

SECTION 2.1

GRID

OUR FIRST PRINCIPLE IS AN EASY ONE: line things up!

You may have heard this common designer complaint:

A designer builds a pixel-perfect mock-up and passes it along to the developer to build. The developer builds the design, but the built design differs from the mock-up by something arbitrary like two pixels!

“Silly designer,” says the developer. “Why does something so small matter? It’s basically the same.”

Here’s the thing—small pixel differences really do matter, especially when it comes to elements on a page. If one element is close to but not completely lined up with another element, it can create a tiny bit of unevenness, and that little bit of chaos leads to a feeling of unease and clutter (Figure 2-2).

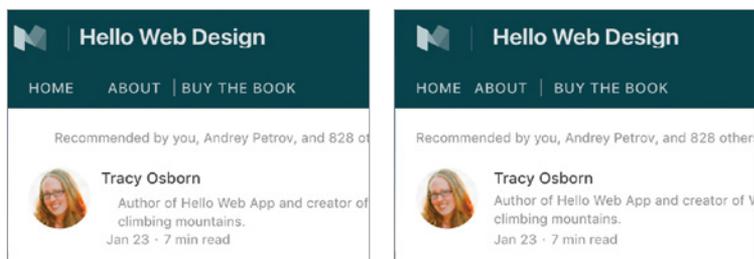


FIGURE 2-2: The two screenshots are basically the same, but the one on the left, where everything is slightly out of line, looks less cohesive and professional.



FIGURE 2-3: The *New York Times* website uses a grid to effectively organize a large number of elements and showcase items on its home page.



FIGURE 2-4: The Gridset website demos a compound 4+6 grid with and without guidelines

The easiest way to make sure



FIGURE 2-5: The 12-column grid system in the Bootstrap CSS framework.

and ordered. Aligning elements to an underlying grid will help you achieve alignment and consistency throughout your design.

The *New York Times* website (**FIGURE 2-3**) uses five main columns to organize a large amount of information. The spaces between the columns highlighted in red are *gutters*. Objects in your grid can span multiple columns, and some objects might break out of the grid, but everything more or less adheres to the grid.

You can use any number of columns in your design (**FIGURE 2-4**), but a 12-column grid (**FIGURE 2-5**) is the most often used for versatility.

A grid also helps you create a plan for your website's layout because it constrains you to place elements within certain areas rather than having the entire page to choose from. Win win win.

In essence: Got a bunch of elements? Line them up (horizontally, vertically, or both) to create a feeling of cohesiveness.

Shortcuts

The theory here is simple—just line things up—but there are many tools that make it easy to ensure we’re using the grid.

Grids in mock-up programs

If you’re working on something that you won’t be building in CSS, you’ll need to add guides to your design.

All mock-up programs—Photoshop, Sketch, or GIMP—allow you to set guides to float over your design, which makes it easy to align elements to the guide.

If you’re working on a website mock-up in something like Photoshop, you can use grid templates to lay out your website using the same columns you’d use in your website framework system (**FIGURE 2-6**).

Guides are also accessible in most slide programs and other simple layouts, such as Keynote (**FIGURE 2-7**).

For slides, you don’t need to add a whole 12-column grid to your system—the slides in **FIGURE 2-7** have only a few guides, just enough to keep elements on separate pages along the same lines.

However, if you’re designing something a bit more complex, you can download quite a few templates with multiple columns already set up, such as the 960.gs (helloworlds.com/hwd/4) system.

Grids for web design

I highly recommend using a CSS framework with a grid included, such as Bootstrap (helloworlds.com/hwd/5), Foundation (helloworlds.com/hwd/6), Skeleton (helloworlds.com/hwd/7), mini.css (helloworlds.com/hwd/8), or PureCSS (helloworlds.com/hwd/9) (**FIGURE 2-8**).

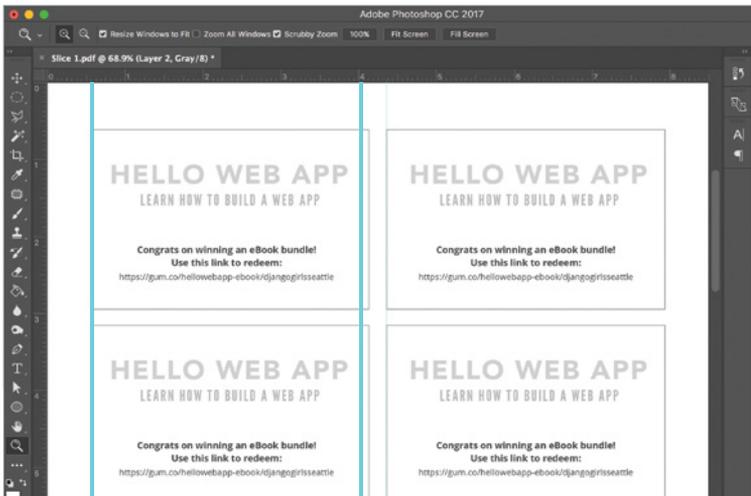


FIGURE 2-6: Guides can be set by dragging from the rulers at the left and top of the screen in Photoshop.

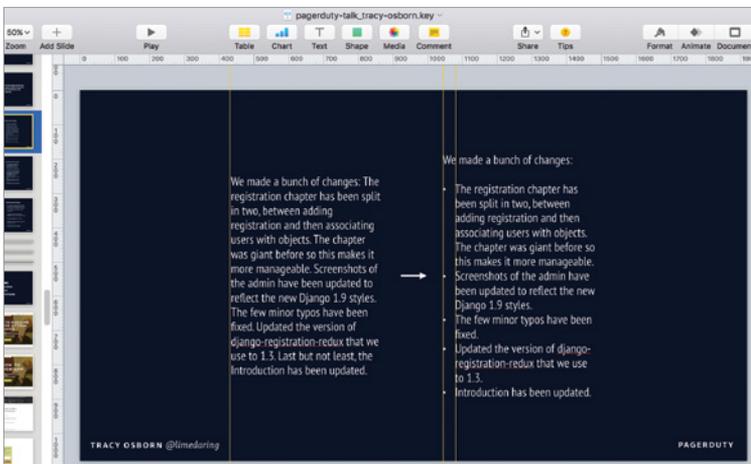


FIGURE 2-7: Keynote also lets you pull out guides from the rulers on the screens. Many other programs have similar options.

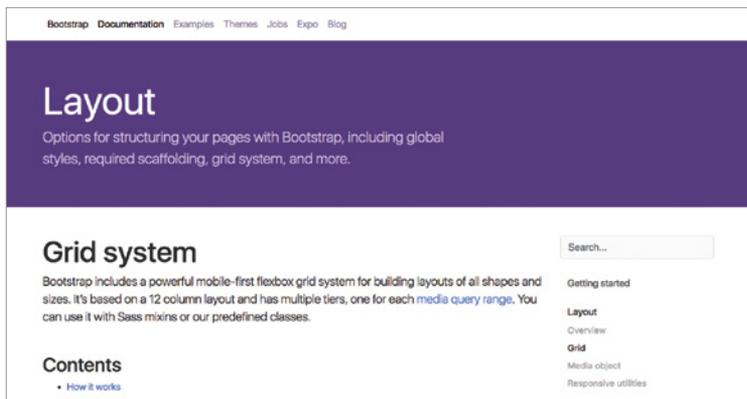


FIGURE 2-8: The grid system included in the Bootstrap CSS framework.



FIGURE 2-9: Some of the CSS classes included in Bootstrap to align and set your objects in the underlying grid.

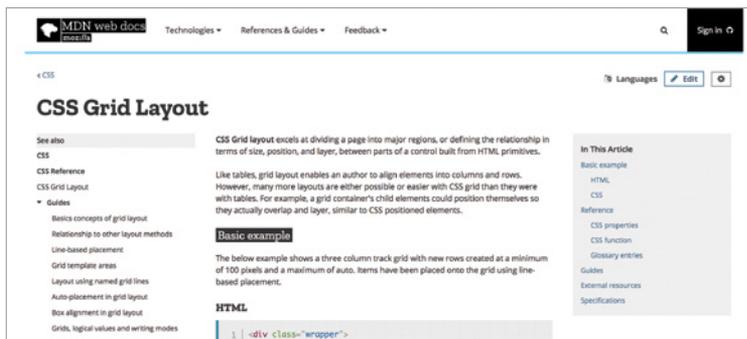


FIGURE 2-10: CSS Grid docs on the Mozilla Developer Network (helloworlds.com/hwd/10).

Login

You can access your account details below by logging in using your credentials.

 Remember me

[Password or username recovery](#)

Login

You can access your account details below by logging in using your credentials.

 Remember me

[Password or username recovery](#)

FIGURE 2-11: On the left, the original widget. On the right, we've lined up the edges of the elements in the box, making it feel a bit cleaner and less chaotic.

By using the HTML classes that constrain your design to the underlying CSS grid, your elements will naturally align with other elements on the page (**FIGURE 2-9**). Just keep in mind that additional margins or padding that might be added in CSS may bump elements out of alignment.

CSS is getting a new element called CSS Grid (how convenient!) that makes it super simple to align elements to a grid without using a CSS framework. At the time of this book's writing, CSS Grid is on the verge of being released and covered by most browsers. While this book doesn't cover CSS, using CSS Grid will make it easier to implement grid-based designs (**FIGURE 2-10**).

Real-life examples

Remember the little widget we introduced at the beginning of this chapter? Let's update the design by lining up the elements (**FIGURE 2-11**).

While the widget still has issues (after all, we're just getting started), you can see that this little change has made a small but significant positive difference. All of the interior elements are now lined up: headline, content, input, button. The input's placeholder text is bumped out a bit because of the padding

in the form, but its containing element aligns to the grid. The overall feeling is just a bit more cohesive, a little less chaotic.

Again, in a nutshell: just line things up. Add guides to align your elements to an invisible grid, and pay attention to the little pixel differences and misalignments that can make a design feel more chaotic. Feel free to break the rules occasionally and go off the grid (*ha*), but using a grid for most of your elements will create a cleaner, more organized design.

Next up, we're going to talk color!