  y = str(input("Enter the Input:"))  print ("Entered Input is", type(x))  print ("Entered Input is", type(y)) |

## Case 16:

Create a list with a range of 10 values starting from 2 to 11 and prove mutability by updating the 3rd element with 100 and prove resizable properties by adding 100 in the 5th position

|  |
| --- |
| list = [2,3,4,5,6,7,8,9,10,11]  print (list)  print (list.count(10))  **to Insert a value in the specific position:**  list.insert(4,100)  **to update a value in the list**  print (list[3])  list[3]= 70  print (list)  To append the value at the last  List.append(77) |

## Case 17:

Create a tuple of 2 fields eg. ("Inceptez","Technologies","Pvt","Ltd"), prove immutability and non resizable nature, access the 2nd and 4th fields and store in another tuple.

|  |
| --- |
| thistuple = ['inceptez','technologies','private','ltd']  print (thistuple[1],thistuple[3])  tuple = thistuple[1],thistuple[3]  print (tuple)  **To insert a element in tuple:**  thistuple = thistuple + ['gokul']  print (thistuple)  thistuple = thistuple[:2] + ['f1']  print (thistuple) |

## Case 19:

Create a list of tuple as given below and delete all duplicate tuples of the list  lst=[("Inceptez","Technologies"),("Apple","Incorporation"),("Inceptez","Technologies"),("Inceptez","Technologies")]

|  |
| --- |
| thistuple = [("Inceptez","Technologies"),("Apple","Incorporation"),("Inceptez","Technologies"),("Inceptez","Technologies")]  thistuple1 = **set**(thistuple)  print (thistuple1)  **set ()** operation will remove the duplicates from the list/tuple |

## Case 20:

Append ("Intel","Corp") in the above de duplicated list

|  |
| --- |
| thistuple = [("Inceptez","Technologies"),("Apple","Incorporation"),("Inceptez","Technologies"),("Inceptez","Technologies")]  thistuple1 = set(thistuple)  thistuple1.**add** (("Intel","Corp"))  print (thistuple1) |

## Case 22:

Create a list of values lst=[10,20,40,30,20], find the first, last values of the list, sort the list in ascending order, sort in descending order, print the minumum and maximum values of the descending sorted list, find the sum of all elements in the list, remove the number 30 and 20 from the list.

|  |
| --- |
| lst = [10,20,30,40,30,20]  lst\_len = len(lst)-1  print (lst\_len)  print (lst[0])  print (lst[lst\_len])  **lst.sort()**  **print (lst)**  **lst.sort(reverse=True)**  **print (lst)**  print (sum(lst))  unwanted\_list = [20,30]  lst1 = [ele for ele in lst if ele not in unwanted\_list]  lst2 = []  print (lst1)  for ele in lst:  if ele not in unwanted\_list:  lst2.append(ele)  print (lst2) |

## Case 23:

Do the above same (step 25) operation in the tuple of elements tup=(10,20,40,30,20)

|  |
| --- |
| lst = (10,20,30,40,30,20)  lst\_len = len(lst)-1  print (lst\_len)  print (lst[0])  print (lst[lst\_len])  **print (sorted(lst))**  **print (sorted(lst, reverse=True))**  print (sum(lst))  unwanted\_list = (20,30)  lst1 = [ele for ele in lst if ele not in unwanted\_list]  lst2 = []  print (lst1)  for ele in lst:  if ele not in unwanted\_list:  lst2.append(ele)  print (lst2) |

## Case 24:

Convert the string to list from str1="Inceptez Technologies Pvt Ltd" to lst\_str1=['Inceptez', 'Technologies', 'Pvt', 'Ltd']

|  |
| --- |
| str1="Inceptez Technologies Pvt Ltd"  lst = list(str1.**split**(" "))  print (lst) |

## Case 26:

Write a program using for loop to print even numbers and odd numbers in the below range of data (generate using range function) [5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20] output should be with even as 6,8,10,12,14,16,18,20 and odd as 5,7,9,11,13,15,17,19.

If we want to print the values from 0 to 10 or starting to one number to mentioned number we can use range function

**Syntax:**

|  |
| --- |
| For i in range (0,10):  Print (i); |

|  |
| --- |
| l = []  g = []  for i in range(2, 21):  if (i%2==0):  l.append(i)  else:  g.append(i)    print (str(l) + "is Even Numbers")  print (str(g) + "is odd Numbers") |

## Case 27:

Write a while loop to loop from 0 till 21 with the increment of 3, the result should be exactly 3,6,9,12,15,18 and store this result in a list

|  |
| --- |
| l = []  i= 1  while (i<21):  if(i%3==0):  l.append(i)  i+=1  print (l) |

## Case 28:

Write a for or while loop to print the cube of 4, result should be 4\*4\*4=64 (initiate some variable outside the loop with 4 and loop through 3 times to achieve the result)

|  |
| --- |
| i = 4  for x in str(i):  x = i\*i\*i  print (x) |

## Case 29:

Create a list as sal\_lst=[10000,20000,30000,10000,15000], loop through the list and add 1000 bonus to the salary and store in another list sal\_bonus\_lst=[11000,21000,31000,11000,16000]

then store the bonus applied salary in another list where sal>11000

|  |
| --- |
| l = [10000,20000,30000,10000,15000]  sal\_bonus\_list = []  for i in range(len(l)):  l[i] += 1000  print (l)  for j in range(len(l)):  if (l[j]>15000):  sal\_bonus\_list.append(l[j])  print (sal\_bonus\_list) |

## Case 30:

Write a do while loop to print “Inceptez technologies” n number of times as per the input you get from the user. Minimum it has to be printed at least one time regardless of the user input.

|  |
| --- |
| a = 'inceptez technologies'  i = int(input("Enter Number of Times: "))  while True:  print (a)  i = i-1  if(i<=0):  break |

## Case 31:

From the given list of list of elements produce the following output using nested for loop

lst1=[[10,20],[30,40,50],[60,70,80]], calculate the sum of all number, calculate the min value and the max value of all the elements in the lst1.

|  |
| --- |
| lst = [[10,20],[30,40,50],[60,70,80]]  lst1 = []  for i in lst:  for j in i:  lst1.append(j)  print (lst1)  print (sum(lst1))  print (min(lst1))  print (max(lst1)) |

## Case 32:

Create a looping construct to create 3 tables upto 10 values. Output should be like this…

1 x 3 = 3

2 x 3 = 6

3 x 3 = 9

.

.

10 x 3 = 30

|  |
| --- |
| x = int(input("Enter the Number:"))  y = int(input("Enter the Number to be Multiplied for:"))  for i in range(1,x):  print (y, 'X' ,i, '=' ,y\*i) |

https://www.programiz.com/python-programming/online-compiler/