

Fast arrays

1-dimensional array

Declaration:

```
int[] array = new int[4];
```


1-dimensional array

Declaration:

```
int[] array = new int[4];
```

Access:

```
int i = array[2];
```


2-dimensional array

Declaration:

```
int[,] array = new int[2,2];
```


2-dimensional array

Declaration:

```
int[,] array = new int[2,2];
```

Access:

```
int i = array[1,1];
```


jagged array

Declaration:

```
int[][] array = new int[2][];
```


jagged array

Declaration:

```
int[][] array = new int[2][];
```

Initialisation:

```
array[0] = new int[2];
```

```
array[1] = new int[2];
```


jagged array

Declaration:

```
int[][] array = new int[2][];
```

Initialisation:

```
array[0] = new int[2];  
array[1] = new int[2];
```

Access:

```
int i = array[1][1];
```


jagged array

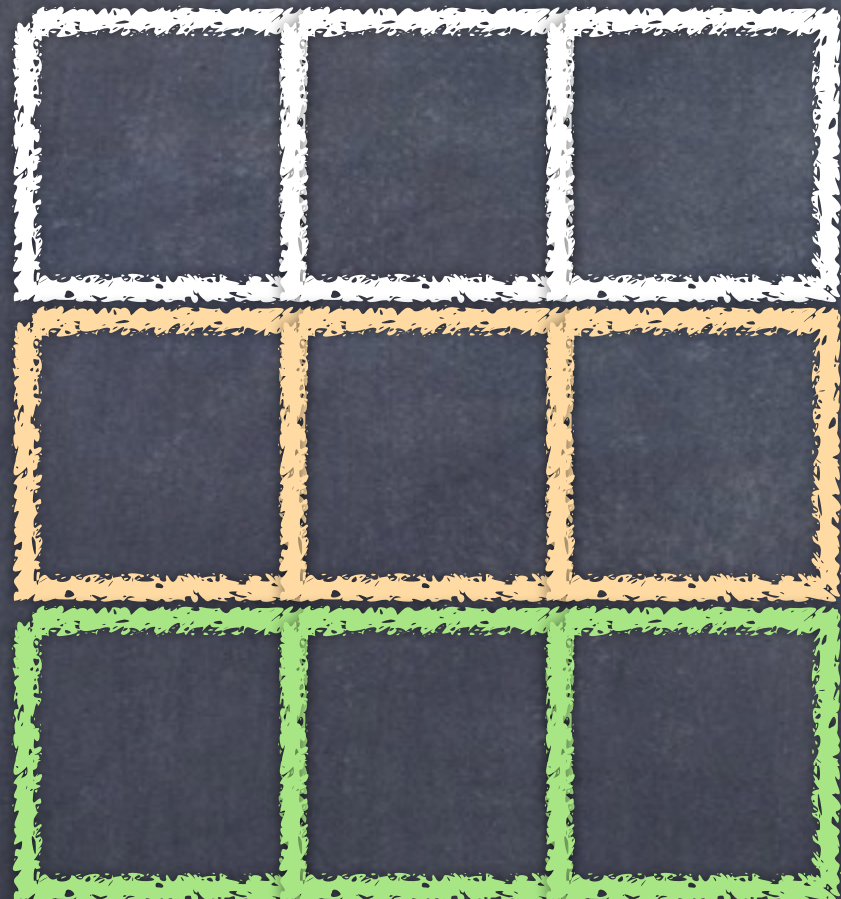


Second row only has 1 element:

```
int[][] row = new int[2][];  
row[0] = new int[2];  
row[1] = new int[1];
```


Array flattening

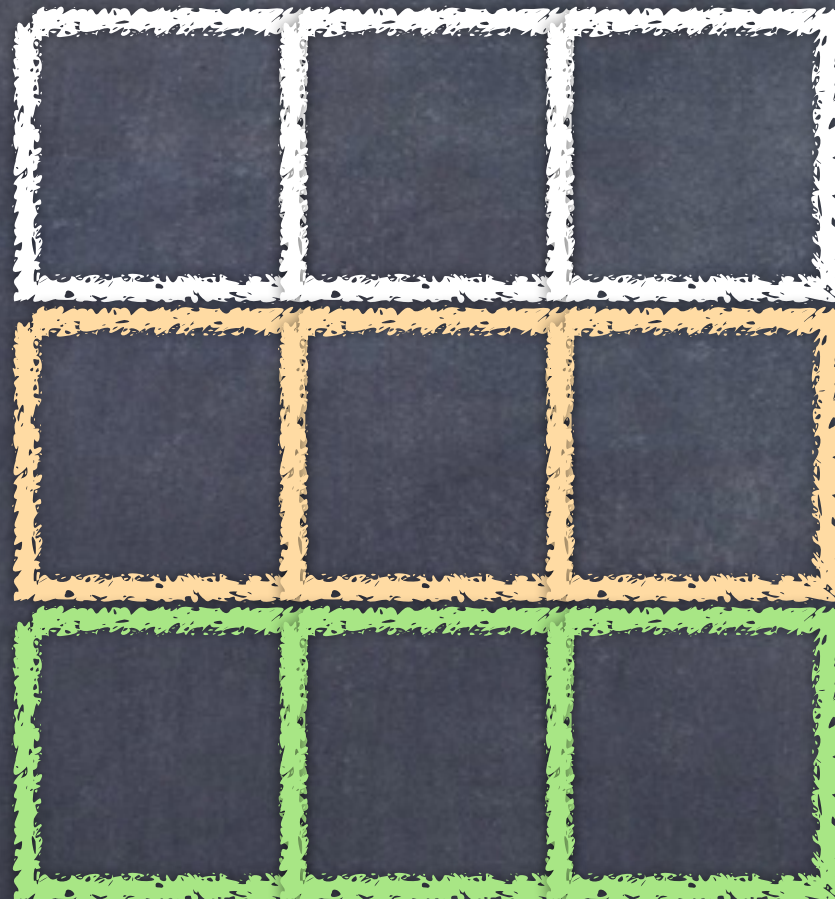
Array flattening



Array flattening



Array flattening



Indexing:

$\text{int index} = 3 * \text{row} + \text{col};$

Fast arrays

- If you only have 1 dimension of data, use 1-dimensional arrays for the best performance.
- If you have 2 dimensions of data, flatten the array.
- If this is not possible, consider using a jagged array.
- If there is no other option, use a 2-dimensional array