50 Pythonic Logic Exercises with Solutions

String Manipulation

1. Reverse a string

```
s = "hello"
res = s[::-1]
print(res)  # 'olleh'
```

2. Check if a string is a palindrome

```
s = "madam"
res = s == s[::-1]
print(res)  # True
```

3. Count vowels in a string

```
s = "pythonic"
res = sum(ch in "aeiou" for ch in s)
print(res)  # 3
```

4. Remove all spaces

```
s = "a b  c"
res = s.replace(" ", "")
print(res)  # 'abc'
```

5. Extract digits from a string

```
s = "ab12c3"
res = ''.join(ch for ch in s if ch.isdigit())
print(res)  # '123'
```

Splitting & Joining

6. Split a sentence into words

```
s = "Python is fun"
res = s.split()
print(res)  # ['Python', 'is', 'fun']
```

7. Join list into string

```
words = ["Python", "is", "cool"]
res = "-".join(words)
print(res)  # 'Python-is-cool'
```

8. Get first letters of words

```
s = "Keep It Simple"
res = ''.join(word[0] for word in s.split())
print(res)  # 'KIS'
```

9. Sort words alphabetically

```
s = "banana apple cherry"
res = " ".join(sorted(s.split()))
print(res)  # 'apple banana cherry'
```

10. Count word frequency

```
s = "one two two three three three"
res = {w: s.split().count(w) for w in set(s.split())}
print(res)  # {'three': 3, 'two': 2, 'one': 1}
```

Slicing

11. Get every 2nd character

```
s = "abcdef"
res = s[::2]
print(res)  # 'ace'
```

12. Slice without first and last

```
s = "Python"
res = s[1:-1]
print(res)  # 'ytho'
```

13. Rotate string by k characters

```
s, k = "abcdef", 2
res = s[k:] + s[:k]
print(res)  # 'cdefab'
```

14. Get last 3 characters

```
s = "hello"
res = s[-3:]
print(res)  # 'llo'
```

15. Remove vowels using slicing & replace

```
s = "beautiful"
res = ''.join(ch for ch in s if ch not in "aeiou")
print(res)  # 'btfl'
```

List Comprehensions

16. Squares of numbers

```
nums = [1, 2, 3, 4]
res = [n**2 for n in nums]
print(res)  # [1, 4, 9, 16]
```

17. Flatten a 2D list

```
matrix = [[1,2],[3,4]]
res = [x for row in matrix for x in row]
print(res)  # [1, 2, 3, 4]
```

18. Get even numbers

```
nums = range(10)
res = [n for n in nums if n % 2 == 0]
print(res)  # [0, 2, 4, 6, 8]
```

19. Extract uppercase letters

```
s = "PyThOn"
res = [ch for ch in s if ch.isupper()]
print(res)  # ['P', 'T', 'O']
```

20. Cartesian product

```
a, b = [1, 2], [3, 4]
res = [(x,y) for x in a for y in b]
print(res)  # [(1, 3), (1, 4), (2, 3), (2, 4)]
```

Map, Filter, Reduce

21. Map: square numbers

```
nums = [1, 2, 3]
res = list(map(lambda x: x**2, nums))
print(res)  # [1, 4, 9]
```

22. Filter: keep odd numbers

```
nums = [1, 2, 3, 4]
res = list(filter(lambda x: x % 2, nums))
print(res)  # [1, 3]
```

23. Reduce: sum of numbers

```
from functools import reduce
nums = [1, 2, 3, 4]
```

```
res = reduce(lambda x,y: x+y, nums)
print(res)  # 10
```

24. Map with string length

```
words = ["hi", "world"]
res = list(map(len, words))
print(res)  # [2, 5]
```

25. Filter palindromes

```
words = ["madam", "apple", "level"]
res = list(filter(lambda w: w == w[::-1], words))
print(res)  # ['madam', 'level']
```

Set & Dictionary Logic

26. Unique characters in string

```
s = "programming"
res = set(s)
print(res)  # {'m', 'i', 'r', 'n', 'a', 'g', 'p', 'o'}
```

27. Common elements of two lists

```
a, b = [1,2,3], [2,3,4]
res = list(set(a) & set(b))
print(res)  # [2, 3]
```

28. Word to length mapping

```
words = ["apple", "banana"]
res = {w: len(w) for w in words}
print(res)  # {'apple': 5, 'banana': 6}
```

29. Count characters in string

```
s = "mississippi"
res = {ch: s.count(ch) for ch in set(s)}
print(res)  # {'s': 4, 'm': 1, 'p': 2, 'i': 4}
```

30. Invert dictionary

```
d = {"a":1, "b":2}
res = {v:k for k,v in d.items()}
print(res)  # {1: 'a', 2: 'b'}
```

Functional Tricks

31. Find max word length

```
words = ["a", "ab", "abc"]
res = max(words, key=len)
print(res)  # 'abc'
```

32. Sort by last character

```
words = ["car", "apple", "dog"]
res = sorted(words, key=lambda w: w[-1])
print(res)  # ['car', 'apple', 'dog']
```

33. Sort dict by values

```
d = {"a":3, "b":1, "c":2}
res = dict(sorted(d.items(), key=lambda x: x[1]))
print(res)  # {'b': 1, 'c': 2, 'a': 3}
```

34. Find all anagrams of “eat”

```
words = ["tea", "bat", "ate", "cat"]
res = [w for w in words if sorted(w) == sorted("eat")]
print(res)  # ['tea', 'ate']
```

35. Zip two lists

```
a, b = [1,2,3], ['a','b','c']
res = dict(zip(a, b))
print(res)  # {1: 'a', 2: 'b', 3: 'c'}
```

Advanced

36. Flatten nested list recursively

```
def flatten(lst):
    for i in lst:
        if isinstance(i, list):
            yield from flatten(i)
        else:
            yield i

res = list(flatten([1, [2, [3, 4]], 5]))
print(res)  # [1, 2, 3, 4, 5]
```

37. Fibonacci with reduce

```
from functools import reduce
n = 7
```

```python
res = reduce(lambda x, _: x+[x[-2]+x[-1]], range(n-2), [0,1])
print(res)  # [0, 1, 1, 2, 3, 5, 8]
```

38. Rotate list

```python
lst, k = [1,2,3,4], 2
res = lst[k:] + lst[:k]
print(res)  # [3, 4, 1, 2]
```

39. Find second largest

```python
nums = [10, 20, 4, 8]
res = sorted(set(nums))[-2]
print(res)  # 10
```

40. Group words by length

```python
words = ["hi", "to", "cat", "dog"]
from collections import defaultdict
d = defaultdict(list)
for w in words: d[len(w)].append(w)
print(dict(d))  # {2: ['hi', 'to'], 3: ['cat', 'dog']}
```

Challenges

41. Reverse words in sentence

```python
s = "Python is fun"
res = " ".join(s.split()[::-1])
print(res)  # 'fun is Python'
```

42. Find missing number in list

```python
nums = [1,2,4,5]
res = set(range(1,6)) - set(nums)
print(res)  # {3}
```

43. Check anagram

```python
a, b = "listen", "silent"
res = sorted(a) == sorted(b)
print(res)  # True
```

44. Remove duplicates while preserving order

```python
nums = [1,2,2,3,1]
res = list(dict.fromkeys(nums))
print(res)  # [1, 2, 3]
```

45. Find most common character

```python
from collections import Counter
s = "aabbbcccc"
res = Counter(s).most_common(1)[0][0]
print(res)  # 'c'
```

46. Longest word in sentence

```python
s = "Python makes coding simple"
res = max(s.split(), key=len)
print(res)  # 'simple'
```

47. Intersection of multiple lists

```python
from functools import reduce
lists = [[1,2,3],[2,3,4],[3,4,5]]
res = list(reduce(set.intersection, map(set, lists)))
print(res)  # [3]
```

48. Find all substrings of string

```python
s = "abc"
res = [s[i:j] for i in range(len(s)) for j in range(i+1, len(s)+1)]
print(res)  # ['a', 'ab', 'abc', 'b', 'bc', 'c']
```

49. Merge two sorted lists

```python
a, b = [1,3,5], [2,4,6]
res = sorted(a+b)
print(res)  # [1, 2, 3, 4, 5, 6]
```

50. Check if all elements are unique

```python
lst = [1,2,3,4,1]
res = len(lst) == len(set(lst))
print(res)  # False
```
