

1. Create a List From A to Z

Method 1: Using `string.ascii_lowercase`

```
import string

letters = list(string.ascii_lowercase)
print(letters)
```

Output

```
['a', 'b', 'c', 'd', ..., 'x', 'y', 'z']
```

2. Create a List Without Importing Anything

```
letters = [chr(i) for i in range(ord('a'), ord('z') + 1)]
print(letters)
```

3. Explanation

`string.ascii_lowercase`

- Python's `string` module contains common letter sets.
- `ascii_lowercase = "abcdefghijklmnopqrstuvwxyz"`

`list()`

- Converts the string into a list of characters.

`chr()`

- Converts a number → character.
- Example: `chr(97)` gives 'a'

`ord()`

- Converts a character → number.
- Example: `ord('a')` gives 97

Range

```
range(ord('a'), ord('z')+1)
```

Creates numbers from 97 → 122 (a to z).

4. Examples Using A-Z List

Example 1: Print A to Z one by one

```
for letter in letters:  
    print(letter)
```

Example 2: Pick a random letter

```
import random  
  
letters = list(string.ascii_lowercase)  
print(random.choice(letters))
```

Example 3: Check if a letter is in the list

```
if 'g' in letters:  
    print("Yes, g is in the list!")
```

Example 4: Find index of a letter

```
print(letters.index('m')) # Output: 12
```

Below is a **beginner-friendly, safe, and complete 1-hour learning plan** to learn **Python string (str) er sob chakriyo method + property**. I'll explain each method **simple Bangla + example** so you can understand quickly.

1-Hour Python String Full Course (All Methods + Properties)

Total time: ≈ 60 minutes

You learn: Most commonly used & interview-level important string methods

👉 Before starting: What is a string? (2 minutes)

Python string:

```
name = "Atiar"
```

- ✓ Ordered
- ✓ Immutable (change kora jayna)
- ✓ Iterable

■ PART-1: String Basic Properties (10 minutes)

1 len() — length ber kore

```
name = "Python"  
print(len(name)) # 6
```

2 Indexing

```
name = "Hello"  
print(name[0]) # H  
print(name[-1]) # o
```

3 Slicing

```
text = "programming"  
print(text[0:5]) # progr  
print(text[5:]) # amming  
print(text[:5]) # progr
```

4 Concatenation

```
a = "Hello"  
b = "World"  
print(a + " " + b)
```

5 Repetition

```
print("Hi" * 3) # HiHiHi
```

PART–2: Most Useful String Methods (45 minutes)

I divided them into groups so you learn fast.

Group-1: Case Changing Methods (5 minutes)

upper() — sob uppercase

```
print("python".upper()) # PYTHON
```

lower() — sob lowercase

```
print("PYTHON".lower()) # python
```

title() — every word capital

```
print("hello world".title()) # Hello World
```

capitalize() — first letter capital

```
print("python code".capitalize()) # Python code
```

swapcase() — small → capital, capital → small

```
print("PyThOn".swapcase()) # pYtHoN
```

Group-2: Search / Find Methods (10 minutes)

find() — location ber kore (na pele -1)

```
print("python".find("th")) # 2
```

rfind() — right side theke search

```
print("pythonpython".rfind("th")) # 8
```

index() — same as find() but error dey

```
print("python".index("th")) # 2
```

count() — kotobar ache ber kore

```
print("banana".count("a")) # 3
```

startswith() — ki diye start?

```
print("hello".startswith("he")) # True
```

endswith() — ki diye end?

```
print("hello".endswith("lo")) # True
```

● Group-3: Replace & Modify Methods (10 minutes)

replace()

```
print("I love python".replace("python", "AI"))
```

strip() — space/remove from both side

```
print(" hello ".strip()) # hello
```

lstrip() — left side remove

```
print(" hi ".lstrip())
```

rstrip() — right side remove

```
print("hi ".rstrip())
```

● Group-4: Split & Join Methods (8 minutes)

split() — text → list

```
print("a,b,c".split(",")) # ['a', 'b', 'c']
```

rsplit() — right side theke split

```
print("1-2-3".rsplit("-", 1)) # ['1-2', '3']
```

splitlines() — line break hole line alada

```
print("Hi\nHello".splitlines())
```

join() — list → string

```
words = ["I", "love", "Python"]
print(" ".join(words)) # I love Python
```

Group-5: Check Type (Boolean) Methods (7 minutes)

isalnum() — letter/number

isalpha() — just letter

isdigit() — just digit

islower()

isupper()

isspace() — only space

istitle()

Examples:

```
print("Python3".isalnum()) # True
print("ABC".isupper()) # True
print(" ".isspace()) # True
```

Group-6: Formatting Methods (5 minutes)

center(width)

```
print("Hi".center(10, "*"))
```

ljust(width)

```
print("Hi".ljust(10, "-"))
```

rjust(width)

```
print("Hi".rjust(10, "-"))
```

format()

```
print("My age is {}".format(17))
```

f-string (best)

```
age = 17  
print(f"My age is {age}")
```

Group-7: Real Advanced Methods (Optional) (5 minutes)

encode()

```
print("hello".encode())
```

expandtabs()

```
print("hi\tthere".expandtabs(4))
```

partition()

```
print("key=value".partition("="))
```

rpartition()

```
print("a=b=c".rpartition("="))
```

END: 1-Hour Practice Task

Try these:

- 1 Count vowels from a string**
- 2 Remove extra spaces**
- 3 Detect if a string is palindrome**
- 4 Word frequency count**
- 5 Convert a line to:**

- all lowercase
- Title case
- Without spaces
- Split into list

Here is the **simplest & best beginner-friendly method** to **count vowels from a string** in Python.

Method 1: Simple Loop

```
text = "I love Python Programming"

vowels = "aeiouAEIOU"
count = 0

for char in text:
    if char in vowels:
        count += 1

print("Total vowels:", count)
```

Output

```
Total vowels: 8
```

Method 2: Using List Comprehension (Short Code)

```
text = "Bangladesh"

vowels = "aeiouAEIOU"

count = sum(1 for c in text if c in vowels)

print(count)
```

Method 3: Count Each Vowel Separately

```
text = "Bangladesh"

vowels = "aeiou"
result = {}

for v in vowels:
    result[v] = text.lower().count(v)

print(result)
```

Output

```
{'a': 3, 'e': 1, 'i': 0, 'o': 0, 'u': 0}
```

What is a Palindrome?

A word or sentence that reads **same forward and backward**.

Examples:

- madam ✓
- level ✓
- racecar ✓
- hello ✗

Method 1: Easiest (Using Reverse)

```
text = "madam"

if text == text[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Method 2: Case-insensitive + remove spaces

Useful for sentences like:

Never odd or even

```
text = "Never odd or even"

# clean the text: remove spaces + lowercase
clean = text.replace(" ", "").lower()

if clean == clean[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Method 3: Using a Loop (Beginner logic)

```
text = "level"
clean = text.lower()

is_palindrome = True

for i in range(len(clean)):
    if clean[i] != clean[-i-1]:
        is_palindrome = False
        break

print("Palindrome" if is_palindrome else "Not Palindrome")
```



Mini Practice Task for You

Check if these are palindrome:

1. racecar
 2. computer
 3. noon
 4. Was it a car or a cat I saw
-
-

✓ Method 1: Basic Word Frequency Count

```
text = "I love Python because Python is powerful and I love coding"

words = text.lower().split()    # lowercase + split into list

freq = {}    # empty dictionary

for w in words:
    if w in freq:
        freq[w] += 1
    else:
        freq[w] = 1

print(freq)
```

Output

```
{'i': 2, 'love': 2, 'python': 2, 'because': 1, 'is': 1, 'powerful': 1,
'and': 1, 'coding': 1}
```

✓ Method 2: Using `collections.Counter` (Best & Short)

```
from collections import Counter

text = "apple banana apple mango banana apple"

freq = Counter(text.lower().split())
print(freq)
```

Output

```
Counter({'apple': 3, 'banana': 2, 'mango': 1})
```

✓ Method 3: Sort by Word Frequency

```
text = "I love Python and I love AI"

words = text.lower().split()
from collections import Counter

freq = Counter(words)

sorted_freq = sorted(freq.items(), key=lambda x: x[1], reverse=True)

print(sorted_freq)
```

Output

```
[('love', 2), ('i', 2), ('python', 1), ('and', 1), ('ai', 1)]
```

📝 Extra Example: Remove punctuation before counting

```
import string

text = "Hello, hello! Python is great; Python is fun."

clean = text.lower()

# remove punctuation
for p in string.punctuation:
    clean = clean.replace(p, "")

words = clean.split()

from collections import Counter

print(Counter(words))
```

Here are **50 real, practical, beginner-to-advanced Python string practice problems** — perfect for mastering all string methods you learned.

50 Real Python String Practice Problems

Basic Level (1–15)

1. Print the length of a string given by the user.
 2. Take a string and print the first and last character.
 3. Check if a string contains the word "python" .
 4. Count how many times the letter "a" appears in a string.
 5. Convert a string to uppercase.
 6. Convert a string to lowercase.
 7. Remove spaces from left, right, and both sides of a string.
 8. Reverse a string.
 9. Check whether a string is empty or not.
 10. Replace all spaces with - .
 11. Split a sentence into words.
 12. Join a list of words into one string with spaces.
 13. Find the index of "@" inside an email string.
 14. Print every character of a string using a loop.
 15. Check if two strings are equal (case-insensitive).
-

Intermediate Level (16–35)

16. Count vowels in a string.
17. Count consonants in a string.
18. Detect if a string is a palindrome.
19. Remove all punctuation from a paragraph.
20. Convert each word's first letter to capital (Title Case).
21. Swap the case of all letters in a string.
22. Count the occurrence of each character in a string.
23. Extract only numeric characters from a string.
24. Extract only alphabet characters from a string.
25. Extract only special characters from a string.
26. Check if a string is alphanumeric.
27. Check if a string starts with "https" .
28. Check if a string ends with ".com" .

-
29. Remove all duplicate characters from a string.
 30. Print characters present at even indexes.
 31. Print characters present at odd indexes.
 32. Replace all repeated characters with `"*"`.
 33. Remove all vowels from a string.
 34. Find the longest word in a sentence.
 35. Count the number of words in a paragraph.
-

Advanced Level (36–50)

36. Word frequency count for a paragraph.
 37. Print top 3 most frequent words.
 38. Given a sentence, reverse the order of the words.
 39. Find the shortest word in a sentence.
 40. Remove extra spaces:
 - Convert `"I love Python"` → `"I love Python"`
 41. Check if a sentence contains all vowels at least once.
 42. Compare two strings ignoring punctuation.
 43. Remove all digits from a string.
 44. Replace every vowel with its uppercase version.
 45. Count how many uppercase letters are in a string.
 46. Count how many lowercase letters are in a string.
 47. Check if a sentence is a pangram (contains A–Z).
 48. Convert a string into a dictionary of letter counts.
 49. Given `"a3b2c4"` → expand it → `"aa bb cccc"` (string decoding).
 50. Validate if a strong password:
 - $\text{length} \geq 8$
 - contains uppercase
 - contains lowercase
 - contains digit
 - contains special character
-

Great — I'll give you full solutions for all 50 string problems (clean, beginner-friendly, well-structured).

⚠️ If you later want these as a **PDF**, just say: “Make PDF”.

Now enjoy the solutions 

✓ SOLUTIONS TO ALL 50 STRING PRACTICE PROBLEMS

➊ Basic Level Solutions (1–15)

1. Length of a string

```
text = input("Enter string: ")
print(len(text))
```

2. First and last character

```
s = input()
print(s[0], s[-1])
```

3. Contains “python”?

```
s = input().lower()
print("python" in s)
```

4. Count occurrences of 'a'

```
s = input().lower()
print(s.count('a'))
```

5. Uppercase

```
print(input().upper())
```

6. Lowercase

```
print(input().lower())
```

7. Strip spaces

```
s = input()
print(s.lstrip())
```

```
print(s.rstrip())
print(s.strip())
```

8. Reverse a string

```
print(input()[:-1])
```

9. Check empty

```
s = input()
print("Empty" if len(s) == 0 else "Not empty")
```

10. Replace spaces with '-'

```
print(input().replace(" ", "-"))
```

11. Split

```
print(input().split())
```

12. Join list of words

```
words = ["I", "love", "Python"]
print(" ".join(words))
```

13. Find '@' in email

```
email = input()
print(email.find("@"))
```

14. Print characters using loop

```
for ch in input():
    print(ch)
```

15. Compare strings (case-insensitive)

```
s1 = input().lower()
s2 = input().lower()
print(s1 == s2)
```

Intermediate Level Solutions (16–35)

16. Count vowels

```
s = input().lower()
vowels = "aeiou"
count = sum(c in vowels for c in s)
print(count)
```

17. Count consonants

```
s = input().lower()
vowels = "aeiou"
count = sum(c.isalpha() and c not in vowels for c in s)
print(count)
```

18. Palindrome

```
s = input().lower()
print("Palindrome" if s == s[::-1] else "Not Palindrome")
```

19. Remove punctuation

```
import string
s = input()
for p in string.punctuation:
    s = s.replace(p, "")
print(s)
```

20. Title Case

```
print(input().title())
```

21. Swap case

```
print(input().swapcase())
```

22. Character frequency

```
s = input()
freq = {}
for c in s:
```

```
    freq[c] = freq.get(c, 0) + 1
print(freq)
```

23. Extract digits

```
s = input()
digits = "".join(c for c in s if c.isdigit())
print(digits)
```

24. Extract alphabet letters

```
s = input()
letters = "".join(c for c in s if c.isalpha())
print(letters)
```

25. Extract special characters

```
s = input()
special = "".join(c for c in s if not c.isalnum())
print(special)
```

26. Check alphanumeric

```
print(input().isalnum())
```

27. Starts with "https"

```
print(input().startswith("https"))
```

28. Ends with ".com"

```
print(input().endswith(".com"))
```

29. Remove duplicate characters

```
s = input()
result = ""
for c in s:
    if c not in result:
        result += c
print(result)
```

30. Characters at even indexes

```
s = input()  
print(s[::-2])
```

31. Characters at odd indexes

```
s = input()  
print(s[1::2])
```

32. Replace repeated characters with "***"

```
s = input()  
result = ""  
seen = set()  
  
for c in s:  
    if c in seen:  
        result += "***"  
    else:  
        result += c  
    seen.add(c)  
  
print(result)
```

33. Remove vowels

```
s = input()  
vowels = "aeiouAEIOU"  
print("".join(c for c in s if c not in vowels))
```

34. Longest word

```
words = input().split()  
print(max(words, key=len))
```

35. Count words

```
print(len(input().split()))
```



Advanced Level Solutions (36–50)

36. Word frequency

```
from collections import Counter
text = input().lower().split()
print(Counter(text))
```

37. Top 3 most frequent words

```
from collections import Counter
words = input().lower().split()
print(Counter(words).most_common(3))
```

38. Reverse word order

```
print(" ".join(input().split()[::-1]))
```

39. Shortest word

```
words = input().split()
print(min(words, key=len))
```

40. Remove extra spaces

```
print(" ".join(input().split()))
```

41. Contains all vowels?

```
s = input().lower()
print(all(v in s for v in "aeiou"))
```

42. Compare ignoring punctuation

```
import string

s1 = input().lower()
s2 = input().lower()

for p in string.punctuation:
    s1 = s1.replace(p, "")
    s2 = s2.replace(p, "")

print(s1 == s2)
```

43. Remove digits

```
print("".join(c for c in input() if not c.isdigit()))
```

44. Convert vowels to uppercase

```
s = input()
vowels = "aeiou"
result = ""

for c in s:
    if c.lower() in vowels:
        result += c.upper()
    else:
        result += c

print(result)
```

45. Count uppercase letters

```
print(sum(c.isupper() for c in input()))
```

46. Count lowercase letters

```
print(sum(c.islower() for c in input()))
```

47. Check pangram (A-Z)

```
s = input().lower()
alphabet = "abcdefghijklmnopqrstuvwxyz"
print(all(a in s for a in alphabet))
```

48. String to dictionary of letter counts

```
s = input()
freq = {}
for c in s:
    freq[c] = freq.get(c, 0) + 1
print(freq)
```

49. Decode string “a3b2c4” → “aa bb cccc”

```
s = input()
result = ""

for i in range(0, len(s), 2):
    char = s[i]
    num = int(s[i+1])
    result += char * num

print(result)
```

50. Strong password checker

```
import string

pwd = input()

is_strong = (
    len(pwd) >= 8 and
    any(c.islower() for c in pwd) and
    any(c.isupper() for c in pwd) and
    any(c.isdigit() for c in pwd) and
    any(c in string.punctuation for c in pwd)
)

print("Strong" if is_strong else "Weak")
```
