

Website Architecture

Lezione 5

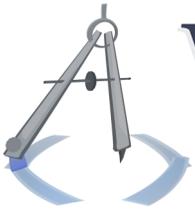
Monitors as Media

Covers the practical limitations of designing for the screen, including advice for maximal robustness. This includes basic study of color reproduction, fonts, typesetting and flow design, and common CSS techniques for layouts.

Michael Serritella

Summer 2010





Website Architecture

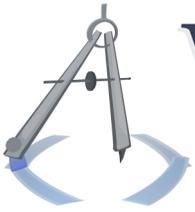
Lezione 5 | Monitors as Media

Intro to Design

People on the Web may use a huge variety of media to view your site. In order to accommodate them, you need to be wise and conservative (or perhaps clever and flexible). And regardless of the choice of media, there are some design considerations you should make so that the user can navigate your site quickly and intuitively. Here we will see some Tips and Tricks on designing beautiful and usable websites.

The advice here generally falls into two categories: the first is how to make your site readable (and "usable") and the second is how to make your site attractive. In making your site usable, you give your user a mental image of the site structure and his/her options as soon as possible, so the user knows where the navigation menu is, where the content is, and how the content is divided (e.g. into blog posts, lists, paragraphs, sections & subsections). You should use visual cues so the user can digest your site and construct this image instantaneously. The second category of advice is how to use the basics of graphic design in order to make your site look professional and appealing.



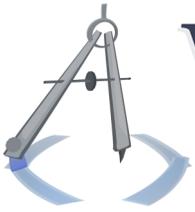


Manifest Destiny & 1920px

In American history, Manifest Destiny was the belief by European settlers that they were destined to expand as far west as possible, perhaps simply because they saw that the open West existed. In web design, monitor resolution has always been an issue - knowing which is most common and making your designs adaptable to those who have larger resolutions and those who have smaller ones. In recent years, we have seen larger and larger monitor resolutions. Most recently, the resolution of 1920 wide by 1080 tall (HDTV/1080p resolution) has become a not-uncommon choice. This has led some web designers to consume the full width and space available, and we may finally have gone overboard.

This section is about the management of real estate on the screen, more specifically focused on the management of width. This is because you have to pick resonable widths for your design elements and stick with them; people are not willing to tolerate scrolling a website horizontally, while you can expand your content vertically without a problem.





Website Architecture

Lezione 5 | Monitors as Media

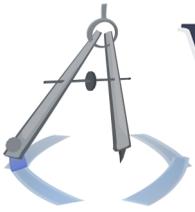
Nicey-nice margins

Perhaps the first lesson is that you generally have plenty of space for your content, so you should feel free to space it out a little; too many sites have narrow margins around everything or a lack of margins altogether. Having margins around the entire page is generally more comfortable for the user. Having margins around each section of content allows the user to recognize the distinct sections of your site quickly and intuitively; if there is almost no space between your site sections, list items, etc., it will be difficult for the user to tell which parts of your site are separate, how many list items there are, etc. On mobile devices, margins help people to click or tap on small items without accidentally clicking on the adjacent one. Just add some margins. Make it nicey-nice.

Division of horizontal space

Even if your site has a landscape orientation (content is wider than it is tall), you should probably not go east forever. Content sections should probably be divided so that their width does not exceed a comfortable amount. There is a limit to how





Website Architecture

Lezione 5 | Monitors as Media

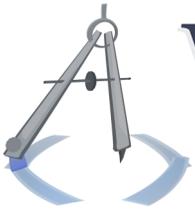
long a line of text can be without disrupting the user's sense of which line he is on and which line comes next; imagine following a very long line of text all the way to the right edge of the page and then going to the left edge and guessing which line comes next. Some in the typesetting industry say that this limit is 66 words per line and that no serious book will break this rule.

If you want to follow this rule in your website, it is not too difficult to approximate. Remember that you can set the width (or `max-width`) of an element to quantity expressed in `exes`, where one `ex` is the width of a lowercase `x`. If you assume the average word is around four letters long, plus spaces in between, you may want to set the maximum width of the main content section to be $297ex$ ($66 * 4.5$). On either side, you may have a column for the navigation menu or a vertical photo or advertisement.

Fixed-width or bounded content

Now, how do you accommodate the variation in width between the screens of different users? The simplest answers are to





Website Architecture

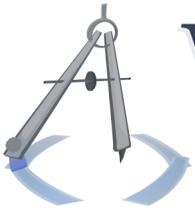
Lezione 5 | Monitors as Media

make the layout stretch to the available width or to pick one width that works for everyone. Picking a fixed width is a generally safe and increasingly popular choice; it is an especially simple way to deal with the emergence of mobile devices such as tablets, which are about as big as a small laptop monitor. In this design, everything in the site exists within one column in the horizontal center of the screen; outside of this may be a simple background, perhaps just a solid color.

A common choice of width for this column is within 960 to 988px. It's a range of magic numbers that happens to work out well for both users' monitors and the division of content into decent-sized columns (often called a "grid layout"). You can visit [The Grid System Generator](#) to see example grids and generate one to be used on your site. If you want to do it yourself, you can create a `<div>` around everything and use the value `auto` for its `margin-left` and `margin-right`.

Fixed/grid layouts are good for beginners because they do not involve many risks. Of course, the risk is that your site looks cookie-cutter, but with an increasing prevalence of grid-based layouts, people might not notice and they may even expect it.



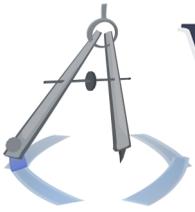


Fluid layouts

On the contrary, you can make a layout which is expressed in terms of the screen width available - e.g. a content area which is 60% as wide as the screen; these layouts are called fluid layouts. This is the next step up from a fixed-width layout and it is usually sensible. The technique is simple: express the width of an element in CSS using a percentage. This generally means that the content of the box takes up that percentage of the available width in the "containing block", which is the whole monitor/canvas by default. If you are curious, read the **appropriate section of the CSS standard**.

There are two general problems in fluid layouts. One is that the flexible content may become too narrow or too wide, depending on the user's resolution or even how the browser is resized. To combat this, you should limit the number of sections which are fluid (probably only the main content section) and give lower and upper bounds on the width using CSS. The second problem is that fluid, column-based layouts are a little difficult with CSS 2.1. For this, you can search for prefab solutions in CSS 2.1 or CSS3, which are more simply expressed.





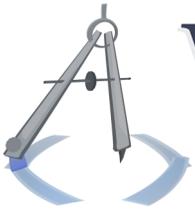
Responsive designs

A new option in solving this problem is called a responsive design. This uses a feature of CSS3 called media queries. Media queries are an extension of the `media` attribute on `<link>` tags used in HTML 4.01 + CSS 2.1 for giving different stylesheets for print, handheld, and other media. With CSS3 media queries, you can give quantitative media restrictions, such as saying that a CSS file is only to be used when the screen is between 769px and 1024px in width. You can specify different CSS files for different screen widths, and the browser will switch to the proper CSS file, even if the user resizes the window while the page is already open. So, the gist of a responsive design is that you make one HTML document that works for all of your different layouts and several CSS files, one for each layout.

You can find an in-depth explanation and an example on the article which brought this feature to light; see [A List Apart](#).

Responsive designs aren't always necessary, though they can be nice to use. You need to be clever in order to make one HTML design that works for several different CSS files; you



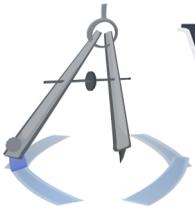


Website Architecture

Lezione 5 | Monitors as Media

need to know CSS fairly thoroughly. There is also a bit of a performance cost, since the browser typically downloads all CSS files and all of their linked resources (e.g. background images and fonts specified in the CSS files), even if the CSS file would not be used because of its media restrictions.





Menus & content organization

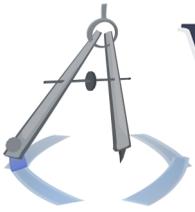
Some of the most simple and effective choices are the organization (grouping, division, and placement) of navigation menus and principal content.

The Magical Numbers 5,6,7,8,9,10

A person's working memory is like a short-term memory, though it is typically a more specific term: the working memory contains the items which a person can think and reason about simultaneously, including making comparisons between those items. Working memory is quite small. For example, if you were to hear a list of 20 numbers and then were asked to repeat the largest 10 of them, you would probably have a hard time. Thus your working memory may only have a capacity of 10 or fewer items. According to early psychological research, the average capacity may be more like 7 "chunks". See discussion on the publication: **The Magical Number Seven, Plus or Minus Two.**

What does this mean for web design? Something. It means





Website Architecture

Lezione 5 | Monitors as Media

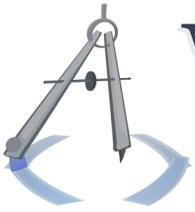
that you probably shouldn't put many more than seven items in your navigation menus. When people come to your site, they may have an idea of what they want, such as "contact information" or "where to buy that water bottle". They will look at your list of menu items and choose which one is the closest to what they want. This is similar to our earlier thought experiment with the numbers. If there are too many choices, the user will take longer to figure this out and will probably become annoyed. Note that you can make a hierarchy of menus if you want - eight on the first level, four or five sub-items within each top-level item, etc.

Avoid horizontal menus

Menus should be vertically oriented in the vast majority of cases. There are three main reasons.

The first is that menus tend to grow, since sites tend to grow over time, and a horizontal menu does not have a lot of real estate into which it can grow. A vertical menu can grow vertically and this only necessitates scrolling, which is considered acceptable. So, vertical menus are more scalable with respect to the growth of site content.





Website Architecture

Lezione 5 | Monitors as Media

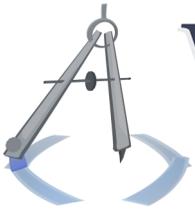
The second is that horizontal growth may break your layout (overstepping its bounds, looking unplanned or ugly), possibly when viewed in smaller resolutions, such as those on mobile devices. Even if it doesn't break the layout, it may cause the user to scroll horizontally, which is not acceptable. So, vertical menus are more scalable with respect to media and usability concerns.

The third reason is that the items in a vertical menu are easier to discern as distinct chunks (see psychology research above), so users will more easily and immediately identify the separate items, how many there are, etc.

Big & Tall content

You may have a sparse amount of content which you would like to trump up. For example, consider Facebook. Most users' comments are only a few words, and the actual text probably consumes a small fraction of the area of each page, yet these comments are the focal points. Perhaps this is an attempt to make people feel satisfied after writing a comment of only ten words?





Website Architecture

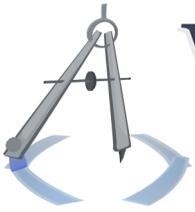
Lezione 5 | Monitors as Media

So, maybe you want to do this. It seems that people like it. Perhaps it is simple enough to say: don't be afraid to pad the hell out of your content. Add margins, vertical and horizontal; add padding; make the text size larger; center it in the middle of nowhere. If you think your site is lacking a focal point or the content looks too shallow, you might want to try this.

Segmented content

Your main content should probably be broken up into segments. If you don't have a few subheadings, or at least paragraphs with visible spaces in between, you're probably doing something wrong. Breaking up the content can give the user a better mental image of the structure of your content, as described in the introduction to this lesson. If the user is reading part of a paragraph and decides that it isn't relevant to him, he will probably want to skip to the "next part". What is the next part? If there is one long, run-on stream of content, the user will start skipping around if you're lucky and will simply leave the site if you're less than lucky.



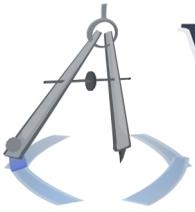


Documents vs. Programs

In general, the layout design and content of sites can usually be put into one of two categories: document-like or program-like; i.e. a site either looks like a document or looks like a program. There are hybrids and compromises, of course, but you can learn a lot from studying the dichotomy presented here. You should probably identify the design goal of the site you're working on, so you can tell which category it falls within and thus which design choices may be natural next steps. There are no hard rules here, just helpful suggestions.

Document-like sites are certainly more common. In these sites, fixed positioning is rarely appropriate, the content should have headings and subheadings, the menu should probably be diminutive, and there is a reduced expectation of interactive effects (e.g. hover styles). In program-like sites, you may want to use fixed positioning, highlight the user's current menu choices (e.g. navigation choices) and hovered-over choices, and have less emphasis on headings and hierarchical content structure.





Common pitfalls of simple sites

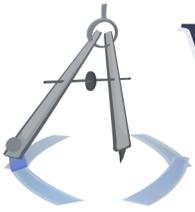
Say you get a site going, and then it's not good. Now what?

Different stuff should look different

As mentioned before, your design should give the user a clear sense of the different sections of your content and how they relate to each other. Most commonly, your main content exists in a hierarchical relationship: parent & child elements in a family tree, making for headings, subheadings, and paragraphs. The actual structure of an HTML document is hierarchical, so why not take advantage of the inherent hierarchy of your content? Manifest it in your layout; make headings look different from paragraphs, paragraphs look different from lists, etc.

One of the most neglected of these opportunities is the opportunity to add an indentation for subordinate content. Add a **margin-left** (or **padding-left**) for subheadings and their associated paragraph text; add a larger margin for the subsections of *those* sections. The user should be able to take a quick look at your page and tell its place within the hierarchy of information.





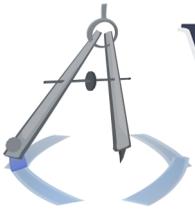
How do you style headings and more important text? You can choose from a large number of styling options. Make the text bigger, underline it, make it italic, bold, bold + italic, a different color, bold + a different color, or even bold + a lighter color, such as bold + gray.

The Wall of Text

Many sites, and perhaps most sites, will run into the problem known as the Wall of Text. You know what it is: a whole page of plain, unformatted text. How to fix this age-old problem?

Break up the content as per the hierarchy and distinct-content recommendations above. If that doesn't work, add images where halfway appropriate and add text formatting that may be completely floral, such as bold font or bold + lighter-color font (dark blue?) for slightly important words. You may also see if you can add a non-repeating graphic in the left or right margin of the content - perhaps even in the background of the content.





Fonts and colors

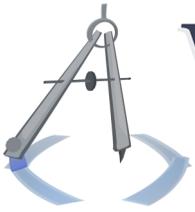
You will need to make special considerations in your font selection and color palette; the monitors of the world's users can be challenging media for a graphic design.

Font categories

Note that there is a dichotomy within fonts: there are serif fonts and sans-serif fonts. Serif fonts are those which have accent markings at the end of the strokes, such as in Times New Roman. Sans-serif fonts are those without serif markings, such as Arial and Comic Sans (now you know why it's called Comic Sans). In CSS, you can use the `font-family` property to instruct the browser to use a specific font or even a default serif or sans-serif font.

For accent text and for large text, the choice of font category doesn't much matter, but for "copy text" - the main text of the content - the choice does matter. Basically, serif fonts are considered easier to read because the "feet" at the end of the letters helps to guide the reader to the next letter; *however*, this is





Website Architecture

Lezione 5 | Monitors as Media

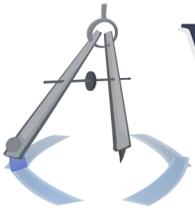
only true if the font is rendered in a crisp and clear fashion. The most common computer monitors are just not there yet, and print media are better at using the fine resolutions necessary for serif fonts. So, for rendering copy text on a computer monitor, sans-serif is usually best. You may still want to throw out all of these recommendations if you are going for a certain effect, such as when using a typewriter font for a site about government documents.

Remember that you may actually give different CSS styles for when a page is printed out; you can use the `@media` rule to define a style for print which uses a serif font.

Font diversity

There are conflicting opinions on this, but you should probably feel alright using a variety of fonts (including possibly serif and sans-serif) for giving your site some artistic interest. Using three fonts is probably the limit.





Font embedding

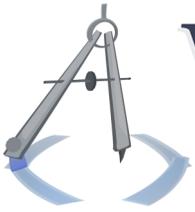
In the past, you could only use a small collection of fonts - probably about a dozen - and expect them to exist on users' computers. Now, you can embed fonts in your website, meaning you provide the font to the user so he can use it when rendering your site. This is becoming more common on websites and is already pretty well supported from browsers, including mobile browsers. To embed fonts, you basically put the file alongside your website files and then refer to it from your CSS.

More specifically, you will use the `@font-face` directive in CSS. Follow the directions on [this blog-post tutorial](#), and you may also want to use [this font converter](#).

The use of contrast

One huge problem is that everyone's monitor is different. People have different makes & models, and they also tune their monitors differently. It is practically impossible to guarantee that a site will look even reasonably equivalent to all people. Some displays will look more green, more blue, more dark, or





Website Architecture

Lezione 5 | Monitors as Media

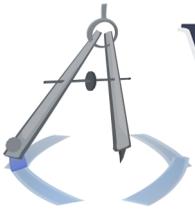
whatever. So, you can't do that. At best, it should be legible and usable by all people.

In order to do this, you must make conservative choices of color and contrast. Do not use colors which are in the top and bottom of the brightness spectrum (such as 15% on either end?), because the monitor may be so bright or dark that your color washes out into a black or white abyss. Use higher contrast - don't make a blue background and light-blue content - especially for text. Monitors do vary in their levels of contrast. This does unfortunately put a damper on your artistic freedom, but the damper is far better than some people being unable to read or use your site.

Cross-media backgrounds

Keep in mind that background colors and images are not printed by default. In other media, they may be similarly marginalized (a mobile device?); you can't be sure. So, you should probably not make them an integral part of your design or your logo or your navigation menu. Assume the background is expendable.





The quest for balance

asdf

Graphic design as a subset of art

arranging stuff on a page; client expectations and usability

Balance of positive and negative space

try to have complementary elements, probably asymmetrical

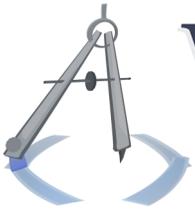
Higher dimensions of graphic design

as a mapping to the problem of spatial balance;

alternative would be to have uniform distribution of an effect ("everything has the same 3px white border"); typically not as good an idea

Color usage (discrete)





Website Architecture

Lezione 5 | Monitors as Media

Contrast usage

Texture

Angles and curvature

Scale of elements

headings/etc; relates to hierarchy

