Java **Arrays**

{ OBJECT , OBJECT , OBJECT , OBJECT , OBJECT , OBJECT , OBJECT }

Primary **Concepts**

Working with **Array Elements**	**Iterating**
The **Arrays** Class	**Two-Dimensional** Arrays

Comparisons with **Javascript**

Javascript

```
var books = [];

var grades = [];

var temps = [71, 68, “red”, 72, 78];
```

Java

```
String[] books = new String[5];

int[] books = new int[25];

int[] temps = {71, 68, 75, 72, 78};
```

Initializing **Arrays**

```
String[] books;
books = new String[5];

String[] books = new String[5];

int[] temps = {71, 68, 75, 72, 78};

int[] temps = new int[25];
```

Initializing **Arrays**

```
String[] books = new String[5]; // literal

final int COUNT = 25; // constant
String[] books = new String[COUNT];

int len = (int) Math.floor(Math.random() * 100);
String[] books = new String[len]; // variable
```

Working with Array **Elements**

Type	Default Value
integer number types	0
decimal number types	0.0
boolean	FALSE
character	\0 (zero)
object	null

```
int[] numbers = new int[3];
numbers[0] = 1;
numbers[1] = 2;
```

```
System.out.println(numbers[0]); // 1
System.out.println(numbers[1]); // 2
System.out.println(numbers[2]); // 0 -- default value
System.out.println(numbers[3]); // ArrayIndexOutOfBoundsException !!!
```

Working with Array **Elements**

```
// using the array initializer syntax
String[] tech = {"MacBook", "iPad", "watch"};

System.out.println(tech.length); // 3

System.out.println(tech[0]); // "MacBook"
System.out.println(tech[1]); // "iPad"
System.out.println(tech[2]); // “watch"

// ArrayIndexOutOfBoundsException !
tech[3] = "iPad";
```

Iterating **Arrays**

```
String[] languages = {"html", "css", "javascript", "java"};

for (int i = 0; i < languages.length; i += 1) {
    System.out.println(languages[i]);
}
```

Iterating **Arrays**

```java
String[] languages = {"html", "css", "javascript", "java"};

for (String language : languages) {
    System.out.println(language);
}

// html
// css
// javascript
// java
```

Iterating **Arrays**

```
int[] numbers = {1, 2, 3, 4, 5};
for (int n : numbers) {
    System.out.println(n);
}
// 1
// 2
// 3
// 4
// 5
```

Arrays Live Coding

Working with Elements
Iterating

Live coding in IntelliJ

The **Arrays** Class

```
import java.util.Arrays;

void Arrays.fill(array, value);
    // Fills all elements with ‘value’
boolean Arrays.equals(array1, array2);
    // Returns true if type and elements are equal
array Arrays.copyOf(array, length);
    // Returns a copy of the array (specified length)
String Arrays.toString(array);
    // Returns a String representation of the array
void Arrays.sort(array);
    // Sorts the elements into ascending order
```

Two-Dimensional Arrays

```java
int[][] matrix = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
};

// access the first element in the second row
System.out.println(matrix[1][0]); // 4
// the last element in the first row
System.out.println(matrix[0][2]); // 3
// the first element in the last row
System.out.println(matrix[2][0]); // 7
```

Two-Dimensional Arrays

```
int[][] matrix = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
};
```

```
for (int[] row : matrix) {
    System.out.println("+---+---+---+");

    System.out.print("| ");

    for (int n : row) {
        System.out.print(n + " | ");
    }

    System.out.println();
}

System.out.println("+---+---+---+");
```

```
// Output
+---+---+---+
| 1 | 2 | 3 |
+---+---+---+
| 4 | 5 | 6 |
+---+---+---+
| 7 | 8 | 9 |
+---+---+---+
```

Arrays Live Coding

The Arrays Class

Two-Dimensional Arrays

Live coding in IntelliJ

wrapup

working with elements - declare type,
define length when initializing

iterating - we can use a regular for loop,
or an enhanced for

Arrays class - manipulate arrays using
built-in Java methods

2D arrays - create matrices of arrays for
readable console output