

## Review

- Strings
  - Strings are objects that hold an array of chars
  - Making Strings
  - ASCII Encoding
  - String methods and functions
  - Comparing Strings
  - Building Strings
  - Splitting Strings into an array
  - Joining the elements of a String array into a String

## WHO Tuberculosis Data

The screenshot shows a web browser displaying the WHO Tuberculosis (TB) data download page. The URL is <http://www.who.int/tb/country/data/download/en/index.html>. The page has a header with the WHO logo and navigation links for Health topics, Data and statistics, Media centre, Publications, Countries, Programmes and projects, and About WHO. Below the header, there is a section titled "Tuberculosis (TB)" with a sub-section "Download data as CSV files". It provides information about the data being provided by countries to WHO and estimates of TB burden generated by WHO for Global Tuberculosis Control 2010 are available for download as comma-separated value (CSV) files. CSV files can be imported into many spreadsheet and database programs, and some spreadsheet such as Excel can open CSV files directly. A link "Download the data dictionary" is also present.

<http://www.who.int/tb/country/data/download/en/index.html>

This screenshot shows two sheets in an Excel spreadsheet. The active sheet is named 'country' and contains data for various countries. The columns include country names, population, and various tuberculosis metrics. The second sheet, 'Sheet1', is titled 'Data dictionary' and lists variable names, datasets, and their definitions, such as 'e\_pop\_num' for population and 'e\_prew\_100k' for estimated incidence per 100,000 population.

This screenshot shows a Notepad++ window displaying the contents of a CSV file named 'reduced.csv'. The file contains a list of countries with their names and corresponding values for various variables, including population and tuberculosis metrics. The file is in plain text format with commas as separators.

## Loading Data From a File

- `loadStrings()` function
  - Reads all data and returns an array of Strings
  - Each String in the array is a separate line from the file

```
// ParseFile1

String[] data;
int count = 0;

void setup() {
  // Load data from a file as array of strings
  data = loadStrings("reduced.csv");
}

void draw() {
  // Continue printing data until run out
  if (count >= data.length) return;
  println(data[count]);
  count++;
}
```

## Split a String based on a single or multiple separator chars

```
String s1 = "12, 34, 56";
String[] as;

void setup() {
  as = split(s1, ",");
  //as = trim(as);
  println( as );
}
```

```
[0] "12"
[1] "34"
[2] "56"
```

```
String s1 = "Data: 12, 34, 56";
String[] as;

void setup() {
  as = splitTokens(s1, ":" );
  //as = trim(as);
  println( as );
}
```

```
[0] "Data"
[1] "12"
[2] "34"
[3] "56"
```

```

// ParseFile2

String[] data;
Item[] items;
int count = 0;

void setup() {
    // Load data as array of strings
    data = loadStrings("reduced.csv");

    // Build object array
    items = new Item[data.length];
    for (int i=0; i<data.length; i++) {
        items[i] = new Item(data[i]);
    }
}

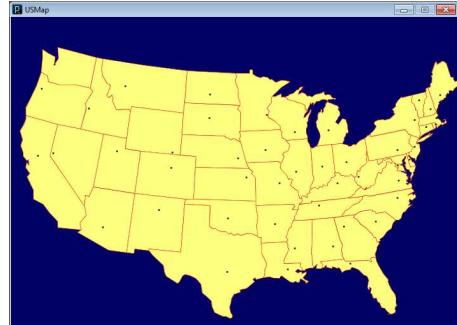
void draw() {
    // Continue printing data until run out
    if (count >= items.length) return;
    items[count].pr();
    count++;
}
}

class Item {
    String country; // Country name
    int year; // Year
    int pop; // Population
    int inc; // Incidences of TB
    // per 100,000
}

Item(String line) {
    String[] data = split(line, ",");
    country = data[0];
    year = int(data[1]);
    pop = int(data[2]);
    inc = int(data[3]);
}

void pr() {
    String msg = "In " + year + ", " + country;
    msg += " had population " + pop;
    msg += " and TB incidences per 100k of " + inc;
    println(msg);
}
}

```



USMap.pde, USMap2.pde

### File Selection

- **selectInput()** function
  - Displays a file chooser allowing user to select a file
  - Returns a complete path to selected file as a String

```

void draw() { }

void mousePressed()
{
    String filepath = selectInput("Please select a data file");
    println(filepath);
}

```

filepath.pde

Also see **selectOutput()**

```

// stripSpaces

void setup() {
    String s = "abc def vghi";
    String s3 = stripSpaces(s);
    println(s3);
}

// Remove spaces from a string
String stripSpaces( String s ) {

    for (int i=s.length()-1; i>=0; i--) {
        char c = s.charAt(i);
        if (c == ' ') {
            String s1 = s.substring(0, i);
            String s2 = s.substring(i+1);
            s = s1 + s2;
        }
    }
    return s;
}

```

stripSpaces.pde