From Data to Pixels

Day 1, Markku Reunanen

Kickoff

Welcome!

This time online

Contents of the course

Requirements, attendance

Machines & Software

Breakout rooms – let's try at least...

OOP recap and some more

Encapsulation – things that belong together go together

Classes and objects

Attributes (variables and arrays) and methods (functions)

Constructor – what we do when an object is created

PImage, PFont, String … we've been using objects before

Inheritance

"All bourbon is whiskey, but not all whiskey is bourbon"

Inheriting a class from another: adding new attributes and methods, modifying existing ones

Base class (keywod *super*) and derived class (also subclass)

In Java/Processing the keyword is *extends*

Protecting class members

private – not visible from outside, not inherited

protected – not visible from outside, inherited

public (default) – visible from outside, inherited

Because of how Processing works these don't exactly work as described above but in plain Java and many other languages yes

Object arrays

We can make arrays out of objects (almost) as with any type:

PImage pics[]=new PImage[10];

Note that this didn't yet create the objects, just a placeholder where we can start putting them

Ratios

a/b = c/d

Solving for a: a = b*c/d

And so on, this is your high (or even elementary school) math

For a column chart:

barheight/height = value/maxvalue

There is *map()* in Processing, but we don't really need it