REVIEW OF PYTHON
VIVA QUESTIONS

1. **What does IDLE stand for? What is Python IDLE?**

   IDLE is an acronym of ***Integrated Development Environment*** and is the standard, most popular Python development environment.

   To run a program, we basically need an editor to write it, an interpreter/compiler to execute it and a debugger to catch and remove the errors. Python IDLE provides all these tools as a bundle. It lets edit, run, browse and debug Python Programs from a single interface and makes it easy to write programs.

2. **What is the difference between Interactive mode and Script mode?**

   Python IDLE can be used in two modes: Interactive mode and Script mode. **Python shell** is an **interactive interpreter**. **Python editor** allows us to work in **script mode** i.e. we can create and edit python source file.

3. **What are tokens?**

   Tokens- Smallest individual unit of a program is called token. Example keywords, identifiers, literals, operators and punctuators.

4. **What are Keywords?**

   They are the words used by Python interpreter to recognize the structure of program. As these words have specific meaning for interpreter, they cannot be used for any other purpose.

   ['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

   NOTE: All these keywords are in small alphabets, except for ***False, None, True***, which are starting with capital alphabets.

5. **What are identifiers and literals?**

   IDENTIFIERS: These are the names given to identify any memory block, program unit or program objects, function, or module. Examples of identifiers are: num, roll_no, name etc.

LITERALS: A fixed numeric or non-numeric value is called a literal. Examples of literals are: 502, -178.76, "Rajan" etc.

6. What is a variable?

Variables are like containers which are used to store the values for some input, intermediate result of calculations or the final result of a mathematical operation in computer memory.

7. What are the naming conventions of identifiers/variables/literals.

- Variable names are ***case sensitive***. For eg. num and NUM are treated as two different variable names.
- ***Keywords*** or words, which have special meaning, should ***not*** be used as the variable names.
- Variable names should be ***short*** and ***meaningful***.
- All variable names must begin with a ***letter*** or an ***underscore (_).***
- After the first initial letter, variable names may contain ***letters*** and ***digits*** (0 to 9) and an ***underscore*** (_), but no spaces or special characters are allowed.

Examples of valid variable names: sum, marks1, first_name, _money

Examples of invalid variables names: marks%, 12grade, class, last-name

8. What do you understand by the terms L-value and R-value?

L-value is a value which has an address. Thus, all variables are l-values since variables have addresses. As the name suggests l-value appears on left hand side but it can appear on right hand side of an assignment operator(=). l-value often represents as identifier.

R-value refers to data value that is stored at some address in memory. An **r-value** is a value which is assigned to an l-value. An r-value appears on right but not on left hand side of an assignment operator(=).

9. What is the default separator and terminator in print() function in Python? How can we change it?

By default, print uses a single space as a separator and a \n as a terminator (appears at the end of the string).
We can use the parameters ***sep*** and ***end*** to change the separator and terminator respectively.

10. Compare and contrast list, tuple and dictionary.

List	Tuple	Dictionary

A list is an ordered collection of comma-separated values (items) within square brackets.	A tuple is an ordered collection of commas separated values. The comma separated values can be enclosed in parenthesis but parenthesis are not mandatory.	A dictionary is an unordered collection of items where each item is a key: value pair. We can also refer to a dictionary as a mapping between a set of keys/indices and a set of values. The entire dictionary is enclosed in curly braces.
it is mutable	it is immutable.	it is mutable
list2=[“Raman”, 100, 200, 300, “Ashwin”]	**tup1=(‘Sunday’, ‘Monday’, 10, 20)**	**dict1={'R':'RAINY' , 'S':'SUMMER', 'W':'WINTER' , 'A':'AUTUMN'}**

11. What is the difference between mutable and immutable data types?

Mutable data types	immutable data types
Variables of mutable data types can be changed after creation and assignment of values.	Variables of immutable data types can not be changed or modified once they are created and assigned the value. If an attempt is made to change the value of variable of immutable datatype, the new is stored in some other memory location and the variable starts pointing to that location.
Some of the **mutable** data types in Python are **list and dictionary**	some of the **immutable** data types are **int, float, decimal, bool, string and tuple**.

12. What are the comments? Declare a single line comment and a multi-line comment.

Comments are the line that compiler ignores to compile or execute. There are two types of comments in Python

- **Single line comment**: This type of comment deactivates only that line where comment is applied. Single line comments are applied with the help of " #". For e.g.

```
# This program calculates the sum of two numbers
>>>n1=10
>>>n2=20
>>>sum= n1 + n2    # sum of numbers 10 and 20
>>>print ("sum=", sum)
```

- **Multi line Comment**: This Type of comment deactivates group of lines when applied. This type of comments is applied with the help of **triple quoted string**.

```
'''This is a
     multiline comment'''
```

or

```
"""This is a
     multiline comment"""
```

13. What are the different types of errors in a Python program?

SYNTAX ERROR: An error in the syntax of writing code that does not conform to the syntax of the programming language is called a syntax error.

LOGICAL ERROR: It is an error in a program's source code that results in incorrect or unexpected result. It is a type of runtime error that may simply produce the wrong output or may cause a program to crash while running.

RUN TIME ERROR: A runtime error is an error that occurs during execution of the program and causes abnormal termination of program.

14. Write functions for the following:

To convert string to int	int()
To convert string to float	float()
To convert numeric data to string	str()

15. What are operators?

Operators performs some action on data. Various types of operators are:

- **Arithmetic** (+,-,*,/,%)
- **Assignment Operator** (=)

- **Relational/comparison** (<,>, <=,>=,==,!=)
- **Logical** (AND, OR, NOT)
- **Identity Operators** (is, is not)
- **Membership Operators** (in, not in)

16.What is difference between /, //, %?

/	//	%
Divides two operands and gives quotient	Integer division	Divides two operands and gives remainder
10/5=2 10/3.0 = 3.3 10.0/3=3.3 10/3=3	5//2=2 5.0//2=2.0	10%5=0

17. What is the difference between * and **?

*	**
Multiplies two operands	Exponentiation
10*5=50	2**4= 16

18.What is the difference between = and ==?

=	==
Assigns the value	Checks if the value of left operand is equal to the value of right operand, if yes then condition becomes true.
x=50	15= = 15, true 16 = = 15, false

19.What is the difference between concatenation and repetition?

Concatenation (+)	**Repetition (*)**
Joins the objects (strings, lists etc.) on either side of the operator	Concatenates multiple copies of the same object (strings, lists etc.) given number of times

>>> **str1="Tech"**
>>> **str2="Era"**
>>> **str1+str2**
'TechEra'
 	
>>> **str3=str2+str1**
	>>> **str1*2**
	'TechTech'

>>> **print(str3)** **EraTech**	

20. What is the difference between is and in operators?

is, is not - **Identity Operators**	in, not in - **Membership Operators**
Identity operators are used to compare two variables whether they are of same type with the same memory location.	Membership operators are used to validate the membership of a value. It tests for membership in a sequence, such as strings, lists, or tuples.
The ***is*** operator evaluates to true if the variables on either side of the operator point to the same memory location and false otherwise. **>>>x = 5** **>>>type(x) is int** **True**	The ***in*** operator is used to check if a value exists in a sequence or not. Evaluates to true if it finds a variable in the specified sequence and false otherwise. **>>>x = [1,2,3,4,5]** **>>>3 in x** **True**
The ***is not*** operator evaluates to true if the variables on either side of the operator point to the different memory location and false otherwise. **>>>x = 5** **>>>type(x) is not int** **False**	The ***not in*** operator evaluates to true if it does not find a variable in the specified sequence and false otherwise. **>>>x = [1,2,3,4,5]** **>>>7 not in x** **True**

21. What is the use of indentation in Python?

Indentation in Python is used to form block of statements also known as suite. It is required for indicating what block of code, a statement belongs to. Indentation is typically used in flow of control statements (like if, for etc.), function, class etc. Although we can give any number of spaces for indentation but all the statements in a block must have the same indent.

Unnecessary indentation results in syntax error and incorrect indentation results in logical error.

22. What is the difference between indexing and slicing?

Indexing is used to retrieve an element from an ordered data type like string, list, tuple etc. Slicing is used to retrieve a subset of values from an ordered data type. A slice of a list is basically its sub-list.

23. What is the difference among insert, append and extend?

insert()	**append()**	**extend()**

It is used to insert data at the given index position.	It is used to add an element at the end of the list.	is used to add one or more element at the end of the list.
z1=[121, "RAMESH", 890, 453.90] **z1.insert(1, "KIRTI")** **OUTPUT** **[121, 'KABIR', 'RAMESH', 890, 453.9]**	**list1=[100, 200, 300, 400, 500]** **list2=["Raman", 100, "Ashwin"]** **list1.extend(list2)** **print(list1)** **OUTPUT** **[100, 200, 300, 400, 500, 'Raman', 100, 'Ashwin']** here list2 is added at the end of the list list1.	**z1=[121, "RAMESH", 890, 453.90]** **z1.insert(1, "KIRTI")** **z1.append(10)** **print(z1)** **OUTPUT** **[121, 'KABIR', 'RAMESH', 890, 453.9,10]** here list2 is added at the end of the list list1.

24. What is the difference among del, remove or pop methods to delete elements from a List?

pop()	**del**	**remove()**
It removes the element from the specified index, and also return the element which was removed. The last element is deleted if no index value is provided in pop ( ).	The del method removes the specified element from the list, but it does not return the deleted value.	The remove method is used when the index value is unknown and the element to be deleted is known.

>>> **list1 =[100, 200, 90, 'Raman', 100, 'Ashwin']**
>>> **list1.pop(2)**
 	
OUTPUT
90
	>>> **list1=[100, 200, 'Raman', 100]**
	>>> **del list1[2]**
	>>> **print(list1)**
	
	OUTPUT
	
	[100, 200, 100]
		>>> **list1=[100, 200, 50, 400, 500, 'Raman', 100, 'Ashwin']**
		>>> **list1.remove(400)**
		>>> **print(list1)**
		
		OUTPUT

		[100, 200, 50, 500, 'Raman', 100, 'Ashwin']

25. What are the major differences between key and value in a dictionary?

key	value
Keys are unique	Values may not be unique
The keys must be of an immutable data type such as strings, numbers, or tuples.	The values of a dictionary can be of any type

26. Give the difference between upper() and isupper() functions of string.

The upper() function returns the copy of the string with all the letters in uppercase. The isupper() function returns True if the string is in uppercase.
Example

```
>>> print(str3.upper())
ERATECH

>>> str5='tech era *2018*'
>>> print(str5.isupper())
False
```

27. Compare random(), randint() and randrange().

Function name	Purpose	Example
random()	Generates a random float number between 0.0 to 1.0.	>>> random.random() 0.21787519275240708
randint()	Returns a random integer between the specified integers.	>>> random.randint(1,50) 21 >>> random.randint(1,50) 35

randrange()	Returns a randomly selected element from the range created by the start, stop and step arguments. The value of start is 0	>>> random.randrange(1,20)
		1
		>>> random.randrange(1,20,5)

	by default. Similarly, the value of step is 1 by default.	16

28. When is the else clause executed in looping statements?

The **else clause** in looping statements is executed when the loop terminates.

29. Compare break and continue.

BREAK	CONTINUE
Break statement causes the loop to break/ terminate immediately.	Continue statement causes the current iteration of the loop to skip and go to the next iteration.
The loop of which, break statement is a part, stops.	All the statements following the continue statement in the loop will not be executed and loop control goes to the next iteration.

s=10;
for i in range (10, 20, 3):
  s+=i
  if(i==16):
  break
  print(s);
print("end");
 	
OUTPUT:
 	
20
33
end
	s=10;
	for i in range (10, 20, 3):
	  s+=i
	  if(i==16):
	  continue
	  print(s);
	print("end");
	
	OUTPUT:
	
	20
	33
	68
	end