Prefix Sum Pattern — Coding Interview Notes (Light Theme)

General Pattern Template

```
def fn(arr):
    prefix = [arr[0]]
    for i in range(1, len(arr)):
        prefix.append(prefix[-1] + arr[i])

    return prefix
```

Concept:
The **Prefix Sum** pattern is a technique for preprocessing cumulative information from an array to allow constant-time queries over contiguous subarrays. It is particularly powerful for range sum queries, difference calculations, or aggregate statistics.

Time Complexity: O(n) to build, O(1) per query.
When to use: When repeated range-sum or difference queries are required.

Key Ideas

1. Each prefix[i] stores the sum of arr[0]..arr[i].
2. Allows O(1) retrieval of any subarray sum using: prefix[r] - prefix[l-1].
3. Can be extended to 2D grids (prefix sum matrix).
4. Ideal for problems involving range updates or cumulative comparisons.

Example 1: Build Prefix Sum Array

Goal: Construct prefix sums where prefix[i] = sum(arr[:i+1]).
Approach: Accumulate progressively using the previous prefix value.

```
def build_prefix_sum(arr):
    prefix = [arr[0]]
    for i in range(1, len(arr)):
        prefix.append(prefix[-1] + arr[i])
    return prefix
```

Example 2: Range Sum Query Using Prefix

Goal: Given an array and many range queries (l, r), return the sum of elements from l to r efficiently.
Approach: Use the prefix sum to compute any range sum in O(1).

```
def range_sum(prefix, l, r):
    if l == 0:
        return prefix[r]
    return prefix[r] - prefix[l - 1]

# Example usage
arr = [2, 3, 5, 1, 6]
prefix = build_prefix_sum(arr)
print(range_sum(prefix, 1, 3))  # Output: 3 + 5 + 1 = 9
```

Example 3: Prefix Sum for Even-Odd Differences

Goal: Compute prefix sums of even and odd numbers separately for range difference queries.
Approach: Maintain two parallel prefix arrays for even and odd indices or values.

```
def even_odd_prefix(arr):
    even_prefix, odd_prefix = [0]*len(arr), [0]*len(arr)
    even_prefix[0] = arr[0] if arr[0] % 2 == 0 else 0
    odd_prefix[0] = arr[0] if arr[0] % 2 != 0 else 0

    for i in range(1, len(arr)):
        even_prefix[i] = even_prefix[i-1] + (arr[i] if arr[i] % 2 == 0 else 0)
        odd_prefix[i] = odd_prefix[i-1] + (arr[i] if arr[i] % 2 != 0 else 0)

    return even_prefix, odd_prefix
```

Summary Table

ConceptUse CaseComplexity Prefix sum arrayPrecompute cumulative sumsO(n) Range sum queryFast sum between indicesO(1) per query Even/Odd prefixTrack subsets separatelyO(n)