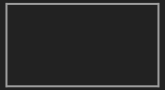
Arrays

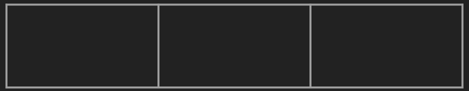
Arrays

Arrays are used for storing multiple elements of the same type. This could be a list of int or String or any other data type.

```
int grade;
```



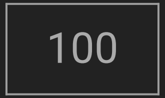
```
int[] grades = new int[3];
```



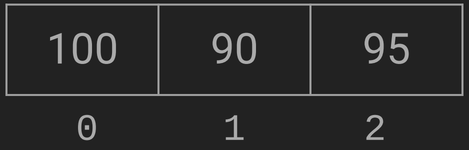
Arrays

You access elements of an array using an index. You will notice this is very similar to Strings (which are an array of characters).

```
int grade = 100;
```



```
int[] grades = new int[3];

grades[0] = 100;
grades[1] = 90;
grades[2] = 95;
```



Arrays

A constant is a good way to set the size of an array (in case you want to change it in the future).

```
final int NUM_GRADES = 5;

int[] grades = new int[NUM_GRADES];

for (int i = 0; i < NUM_GRADES; i++) {
    grades[i] = scan.nextInt();
}
```

Arrays

You can set the size of an array at run-time and use its length property (notice this is different than String’s length() method). Once you initialize an array, you can not alter its size.

```
System.out.println("How many grades are you entering?");
int gradeCount = scan.nextInt();

int[] grades = new int[gradeCount];

for (int i = 0; i < grades.length; i++) {
    grades[i] = scan.nextInt();
}
```

Arrays - Iterating

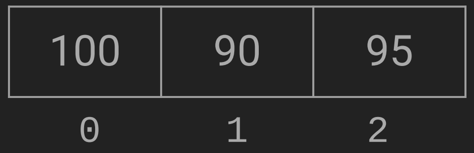
You can iterate through the elements in an array using loops.

```
for (int i = 0; i < grades.length; i++) {
     System.out.println(grades[i]);
}

String search = "Cyberpunk 2077";
int i = 0;

while (i < names.length) {
     if (names[i].equals(search)) {
          System.out.println("You’re breathtaking!");
          break;
     }
     i++;
}
```

Arrays - Initializing

You can initialize an array with values and change them later if you want.

```
int[] grades = { 100, 90, 95 };
```



Arrays and Strings

The code below creates an array of 3 Strings.

```
String[] names = new String[3];

names[0] = "Iron man";
names[1] = "Spider-man";
names[2] = "Batman";
```

Arrays and Strings

You can initialize an array of strings with values. Notice how we can select a random string from an array. (yeah, I know it would have been great to know this weeks ago).

```
String[] names = { "Iron man", "Spider-man", "Batman" };

int which = rand.nextInt(names.length);

String hero = names[which];

System.out.println("Hero Name: " + hero);
System.out.println("Hero name length: " + hero.length());
```

Let’s Code

Don’t Forget!

Check the syllabus / schedule for reading assignments and due dates!

Review

"for each"

"for each"

You can iterate through each element in an array using an enhanced loop. You can read this as "for each" element in array.

```
int[] grades = { 100, 80, 95, 90, 75, 93 };

for (int grade : grades) {
    System.out.println(grade);
}
```

"for each"

This works with other data types such as Strings.

```
String[] departmentList = { "CS", "ENG", "MAT" };

for (String department : departmentList) {
    if (department.equals("CS")) {
        System.out.println("I Love Computer Science!");
    } else {
        System.out.println("That’s cool I guess.");
    }
}


// Notice that department is only in the scope of the loop.
```

Multiple Arrays

Multiple Arrays

You may want to have multiple arrays of the **same size** to store multiple properties.

```
final int NUMBER_OF_STUDENTS = 20;

String[] studentNames = new String[NUMBER_OF_STUDENTS];
double[] studentGPAs = new double[NUMBER_OF_STUDENTS];

for (int i = 0; i < NUMBER_OF_STUDENTS; i++) {
    System.out.println("Enter Name: ");
    studentNames[i] = scan.nextLine();

    System.out.println("Enter GPA: ");
    studentGPAs[i] = scan.nextDouble();
}
```

Comparing Arrays

You may want to iterate through 2 arrays and compare them

```
for (int i = 0; i < NUMBER_OF_STUDENTS; i++) {
    if (springGrades[i] > fallGrades[i]) {
        System.out.println("I see Improvement!");
    }
}
```

Updating Array Elements

Updating Array Elements

You can update the values of elements in an array.

```
// Initialize the list
for (int i = 0; i < numbers.length; i++) {
    numbers[i] = i;
}


// Update it
for (int i = 0; i < numbers.length; i++) {
    numbers[i] = numbers[i] * 2;
}
```

Swapping the Values of Two Variables

Swapping Values

Sometimes we need to swap values. We need to be careful not to lose information.

```
// Will not work!
int x = 5;
int y = 13;

x = y;
y = x;


Result:
x = 13
y = 13
```

```
// Use a temp variable!
    int x = 5;
    int y = 13;
    int temp;

    temp = x;    // Save X
    x = y;
    y = temp;

    Result:
    temp = 5
    x = 13
    y = 5
```

Swapping Array Elements

You need a temp variable to swap array elements.

```
// Will not work!
a[0] = a[4];
a[4] = a[0];
```

```
// Use a temp variable!
    int temp;

    temp = a[0];
    a[0] = a[4];
    a[4] = temp;
```

2-Dimensional Arrays

2-Dimensional Arrays

So far, your array has been a simple list. You can have a two dimensional array which is an array of arrays!

```
int[][] ticTacToe = new int[3][3];
```

0, 0	0, 1	0, 2
1, 0	1, 1	1, 2
2, 0	2, 1	2, 2

2-Dimensional Arrays

```
int[][] grid = new int[3][3];

for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        grid[i][j] = i + j;
    }
}
```

0	1	2
1	2	3
2	3	4

i	j
0	0
0	1
0	2
1	0
1	1
1	2
2	0
2	1
2	2

2-Dimensional Arrays

You can initialize a 2-dimensional array in a similar way to a regular array.

```
char[][] board = { {'X', '.', 'O'}, {'O', 'X', '.'}, {'.', '.', 'X '} };

for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        System.out.print(board[i][j]);
    }
    System.out.println();
}

Result:
X.O
OX.
..X
```

Let's Code

Don't Forget!

Check the syllabus / schedule for reading assignments and due dates!