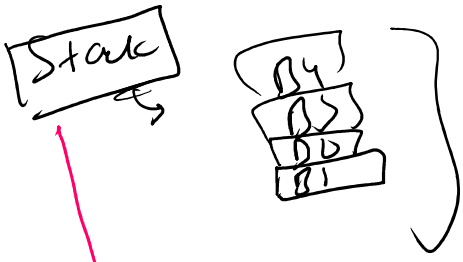
1.8 - Arrays - 1

Saturday, September 6, 2025 2:32 PM

Call Stack in Functions

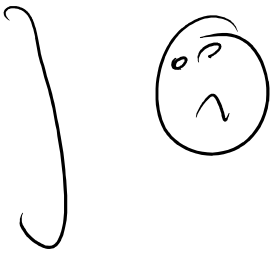



```
main(){
  fun1()
  fun2()
}
```

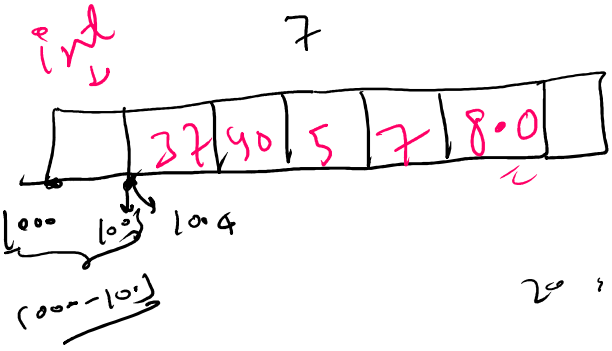
?

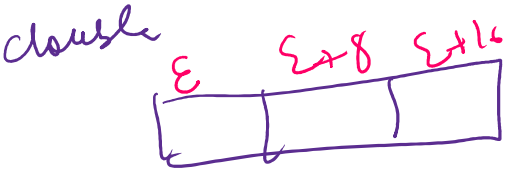
Arrays

```
int x1=100
int x2 =
  :
int x100
```



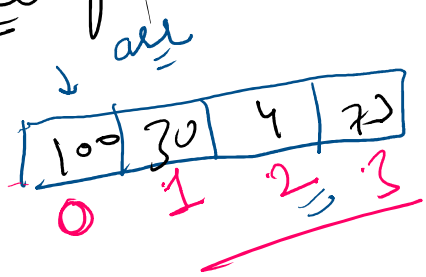
- Array is a data structure which stores multiple values of similar data types





Concept of indexing

Concept of indexing

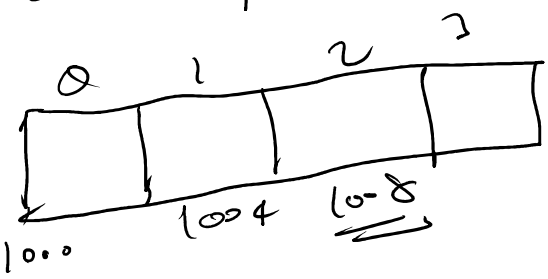


arr[2]

* indexing starts from 0

Why indexing starts from zero

arr



arr[0]
arr[1]
arr[2]
arr[3]

* Name of the array stores starting address of the array

arr[0] 1000 + 0X4 ⇒ 1000

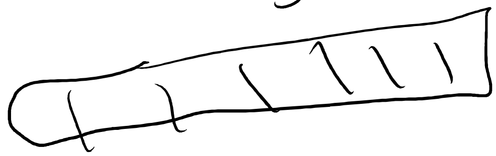
arr[1] 1000 + 1X4 ⇒ 1004

arr[2] 1000 + 2X4 = 1008

How to use arrays in code

```
int a = 10;
int[] arr = new int[5];
```

float[] a = new float[10];



⇒ Arrays are initialized with default values

int = 0
float = 0.0
double = 0.0

char = null
boolean = false

① If we try to access an index out of the range of array size, we get ArrayIndexOutOfBoundsException

[n] → [0, n-1]
(10) → [0-9]

Max in array

3 -2 5 1 4

-2
-2

5

cm = 5

int curMax = ar[0];

for (i = 1; i < n; i++) {

```
int curr...
for (i = 1; i < n; i++) {
    if (arr[i] > curr max) {
        curr max = arr[i];
    }
}
```

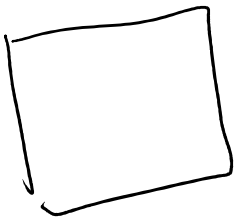
- arr.length → gives the size of the array.

How memory allocation happens

```
int arr[] = new int[10];
```

arr → reference

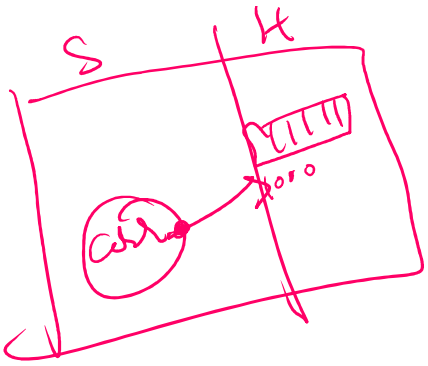
new int[10] → creating a object



Ram → Stack memory | static memory

→ Heap memory | dynamic memory

- Objects are always created in heap memory
- References ——— Stack



① stack memory allocation happens on compile time

② Heap ——————→ on run time →