

**Art 742b: Networks & Transactions  
Spring 2012**

**Green Hall, Room 209**

**Lab: Mondays 1:30–3:00**

**Studio: Mondays 3:00–5:30**

**Class website: <http://art.yale.edu/Art742b>**

**Class email list: [networks1@panlists.yale.edu](mailto:networks1@panlists.yale.edu)**

**Dan Michaelson: [dan.michaelson@yale.edu](mailto:dan.michaelson@yale.edu)**

**AIM or Skype: danmichaelson**

**203-606-4492**

**TAs:**

**Weiye Li: [weiye.li@yale.edu](mailto:weiye.li@yale.edu)**

**Harry Gassel: [harry.gassel@yale.edu](mailto:harry.gassel@yale.edu)**

**Class goal**

How can graphic design influence and be influenced by the unpredictable encounters between one group and another? Or between quantities of unknown users on one side, and vast webs of fluctuating information on the other? In the Networks & Transactions sequence we will develop together a typography appropriate for these pervasive conditions of the modern world, by means of close synthesis of reading, hands-on practice, visual design, and the design of social, network, and movement forms. This is the first class of the sequence.

Attendance each week is essential, as is weekly programming practice between classes, as well as weekly design progress. I will also hand out a reading most weeks, which you are expected to read and discuss the following week.

## Why PHP?

In the programming lab each Monday, you'll learn PHP, a very popular programming language. No previous programming experience, of any kind, is assumed. Each week you'll learn a little more PHP through hands-on lab demonstrations. And you will complete design assignments of increasing complexity that reinforce those skills, and which also allow you to engage important design issues by confronting unavoidable questions posed by the logic of your computer and its network.

I've selected PHP for three reasons. First, PHP is a simple language to learn and not confusing. For non-programmers its syntax is easier to learn than many languages, and to use it all you need is a text editor and a web server, which has been set up for us.

Second, PHP projects are connected nodes. Your PHP programs have seamless access to content from the internet and from databases, with little difficulty-- a major focus of the class. Furthermore, your projects can be used immediately by anyone in the world with any web browser, without any special effort on your part to publish or deploy the project. PHP can produce HTML of any design, as well as images and PDFs. Users don't need a plugin to use your project, nor do they need to download your software.

Third, learning PHP is worth your time. It is a widely adopted and useful language, not a teaching language. PHP is free and open-source, and has excellent documentation. It is most well known as a language that enables the dynamic content for websites such as Yahoo; more than 15 million websites use custom PHP programs to generate web pages comprised of dynamic content. But, as in our class, PHP can also be used for networked projects other than traditional websites. Perhaps more important than PHP's ubiquity, is that as a fellow offspring of the C programming language, PHP's basic syntax and operation is similar to sister languages ActionScript (used by Flash), JavaScript (used in web browsers), Java, and Perl; so once you thoroughly learn PHP in this class, you'll find it very easy to pick up other environments in the future, such as Flash in Networks & Transactions 2.

## Required text

Robin Nixon. *Learning PHP, MySQL, and JavaScript*. O'Reilly, 2009.

<http://www.amazon.com/Learning-MySQL-JavaScript-Step-Step/dp/0596157134>

Also available online through Yale subscription:

<http://proquest.safaribooksonline.com/book/databases/mysql/9780596803605/firstchapter>

## Required software

TextWrangler:

<http://barebones.com/products/textwrangler/download>

You must also obtain an account on Bert, the Art School development server, by requesting it from Johannes in person. Your account needs "shell access."

## Assignments

### *Assignment 1. Junction*

This project need not use any programming.

Duration: 3 weeks.

“The contemporary image is typified precisely by its generative power; it is no longer a trace (retroactive) but a programme (active).” – Nicolas Bourriard, *Relational Aesthetics* (1998)

Identify two flows in the city of New Haven (or New York). For example, cups of coffee sold at Dunkin Donuts, live music listings, bus schedules, historical change in areas of the city, weather patterns, or the circulation of people within and between neighborhoods.

Consider how these two flows might intersect. For example, could a display in a bodega tell customers about lectures at Yale. How would its design engage both the bodega and the university?

What is an appropriate and beautiful typography and movement for the junction you’re creating? **Consider especially:** How can elisions in your design, negative space in its movement, or syncopation in its rhythm help communicate more powerfully than what’s said explicitly? (See also, connotation vs. denotation.)

### *Assignment 2. Weather.*

Design a display of current weather conditions and/or the weather forecast-- either for a single city (such as New Haven) or for multiple cities (which, if you wish, could be selected by the user). What kind of design system will you create to accommodate all possible weather conditions? Consider how your own agency as a designer can enter into such a project: as a system, what is the weather in your opinion? Yahoo provides a web service which will provide weather data to your program; I will show you how to use it. **Consider especially:** Graphic design as a kit of parts. At the same time, remember you are also formulating a distinct attitude about the weather as a system or flux, from the perspective of your thesis, your practice, or what you feel graphic design is today...

### *Assignment 3. Content management system.*

Create a simple content management system. Consider who may edit it (only you, a select group, a large group, or anyone), and who will be viewing it (only you, a select group, or a large group). What is the relationship between the following aspects:

- content
- content structure
- user base
- the template you create for input
- and the template you create for output and links among content?

**Consider especially:** Like a weather system, how does your system evolve over time?

## Readings

### Week 1

Josef Albers, "Regarding the Economy of Typeface"

Wim Cuyvers, "Planks and Bricks"

### Week 2

Excerpt from Thomas Pynchon, *The Crying of Lot 49*

Paul Elliman, "Token Resistance"

### Week 3

Dan Michaelson, "On Exactitude in Science"

### Week 4

Excerpt from James Gleick, *Chaos*

### Week 5

Excerpt from Hans Ulrich Obrist, *Cedric Price*

Excerpt from Olafur Eliasson, *Take Your Time Volume 2: Printed Matter*

### Week 6

Clay Shirky, "Group as User: Flaming and the Design of Social Software"