# Python Viva Questions - Module 1 & 2

1. **What is Python?**

   Python is a high-level, interpreted, and general-purpose programming language known for its simplicity and readability.

2. **Who developed Python and when?**

   Python was developed by **Guido van Rossum** in **1991**.

3. **What are the key features of Python?**

   Easy to learn and use
   Interpreted language
   Object-oriented
   Portable and extensible
   Large standard library

4. **What are Python identifiers?**
   Identifiers are the names used to identify variables, functions, classes, or other objects in Python.

5. **What are keywords in Python?**

   Keywords are reserved words that have special meanings in Python (e.g., if, else, while, for, def, class).

6. **What are variables in Python?**

   Variables are used to store data values. They are created automatically when a value is assigned.

7. **How do you take input from the user in Python?**

   Using the **input()** function.
   Example:
   name = input("Enter your name: ")

8. **How do you display output in Python?**

   Using the **print()** function.
   Example:
   print("Hello, World!")

**9. What are data types in Python?**

Common data types are:

int (Integer)

float (Floating-point)

str (String)

bool (Boolean)

list, tuple, set, dict

**10. Are strings mutable in Python?**

No. **Strings are immutable**, meaning their content cannot be changed after creation.

**11. How can you access characters in a string?**

By using **indexing**. Index starts at **0**.

```
text = "Hello"
print(text[1])  # Output: e
```

**12. What is string slicing?**

Slicing is used to extract a part of a string.

```
text = "Python"
print(text[0:3])  # Output: Pyt
```

**13. What does negative indexing mean in strings?**

Negative indexing means counting from the **end of the string**.

```
text = "Python"
print(text[-1])  # Output: n
```

**14. How do you find the length of a string?**

Using the **len()** function.

**15. How do you check if a substring exists in a string?**

Using the **in** operator.

```
"Py" in "Python"  # Output: True
```

**16. What is string formatting in Python?**

String formatting allows inserting variables into strings.
Example using f-string:

```
name = "Alice"
print(f"Hello, {name}!")
```

**17. What does the count() method do?**

It returns the number of times a substring appears in the string.

```
text = "banana"
print(text.count("a"))  # Output: 3
```

**18. What is type conversion?**

Type conversion is converting one data type to another.
Example:

```
x = int("5")  # converts string to integer
```

**19. What is the difference between mutable and immutable objects?**

**Mutable:** Can be changed after creation (e.g., list, dict, set)

**Immutable:** Cannot be changed (e.g., str, tuple, int)

**20. What is a list in Python?**

A **list** is an ordered, mutable (changeable) collection of elements that can hold **items of different data types**.
Example:

```
my_list = [10, "Hello", 3.5]
```

**21. How do you create a list in Python?**

Lists are created using **square brackets [ ]**.

```
fruits = ["apple", "banana", "cherry"]
```

**22. What is the main feature of a list?**

Lists are **ordered**, **mutable**, and can contain **duplicate** elements.

**23. How can you access elements of a list?**

By using **indexing**. Indexing starts from **0**.

```
fruits = ["apple", "banana", "cherry"]
print(fruits[1])  # Output: banana
```

**24. What is negative indexing in lists?**

Negative indexing starts counting from the **end** of the list.

```
fruits = ["apple", "banana", "cherry"]
print(fruits[-1])  # Output: cherry
```

**25. How do you add elements to a list?**

**append():** Adds one element at the end.

**insert():** Adds an element at a specific position.

**extend():** Adds multiple elements from another list.

Example:

```
fruits.append("mango")
fruits.insert(1, "orange")
```

**26. How do you remove elements from a list?**

remove(item) – Removes the first matching element.

pop(index) – Removes an element by index.

clear() – Removes all elements.

Example:

```
fruits.remove("banana")
fruits.pop(0)
fruits.clear()
```

**27. How do you change a value in a list?**

Assign a new value to the specific index.
fruits[1] = "grape"

**28. How do you find the length of a list?**

Using the **len()** function.
len(fruits)

**29. How can you check if an item exists in a list?**

Using the **in** keyword.
if "apple" in fruits:
print("Yes, apple is in the list")

**30. What is list slicing?**

It is used to access a **subset** of list elements.
numbers = [10, 20, 30, 40, 50]
print(numbers[1:4])  # Output: [20, 30, 40]

**31. How do you copy a list?**

Use the **copy()** method or slicing.
new_list = fruits.copy()

**32. What is the difference between append() and extend()?**

append() adds one element at the end.

extend() adds multiple elements from another iterable.

a = [1, 2]
b = [3, 4]
a.append(b)   # [1, 2, [3, 4]]
a.extend(b)   # [1, 2, 3, 4]

**33. How do you sort a list in Python?**

Use the **sort()** method or **sorted()** function.
numbers.sort()

**34. How do you reverse a list?**

Using the reverse() method.

Using slicing [::-1].

```
numbers.reverse()
# or
numbers[::-1]
```

**35. What is the difference between sort() and sorted()?**

sort() modifies the original list.

sorted() returns a new sorted list without changing the original one.

**36. Can a list contain another list?**

Yes. Such a list is called a **nested list**.
nested = [[1, 2], [3, 4], [5, 6]]

**37. How can you iterate through a list?**

Using a **for loop**.

```
for item in fruits:
print(item)
```

**38. What is a list comprehension?**

It's a short way to create lists using a single line of code.

**39. What is a tuple in Python?**

A tuple is an ordered, immutable collection of items.
Example:
my_tuple = (1, 2, 3)

**40. What is a set in Python?**

A set is an unordered collection of unique items.
Example:
my_set = {1, 2, 3}

**41. What is a dictionary in Python?**

A dictionary is an unordered collection of key–value pairs.
Example:
my_dict = {"name": "Alice", "age": 20}

**42. What is indentation in Python?**

Indentation is the space used at the beginning of a line to define blocks of code. Python uses indentation instead of braces {}.

**43. What are conditional statements in Python?**

They are used to perform different actions based on conditions:
if, elif, else.

**44. What are loops in Python?**

Loops are used to execute a block of code repeatedly.

**for loop**

**while loop**

**45. What is a function in Python?**

A function is a block of code that performs a specific task and can be reused.
Example:
def greet():
print("Hello!")

**46. What are comments in Python?**

Comments are lines ignored by the interpreter, used for documentation.

Single-line: # comment

Multi-line: triple quotes (''' ... ''')

| Feature | List | Tuple | Set | Dictionary |
|---|---|---|---|---|
| **Definition** | Ordered collection of elements | Ordered, immutable collection of elements | Unordered collection of unique elements | Unordered collection of key–value pairs |
| **Syntax** | [ ] (square brackets) | ( ) (parentheses) | { } (curly braces) | {key: value} |
| **Mutability** | Mutable (can be changed) | Immutable (cannot be changed) | Mutable (can add/remove items) | Mutable (can change values) |
| **Order Maintained** | Yes | Yes | No (unordered) | Yes (insertion order from Python 3.7+) |
| **Duplicates Allowed** | Yes | Yes | No | Keys – No, Values – Yes |
| **Indexing / Slicing** | Supported | Supported | Not supported | Keys used instead of index |
| **Elements Access** | By index (e.g., list[0]) | By index (e.g., tuple[0]) | By looping (no index) | By key (e.g., dict['key']) |
| **Example** | [10, 20, 30] | (10, 20, 30) | {10, 20, 30} | {'a': 10, 'b': 20} |