# Importing from a Data Source into an Array



Simon Robinson
LEAD SOFTWARE DEVELOPER

@TechieSimon www.SimonRobinson.com



#### Overview

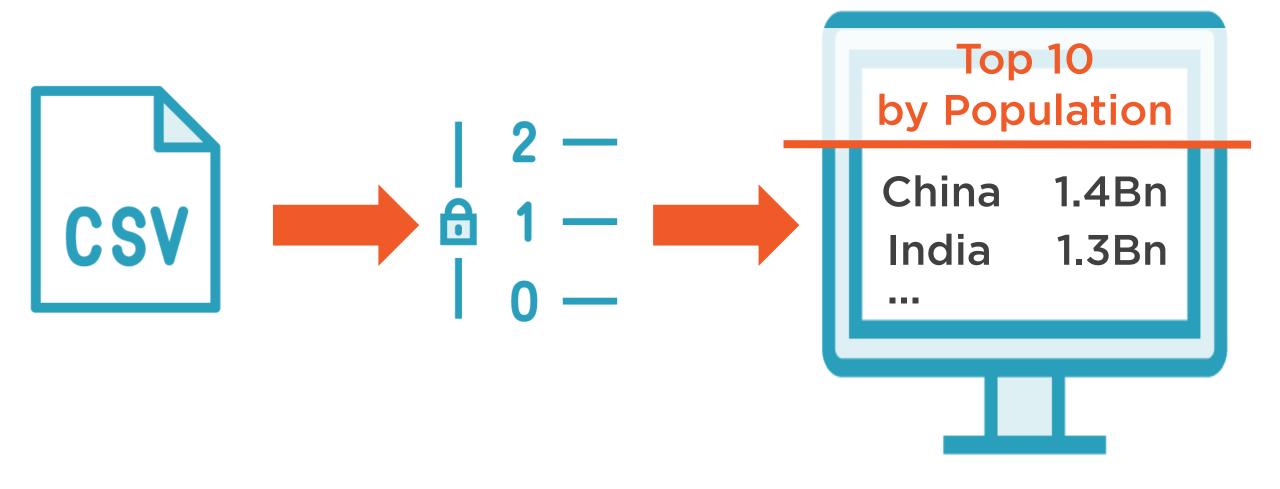


#### Introduce sample demo

- Dynamically put data in an array
- Uninitialized array contains nulls
- Arrays are ubiquitous



#### Countries Demo App



#### Demo



Read top 10 countries from CSV file



# CSV/web Demo



#### Previously...

```
string[] daysOfWeek = {
    "Monday",
    "Tuesday",
    "Wednesday",
    "Thursday",
    "Friday",
    "Saturday",
    "Sunday"
};
```

This is a Collection Initializer (Array initializer)

Can't do this
if you don't know
the values
when the array is instantiated



# CSV/web Demo



#### Formatting Population

1339180127



Round

1339000000



**Space** 

1 339 000 000



## Demos



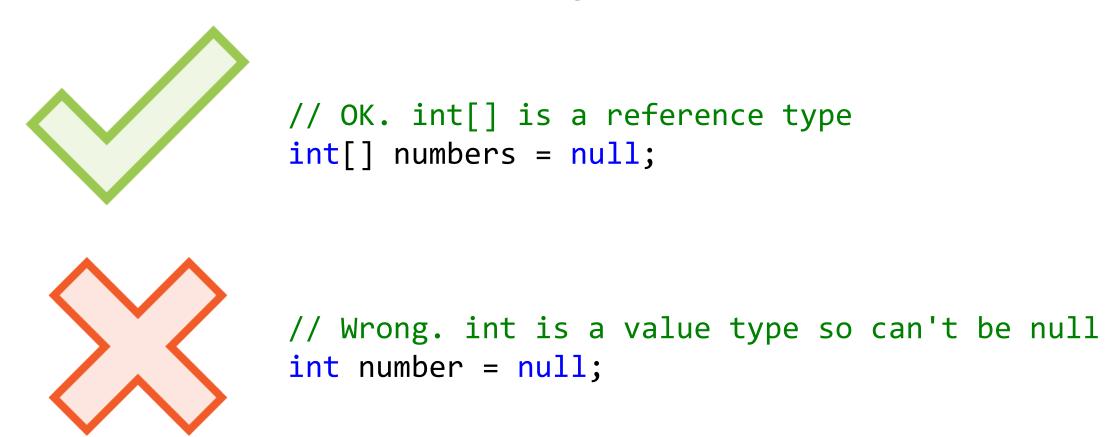
#### Instantiating an Array

```
Country[] countries = null;
```

Declaring, not instantiating



#### Arrays Are Always Reference Types



#### Instantiating an Array

```
// nValues is an int
Country[] countries = new Country[nValues];
```

countries will contain all nulls

Minimum information you must provide

```
// nValues is an int
int[] ints = new int[nValues];
```

ints will contain all zeros



### Instantiating an Array

```
// country1, country2 etc. are of type Country
Country[] countries = new Country[]
{
      country1, country2, country3, country4
}
```

Specifying all values



#### Summary



#### Demo: Import data into an array

- Initialize an array by size
- Array starts full of null/default values
- Can populate with for loop
- Arrays used in .NET Framework

