Perfect 🔥 — you're asking for a **DSA (Data Structures & Algorithms)** quick reference of **commonly used functions/methods** for **Arrays, Strings, ArrayList, HashMap, HashSet, and other core Java utilities** — which is *super* useful for LeetCode and interviews.

Here's a **complete cheat sheet** with categories 👇

🧩 1. Arrays (Primitive or Object Arrays)

import java.util.Arrays;

Common Methods:

Method	Description	Example
Arrays.sort(arr)	Sorts the array in ascending order	Arrays.sort(arr);
Arrays.sort(arr, Collections.reverseOrder())	Sorts array in descending order (for objects)	Arrays.sort(arr, Collections.reverseOrder());
Arrays.toString(arr)	Converts array to readable string	System.out.println(Arrays.toString(arr));
Arrays.equals(arr1, arr2)	Compares two arrays	Arrays.equals(a, b)
Arrays.fill(arr, val)	Fills entire array with value	Arrays.fill(arr, 0);
Arrays.copyOf(arr, newLength)	Copies array	int[] b = Arrays.copyOf(a, 5);
Arrays.binarySearch(arr, key)	Binary search (sorted array only)	int idx = Arrays.binarySearch(arr, 10);
arr.length	Gives length of array	int n = arr.length;

📝 2. Strings

String s = "hello";

Common Methods:

Method	**Description**	**Example**
s.length()	Returns string length	int len = s.length();
s.charAt(i)	Returns character at index i	char c = s.charAt(0);
s.substring(start, end)	Extract substring	s.substring(1, 3);
s.equals(str)	Compare content	s.equals("hi");
s.equalsIgnoreCase(str)	Case-insensitive compare	
s.contains(str)	Checks substring presence	s.contains("el");
s.indexOf(ch)	Finds first index	s.indexOf('e');
s.lastIndexOf(ch)	Finds last index	s.lastIndexOf('l');
s.replace(old, new)	Replace all occurrences	s.replace('l','x');
s.toLowerCase() / s.toUpperCase()	Convert case	
s.trim()	Removes spaces at ends	
s.split(" ")	Splits string into array	String[] arr = s.split(" ");
new StringBuilder(s).reverse()	Reverse string	

🧮 3. ArrayList

```
import java.util.ArrayList;
```

```
ArrayList<Integer> list = new ArrayList<>();
```

Common Methods:

Method	**Description**	**Example**
list.add(val)	Add element	list.add(5);
list.add(index, val)	Add at specific index	list.add(1, 10);
list.get(i)	Get element	int x = list.get(0);
list.set(i, val)	Replace element	list.set(1, 20);
list.remove(i)	Remove element at index	list.remove(0);
list.size()	Get size	
list.contains(x)	Check if present	
list.isEmpty()	Check if empty	
Collections.sort(list)	Sort ascending	
Collections.reverse(list)	Reverse list	
list.clear()	Remove all elements	

🗺️ 4. HashMap

import java.util.HashMap;

HashMap<Integer, String> map = new HashMap<>();

Common Methods:

Method	**Description**	**Example**
map.put(key, value)	Insert key-value pair	map.put(1, "One");
map.get(key)	Get value	map.get(1);
map.containsKey(key)	Check key exists	
map.containsValue(val)	Check value exists	
map.remove(key)	Remove entry	
map.size()	Number of entries	
map.keySet()	All keys	for(int k : map.keySet())
map.values()	All values	for(String v : map.values())
map.entrySet()	All entries	for(Map.Entry<Integer, String> e : map.entrySet())
map.clear()	Clears map	

🔢 5. HashSet

```
import java.util.HashSet;

HashSet<Integer> set = new HashSet<>();
```

Common Methods:

Method	Description	Example
set.add(x)	Add element	
set.remove(x)	Remove element	
set.contains(x)	Check existence	
set.size()	Count	
set.isEmpty()	Check empty	
set.clear()	Remove all	
set.addAll(otherSet)	Union	
set.retainAll(otherSet)	Intersection	

---

6. Stack

import java.util.Stack;

Stack<Integer> st = new Stack<>();

Method	Description	Example
st.push(x)	Push element	
st.pop()	Remove and return top	
st.peek()	Return top without removing	
st.isEmpty()	Check if empty	
st.size()	Get size	

---

7. Queue (LinkedList as Queue)

import java.util.Queue;

import java.util.LinkedList;

Queue<Integer> q = new LinkedList<>();

Method	Description	Example
q.add(x)	Enqueue	
q.poll()	Dequeue	
q.peek()	Front element	
q.isEmpty()	Check empty	

🪶 8. PriorityQueue (Min/Max Heap)

```
import java.util.PriorityQueue;

PriorityQueue<Integer> pq = new PriorityQueue<>(); // min-heap

PriorityQueue<Integer> maxpq = new PriorityQueue<>(Collections.reverseOrder()); // max-heap
```

Method	Description	Example
pq.add(x)	Add element	
pq.poll()	Remove smallest/largest	
pq.peek()	View top	
pq.size()	Get size	

⚙️ 9. Collections Utility

```
import java.util.Collections;
```

Method	Description	Example
Collections.sort(list)	Sort list	
Collections.reverse(list)	Reverse order	
Collections.max(list)	Maximum element	
Collections.min(list)	Minimum element	
Collections.frequency(list, val)	Count occurrences	

💡 10. Integer / Character Helper Methods

```
Integer.parseInt("123");   // String to int

String.valueOf(123);       // int to String

Character.isDigit(ch);     // Check digit

Character.isLetter(ch);    // Check letter

Character.toLowerCase(ch); // Convert case
```