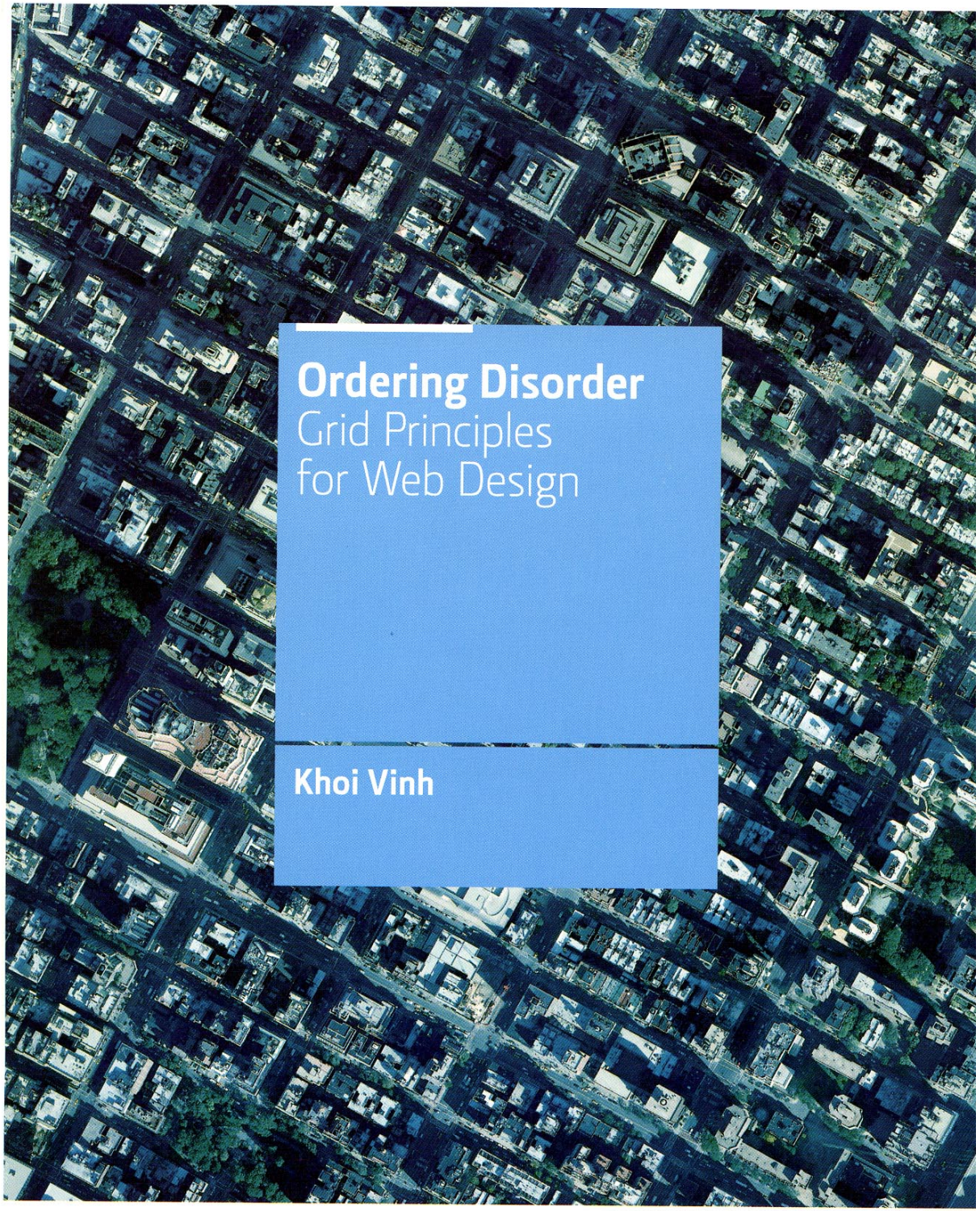


Ordering Disorder
Grid Principles for Web Design

Khoi Vinh

Among experienced graphic designers working in all manner of media, the many benefits of designing with typographic grids are well known. It's worthwhile to recount the major ones here:

- Grids add order, continuity, and harmony to the presentation of information.
- Grids allow an audience to predict where to find information, which aids in the communication of that information.
- Grids make it easier to add new content in a manner consistent with the overall vision of the original presentation.
- Grids facilitate collaboration on the design of a single solution without compromising the overall vision of that solution.

These are just some of the points that you'll read in advocacy of grid-based design, but until recently they've been written primarily in the context of how grids benefit traditional graphic design. To this day, there's relatively little writing about the grid as it applies to digital media and user interfaces, a fact that I find puzzling.

After all, a graphical presentation on a computer monitor is already rendered through a de facto grid system of very fine proportions: a monitor display is composed of seventy-two tiny pixels per inch, arranged vertically and horizontally like an incredibly tight sheet of graph paper—a grid, in fact. What's more, this is a grid authored by mathematics.

the principles underlying typographic grids are significantly different on the Web:

- A grid solution on the Web is critically dependent on the technology available to the user for its successful rendering.
- Typography is, at least for the time being, fundamentally unstable online, varying greatly from user to user, and potentially frustrating the ability to produce desired results with the grid.
- There is no canonical size for a Web browser, both because the physical size of users' monitors will vary, and because the user may actually have her browser window reduced or expanded beyond its default size. As a result, the grid lacks the same luxury of a fixed, knowable sheet size found on the printed page.
- User inputs, preferences, and settings—which can be made both passively and actively—can drastically alter the requirements for an online grid solution.
- While precision placement of elements is possible on the Web, the grid is limited to increments of pixels, and pixels themselves vary in scale from device to device.

Given the highly variable nature of Web design, we can begin to think of print design being relatively definitive in nature. Analog design solutions are declarative, whereas Web design solutions are conversational. A solution for a Web design problem embodies a kind of conversation among the designer's intention, the technology at hand, and the user. What's more, communication on the Web (and in all digital channels) is increasingly concerned with facilitating conversations among users, creating frameworks within which highly unpredictable kinds of content can flow back and forth between people. It's useful to think of Web design as being similar but fundamentally different from what came before it in the analog world.

Chapter 3
Process

The balance of this book focuses on how to build a grid, including an overview of the steps to a working solution. Before we begin, though, it's useful to outline principles that every designer should keep in mind.

A grid should focus on problem solving first and aesthetics second. A grid can provide such a seductive aesthetic enhancement to any design that it's tempting to focus on its beauty rather than its utility. Many designers become preoccupied with the beauty of the grid and contort content or functionality to squeeze it in, regardless of how uncomfortable or ill-suited it may be. But the most successful grids are built in service of well-defined problems. Whether they're communication problems, organizational problems, or transactional problems, a grid derives its beauty from how well its resolves those challenges.

A grid is a component of the user experience. A grid is not a tool to impose complete control over a user's experience of a website. Rather, a grid is a framework within which the user can control his own experiences. Designers should not force every element and interaction to occur within the grid, nor should they allow the user experience to be unformed and unpredictable. It's the designer's job to make certain decisions for the user—not every decision, but enough so the user can accomplish his goals unhindered. The grid is a tool for that job.

The simpler the grid, the more effective it is. The principles described in this book can be used to create grids made up of sixteen, twenty, or even more units, in any combination of columns of uniform or irregular width. However, the fact of the matter is that the designer should always strive to create the simplest grid possible. As we'll see, mathematic precision is a key element of good grid design, but mathematical *usefulness* is just as important. The formulas you use to calculate combined units and columns should be fairly straightforward, even simple enough to do in your head or to explain quickly to a colleague. The simpler a grid system is to use and to explain to others, the more users will benefit from that grid system.

Steps

Here, then, are the major steps in designing a grid solution:

1. **Research and requirements**
2. **Wireframes**
3. **Preparatory design**
   - Pencil sketches
   - Units, columns, baseline development and calculations
   - Page sketches
4. **Comps**
5. **Production (code)**

Keep in mind that though our thinking should be rigorous, our process need not be. Though the order in which I've presented these steps has been useful to me, it may not necessarily work for every designer. Not every step must be followed in exactly the same order, for a specific amount of time, in a specific way, or even at all. For example, step 3, preparatory design, is a set of three different activities that are often done concurrently, with the designer switching between pencil and paper, software, and back-of-the-envelope math as necessary. What matters is not the rote repetition of these steps, but following the principles of good grid design throughout.

All the same, it's worth spending some time up front to discuss two critical steps in this process. The first is research, and the second is sketching.

Research and constraints

We first judge truly good design not by its beauty or its innovativeness or its efficiency, but rather by how well it responds to its original problem. Successful solutions demand that the designer grasp the problem presented to her and the constraints within which she's working. The designer has to ask and understand the answer to questions such as: *who* is the audience, *what* is the context, *when* will the solution be encountered, *how* will the solution be used—and even *why* is the solution necessary?

These questions can be difficult to answer, and the answers themselves are often unclear or difficult to parse. A designer must be persistent in asking them, in pressing for good and accurate answers, and in thoroughly examining and comprehending those answers.

Because a grid can give us such a head start in creating solutions, it can be tempting to forgo this stage of the process. Once a designer masters the rudiments of grids, it becomes much easier to start the mechanical process of constructing units and columns than to do the hard work of asking and answering these questions.

But nearly every design problem demands a period of thoughtful study before the search for a solution begins. Without a clear sense of the challenge at hand, any design work—including the development of the grid—is done in vain. It's a much more productive use of time to do research at the beginning of a project than to jump straight to the design.

Grid-based designs are no different. The more completely the problem is investigated, the better the grid will be. Well-researched grids maximize the creative options available to the designer. They also anticipate and avoid the traps of prematurely constructed grids: inappropriately structured units and columns, grids that are good for some aspects of the problem but inadequate for others, grids that fail to account for constraints that may not be obvious at the outset, grids that prove so unworkable that they need to be rebuilt at inappropriate times, and grids that prove unusable for collaborators.

What kind of constraints should the designer look for? They fall into three main categories:

- **Technical constraints** determine the delivery of the design solution. They include the target screen resolution and the generation or "modernity" of the target Web browser, two critical factors for any design. Often, technical constraints regarding a site's publishing system are important elements as well; the designer needs to consider limitations that such systems might impose on how the content is output. A publishing system frequently affects how content creators produce content for publishing, the workflow, which in turn influences the kind of design solution that can be put into place.

- **Business constraints** determine the very purpose of the solution. Whether it's to increase visitor traffic, time spent on a site, click-through performance to advertisements, or conversions of site visitors to customers, these goals are the most important imperatives for any design solution. The designer should consider branding, positioning, and marketing considerations as well. Finally, she should fully assess the business's ability to maintain the solution she creates: who will need to work with the grid after it's completed, and what are their skills.

- **Content and editorial constraints** determine the production of the content. They account for the different forms content might take, such as the types of articles, their length and the length of their headlines and summaries, pull-quotes, images and embedded content such as video and interactive elements, data tables and charts, and so forth.

Of course, designers will bemoan the inconvenience of constraints, or perhaps the thorniness of some of the particular constraints they must contend with. If only those constraints were lifted, if only the problem were slightly different, then the solution would be much easier to arrive at or more elegant in nature.

However, these constraints have a silver lining: in some ways they might make a problem more difficult, but they can also make it easier to arrive at a design. Comprehensive solutions like grids can often benefit from being built around one or two nonnegotiable constraints, immovable requirements that can't be easily altered during the design process. To begin with, they can directly influence the proportions of a grid, the very sizes of the units, columns, and regions that the designer constructs. These kinds of constraints might appear to limit the options available to a designer, they very often also have the effect of increasing a designer's inventiveness. The more wide open a design problem and the less restrictive the constraints, the less a designer is likely to make those insightful leaps of logic that are the hallmark of great design. Nonnegotiable constraints can help spur a designer to do this. Whether it's locking in a particular dimension, a technological imperative, an advertising unit, or some other factor that a designer must work around rather than conveniently modify to her own needs—having one immovable requirement can be enormously useful.

Sketching

Having spent so many paragraphs belaboring the importance of thoroughly researching a problem, I can make this next point more succinctly: sketching on paper is an essential tool for thorough design problem solving, and it's particularly helpful in developing grids. The simple act of quickly and loosely drawing out speculative combinations of columns and potential layouts can save vast amounts of time and often leads to much more creatively fertile grid solutions than simply jumping ahead to designing or even coding a grid.

I can't emphasize enough the power and usefulness of using old-fashioned pencil and paper to work out problems, to brainstorm potential solutions, and to explore promising or even not-so-promising ideas that may be too costly or time consuming to test otherwise. In fact, the most important aspect of sketching is not so much making marks on paper, but rather being able to run through many ideas quickly, with little cost. Remember, you have no expectation that the sketches will amount to anything more than just sketches. Sketches don't need to be pretty.

As mentioned earlier, it's also important to keep in mind that sketching need not be a discrete phase of constructing a grid that begins and ends at specific points. Sketching can happen at any phase throughout the project, at multiple levels of completion—though of course it's most useful to sketch earlier, so that more ideas and possibilities can be run through very quickly. Keeping a pencil and a pad of paper handy at all times is sure to prove invaluable.

Terminology

The vocabulary that describes the various components of a grid might seem simple, but it can also be surprisingly unspecific. For example: the notion of a column seems straightforward enough, but on a page based on an eight-column grid, a designer might create a layout with only two columns of text, rendering the meaning of that term imprecise. Even books about the craft of grid-based design don't always agree on terminology, with some using terms (e.g., regions, fields) that are missing from others. For the purposes of this book, then, it's important to establish some common ground terminology as we proceed further into more practical discussions of grids.

UNITS

The building block of any grid, a unit is the smallest vertical division of the page (i.e., units are measured in width), upon which columns are built. Units are typically too narrow to house most textual content.

COLUMNS

Columns are groups of units, combined together to create workable areas for the presentation of content. Most text columns, for example, require two or more units to be workable. A grid system of, say, sixteen units can be combined into two columns of eight units each, or four columns of four units each, and so on.

REGIONS

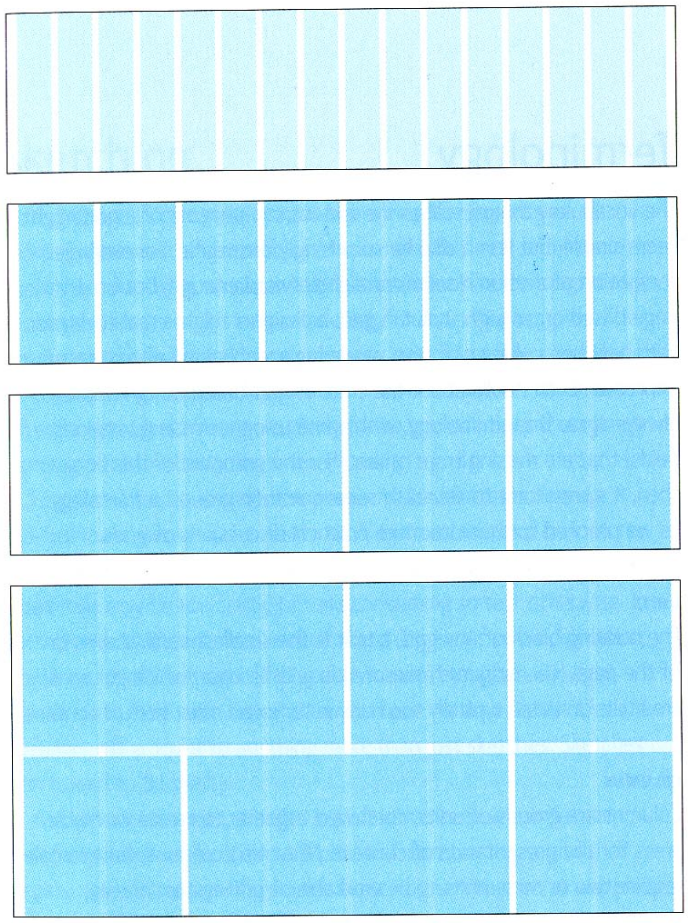
Regions are groupings of similar columns that form parts of the page. For example, in a four-column grid, the first three columns from the left might make up a single region for the display of one kind of content, and the remaining column might form another region.

16 units

8 columns

3 regions

2 fields



FIELDS

Fields are horizontal divisions of the page (i.e., fields are measured in height) that help a designer to visually pace the placement of elements on the Y-axis. Fields can be calculated in many ways, but using the golden ratio is one of the most effective methods.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur scelerisque odio vitae lectus tincidunt porta pellentesque urna placerat.

The baseline grid is based on the invisible lines on which letterforms rest.

BASELINE GRID

In traditional typography, the baseline is the invisible line on which letterforms rest, e.g., the bottom edge of a capital *E* rests on the baseline, while the descender of a lowercase *p* will fall below the baseline. The baseline grid is formed by a uniform, top-to-bottom repetition of baselines spaced apart according to the leading or line-spacing of the text.

HORIZONTAL VS. VERTICAL ORIENTATIONS

These concepts are notoriously easy to confuse (a unit can be thought of as either a horizontal or vertical division of a page, depending on one's point of vew), so this book refers instead to the columnar grid (divisions of the page measured in width) and the baseline grid and regions (divisions of the page measured in height).

GUTTERS

Gutters are the empty spaces between units and columns. When units are combined into columns, they incorporate the gutters between them, but not the space to the left of the leftmost unit nor the space to the right of the rightmost unit.

MARGINS AND PADDING

Margins are the space outside a unit or column. Padding is the space within a unit or column. Margins are generally used to create gutters, while padding is generally used to create a small, visible inset within a block of text inside a column.

ELEMENTS

An element is any single component of a layout. Examples include a headline, a block of text, a photo, or a button.

MODULES

Modules are groups of elements, combined to form discrete blocks of content or functionality. A registration form, for example, is a module composed of several constituent elements such as a label, a form field, a button, and so forth.

Chapter 4
Execution

There's no better way to learn how to design with a grid than to actually roll up your sleeves and try it. Now is the time to put all that theory, history, and preparatory study to work. This section focuses on creating a practical solution to the challenge of designing a website.

There is, of course, no such thing as a typical website or even a typical Web design problem. But the job we'll focus on in this section combines many common types of pages, including a blog, profile pages, and a home page that knits together disparate content types. The project also illustrates many different design challenges that will be instructive in showcasing a general approach to real-world grid design. Taking all of the project's constraints into account, we'll construct a single grid system and apply it to the various page types found throughout the site.

A side note: this section, and indeed this book, were written and designed so that readers can start reading anywhere, and I fully expect that many will buy this book exclusively so that they can skip ahead to this very section. There's nothing wrong with that, of course. Still, it's worth mentioning that, just as the ideas and context provided in the preceding chapters are of limited use without the practical advice that follows here, it's also true that the how-to instruction in this section will be of limited use without the ideas that precede it. The next *120* or so pages will provide you with the tools, but the preceding chapters are meant to provide you with the creative skills necessary to make good use of these tools. Don't cheat yourself!

OVERVIEW OF PROJECTS

The job at hand is to design an online design journal—a website by, for, and about designers. It will combine editorial content in the form of blog publishing and a social networking layer into a cohesive user experience.

The site is composed of four major templates. We'll take an in-depth look at how to construct a grid that works for all of them, then apply that grid to the design of each. As we work through each template, the various constraints and lessons will build on one another:

- Blog article page
- Category/catalog page
- Profile page
- Home/gateway page

Later in the book, we'll extend that solution across a number of secondary templates, which will help demonstrate the flexibility of our system. These include:

- Messaging pages (e.g., error pages)
- Blog index pages
- Preferences pages
- Calendar display pages
- Email templates

Since we're dealing in the hypothetical, we'll also assume a make-believe name for our design journal: Designery.us. This gives us some grounding in the branding requirements for our project, which we'll discuss in greater detail soon.

Getting started

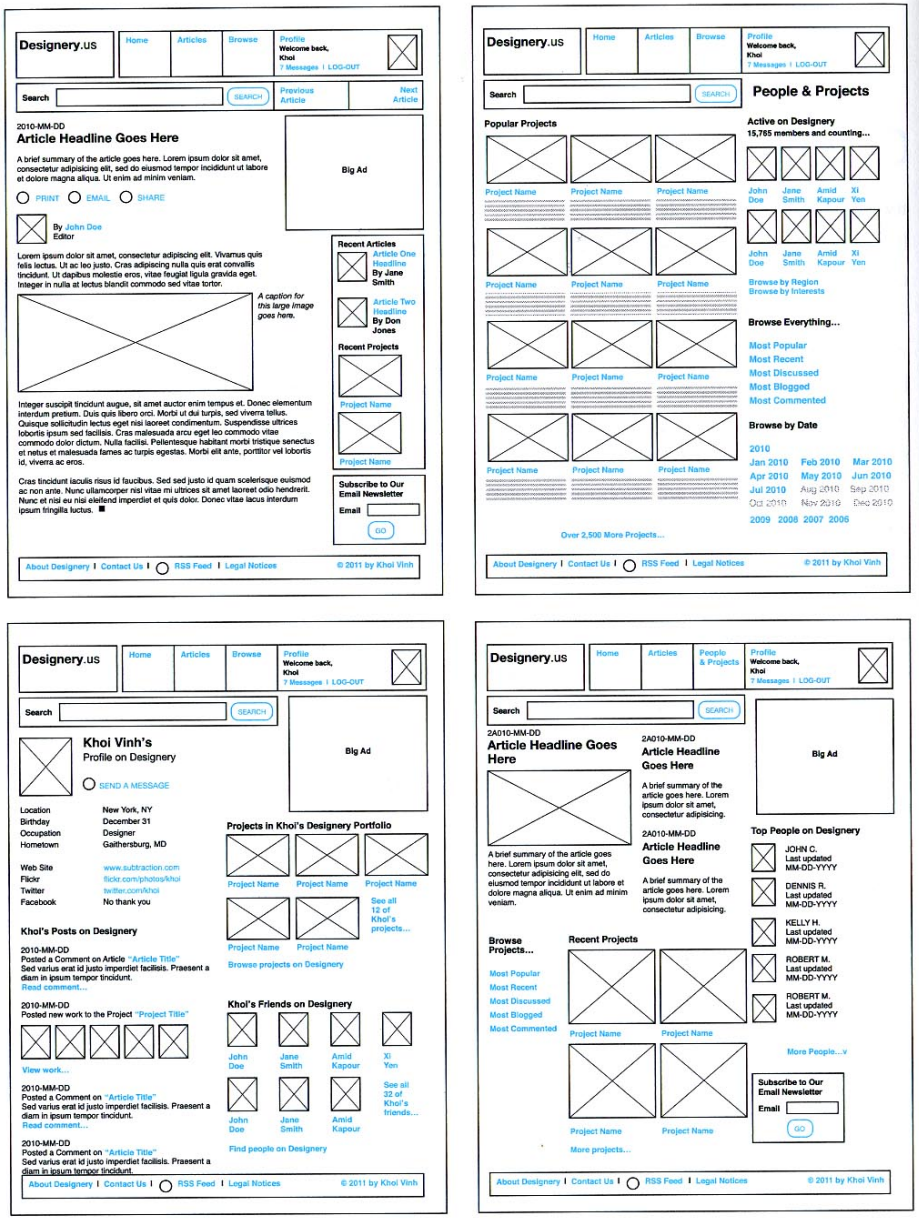
UNDERSTANDING THE REQUIREMENTS

If there's one lesson you should take away from this book, it's that understanding a project's constraints is the first and most critical step in crafting a solution. We must identify the criteria that determine whether the outcome we create will be successful, or even workable Until that is accomplished, there's little point in thinking about the grid we might develop, much less in unpacking our digital design tools to start choosing typefaces and colors.

For Designery.us, we will suppose the preexistence of a set of well-documented wireframes prepared by an information architect. Wireframes are schematic drawings that capture the features and types of content that are required for each template. These wireframes do much of the job for us; in fact they function as documentation of many (though not all) of the constraints that inform the project.

At this point, we won't account for the constraints presented by every template throughout the site. Instead, we'll focus on the four major templates that will form the basis of the majority of pages. If we design our solution thoughtfully, we trust that whatever grid we develop can be easily applied or adapted to the less critical templates later on.

In reviewing the wireframes for those four main templates, we can identify two principle constraints. These, helpfully, are fairly clear cut and reasonably typical of many Web design challenges: the viewport size and the advertising unit.



Opposite: Our hypothetical project is based on these four wireframes for the article, category, profile and home pages.

THE VIEWPORT

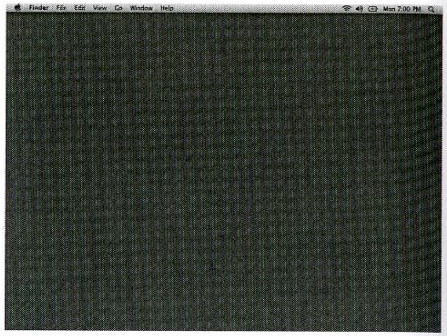
It seems unlikely that we'll ever have a single, standard size that all Web browsers will adhere to. Even as desktop screens get ever larger, our attention will continue to be divided among the much smaller screens on our mobile devices, the medium-sized screens that are becoming common with tablet computing devices, or the large-scale screens on our high-definition televisions. (Paradoxically designers must treat large television screens like lower-resolution devices because users sit far away from them.)

However, as this book goes to press, at a still early juncture in the Web's development, it seems reasonable to say that 1024 pixels wide by 768 pixels tall is as close as we come to a canonical size. Or, rather, it's the most utilitarian of the handful of standard screen resolutions we can work within. It's neither too small for most reasonably sized displays nor too large for the increasing power and resolution of handheld devices.

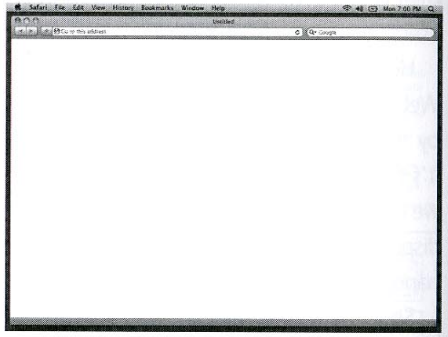
So the first constraint is that our design must fit into a 1024 x 768 screen.

Of course, a 1024 x 768 screen doesn't translate directly into a viewport of that same size for a web page. Thanks to menu bars, the tendency of most browsers to be sized slightly smaller than the space available to them, and other bits of user interface overhead, most browsers have a natural "posture" about 20 percent smaller than the total screen real estate. Within that browser window, we'll also want to accommodate some minor padding on the left and right sides of the window, so our design doesn't abut either edge, reducing the available viewport even more.

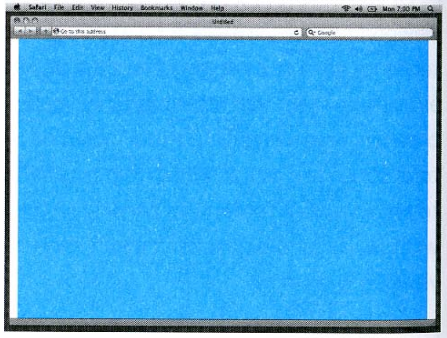
A 1024 x 768 screen is as close as we have to a standard screen resolution. This will serve as one of the constraints for our design.



A browser's natural posture inside a screen makes roughly 80 percent of the screen available to users for viewing any web page. This makes for a space within the browser of about 974 x 650 pixels.



Assuming some visual padding on the left and right edges of the browser, this reduces the available screen real estate—or "live area" to approximately 960 x 650 pixels.



THE ADVERTISING UNIT

Looking at our wireframes, we see right away one shared requirement: a rectangular ad unit is a prominent feature. It may not appear on every page, but we can recognize immediately that whatever grid we construct needs to work with this ad unit. This ensures consistency across the site.

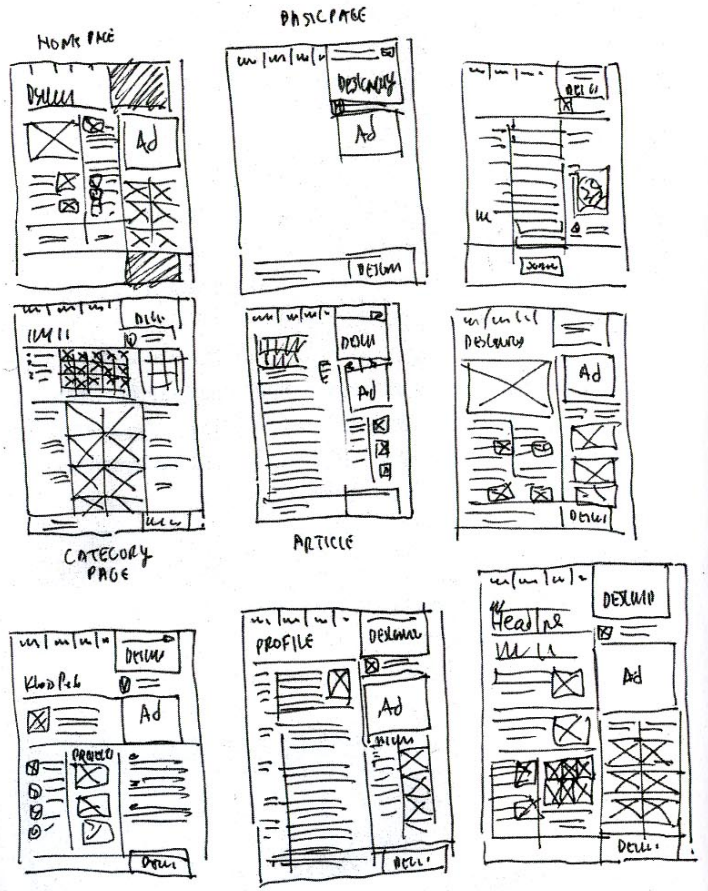
For many Web designers, ads are the bane of our craft. They present a particularly touchy kind of constraint that often seems antithetical to creating well-designed user experiences. By their nature, ads are intended to disrupt the flow of the user experience that designers try to orchestrate in our solutions.

The rather obvious fact of the matter, however, is that for many sites, advertising units are their very lifeblood; they provide the revenue that allows the sites to run. In spite of their inconvenient nature, they're a very real and important constraint.

That also happens to make them a very useful constraint, too. We have discussed the paradox that the less mutable the constraint, the better the design problem. A prominent advertising unit is helpful to us in that it provides a foundation on which to build our grid; it effectively informs the way the grid is shaped.

For our purposes, we'll use the Internet Advertising Bureau's "big ad" unit, which is 336 pixels wide by 280 pixels tall, as our second constraint. As it happens, the width of this particular unit specification can also accommodate at least two other sizes of ads: the 300 x 250 medium rectangle unit and the 300 x 600 half-page unit. That is, a design created around the big ad unit should be fairly easy to use for these other ad sizes.

The designs for each of the pages in our site begin with hand-drawn sketches.



Sketching the solution

As with every design problem, the best path to a solution starts with sketching. Using pencil and paper, we can run quickly through several different approaches for a design framework that will work across the site. Again, the focus is on the four main templates, and in fact we only need to create sketches for those pages, but it can be helpful to keep the secondary templates in mind during this process as well.

In sketching, the goal is not to create a drawing of a completed solution, but rather to think as creatively as possible through approximation. Sketches should be fast, loose, and small in scale—a single sheet of paper is more than enough room for at least a half a dozen sketches done in quick succession. It's not important at this point to sketch an exact grid or to determine the number of units in the solution. Instead, focus on the number of columns to be used on a given page, regardless of how complex or difficult it will be to create those columns mathematically. In sketching, we want to remain as free as possible to come up with creative solutions, and let those ideas in turn determine the grid we'll construct later.

After several rounds of this visual brainstorming, we'll settle on these rough sketches to guide us as we begin to construct the grid.