

Java Wrapper Class

byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean
char	Character

Array

	C++	Java	Python3
new	<code>int x = [0];</code>	<code>int[] x = new int[]{0};</code>	<code>x = [0]</code>
length	<code>std::size(x) // C++17</code>	<code>x.length</code>	<code>len(x)</code>

String

	C++	Java	Python3
new	<code>char* s = "string";</code>	<code>String s = "string";</code>	<code>s = "string"</code>
get	<code>char c = s[0];</code>	<code>char c = s.charAt(0);</code>	<code>c = s[0];</code>
length	<code>strlen(s)</code>	<code>s.length()</code>	<code>len(s)</code>
sub string	<code>char* sub = malloc(10); strncpy(sub, s+begin, end-begin);</code>	<code>String sub = s.substring(begin, end);</code>	<code>s[begin, end]</code>
check substring	<code>if (strstr(sub, s))</code>	<code>if (s.contains(sub))</code>	<code>if sub in s:</code>

Stack

	C++	Java	Python3
include	<code>#include <stack></code>	<code>import java.util.*;</code>	
new	<code>std::stack<int> s;</code>	<code>Stack<Integer> s = new Stack<Integer>();</code>	<code>s = []</code>
push	<code>s.push(i)</code>	<code>s.push(i);</code>	<code>s.append(i)</code>
pop	<code>s.pop();</code>	<code>popped = s.pop();</code>	<code>popped = s.pop()</code>
top	<code>s.top()</code>	<code>s.peak()</code>	<code>s[-1]</code>
size	<code>s.size()</code>	<code>s.size()</code>	<code>len(s)</code>
check empty	<code>if (s.empty())</code>	<code>if (s.empty())</code>	<code>if not s:</code>



By **Ptero**
cheatography.com/ptero/

Not published yet.
 Last updated 3rd July, 2024.
 Page 1 of 3.

Sponsored by **ApolloPad.com**
 Everyone has a novel in them. Finish Yours!
<https://apollopad.com>

Queue			
	C++	Java	Python3
include	<code>#include <queue></code>	<code>import java.util.*;</code>	
new	<code>std::queue<int> q;</code>	<code>Queue<Integer> q = new LinkedList<>();</code>	<code>q = []</code>
push	<code>q.push(i);</code>	<code>q.add(i);</code>	<code>q.append(i)</code>
pop	<code>q.pop()</code>	<code>popped = q.remove();</code>	<code>popped = q.pop(0)</code>
front	<code>q.front()</code>	<code>q.peek()</code>	<code>q[0]</code>
size	<code>q.size()</code>	<code>q.size()</code>	<code>len(q)</code>
check empty	<code>if (q.empty())</code>	<code>if (q.isEmpty())</code>	<code>if not q:</code>

Heap			
	C++	Java	Python3
include	<code>#include <queue></code>	<code>import java.util.*;</code>	<code>from queue import Priority Queue</code>
new min heap	<code>std::priority_queue<int, std::vector<int>, std::greater- <int>> h;</code>	<code>Queue<Integer> h = new PriorityQueu- e<>();</code>	<code>h = PriorityQueue()</code>
new max heap	<code>std::priority_queue<int> h;</code>	<code>Queue<Integer> h = new PriorityQueue<>(Collections.reverseOrder());</code>	
push	<code>h.push(i);</code>	<code>h.add(i);</code>	<code>h.put((1, "Harry"))</code>
pop	<code>h.pop();</code>	<code>Integer popped = h.poll();</code>	<code>popped = h.get()</code>
top	<code>h.top()</code>	<code>h.peek()</code>	<code>popped = h.get() h.put(popped)</code>
size	<code>h.size()</code>	<code>h.size()</code>	<code>h.qsize()</code>
check empty	<code>if (h.empty())</code>	<code>if (h.isEmpty())</code>	<code>if h.empty():</code>

HashSet			
	C++	Java	Python3
include	<code>#include <unordered_set></code>	<code>import java.util.*;</code>	
new	<code>std::unordered_set<int> s;</code>	<code>Set<Integer> s = new HashSet<>();</code>	<code>s = set()</code>
add	<code>s.insert(x);</code>	<code>s.add(x);</code>	<code>s.add(x)</code>
delete	<code>s.erase(x);</code>	<code>s.remove(x);</code>	<code>s.discard(x)</code>
check in set	<code>std::unordered_set<int>::const_iterator it = s.find(x); if (it != s.end())</code>	<code>if (s.contains(x))</code>	<code>if x in s:</code>
size	<code>s.size()</code>	<code>s.size()</code>	<code>len(s)</code>
check empty	<code>if (s.empty())</code>	<code>if (s.isEmpty())</code>	<code>if not s:</code>



HashMap			
	C++	Java	Python3
include	<code>#include <unordered_map></code>	<code>import java.util.*;</code>	
new	<code>std::unordered_map<int, int> t;</code>	<code>Hashtable<Integer, Integer> t = new Hashtable<> ();</code>	<code>d = {}</code>
add	<code>t.insert(std::make_pair<int, int>(key, val));</code>	<code>t.put(key, val);</code>	<code>d[key] = val</code>
get	<code>int val = t.at(key);</code>	<code>Integer val = t.get(key);</code>	<code>val = d[key]</code>
delete	<code>t.erase(key);</code>	<code>t.remove(key);</code>	<code>del d[key]</code>
iterate		<code>for (Map.Entry<Integer, Integer> set : t.entrySet()) { key=set.getKey(); val=set.getValue(); }</code>	
iterate		<code>t.forEach((key, val)->{...})</code>	<code>for key in d:</code>
check in table	<code>std::unordered_map<int, int>::const_iterator it = t.find(key); if (it != t.end())</code>	<code>if (t.containsKey(key))</code>	<code>if key in d.keys():</code>
size	<code>s.size()</code>	<code>s.size()</code>	<code>len(s)</code>
check empty	<code>if (s.empty())</code>	<code>if (s.isEmpty())</code>	<code>if not s:</code>



By **Ptero**
cheatography.com/ptero/

Not published yet.
Last updated 3rd July, 2024.
Page 3 of 3.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish Yours!
<https://apollopad.com>