Arrays

Yesterday :

- Nested Loops
- Printing Patterns
- Break , Continue.

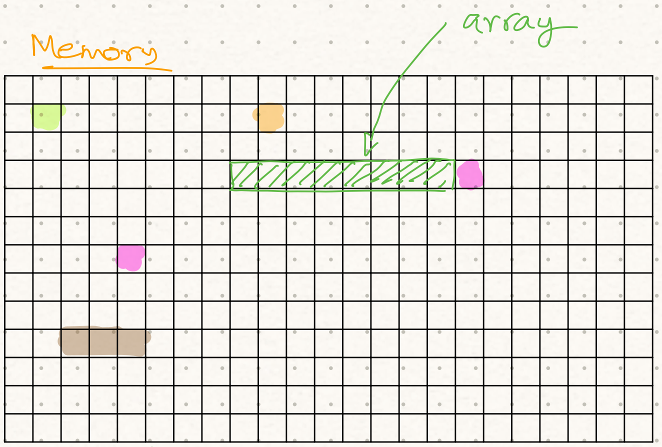
Arrays

- What ?
- Why ?
- How to create arrays in Java ?
- Problems.

```
String name1 = "Prateek";
String name2 = " Nrupol";
String name3 = "Yogesh";
String name4 = " Rajat";
String name5 = " Varun";
```

names	"Prateek"	"Nrupul"	"Yogesh"	"Rajat"	"Varun"

Arrays : a contiguous location in Memory to store a sequence of values (of the same Datatype)



Array Indexing

index

	0	1	2	3	4
names	"Prateek"	"Nrupul"	"Yogesh"	"Rajat"	"Varun"

↑ array name

elements of the array

5 elements

index

s.o.p(names[2]) → "Yogesh"

array name

Length (size) of array is 5.

A[-1] ✗

A[n] "Index Out of Bound Exception".

s.o.p (names[0]) --> "Prateek"

	0	1	2	3	.	.	.	.	i	.	.	.	.	n-1
A														

n elements

--> A[i] : element at index i

Java Syntax :

0	1	2	3	4	5	6

1. Declaring arrays
2. Creating Arrays (allocating memory)
3. Initialise Arrays

1. Declaring arrays

```
String[] names;
 int[]    numbers;
  double[]  values;
```

(Does not create arrays in memory)

2. Creating arrays

length of array

```
String[]  names  =  new  String[5];
 int  []    numbers  =  new  int[100];
boolean[]  flags  =  new  boolean [10];
```

T	F	T	F	F	T	F	T	

3. Initialise Arrays

```
int  n = 10;

int[] nums  = new  int[n];
```

	0	1	2	3	4	5	6	7	8	9
nums	0	0	0	0	0	0	0	0	0	0

default value.

10 elements

```
nums [0] = 1;
nums [2] = 6;
```

nums [9] = 9+1
= 10

nums [2]
= 2+1
= 3

i
0

	0	1	2	3	4	5	6	7	8	9
nums	1	2	3	4	5	6	7	8	9	10

i : ~~0~~ ~~1~~ ~~2~~ . . . ~~9~~ 10

nums [0] = 0+1
= 1

nums [1] = 1+1
= 2

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

for(int i = 0; i < 10; i++){
    nums[i] = i+1
}
```

$(4+1)^2$

$(5+1)^2 = 6 \times 6 = 36$

						i					
		0	1	2	3	4	5	6	7	8	9
Squares		1	4	9	16	25	36	49	64	81	100

```
int[] squares = new int[10];
 for (int i = 0; i < squares.length; i++) {
        squares[i] = (i+1) * (i+1);
 }
```

```
int[]   nums = {10, 8, 24, -5, 3};
```

	0	1	2	3	4
nums	10	8	24	-5	3

```
String[] names = {"Prateek", "Noopur", "Yogesh",
                           "Rajat", "varun"};

s.o.p (names.length);  // 5

 for (int i = 0; i < names.length; i++) {
          s.o.P (names[i]);

   }
```

Problem: Find the max value in the array.

			i				
	0	1	2	3	4	5	6
A	3	-5	12	6	16	-4	0

if element > max
update max

```
int max = A[0]; *
for (int i = 1 ; i < A.length ; i++) {
        if (A[i] > max) {
            max = A[i]
        }
}
print (max)
```

A[1] > max

-5 > 3

max = ~~3~~ ~~12~~ 16

i	i < A.length 7	Body	i++
1	1 < 7 ✓	A[1] > max ✗	2
2	2 < 7 ✓	A[2] > max ✓ $\overset{12}{\text{max}} = \overset{3}{A[2]}$	3
3	3 < 7 ✓	A[3] > max 6 > 12 ✗	4
4	4 < 7 ✓	A[4] > max 16 > 12 ✓ max = A[4]	5
5	5 < 7 ✓	A[5] > max ✗	6
6	6 < 7 ✓	A[6] > max ✗	7
7	7 < 7 ✗		

Prob: Find the sum of elements of the array.

sum: ~~0~~ ~~3~~ ~~-2~~ ~~10~~ 16

	0	1	2	3	4	5	6
A	3	-5	12	6	16	-4	0

3 + (-5) + 12
+ 6 + 16 + (-4)
+ 0
= 28

```
int sum = 0 ;
 for (int i = 0; i < A.length; i++) {
        sum = sum + A[i] ;      // sum += A[i].
 }
→ s.o.p (sum);
```

Sum: ~~0~~ ~~3~~ ~~-2~~ 10

A

0	1	2	3	4	5	6
3	-5	12	6	16	-4	0

```
int sum = 0 ;
for (int i = 0; i<A.length; i++) {
        sum  =  sum + A[i] ;
 }
→ s.o.p (sum);
```

i	i< A.length 7	Body	i++
0	0<7✓	Sum = 0+3 =3	1
1	1<7 ✓	sum = 3+(-5) = -2	2
2	2<7 ✓	Sum = -2 + 12 =10	3
3			
4			
⋮			
6			
7	7<7 ✗		