Working with Text

Fonts

- PFont.list();
  - Returns a String array of available font names

```
println( PFont.list() );
```

- createFont(*name*, *size*);
- createFont(*name*, *size*, *smooth*);
  - Returns a PFont object to be used by Processing

- textFont(*font*)
- textFont(*font*, *size*);
  - Sets the current font

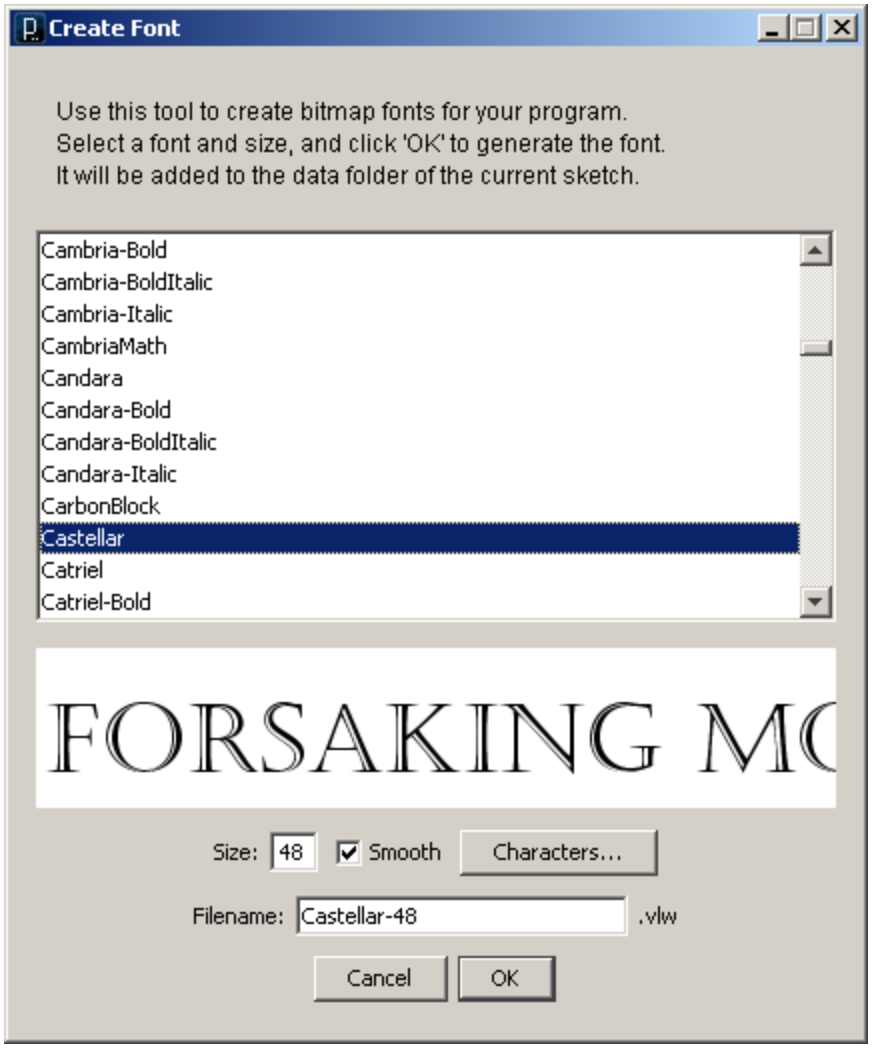
- textAscent() / textDescent();
  - Returns height of current font above / below baseline

- textWidth(*string*)
  - Returns the width of a string using the current font

Create Font Utility

- Load created font files into a PFont object using…

```
loadFont( filename );
```

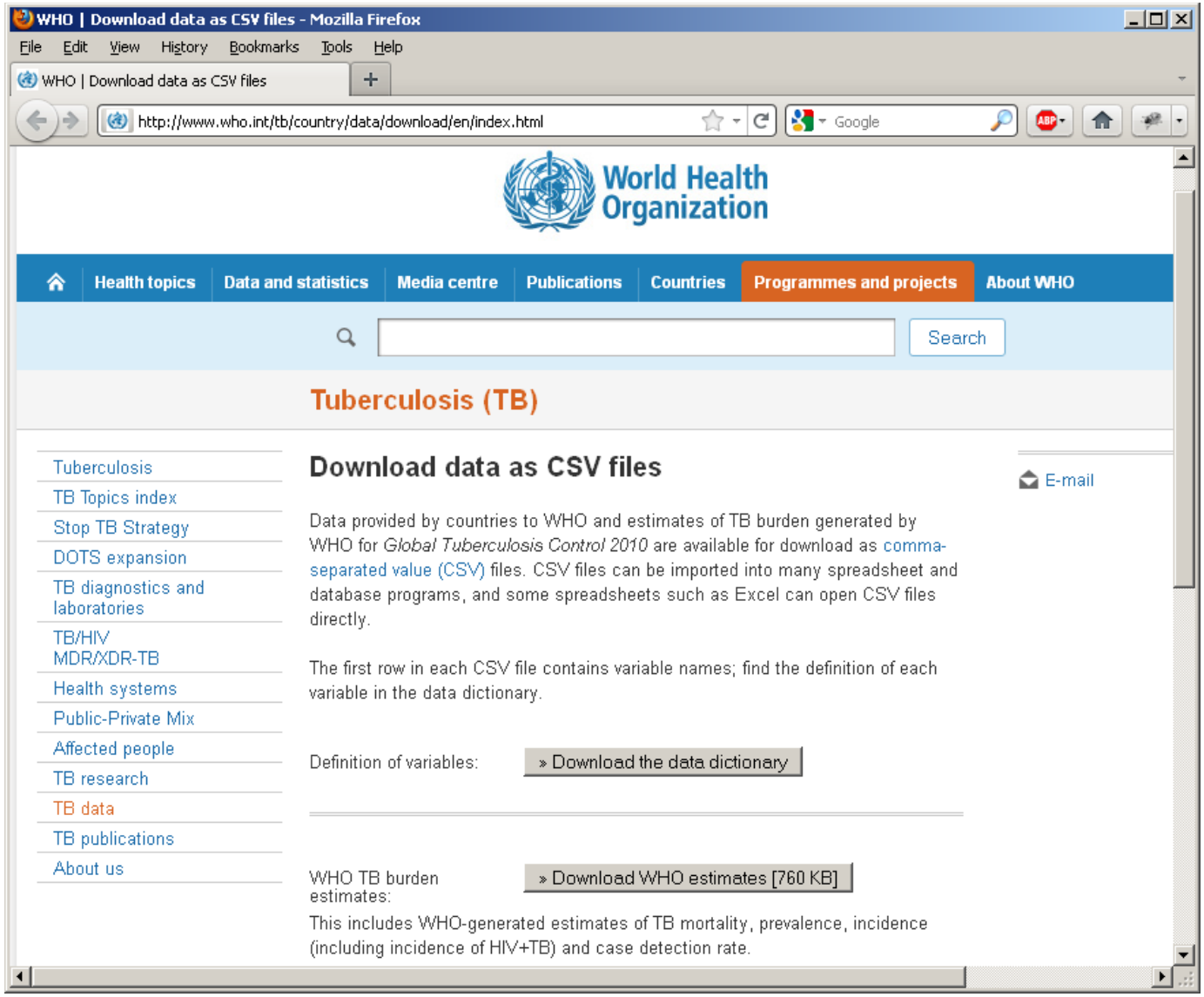


```
// Ticker

String headline = "This is a Headline!";

PFont f;
float x, y;
float w;
float fs = 48;

void setup() {
  size(800, 600);
  f = createFont("Amienne", fs, true);
  x = width;
  y = height/2;
}

void draw() {
  background(255);
  textFont(f, fs);
  w = textWidth(headline);
  fill(0);
  text(headline, x, y);

  x = x - 3;

  if (x < -w) {
    x = width;
    fs = (int)random(20, 48);
  }
}
```

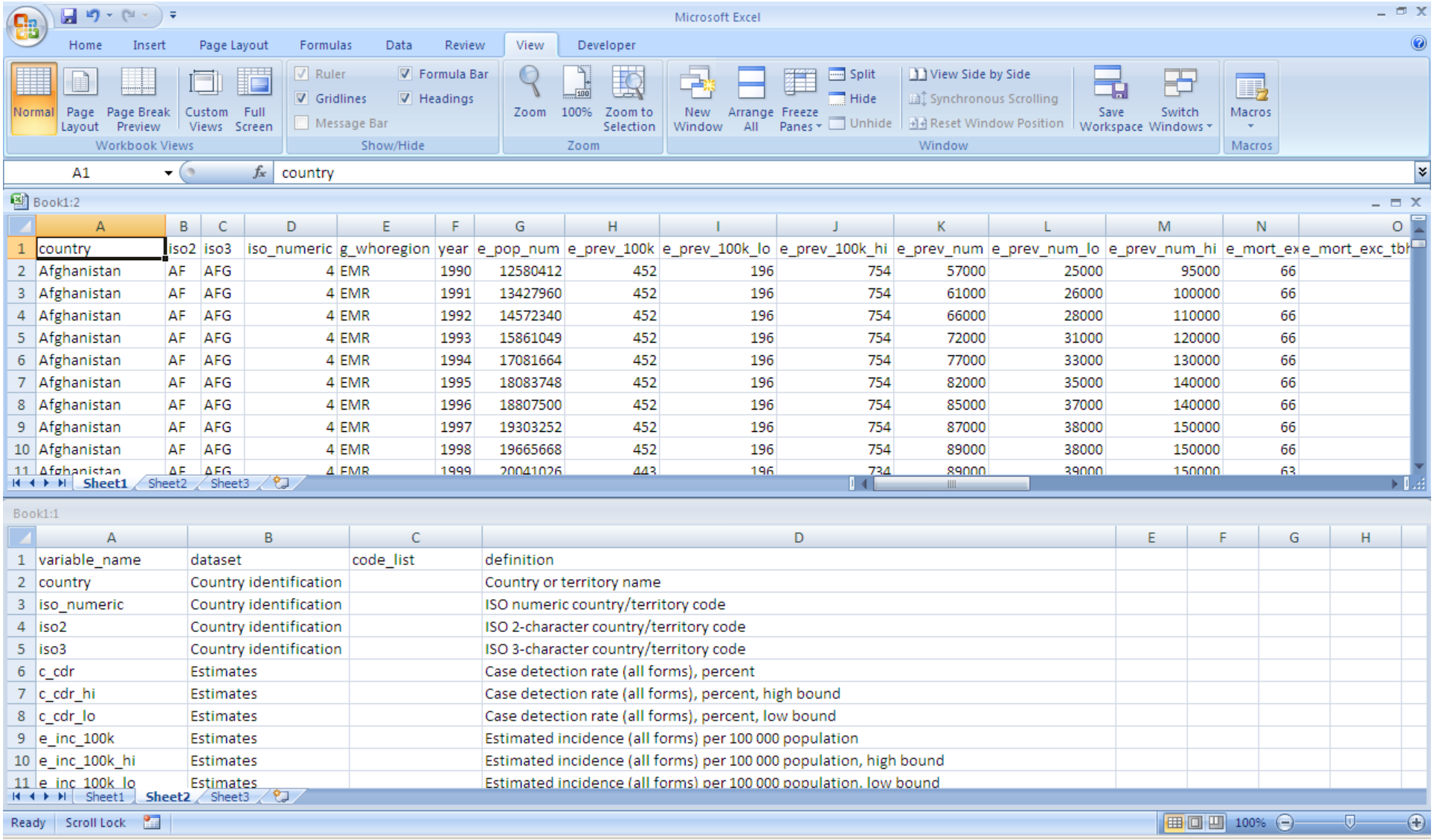
```
// TextFun

String headline = "Bryn Mawr";
float theta, delta;
PFont f;
float x, y;
float w;
float fs = 32;

void setup() {
  size(800, 600, P3D);
  f = createFont("Arial", fs, true);
  textAlign(CENTER);
  x = width/2;
  y = height/2;
  theta = 0;
  delta = 0.05;
}

void draw() {
  background(255);

  translate(x, y);
  rotateY(theta);
  textFont(f, fs);
  w = textWidth(headline);
  fill(0);
  text(headline, 0, 0);
  theta = theta + delta;
}
```

```
// TextCrawl
// TS Eliot - The Love Song of J Alfred Prufrock

String s = "LET us go then, …";
PFont f;
float y;

void setup() {
  size(800, 600, P3D);
  f = createFont("Arial Bold", 20, true);
  y = height;
  println(PFont.list());
}

void draw() {
  background(0);
  fill(255, 255, 0);
  translate(width/2, y);
  rotateX(PI/4);
  textFont(f);
  textAlign(CENTER);
  text(s, 0, 0);
  y--;

  if (y < 0) y = height;

}
```

WHO Tuberculosis Data



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



Book1:2

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	country	iso2	iso3	iso_numeric	g_whoregion	year	e_pop_num	e_prev_100k	e_prev_100k_lo	e_prev_100k_hi	e_prev_num	e_prev_num_lo	e_prev_num_hi	e_mort_ex	e_mort_exc_tbh
2	Afghanistan	AF	AFG	4	EMR	1990	12580412	452	196	754	57000	25000	95000	66	
3	Afghanistan	AF	AFG	4	EMR	1991	13427960	452	196	754	61000	26000	100000	66	
4	Afghanistan	AF	AFG	4	EMR	1992	14572340	452	196	754	66000	28000	110000	66	
5	Afghanistan	AF	AFG	4	EMR	1993	15861049	452	196	754	72000	31000	120000	66	
6	Afghanistan	AF	AFG	4	EMR	1994	17081664	452	196	754	77000	33000	130000	66	
7	Afghanistan	AF	AFG	4	EMR	1995	18083748	452	196	754	82000	35000	140000	66	
8	Afghanistan	AF	AFG	4	EMR	1996	18807500	452	196	754	85000	37000	140000	66	
9	Afghanistan	AF	AFG	4	EMR	1997	19303252	452	196	754	87000	38000	150000	66	
10	Afghanistan	AF	AFG	4	EMR	1998	19665668	452	196	754	89000	38000	150000	66	
11	Afghanistan	AF	AFG	4	EMR	1999	20041026	443	196	734	89000	39000	150000	63	

Book1:1

	A	B	C	D	E	F	G	H
1	variable_name	dataset	code_list	definition				
2	country	Country identification		Country or territory name				
3	iso_numeric	Country identification		ISO numeric country/territory code				
4	iso2	Country identification		ISO 2-character country/territory code				
5	iso3	Country identification		ISO 3-character country/territory code				
6	c_cdr	Estimates		Case detection rate (all forms), percent				
7	c_cdr_hi	Estimates		Case detection rate (all forms), percent, high bound				
8	c_cdr_lo	Estimates		Case detection rate (all forms), percent, low bound				
9	e_inc_100k	Estimates		Estimated incidence (all forms) per 100 000 population				
10	e_inc_100k_hi	Estimates		Estimated incidence (all forms) per 100 000 population, high bound				
11	e_inc_100k_lo	Estimates		Estimated incidence (all forms) per 100 000 population, low bound				

```
// 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++;
}
```

```
// 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 = line.split(",");
    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);
  }  }
}
```

Data Sources

http://www.data.gov/
http://archive.ics.uci.edu/ml/
http://opendata.socrata.com/
http://www.reddit.com/r/datasets

Data Source Lists

http://www.quora.com/Data/Where-can-I-get-large-datasets-open-to-the-public
http://www.readwriteweb.com/archives/where_to_find_open_data_on_the.php

Ideas for Visualizations

http://www.visual-literacy.org/periodic_table/periodic_table.html

GapMinder

http://www.gapminder.org

http://www.gapminder.org/videos/hans-rosling-on-cnn-us-in-a-converging-world/

Hans Roesling

Karolinska Institutet

Stockholm, Sweden