WORKSHEET 2 PYTHON

1. Which of the following is not a core datatype in python?

   A) list
   B) struct
   C) tuple
   D) set

   Ans: B) struct

2. Which of the following is an invalid variable name in python?

   A) _init_
   B) no_1
   C) 1_no
   D) _1

   Ans: C) 1_no

3. Which one of the following is a keyword in python?

   A) in
   B) _init_
   C) on
   D) foo

   Ans: A) in

4. In which of the following manner are the operators of the same precedence executed in python?

   A) Left to Right
   B) BODMAS
   C) Right to Left
   D) None of these

   Ans: A) Left to Right

5. Arrange the following in decreasing order of the precedence when they appear in an expression in python? i) Multiplication ii) Division iii) Exponential iv) Parentheses

   A) iii – iv – ii – i
   B) iii – iv – i – ii
   C) iv – iii – ii – i

D) iii – ii – i – iv

Ans: C) iv – iii – ii – i

6. (28//6)**3/3%3 =?

A) 7.1111…
B) 0
C) 0.3333…
D) 1

Ans: C) 0.3333…

7. a = input("Enter an integer"). What will be the data type of a?

A) int
B) str
C) float
D) double

Ans: B) str

8. Which of the following statements are correct?

A) Division and multiplication have same precedence in python
B) Python's operators' precedence is based on PEDMAS
C) Python's operators' precedence is based on VBODMAS
D) In case of operators' having the same precedence, the one on the left side is executed first.

Ans: A) Division and multiplication have same precedence in python
B) Python's operators' precedence is based on PEDMAS
D) In case of operators' having the same precedence, the one on the left side is executed first

9. Which of the following is(are) valid statement(s) in python?

A) abc = 1,000,000
B) a b c = 1000 2000 3000
C) a,b,c = 1000, 2000, 3000
D) a_b_c = 1,000,000

Ans: A) abc = 1,000,000
C) a,b,c = 1000, 2000, 3000
D) a_b_c = 1,000,000

10. Which of the following is not equal to x16 in python?

A) x**4**4
B) x**16
C) x^16
D) (x**4)**4

Ans: A) x4**4**
C) x^16

11. Differentiate between a list, tuple, set and dictionary

List: Lists are one of the most commonly used data structures provided by python; they are a collection of iterable, mutable and ordered data. They can contain duplicate data.

Tuple: Tuples are similar to lists. This collection also has iterable, ordered, and can contain repetitive data, just like lists. But unlike lists, tuples are immutable.

Set: Set is another data structure that holds a collection of unordered, iterable and mutable data. But it only contains unique elements.

Dictionary: Dictionary refers to a collection (unordered) of various data types. It contains key: value pairs. The key value in a dictionary is present to make it comparatively more optimized.

Difference Between List, Tuple, Set and Dictionary

Parameters	List	Tuple	Set	Dictionary
Mutable	It is mutable	It is immutable, we can't make any changes to the tuple	It is mutable	It is mutable
Ordered	It is ordered	It is ordered	It is ordered	It is ordered
Indexing	Indexing is possible	Indexing is possible	Indexing is not possible	Indexing is possible
Duplicate Data	It allows duplicate elements	It allows duplicate elements	It does not allow any duplicate elements	It does not allow any duplicate elements

12. Are strings mutable in python? Suppose you have a string “I+Love+Python”, write a small code to replace ‘+’ with space in python

Strings are immutable in python

code to replace ‘+’ with space in python

n = ("I+Love+Python")
n.replace("+"," ")

 # Output: 'I Love Python'

```
In [42]:  1 n = ("I+Love+Python")

In [43]:  1 n.replace("+"," ")
Out[43]: 'I Love Python'
```

13. What does the function **ord()** do in python? Explain with an example. Also, write down the function for getting the data type of a variable in python.

ord() - Return the Unicode code point for a one-character string.
The ord() function returns an integer representing the Unicode code point of the character when an argument is a Unicode object, or the value of the byte when the argument is an 8-bit string.

Example1:
a = ord('R')
print(a)
 # Output: 82

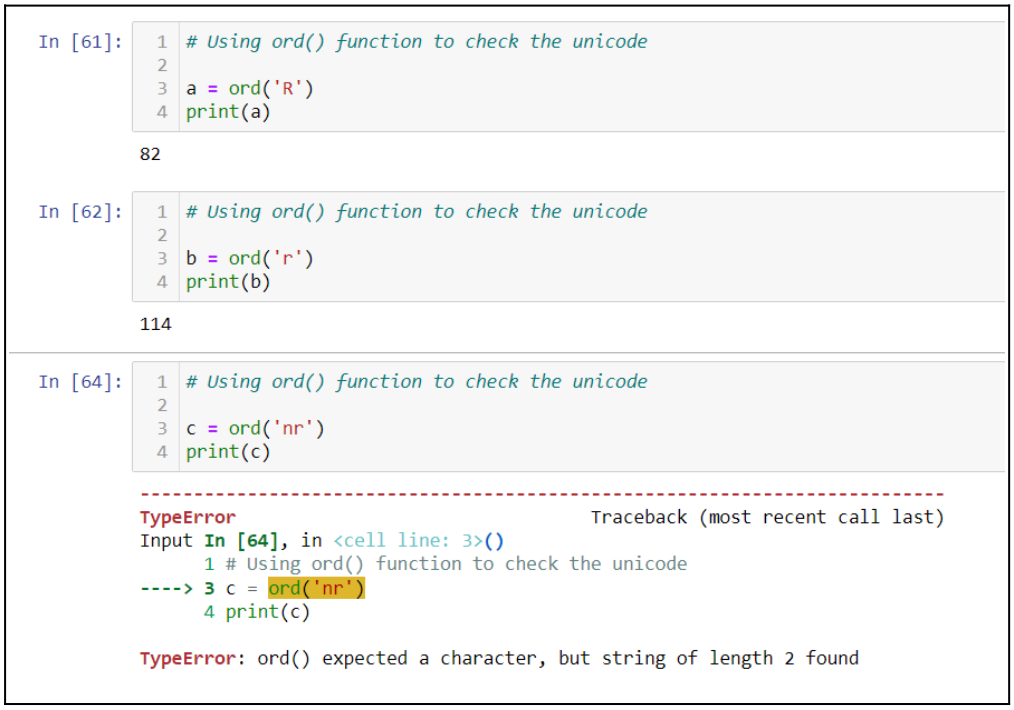
Example2:
b = ord('r')
print(b)
 # Output: 114

Example3:
c = ord('nr')
print(c)

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [64], in <cell line: 3>()
      1 # Using ord() function to check the unicode
----> 3 c = ord('nr')
      4 print(c)

TypeError: ord() expected a character, but string of length 2 found
```

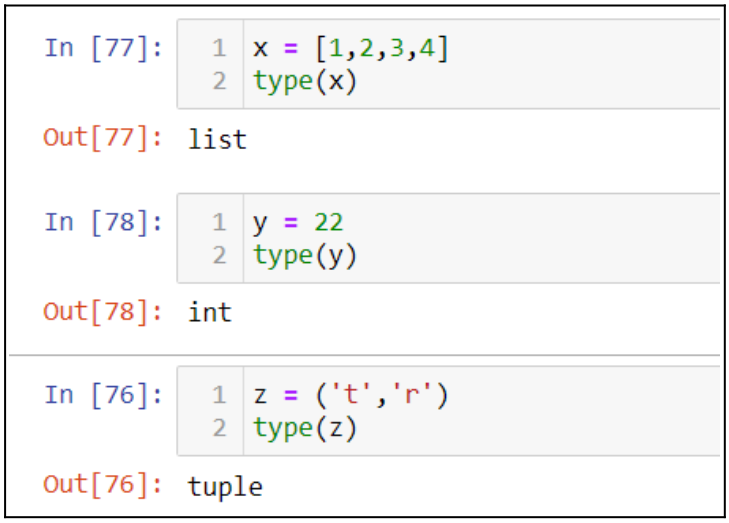
In above examples, we can see the Unicode returned for the string characters, python is case sensitive as we can see in example 1 and 2 Unicode are different.
For example 3 we are getting error as the ord() allows string of length 1.



function for getting the data type of a variable in python.

type() - function will return the type of variable whether it is int, str, float or list based on the variable

Example:



14. Write a python program to solve a quadratic equation of the form ax^2+bx+c=0. Where a, b and c are to be taken by user input. Handle the erroneous input, such as ‘a’ should not be equal to 0.

```
# Python program to find roots of quadratic equation

import math

def equationtroots(a,b,c):              # function for finding roots
    dis = b * b - 4 * a * c             # calculating discriminant using formula
    sqrtval = math.sqrt(abs(dis))

    if dis > 0:                         # checking condition for discriminant
        print(" real and different roots ")
        print((-b + sqrtval)/(2 * a))
        print((-b - sqrtval)/(2 * a))

    elif dis == 0:
        print(" real and same roots")
        print(-b / (2 * a))

    else:                               # when discriminant is less than 0
        print("Complex Roots")
        print(- b / (2 * a), " + i", sqrtval)
        print(- b / (2 * a), " - i", sqrtval)

# input
a = 2
b = 5
c = 1

# If a is 0, then incorrect equation
if a == 0:
        print("Input correct quadratic equation")
else:
    equationroots(a,b,c)
```

 # Output

real and different roots
-0.21922359359558485
-2.2807764064044154

In [123]:

```
# Python program to find roots of quadratic equation

import math

def equationtroots(a,b,c):                  # function for finding roots
    dis = b * b - 4 * a * c                 # calculating discriminant using formula
    sqrtval = math.sqrt(abs(dis))

    if dis > 0:                             # checking condition for discriminant
        print(" real and different roots ")
        print((-b + sqrtval)/(2 * a))
        print((-b - sqrtval)/(2 * a))

    elif dis == 0:
        print(" real and same roots")
        print(-b / (2 * a))

    else:                                   # when discriminant is less than 0
        print("Complex Roots")
        print(- b / (2 * a), " + i", sqrtval)
        print(- b / (2 * a), " - i", sqrtval)

# input
a = 2
b = 5
c = 1

# If a is 0, then incorrect equation
if a == 0:
        print("Input correct quadratic equation")
else:
    equationroots(a,b,c)
```

```
 real and different roots
-0.21922359359558485
-2.2807764064044154
```

15. Write a python program to find the sum of first ‘n’ natural numbers without using any loop. Ask users to input the value of ‘n’

```
def naturalnumbers(n):
    return n*(n+1)/2
n = int(input("Enter n = "))
naturalnumbers(n)
```

 # Output Enter n = 25
325.0

In [126]:

```
# finding sum of first n natural numbers

def naturalnumbers(n):
    return n*(n+1)/2
n = int(input("Enter n = "))
naturalnumbers(n)
```

```
Enter n = 25
```

Out[126]:

```
325.0
```