

Web Design: Introduction and Theory

SUNDAY, MARCH 7, 2010
65 9th STREET, BROOKLYN

LOUISE MA
LOU@HELLOLOUISE.COM

Takeaway from last Sunday's class:

STORYTELLING AND ARCHITECTURE

Plan, plan, plan.

TIME

People don't read on the internet.

IMMEDIACY

Oh! This site is about _____.

FLOW

Above the fold.

ORIENTATION

Where am I?

RESPONSIVENESS

Can I click that?

BASIC LANGUAGES

HTML

CSS

A LITTLE MORE THAN BASIC

Javascript

Flash

CMS

Wordpress

Expression Engine

Tumblr

Indehibit

Plan, plan, plan.

Before diving into the design and sitting down at the computer to mock up your site in Photoshop, take a moment (or longer) to assess these pivotal questions:

What does my audience want to accomplish with this site?

What do I want to accomplish with this site?

What kind of content am I serving?

How often will I update?

Do I make big changes each time I update?

Who is my audience?

Do they have handicaps?

Under what circumstance is someone reaching my site?

Where are they coming from?

What will my audience expect to see when they visit this site?

Answers to these questions will help determine what to put in your site and how to design for it. After you've determined this, the next step is site architecture. This is where you sketch out where all the pages are in your site, and then how they link to one another. This step gives you a quick idea of how large your site is. After you have a map set up you can start to focus on sketching each page. At this point you'll have a good idea of what absolutely needs to be on each page. No design happens yet, because you're still fleshing out the content. When these are all finalized, the design can begin.

REAL WORLD EXAMPLE:

Architecture, blue prints.

You wouldn't try to decorate a house that hasn't been built, and definitely not one that hasn't had its blue prints drawn up. The site architecture step is called "architecture" because that is exactly what you are doing... understanding what it has to do and what its layout looks like.



People don't read on the internet.

Time is of the essence. It is one of the few limited resources working on the web. It doesn't take a lot to make sites (i.e., hard labor or raw materials) but conversely it doesn't take much to consume them.

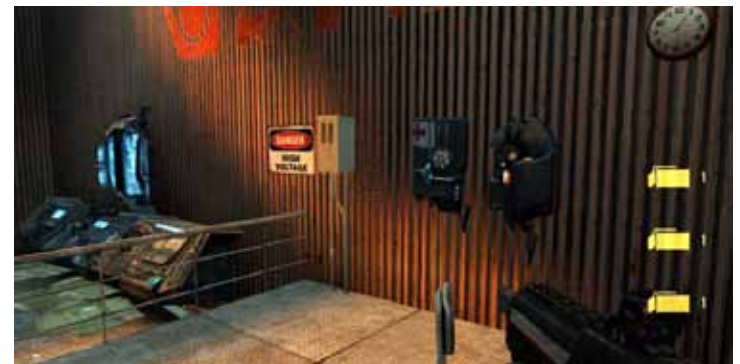
Readers will not pore through your site and read each word. It's more likely that they will scan, looking for that nugget of information they're searching for.

Just how little do they read? About one in five words, to be exact.

REAL WORLD EXAMPLE:

Trying to find health/food in video games.

Pressed for time, players in a video game are constantly scanning for beacons where they can gather health and points. Where are the beacons in your site and how do you identify those? Remember, sites don't come with instruction manuals. If you need to instruct someone how to use your site, then there is a glaring problem in the design.



Oh! This site is about _____.

Since time is so limited, you have only a handful of seconds — not only to make a design impression, but more importantly, to explain what your site is about. Can your mom glance at your site and say “Oh! It’s about ____.” You’d be surprised at what seems obvious to you is a total mystery to others.

REAL WORLD EXAMPLE:

Old Japanese *Kanban*; fake plastic food

It couldn’t be more clear what these shops sell. Yes, it can seem stupid simple, but often that’s what it takes. If readers are scanning your site like a billboard, it should function like a well designed sign. Which means obvious and to the point.



This paint shop sold artist pigment and other paints.

Above the fold.

The term “above the fold” is adapted from the newspaper industry. Remember how the Times, for instance, always arrives as a folded stack? The headline is always above the fold, and also the masthead and graphic. Online, this means all of your important information goes above the fold. Important information definitely includes your site navigation, and the logo.

The fold line hits where the page disappears below the bottom edge of your monitor. Users shouldn't have to scroll to get to essential links, such as an entry deeper into your site. A link above the fold will take them somewhere else, if you link out of your site. Then nothing at the bottom gets seen.

REAL WORLD EXAMPLE:

Feng shui

Feng Shui is all about guiding the flow of chi in your home or office. You want it to flow into certain places, not into others. You want it to pass through areas of importance. This principle is the same with web design. Help your visitors find what they're looking for, or encourage them to go where you want them. Don't let them wander away before showing them what they might need to know.



Where am I?

The back button is one of the most often used button on browsers. We are constantly backtracking and using this tool to reorient ourselves. How do you help your visitor orient him or herself in your site? Web space is ambiguous — it's like a flat piece of paper, but the linked structure makes it more like a labyrinthine piece of architecture. Help your users by using a consistent navigation throughout the site, and let them know where they are in context of the site. This helps your site seem smaller, less daunting for your guests.

REAL WORLD EXAMPLE:

Mall directories



Can I click that?

The hyperlink is the essence of the web.

That's why it's so important to make it clear to users what is clickable and what is not. After all, how can they get to your bio, play a video, or donate money to your cause if they don't know where to click? Take a moment to assess your site's aesthetics, and then how to make the buttons "pop" in the design. They don't necessarily have to be dimensional, but it is a trusted convention.

REAL WORLD EXAMPLE:

Nintendo NES controller

Remember the simplicity of this controller? In real life when we press a button, the object in our hands makes a slight physical shift that lets us know it acknowledges our interaction with it. Think about how responsive the interactive items on your site is. The more responsive it is, the more inviting it is to the visitor.



The basic language of web: HTML and CSS

These are languages you should learn if you are designing often for the web. You may not necessarily have to be fluent, but knowing the basics will help you design better looking, more useful sites.

The following are the fundamentals.

When coding a website, you are coding for two types of readers.

HTML is *content*, important for the *computer*.

```
> Here's a girl  
> Here's a dog  
> They're on a beach
```

CSS is *presentation*, important for *humans*.



HTML AND CSS (CONT'D)

HTML tells the browser there is a sentence of text, and what the text says.

Here is that sample sentence!

```
<p class="example">Here is that sample sentence!</p>
```

CSS tells the browser that the sentence of text is set in 24pt Helvetica, it's centered, and it's blue!

Here is that
sample sentence!

```
p.example {  
  font-size: 24pt;  
  font-family: Helvetica, sans-serif;  
  text-align: center;  
  color: blue;  
}
```

In HTML, there is a variety of code that tells the browser what you're putting there. They specify what kind of object it is. Here are a few:

`<p>`
is a paragraph of text

``
is an image

`<h1>`
is the big header, the title of your site

`<h2>`
is the next biggest header, or the subheader
(header hierarchies keep going, but it's best to stop at about 4 or 5)

`<input>`
is a form field, where you input information
(such as username and password fields, for example)

`<table>`
is tabular data, like a size chart or nutritional information
(shouldn't be used to position or place things on the site)

`<div>`
is kind of abstract
It's best described as a container to fit your other tags into. This you can use to position, place, or style things on the site.

HTML AND CSS (CONT'D)

When coding a site, it's of utmost importance that images are images, tabular data fits into tables, and so on. Search engines like Google that crawl the web rely on developers' adherence to these rules to make sense of a page's content.

A problem arises when the designer building the site confuses one object for another.

FOR EXAMPLE:

The title of Roger's site is "Roger's Fancy Cheeses".

"Roger's Fancy Cheeses" is set in a beautifully designed typeface, with wonderful ornamentations around it.

The problem here is that it's an image in HTML, when it is technically a header. It should be an `<h1>`, not ``.

To the naked eye, it's correct. But when a browser reads it, and more importantly — Google — Roger's site *doesn't have a title*. When a potential cheese buyer searches for Roger's site in Google, the search engine has to rely on all the other cues on his site that helps it determine that the site is his. And that's wrong.

INCORRECT:

To humans...



HTML

```

```

To Google and screen readers (no CSS styling, no images enabled):

CORRECT:

To humans...



To Google and screen readers (no CSS styling, no images enabled):

Roger's Fancy Cheeses

HTML

```
<h1>Roger's Fancy Cheeses</h1>
```

CSS

```
h1 {  
    background: url('title.jpg');  
    width: 600px;  
    height: 300px;  
    text-indent: -99999px;  
    overflow: hidden;  
}
```

A little more than basic: Javascript and Flash

Now that you understand the basics of HTML and CSS, here's a bit on Javascript and Flash.

Javascript is the language that does all those extra things that HTML can't do. Sounds vague? Here are some examples of what it does:

ANIMATION

Image slide shows, transitions

Opening and closing "drawers"

Drag and drop gadgets in your iGoogle home

BROWSER TIE-IN

Open pop-up windows when you enter or leave a site

Pop a warning dialog when you forget to fill in some information

Javascript has come a long way. It used to be that you got little "applets" to go on your page that maybe added a moving reflection to an image. Remember those? Now you can tell your HTML page to call a Javascript "library" that's good for all kinds of tools.

There are many popular Javascript libraries, the most commonly used being MooTools or JQuery. JQuery is a favorite of designers' because it's simple to install and use. Once you have a JQuery library referenced from your web page, you can apply individual scripts — to animate a slideshow, or hide a drawer, for instance. It's a modular system that's less taxing on your server.

In many instances, Javascript is now a viable alternative to Flash when handling transitions and custom fonts.

JAVASCRIPT AND FLASH (CONT'D)

Flash is a whole other animal. The language that drives Flash is called Actionscript, and is generally considered much harder to pick up than HTML or CSS. It's easier to understand Actionscript if you have a firm grasp of Javascript, or OOP (Object Oriented Programming).

Flash encounters a similar problem with designers confusing objects within HTML.

You can use custom fonts, transition beautifully from page to page in a Flash site. But to Google or a browser, all that's there is an empty box. To a human it's rich with graphics and information, but to the computer there is nothing.

Flash is best suited for online games, video, or even audio. It's also acceptable in small instances in a regular website, such as an animation or a small gadget (like a playlist). But it's not semantic to fit your entire site into a flash file. HTML, CSS, and Javascript are now advanced enough to rival the flexibility of a Flash environment.

HUMANS SEE:



GOOGLE SEES: (NO FLASH, NO IMAGES, NO JAVASCRIPT)

Strona, którą właśnie chcesz przeglądać zrealizowana została w technologii Flash i wymaga odpowiedniej wersji pluginu Adobe Flash Player - **9.28 lub nowszej** oraz włączonej obsługi JavaScript.

Player powinien zainstalować się automatycznie. Jeśli pojawi się okienko o nazwie "Adobe Flash Player Update" kliknij "Yes". Podczas instalacji może być wymagane zamknięcie przeglądarki.

Jeśli automatyczna instalacja nie jest możliwa na twoim systemie, kliknij poniższy link:
[Pobierz i zainstaluj najnowszą wersję pluginu Adobe Flash Player](#)

Jeśli jesteś pewien, że masz zainstalowaną odpowiednią wersję Flash Playera - **[pomiń detekcję](#)**

It's a message that apologizes and says you need to have Flash enabled to see the site. Yeesh.

CMS to the rescue.

It used to be that when you made a site, you painstakingly made each individual page. Copy and paste were your best friends — that’s how you copied the code for the logo and navigation, for instance, from one page to another.

As the web grew bigger and sites became more robust, it no longer made sense to maintain each individual page of a site. The people who made large websites began to see the pages as “shells” that the content fit into.

These shells are now officially called “templates,” and what goes in them is controlled by CMS’s, or Content Management Systems.

A CMS is an interface a site admin or webmaster uses to add, edit, and sort the information on a site. If the site were a blog, the CMS is where the blogger(s) would go to post an entry, and edit or delete existing ones. If you use sites like Facebook, Flickr, or MySpace, you already know how to use a CMS — these sites have the CMS aspect built right into their interfaces, since they host each user’s information.

There’s a wealth of CMS’s to choose from. These are the most popular.

Wordpress

Wordpress is free and open source.

This means that all the code you’d use to spruce up your site is free to use. To really use Wordpress to its full potential requires some understanding of PHP. PHP is often used as a copy-and-paste tool, meaning that you’d use a snippet of PHP code in a webpage to act as a placeholder for some other text (or whole webpage). But, it’s not absolutely necessary if you expect to stick to whatever themes are available to you. If you intend on posting in a blog format and don’t know much code, it is perfect.

The learning curve for Wordpress is relatively painless, overall. You’d download a package of PHP files to put on your server, and from there you can start to run Wordpress, add content, and change themes. If you needed help with Wordpress, there is a wealth of documentation that goes into how to troubleshoot or add functionality to your site. It just takes time to find your answer, since the resources are so wide and can sometimes be scattered.

You can read more about using Wordpress at the blog post about Dan Phiffer’s workshop at Trade School:

[HTTP://BLOG.OURGOODS.ORG/POST/388241652/WEBSITES-WITH-WORDPRESS](http://blog.ourgoods.org/post/388241652/websites-with-wordpress)

Expression Engine

Expression Engine has a free package (comparable to Wordpress), a pro package for \$100, and a commercial license for \$250.

Quickly growing in popularity is a beefy CMS called Expression Engine, of which I am a fan. If you are keeping a personal site, a blog, or a portfolio site where you're not operating a shopping cart, it's acceptable to use the free Core package, which has all the functionality of the paid packages, aside from the on-the-call tech support service.

Expression Engine has a code of its own, called EE code. It's great for designers and developers who are familiar with coding, and EE code is much more intuitive than PHP. As with Wordpress, you download a package of files and upload it to your server.

The beauty of EE lays in its conceptual framework. It operates on two main groups of information: Templates and Weblogs. Templates are exactly as they seem — templates to fit your content into. That is where all the HTML, CSS, and custom EE code goes. Weblogs is the “grouping” of information. If your site has two compartments: one for a portfolio of drawings and one for news, you'd make two weblogs — one for each. Within those weblogs you can decide what information you want to enter for each type of item. For each drawing, you have the image, the image's title, the date the image was uploaded, the medium you used for the drawing, etc. It's fully customizable. And, since the template is completely your own creation, there often is no dead giveaway if a site is powered by EE (as opposed to Wordpress, which doles out prefabricated themes).

There's a very strong support network for EE, even for users who are on core packages. There are lots of resources for plugins, modules, and extensions as well — gadgets to add functionality to your site.

Tumblr

Free.

For the novice designer or site owner who knows minimal code I highly recommend Tumblr. Tumblr is essentially one big CMS (like Blogger or Blogspot) so there are no files to download and set up on your server.

It's very simple to buy a domain name, and have Tumblr point to it. Many small businesses looking to set up their sites in a hurry find Tumblr to be a good solution. The service is free, and they offer close to hundreds of different themes that work in many different ways: portfolio site, blog, brochure site, etc.

There is almost no learning curve for Tumblr, and it's the first thing I recommend to people who want a quality site up and running for little to no cost.

Indexhibit

Free.

Created by the artist Daniel Eatock, Indexhibit is a sort of newcomer to the world of CMS. The user base is almost exclusively artists and designers, and the look of an Indexhibit is very particular: one narrow left navigation with the rest of site spanning as wide as a browser allows. It makes a lot of sense if you're mainly showing large images of your work, in a left-sidebar portfolio format. Because it is so simple and straightforward it works well for many artists as it lets their work shine first and foremost on the site.

Again, Indexhibit comes as a package of files you download and set up on your server. In this regard it is exactly like Wordpress but comes with no choice of prefabricated themes. Information is organized by "exhibitions." Once you set up an exhibition, you can choose the format and start uploading images. The options are few, and it takes almost no coding to make small change the presentation. More advanced users can alter the code to made greater design changes.

Support for this CMS appears to be limited, since the user base is so small and specific. There are few tutorials on the site and most of the live help that happens, happens on the forum, which at least seems to be active.

There are a number of other CMS's to choose from, but none that I would recommend as strongly as these. Some developers swear by Moveable Type, though it is being phased out as you read this. Some swear by Drupal, but it's a sprawling engine with a steep, steep learning curve. Others, like Joomla, are heavily flawed and outdated.

As you may have figured out by now, a CMS is a kind of software that can live quite closely to your heart. Site editors either come to love or hate their CMS's, since it can make or break the experience of operating a website.

I would recommend starting with easiest ones, like Tumblr or Indexhibit, and work from there in building more sophisticated platforms once you're comfortable with CMS's in general.

Good luck, and thanks!